William Y. Chang
Hosame Abu-Amara
Jessica Sanford

Transforming Enterprise Cloud Services

Springer

Transforming Enterprise Cloud Services

William Y. Chang • Hosame Abu-Amara
Jessica Feng Sanford

Transforming Enterprise Cloud Services

Springer

William Y. Chang
Irvine, California
U.S.A.

Jessica Feng Sanford
Los Angeles, California
U.S.A.

Hosame Abu-Amara
Irvine, California
U.S.A.

ISBN 978-90-481-9845-0 e-ISBN 978-90-481-9846-7
DOI 10.1007/978-90-481-9846-7
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2010937765

Cover design: eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my wife, Kathy, my son, Robert, and all sentient beings

–William

To my family

–Hosame

To my beloved Grandpa and my husband Brandon

–Jessica

Foreword by Martin Creaner

Cloud services, such as Cloud Computing, storage, and applications, represent a significant evolution in the use and provision of digital information services for business effectiveness. While market hype and growth in Cloud services is at an all time high, the development of a commercial market of sellers and buyers of such services has only just begun.

Already the market is becoming littered with a confusing array of technical features, names, terms, and proprietary approaches. While the market holds great promise, it will not grow if real customers' needs are not addressed, or if services do not deliver the quality, value, and security that they promise.

The idea of 'renting' computing capabilities is not new. In fact, the global telecom network is in many regards the original Cloud-on-demand services running over a shared infrastructure. What makes Cloud-based services new and exciting is the level of user control and the ability to blend in-house computing with services in the Cloud to create a seamless, transparent, and highly cost-effective information environment. Cloud services represent a key enabler of financial benefit for all types of *Service Oriented Enterprises* (SOEs), providing competitive flexibility and efficiency.

There are many technological and business roadblocks that stand in the way of innovation and widespread adoption of Cloud capabilities. Unless these barriers can be removed, and economic and operational cost pressures addressed, Cloud services will never realize their commercial potential. This is where the TM Forum comes in. It is becoming apparent that there are a number of key issues that are emerging that need to be addressed in order for the Cloud to be a commercial success, including security, performance, governance, portability between Clouds, and overall transparency. Many argue that most of these issues—particularly the fears over security—are more imagined than real. That may be true, but unless they are addressed to the satisfaction of the buyer community, the Cloud will fail to make it into the big leagues where major corporations are surrendering significant portions of their infrastructure to the Cloud.

The TM Forum has a proven track record of leading the communications and IT industries through market challenges and helping the market create, deliver, and monetize new business opportunities. To this end, the TM Forum has initiated a wide-ranging Cloud Services Program to bring together the elements necessary for realizing a successful Cloud services market, including an Ecosystem of Cloud buyers and sellers that will enable commercialization of this major business opportunity and a series of collaboration teams looking at issues such as governance, Cloud performance metrics and benchmarking, portability, and transparency of Cloud services.

In this book, William Chang, Hosame Abu-Amara, and Jessica Feng Sanford take an important step in providing some clarity to the increasingly confusing Cloud world. Beginning with a general introduction to the Cloud and the business opportunities, the authors then goes on to explain the various architectural approaches being adopted, including the Public Cloud, Private Cloud, Hybrid Cloud, and Community Cloud. Important topics such as how the TM Forum's core standards, such as the Business Process and Information frameworks, can be applied to the Cloud are addressed in some detail. Likewise, the book applies the well established Service Level Agreement handbook to the challenges of creating SLAs in a Cloud world. Security and policy management in a Cloud environment are explored from many different aspects and in great detail.

Overall, this book provides an important link between the communications management mindset and the enterprise Cloud world. It also shows how some of the lessons learned over the past 20 years of optimizing communications technical and business effectiveness can be applied to the emerging Cloud challenges. I hope that you enjoy the read and gain some valuable insights into how to exploit the Cloud to meet your specific challenges.

President, Telemanagement Forum (TM Forum) Martin Creaner

Foreword by Miodrag Potkonjak

There are three main questions that I always ask myself before I buy and more importantly read a book. After all, reading any book takes so much time that even if I value my time at minimum wage, it almost always costs significantly more than the book itself. The three questions are:

Why a book on this topic? Why a book from these authors? Why exactly this book?

In the case of this book, the first two questions can obviously be instantiated as “Why a book on Cloud Computing?” and “Why a book by William Y. Chang, Hosame Abu-Amara and Jessica Feng Sanford?”

The first question is simultaneously, to paraphrase Dickens, the easiest and the most difficult to answer. It is, maybe surprisingly, also often the most important criteria. After all, if we want to learn how to cook, we will not buy a neurosurgery textbook, and if are preparing for a neurosurgery exam, a cookbook will not help us much.

Cloud Computing is a topic of great interest to a wide business, managerial, technical, and scientific audience.

At least from an economic point of view at the simplest, but very important level, Cloud Computing is a large and rapidly growing market segment.

It is already more than $14 billion and will grow to more than $46 billion in the next four years. From a theoretical economic way, Cloud Computing is best explained with economy of scale and Nobel prize winner Ronald Coase’s theory that each firm will expand its operations only in directions that are profitable. This is usually once a certain product becomes a commodity, e.g., electricity or gas or now IT, computing, and data storage. Non-specialized companies are better off buying commodity services.

At the very basic and fundamental viewpoint, this is a book about the benefits and problems of sharing. Sharing is a universal concept on which dominating industrial, economic, and government operation is based. We share highways, streets, public transportation, hotels, health services, communication infrastructure, etc.

One can argue that the history of computers has been tremendously influenced by the question about what and when to share. It seems that every few years the paradigm drastically changes. For example, on one hand, sharing is emphasized in computational paradigms such as mainframes, supercomputers, utility platforms, grids, the Internet, data Clouds, and sensor networks. On the other hand, we have prominent platforms that emphasize the benefits of individual resources, such as workstations, personal computers, laptops, and the most popular ever computing and communication system—cell phones. Cloud Computing is an ultimate large-scale sharing phase in centralized computing, much like a telecommunication network. As a matter of fact, its name was inspired by Cloud s in AT&T commercials. Specifically, the founders of once famous, then troubled, and eventually successful start-up Loud Computing got inspiration from the AT&T commercials.

This is also a book on the most complex human-made system. The most complex mechanical artifact is an airplane that has a couple of million mechanical parts. Even very obsolete processors have tens of millions of transistors. Modern datacenters have 10^15 transistors and even more interconnects. Specifically, the largest datacenters have almost a million computers, each with almost a billion transistors in processors and even more in storage elements., They operate at the speed of several GHz. However, real complexity is in the system software, and we should expect that soon comb filters optical communication will connect datacenters at petaHertz speeds. In addition, each day 20+ petaBytes of new data is stored in datacenters. From a system software point of view, datacenters are mainly about system management (e.g., thermal and load balancing) and even more about virtualization. Both topics are exceptionally important and interesting.

From an energy and green computing viewpoint, a single observation is sufficient to induce a great deal of interest: 40% of the overall cost of a datacenter during its life time is spent on energy. Security and privacy are widely recognized as emerging premier desiderata.

An intriguing and difficult to answer question is the impact of Cloud Computing on these requirements. There are arguments that both centralized and distributed can facilitate or impede security. Even outside of computing, historical lesson are contradictory. During World War II, more than two million people died in Leningrad mainly because the Soviet government decided to centralize storage of all food that was burnt by German air attacks.

On the other hand, one of the richest people of all time stated many times that the strategy of putting eggs in many baskets is not likely to result in good economic management. According to him, eggs should be all placed in a single basket and one should take great care about that single datacenter, I mean, basket.

Finally, it is interesting to point out that some of the coolest patents (read innovation with short term economic potential) are related to Cloud Computing and datacenters in particular. For example, to create a green computing datacenter, Google patented the idea to place a datacenter on a stationary ship that is placed in cold water (e.g., Bay Area) near the coast. In view of the fact that cooling costs up to 1/3 of the operational cost, the benefits of this strategy may be significant.

In summary, it does not matter what somebody's primary points of view are, Cloud Computing is an important and interesting topic to study.

The second question, why a book by these authors, can be also answered at several levels.

Academic authors not so seldomly overemphasize research, conceptual, and optimization issues at the expense of a less sophisticated, but more practical discussion. Industry people on other hand are often overimpressed by technical details of ephemeral importance. Researchers and advisers from consulting companies are in an ideal position to strike a balance between current and long terms issues, between foundations and applications, between so many potential ways to emphasize and address specific topics.

More importantly, they have deep grasps of not only all technical and technological problems in Cloud Computing, but also of all management issues. All three authors have unique talent to explain complex issues using conceptually clear and engaging writing styles. Finally, William, Hosame, and Jessica have amazingly comprehensive and diverse relevant industrial and research credentials.

The final of the three questions is why this book? To the best of my knowledge, this is the first real book on the tremendously important topic. It has a broad and in-depth treatment of all essential aspects. It has the right balance between foundations and practical issues, between current state-of-the-art and long term trends. It treats the topic in a layered and multidimensional way, enabling both fast conceptual and detailed technical knowledge. Many classes of readers will learn a lot about Cloud Computing, many of them will be able to directly use new knowledge in their everyday work.

So, in summary, this is an excellent book by highly qualified authors on one of the premier high impact topics.

Computer Science Department,
University of California, Los Angeles

Professor Miodrag Potkonjak

Preface

The broad scope of Cloud Computing is creating a technology, business, sociological, and economic renaissance. It delivers the promise of making services available quickly with rather little effort. Cloud Computing allows almost anyone, anywhere, at anytime to interact with these service offerings. Cloud Computing creates a unique opportunity for its users that allows anyone with an idea to have a chance to deliver it to a mass market base. As Cloud Computing continues to evolve and penetrate different industries, it is inevitable that the scope and definition of Cloud Computing becomes very subjective, based on providers' and customers' perspective of applications. For instance, Information Technology (IT) professionals perceive a Cloud as an unlimited, on-demand, flexible computing fabric that is always available to support their needs. Cloud users experience Cloud services as virtual, off-premise applications provided by Cloud service providers. To an end user, a provider offering a set of services or applications in the Cloud can manage these offerings remotely. Despite these discrepancies, there is a general consensus that Cloud Computing includes technology that uses the Internet and collaborated servers to integrate data, applications, and computing resources. With proper Cloud access, such technology allows consumers and businesses to access their personal files on any computer without having to install special tools.

Cloud Computing facilitates efficient operations and management of computing technologies by federating storage, memory, processing, and bandwidth. In the mainstream IT industry, Cloud Computing is broken down into three segments: *applications*, *platforms*, and *infrastructure*. As an evolution of existing enterprise IT environments and services, Cloud Computing services store data (archive, backup, general-purpose), deliver applications (Software as a Service (SaaS)), support software development (Platform as a Service (PaaS)), and deliver access to Internet Infrastructure (Infrastructure as a Service (IaaS)) or to IT resources (Hardware as a Service (HaaS)). There is also the provisioning of an integrated IT that allows a user or a group of users to access federated IT datacenters (IT as a Service (ITaaS)).

Comparing Cloud Computing to other computing technologies, the Grid Computing technology is about a massively scaled infrastructure that is capable of processing huge amounts of data very quickly. The Utility Computing technology on the other hand is about a model of pricing and delivery. In other words, Utility Computing allows its customers to access computing resources as needed, and only pay for the consumed resources. Grid and Utility Computing are two mature technologies and have offered these capabilities for quite some time. What is revolutionary in the Cloud approach is its flexibility that allows enterprises to eliminate their overcapacity by using external or reusable environments to address peak demands. The evolutionary aspect of the Cloud approach, especially PaaS and IaaS, is the level of automation, making traditional IT provisioning easier and faster. As for SaaS, the Cloud approach provides an innovative method that supports a self-service nature and allows unique cross-provider service automation. These are considered an evolution from existing approaches.

From the traditional IT practices, Cloud technology enables enterprises to outsource their computing requirements and receive the needed services through external providers. In other words, outsourcing computing infrastructures to Cloud platforms benefits the enterprises by reducing costs and increasing services. Using such an option, enterprises no longer have to worry about customer support, hosting facilities, hardware deployment, or operational management. This change allows enterprises to leverage massive amounts of resources from the Cloud industry that are architected using high-end networks, servers, and storage, along with built-in fault tolerance. Therefore, whether adopting the new operational model with private or public network services, transforming to Cloud technology over a traditional deployment can yield obvious advantages in scalability, flexibility, reliability, fast setup, affordability, and environmental efficiency. Outsourcing IT infrastructure offloads enterprise resources that deal with IT services, allowing enterprises to focus on their core business or reduce overall business costs. The motivations of moving to the new technology paradigm make changes in the business paradigm possible as well. For example, this new method becomes important to program applications where all users must be served in an economic way. This is called "*multi-tenancy*." Multi-tenancy allows service users to "own" their data and configuration in a virtually partitioned environment, where one application instance can serve multiple users. Although this model is similar to the traditional Online-Transaction Processing software (OLTP) from a user's perspective, in actuality, the virtualization technology detaches applications from the underlying platforms/infrastructure, thus making the supporting resources virtually unlimited. Using SaaS in a large enterprise, the data of multiple users is effectively intermingled in the same collaborated and integrated database. As a result, the adoption of the Cloud paradigm makes it possible for enterprises to serve millions of users and forces enterprises to rethink of their application development and even their business cultures. As more multi-tenant applications become available, competing enterprises may desire to take advantage of this new technology, but in the meantime address security, data privacy, and availability aspects as their differentiators.

In addition to different service adaptations, evolutions of Cloud technologies also provide several deployment alternatives that further influence enterprise business models and present different process and management challenges. Three waves of changes are emerging in many industries:

1. In the first wave of changes, vertically integrated processes and technology infrastructures are established in large enterprises. They are based on proprietary and internal architectures. Essentially, providers deliver islands of Cloud services in order to pioneer these new methods using their own standards to achieve leading roles in their industries.
2. In the second wave of changes, experienced enterprises and service providers leverage business values from each other and start to form Cloud ecosystems for their Community of Interest (CoI). Although the evolving supply chains are vertically integrated, the collaboration is mainly based on proprietary agreements. The value chains (ecosystems) focus on the improvement of the cross-domain *Quality of Service* (QoS) and efficiency of management automation.
3. The third wave of changes arises once the vertical integration gains traction. Cross-providers, policy, and security in this wave are more mature and accessible by others. Smaller providers federate horizontally to gain economies of scale, while enterprises leverage horizontal federations for peak capacity. More choices of services and technologies are available at each layer of the Cloud, integration standards will drive even the fast pace of service developments. Service agents will trade service resources without a user's knowledge.

As the maturity of the technological level and business model in Cloud services advances, the selection of an advanced wave over its previous generation is not a necessity for every enterprise. This is because enterprises gain different degrees of business advantages when carefully selecting technology solutions that offer the closest synergy with their orders of focus and operational strategies. This can be executed beautifully without having to always follow the latest and most comprehensive architecture. More importantly, these enterprises must be ready to fully appreciate the values and features of the Cloud vendors that are most suitable for their business, and be willing to transform their corporation to leverage these investments. Such transformation in an enterprise will include four major areas: *people*, *organization*, *process*, *and technology*. In a nutshell, transformational leadership involving people and organization can ensure the enterprise maintains a common vision and strategy. Through organizational and cultural changes, the enterprise can blend the Cloud concept more deeply into their revised business values and objectives. This transformation leads enterprise stakeholders to architect their business missions and objectives based on the most appropriate Cloud technologies. Meanwhile, the enterprise management process will synchronize the enterprise business with the newly acquired Cloud technologies, using a common best practice to effectively and efficiently achieve the goals of the transformation. Ultimately, these best practices will facilitate faster adoption of other compatible business models, as well as provide a framework for contributing innovative input back to the Cloud ecosystem. All these benefits force enterprises to rethink their organizations, communications, worker skills and knowledge, decision making processes, new and old business paradigms, and approaches for the adaptation.

Throughout this book, the authors present opportunities from different angles that allow enterprises to take advantage of this new business, process, and technology model and to benefit from their practices. Concepts illustrated in this book are derived from many developing standards and implementations. To elaborate these developments, the authors also provide assessments and recommendations against these topics to add value to the subject matter. It is the authors' belief that true network-centric enterprise services will realize greater collaboration possibilities, allowing enterprises to promote inter-organizational, inter-community, global, and private sector interactions.

As we have seen, accelerated changes force enterprises to reconsider the way they operate. The authors hope this book can help enterprises draw appropriate attention on the key areas of transformation, and lay down workable options with sufficient rationales to make the best decisions. For engineers or students who are interested in this topic, this book hopes to provide a clear picture of Cloud Computing and its applications, as well as to provide a comprehensive perspective on the trend of this subject.

Finally, the contributing author(s) for each chapter is denoted after the chapter title by a subscript number that represents the author order on the cover page (e.g., Title$_{1,2}$).

Irvine, California	William Y. Chang
Irvine, California	Hosame Abu-Amara
Los Angeles, California	Jessica Feng Sanford

Acknowledgments

Many people have played a part in the development of this book. The authors would like to thank Mark de Jongh for agreeing to publish this book with Springer and Cindy Zitter for doing an excellent job of coordinating the review of the manuscript to get it to the final product.

Our gratitude is extended to Edward Chow (NASA Jet Propulsion Laboratory) and Henry J. Sienkiewicz (Chief Information Officer, DISA), whose insightful information helped us formulate the core idea of this book. A special recognition to our colleagues at Booz Allen Hamilton, in particular Sean Harnedy, for smoothing the technical prose and making excellent suggestions that led to many improvements to the book.

This book could not have been completed without the support from our friends at TeleManagement Forum). We are particularly grateful to Christy Coffey for her tireless sponsorship of the book from the initial stage to the production and marketing efforts. The support from the TM Forum Defense & Government Catalyst team helped us keep up-to-date with the standards and trends of the Cloud development. We would like to acknowledge Mike Kelly, Paul McCullin, Joyce Pearce, Juan A. Nodar (TM Forum), Martin Huddleston (Defence Science and Technology Laboratories, UK), Halmai Tamas (NATO C3A), Xi Jiang (the Johns Hopkins University Applied Physics Laboratory), Russell Wurth (Cadence, LLC), and John Wilmes (Progress Software) for providing invaluable insight into the standards, which added a lot of weight to the content.

Finally, we would especially like to thank Mr. Martin Creaner at TM Forum and Professor Miodrag Potkonjak at UCLA for writing the wonderful forewords for this book.

Contents

1 Introduction to Enterprise Services and Cloud Resources$_1$ 1

1.1 Introduction to Enterprises 2

1.1.1 Enterprise Resources 2

1.1.2 Enterprise Architecture 3

1.2 Definitions of Cloud, Services, and Ecosystem 7

1.2.1 The Cloud 7

1.2.2 Cloud Services 8

1.2.3 Cloud Technologies 9

1.2.4 Cloud Ecosystem 10

1.3 History of Cloud and Enterprise Services 15

1.3.1 Initial Establishment 16

1.3.2 Early Developments 17

1.3.3 Recent Major Developments 18

1.3.4 Network-Centric Operations 20

1.4 Cloud Enablers 22

1.4.1 Service Architecture and Abstraction 23

1.4.2 Virtualization 26

1.4.3 Web Technologies 29

1.4.4 Key Cloud Characteristics 32

1.5 Enterprise Transformation 34

1.5.1 People and Organization 35

1.5.2 Process 37

1.5.3 Technology 37

1.6 General Framework & Book Origination 38

References 41

2 Cloud Service Business Scenarios and Market Analysis$_1$ 43

2.1 Overview 43

2.2 Cloud Use Cases and Applications 45

2.2.1 Public Cloud 46

2.2.2 Community Cloud 47

2.2.3 Private Cloud 48

2.2.4 Hybrid Cloud 48

2.3 General Information Technologies 49

2.3.1 Software Services 54

2.3.2 Platform Services 55

2.3.3 Infrastructure Services 56

2.4 Commercial Markets and Applications 56

2.4.1 Marketing 58

2.4.2 Sales 60

2.4.3 Finance 60

2.4.4 Financial Industry 61

2.4.5 Telecommunications Industry 62

2.5 US Government and Defense 64

2.5.1 Federal Chief Information Officers Council 65

2.5.2 General Services Administration (GSA) 67

2.5.3 National Business Center (NBC) 67

2.5.4 National Institute of Standards and Technology 69

2.5.5 The U.S. Department of Defense 70

2.6 Scientific, Educational, and Others 75

2.6.1 US Department of Energy (DOE) and Magellan 76

2.6.2 NASA Nebula 77

2.6.3 Education 79

2.6.4 Other International Organizations 81

2.7 Conclusion 84

References 85

3 Cloud Service Architecture and Related Standards$_{3,2}$ 87

3.1 Overview 88

3.2 Types of Cloud Services 91

3.2.1 Software as a Service 91

3.2.2 Platform as a Service 94

3.2.3 Infrastructure as a Service/Hardware as a Service 96

3.3 Holistic Enterprise Architecture and Cloud Services 100

3.3.1 Service and Business Layer 101

3.3.2 Data and Information Layer 101

3.3.3 Integration Layer 102

3.3.4 Technology and Tool Layer 102

3.4 Enterprise Architecture and Cloud Transformations 103

3.4.1 Enterprise Architecture Styles 103

3.4.2 Architecture Transformation 105

3.5 Cloud Architectures and Vendor Implementations 109

3.5.1 Public Cloud 110

3.5.2 Private Cloud 111

3.5.3 Hybrid Cloud 113

3.6 Cloud Related Standards and Forums 114
3.6.1 Open Grid Forum 114
3.6.2 Open Virtualization Format 115
3.6.3 HTTP 116
3.6.4 XML and JSON 117
3.6.5 AJAX 117
3.6.6 HTML5 118
3.6.7 Web Syndication 118
3.6.8 XMPP 120
3.6.9 REST 121
3.6.10 Security and Data Privacy Standards 122
3.7 Enterprise Transformation Implications 125
3.7.1 Information Framework 128
3.7.2 Process Framework 129
3.7.3 Service Level Management 129
3.8 Conclusion 130
References 130

4 Challenges of Enterprise Cloud Services$_1$ 133
4.1 Overview 134
4.2 Non-Technical Challenges 137
4.2.1 Financial 138
4.2.2 Enterprise Scalability 138
4.2.3 Business Operations 141
4.2.4 Organizational 142
4.3 Software Services Perspective 143
4.3.1 User Data 143
4.3.2 Data and Applications 145
4.3.3 Integrity 145
4.4 Platform Services Perspective 150
4.4.1 Data and Information 150
4.4.2 Platform Service Framework 152
4.4.3 Platform Integration 154
4.5 Infrastructure Services Perspective 155
4.5.1 General Infrastructure 156
4.5.2 Service Performance 160
4.6 Security Challenges 161
4.6.1 Data 163
4.6.2 Secured Access 168
4.6.3 Data Governance 169
4.6.4 Data Leakage 171
4.6.5 Security Framework 172
4.7 Operational and Management Challenges 173
4.7.1 Strategy and Service Planning 174

4.7.2 Service Fulfillment 179
4.7.3 Service Assurance 181
4.8 Conclusion 184
References 185

5 Networked Service Management$_2$ 189
5.1 Overview 189
5.2 Software as a Service 190
5.2.1 Software as a Service Licensing Models 190
5.2.2 Transforming Enterprise Architectures to Service-Centric Architectures 191
5.2.3 Enterprise Integration Architecture to Access Software as a Service Applications 193
5.2.4 Enterprise Composition Architecture to Access Software as a Service Applications 198
5.2.5 Transformation Reference Architecture for Enterprises 199
5.2.6 SaaS Data Architecture 203
5.3 Hardware as a Service/Infrastructure as a Service 205
5.3.1 IaaS Hierarchy 205
5.3.2 POD Architecture 207
5.3.3 Transforming Enterprises to Use IaaS 208
5.4 Platform as a Service 213
5.4.1 Implications of PaaS on Transforming Enterprises 213
5.4.2 Example PaaS Techniques 214
5.4.3 Public Cloud vs. Private Cloud 215
5.5 Service Definition and Instance Management 219
5.5.1 Virtualization and Cloud Infrastructure 220
5.5.2 Virtualization-Optimized Cloud Infrastructure 222
5.6 Service Level and Quality Management 223
5.6.1 Specification of Service and Quality Levels 224
5.6.2 Cloud Service Level and Quality Management Architecture 228
5.7 Conclusion 228
References 229

6 Cross-Domain Policy-Based Management$_{1,3}$ 231
6.1 Overview 232
6.2 PBM Benefits and Potential Applications 233
6.2.1 The Benefits and Business Drivers of PBM 234
6.2.2 PBM Support OSS and BSS 234
6.3 PBM Standards and Commercial Implementations 236
6.3.1 TM Forum SID's Policy Aggregate Business Entities 236
6.3.2 IETF Policy Workgroup 240
6.3.3 Market Players 241
6.4 Policy and Management Framework 243

6.4.1 Policy Template 244
6.4.2 Policy Implementation and Usage 245
6.4.3 Policy Management and Policy Engine 248
6.5 Transforming PBM to a Cloud Environment 249
6.5.1 Cloud-Focused Policy Stack 250
6.5.2 Design Considerations 253
6.5.3 Implementation Considerations 254
6.5.4 Service Policy and SLA 256
6.5.5 Service Policy and Resource Allocations 258
6.5.6 Security in Cloud Policy Management 258
6.6 Externalizing Policy and Management 260
6.6.1 Policy Negotiation 261
6.6.2 Automated Policy Negotiation 264
6.6.3 Policy Adaptation 266
6.7 Conclusion 269
References 270

7 Building and Configuring Enterprise Cloud Services$_{3,2}$ 273
7.1 Overview 274
7.2 Design Principles and Deployment Options 276
7.2.1 Service Automation 277
7.2.2 Adapting to High Utilization and Rapid Growth 279
7.3 Standards-Based Business Process Framework 281
7.3.1 The ITIL and eTOM Frameworks 282
7.3.2 Level Zero Key Concept 282
7.3.3 Level One Processes 283
7.3.4 Level Two and Three Processes 285
7.3.5 Improvements to Current eTOM for Cloud Services 288
7.4 Standards-Based Information Framework 289
7.4.1 The SID Business View 289
7.4.2 SID Domains and Level One ABEs 291
7.4.3 Service Domains and Level Two ABEs 291
7.4.4 Improvements to the Current SID for Cloud Services 292
7.5 Technology-Neutral, Service-Centric Architecture 293
7.5.1 Next-generation Datacenter Management 294
7.5.2 Architectural Planning, Simplification, and Transformation 295
7.6 Conclusion 307
References 307

8 Service Monitoring and Quality Assurance$_2$ 309
8.1 Overview 309
8.2 Enterprise Quality and Performance 310
8.2.1 Service Level Agreements, Enterprises, and Customer Experiences 310

8.2.2 Key Quality Indicators and Key Performance Indicators ... 313
8.2.3 Sample Key Quality Indicators and Key Performance Indicators ... 314
8.2.4 Quality Equations and Measurement ... 316
8.3 Service Quality Management ... 318
8.3.1 Value-Chain SQM ... 319
8.3.2 SQM Metrics ... 322
8.4 Probes ... 323
8.5 SLA Management and Reporting ... 324
8.5.1 SLA Monitoring and Reporting Process ... 325
8.5.2 SLA Reporting Mechanisms ... 326
8.6 Enterprise SLA Negotiation ... 327
8.6.1 SLA Development Process ... 327
8.6.2 Form of an Enterprise SLA ... 329
8.7 Policies and Monitoring ... 332
8.7.1 Monitoring Agents ... 334
8.7.2 Manageability and Operability ... 336
8.8 Conclusion ... 338
References ... 340

9 Security for Enterprise Cloud Services$_2$ ... 341
9.1 Overview ... 342
9.2 Security for Cloud Services and Infrastructure ... 343
9.2.1 Authorization and Role-Based Access Control ... 343
9.2.2 Cloud Security Services ... 347
9.2.3 Integration of Role-Based Architecture in the Web ... 349
9.3 Security for Enterprises that Use Cloud Services ... 350
9.3.1 Federated Identity Management Architecture ... 351
9.3.2 Side Channel Attacks and Counter-Measures ... 353
9.4 Intrusion Detection in Cloud Computing ... 356
9.4.1 Types of Raw Data Collected ... 359
9.4.2 Distributed Intrusion Detection Architecture ... 360
9.4.3 Fusion-Based Intrusion Detection Systems ... 362
9.5 Security for Cloud Service Management ... 368
9.5.1 Security for APIs ... 368
9.5.2 Security for Service Containers ... 369
9.6 Measures for Cross-Virtual Machine Security ... 372
9.6.1 Virtual Machine Security ... 372
9.6.2 File System Security Management ... 373
9.6.3 Virtual Machine Image Security ... 376
9.7 Conclusion ... 378
References ... 382

10 Enterprise Cloud Service Applications and Transformations$_{1,2}$ ... 385
10.1 Overview ... 386
10.2 Business and Technology Transformation ... 388

10.2.1 Establish Strategic Promises 388
10.2.2 Plan for New Business Models 392
10.2.3 Establish a Technical Innovation Culture 394
10.3 The New Form of Software and Service 395
10.3.1 End Users' Expectations 396
10.3.2 Expanding Service Categories 396
10.3.3 More Destiny Sharing Interactions 397
10.3.4 Evolving Web Applications 397
10.3.5 Integrating Enterprise SaaS with Cloud Services 398
10.4 Platform Integrations and Collaborations 399
10.4.1 New Applications Development Functions 399
10.4.2 Software Development Standards 400
10.4.3 New Software Packaging Focus 401
10.4.4 New Relationship with Hardware Resources 401
10.4.5 Integrating Enterprise and Cloud PaaS 402
10.5 Infrastructure Transformations 402
10.5.1 Customizable Service Resources 403
10.5.2 Improved Infrastructure 403
10.5.3 Customer Portal and Rapid Provisioning 404
10.5.4 Integrating Enterprise and Cloud IaaS 404
10.6 Cloud Management and Operational Framework 405
10.6.1 Management Paradigms 405
10.6.2 Service Management Automation 406
10.6.3 Changing Process Management 407
10.6.4 Integrating Enterprise and Cloud Governance 408
10.6.5 Integrating Enterprise and Cloud Quality Assurance 409
10.7 Cloud Security and Information Assurance 411
10.7.1 New Applications of Information Assurance 411
10.7.2 Security in Different Service Layers 413
10.8 Final Notes 415
References 415

Authors' Bios 417

Index 419

Chapter 1
Introduction to Enterprise Services and Cloud Resources[1]

Cloud Computing is the latest revolution in the *Information Technology* (IT) industry, following the personal computer revolution and the Internet revolution. This new technology not only matters to the IT industry, but also to technology consumers because its services will soon be directly accessible to consumers' daily appliance-level devices.

Cloud Computing is named after the Cloud representation of the Internet commonly depicted on a network diagram. Its concept broadly implies using the Internet to allow people to have access to virtualized resources, whereby users can manage and control their purchased services. This technology, sometimes also associated with Grid Computing, can be seen as a reincarnation of centralized data processing and storage, as paralleled by the mainframe. It is a resource delivery and usage model, meaning it gets resources via the network on-demand and at scale in a multi-tenant environment. The prime revolutionary aspect of Cloud Computing is its ability to deploy location-independent services. Although the model is similar to a large network of computers that is managed by large organizations that provide services to smaller organizations or individual clients, s*ervice* c*onsumers* (SCs) are no longer locked-in with their providers. This revolutionary technology enables users to switch providers easily and quickly due to its open nature.

Although from a user's or application developer's perspective only the Cloud is referenced, the managing *service providers* (SPs) who provide software, hardware, *Operating Systems* (OS), and networking services now face new process and technology challenges that never existed before. The main challenge includes managing various infrastructures across multiple organizations consisting of frameworks that now include self-healing, self-monitoring, and automatic reconfiguring mission-critical applications.

In this chapter, the authors intend to draw a foundation for the Cloud service environment from an enterprise perspective to illustrate the business, technical, process, and organizational challenges of this new revolution. This chapter will serve as the foundation for the solution discussions in the following chapters.

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_1, © Springer Science+Business Media B.V. 2010

1.1 Introduction to Enterprises

"Enterprise" in the context of this book is defined as an organization or cross-organizational entity performing within a specific business scope and mission. Business operations in an enterprise environment are heavily influenced by dynamic patterns of collaboration and are associated with different levels of accountability. This requires information integration across the management processes, operational processes, and supporting processes that are scattered across many functional areas as different services.

1.1.1 Enterprise Resources

An enterprise includes interdependent resources such as people, organizations, processes, and technology. The People category resource is represented as an abstract collection of knowledge, expertise, skill, experience, and expectation. The Organization category resource is further classified and allocated into different organizations depending upon their specific business missions. The Process category resource handles business processes as well as products, applications, and data. The Technology category resource includes software, hardware, and networking infrastructure, that coordinates business functions and shares information in support of a common mission or set of related missions [1].

Although often associated strictly with IT, these four interdependent resources relate more broadly to the practice of business operations. These four resources define the business mission and objective, the information necessary to perform the executable tasks, and the technologies necessary to carry out the task. With the same or different sets of people, the processes for implementing new technologies are executed in order to align them with the organization's core goals and strategic direction.

As shown in Fig. 1.1, the business goals of a SOE are supported by the four resources in a protective environment for effective and efficient management of data and technology for conducting business [1].

1. Business assurance comes from the governance functions for setting priorities for investments, efforts, and usage of information and data in accordance with regulations and guidance.
2. Technology resource covers enterprise-wide technical infrastructure that supports access, use, management, and delivery of data, applications, hardware, software, networking, and key performance indicators for seamless business operations.
3. Decision making can be improved by collaborating approaches, data, tools, and knowledge from the virtual environment into the business environment.
4. Business intelligence and management data provide the enterprise partners, value-chain stakeholders, and enterprise users the information they need to carry out shared business processes and make decisions.

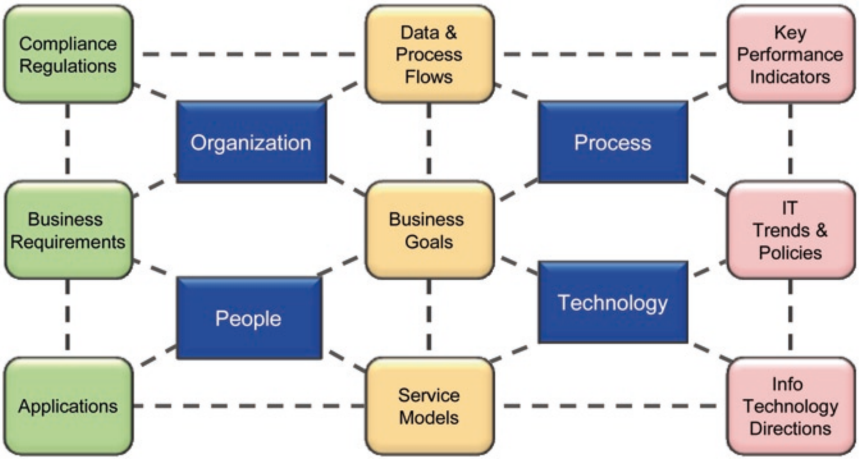


Fig. 1.1 Enterprise resource relationships

5. Applications provide enterprise partners, value-chain stakeholders, and enterprise users with applications (via service interfaces) to access, manage, use, analyze, present, and interpret enterprise data to conduct business.
6. Service models are situated among technology, people, and applications to establish an abstract and unified service view for the management communities inside or outside of (for SPs or value-chain partners) the enterprise to effectively manage the data and technologies.

A *service* is typically defined as a collection of attributes and behaviors that can be provided by an enterprise resource for use by any of the enterprise resources through well defined interfaces. To satisfy enterprise customers and users, the technology resource must provide appropriate, value-added supporting services in addition to a basic offering. For instance, in the telecommunications and information services industries, infrastructure services can be classified as a set of capabilities provided by a set of systems or utilities to their SCs. Such service offerings may include telecommunications or network transport services; services that handle information resources including the storage, retrieval, manipulation and visualization specific to the resource; and management services including fault, configuration, accounting, performance, and security functionalities, as well as service lifecycle management, service instance management, and user life cycle management.

1.1.2 Enterprise Architecture

Enterprise Architecture (EA) is the science of designing an enterprise that describes, documents, and rationalizes business capabilities, strategies, metrics, processes, structure, and resources. These aspects of the business are organized logically by

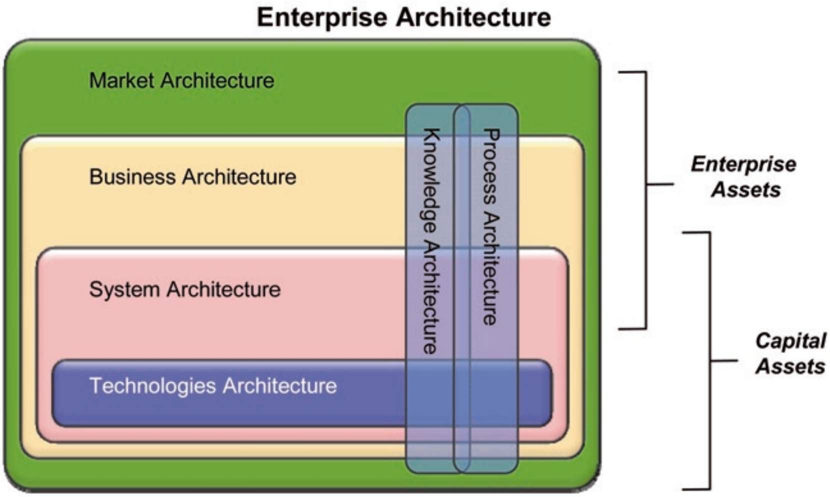


Fig. 1.2 Components of enterprise architecture

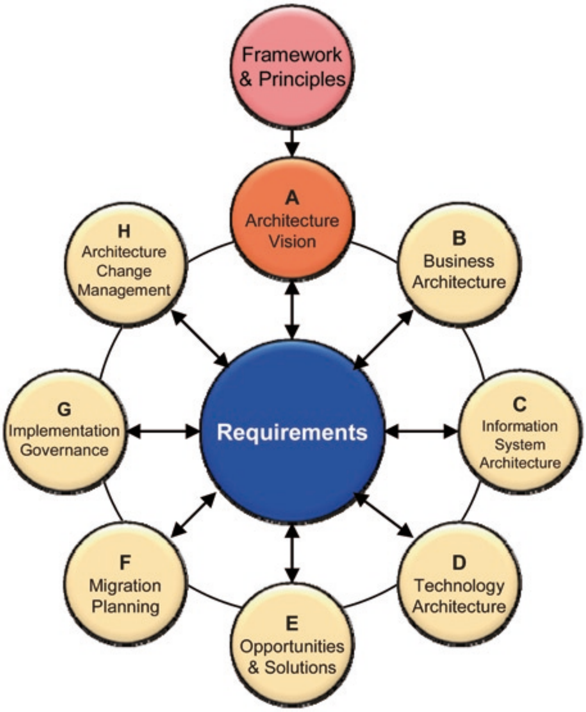
enterprise architects using a multitude of artifacts such as documents and models. The main purpose of an EA is to provide a comprehensive structure to govern an organization's operations and assets, such as IT assets and intellectual assets. This makes it an important component of an enterprise. The EA of an organization can dictate how information and technology can support the organization's business operations, goals, and capabilities effectively and efficiently. Figure 1.2 illustrates a normalized layer of architectures in an enterprise, showing technology and system architectures as actualized capital assets and business and market architectures as enterprise-level assets. Process and knowledge architectures cut through these four architecture layers and chain the enterprise's business culture, core values, and missions together. Philosophically speaking, a well-designed, well-defined, well-understood, and well-documented EA can enable an organization to respond and adapt quickly to any change in the environment in which the organization operates.

A number of standards and frameworks exist that aim to provide foundations of EA design. One of the most recognized and widely adopted standards is The *Open Group Architectural Framework* (TOGAF). TOGAF first emerged in the mid-1990s and continuously evolved into a detailed method and set of supporting resources for developing an EA. This evolution was created by representatives of some of the world's leading IT customer and vendor organizations.

In the telecommunications industry, the *Enhanced Telecommunications Operations Map* (eTOM), published by Telecommunications Management Forum (TM Forum), is the most widely used and accepted standard for business processes in the industry.

TOGAF's Version 8 Enterprise Edition is devoted to EA, including organization-wide management tools for information and communication systems. The core of the TOGAF framework is the *Architecture Development Method* (ADM), a

Fig. 1.3 TOGAF's ADM



model process with input and output descriptions for the process phases. The ADM consists of cyclical architecture development phases that concentrate on either the development of a viewpoint, an architecture, or tasks related to architecture management. There are eight main phases in this model where the EA framework and architecture principles are considered fixed in the recommended execution. These phases are illustrated in Fig. 1.3 [2]. They are:

1. *Architecture vision (analysis) phase*, includes the organization of the project, scope and domain requirements, constraints, and business scenarios (if applicable).
2. *Business architecture phase*, includes the current baseline architecture, target architecture, and gap analysis.
3. *Information systems architecture*, includes: (1) data architecture with the data type, data sources, and data model in accordance with the Business architecture, and (2) applications architecture that meets the specified business requirements and data model.
4. *Technology architecture phase*, includes the baseline architecture and the target technology architecture.
5. *Opportunities and solutions (evaluation) phase,* selects solutions.
6. *Migration planning phase*, checks dependencies in the environment and preparing for implementation of the target architecture.

7. *Implementation and Governance*, administers the implementation and deployment phase of the development project.
8. *Architecture change management (maintenance) phase*, creates new baselines, monitoring changes in the business environment, and identifying new technology opportunities.

eTOM, also called the TM Forum Process Framework, describes the full scope of business processes required by a SP and defines key elements of the process model and how these elements interact with each other. This framework is a common companion of the Information Technology Infrastructure Library (ITIL), an analogous standard and framework for best practices in IT. Figure 1.4 portrays the Level-1 processes of the Enterprise Management portion of the eTOM model, which represents a common enterprise process framework across different industries. In this model, the process of an enterprise can be a portion or a complete set of the following areas: *strategic* and *enterprise planning*, *enterprise risk management*, *enterprise effectiveness management*, *knowledge and research management*, *financial and asset management*, *stakeholder and external relations management*, and *Human Resources Management (HRM)*. Many of these process categories can provide needed details to complement the ADM illustrated earlier. The eTOM model provides a multiple layer drill down, allowing different applications to select the most relevant process for their needs. Further details of the eTOM model and its corresponding Information Framework will be covered in Chap. 7 [3].

EA is a means for an enterprise to plan, implement, and manage its IT solution for the most effective business results. Through well defined guidance and best practices, an enterprise can control whether or not it establishes a sound organizational structure, culture, and methodology to adopt the latest innovations, now

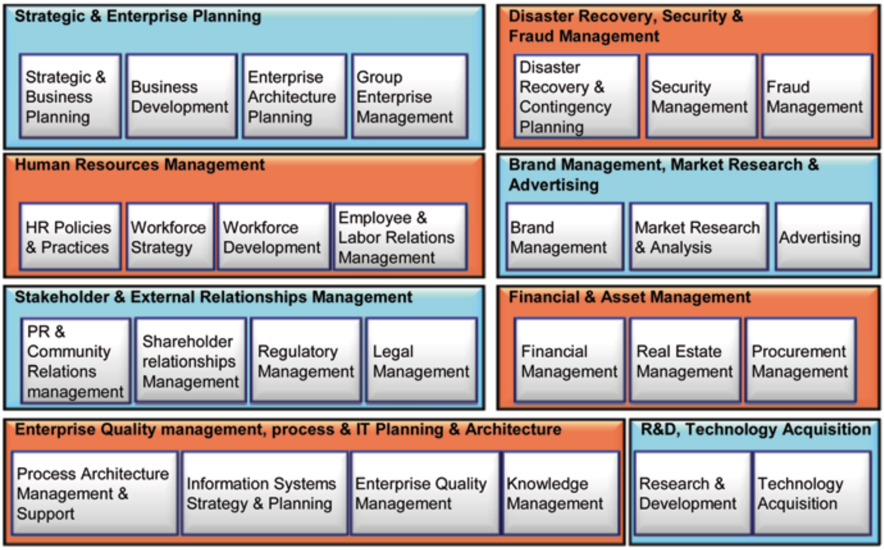

Strategic & Enterprise Planning
Strategic & Business Planning
Business Development
Enterprise Architecture Planning
Group Enterprise Management
Disaster Recovery, Security & Fraud Management
Disaster Recovery & Contingency Planning
Security Management
Fraud Management
Human Resources Management
HR Policies & Practices
Workforce Strategy
Workforce Development
Employee & Labor Relations Management
Brand Management, Market Research & Advertising
Brand Management
Market Research & Analysis
Advertising
Stakeholder & External Relationships Management
PR & Community Relations management
Shareholder relationships Management
Regulatory Management
Legal Management
Financial & Asset Management
Financial Management
Real Estate Management
Procurement Management
Enterprise Quality management, process & IT Planning & Architecture
Process Architecture Management & Support
Information Systems Strategy & Planning
Enterprise Quality Management
Knowledge Management
R&D, Technology Acquisition
Research & Development
Technology Acquisition

Fig. 1.4 TM Forum's business process framework—eTOM

including Cloud technology. Furthermore, depending upon the business paradigm driven by the new technology, an enterprise can establish a discipline to manage its value-chain partners using either centralized or federated approaches more efficiently. Through the standardized governing process, an enterprise can guarantee its clients the best quality and timely service features to achieve high customer satisfaction. Although the enterprise transformation discussions in this book will adopt many concepts from both the TOGAF and TM Forum models, they will not follow the structure described exactly. Later sections will discuss the framework of enterprise transformation, let us first look at the definitions of Cloud-related resources.

1.2 Definitions of Cloud, Services, and Ecosystem

Cloud Computing is a composite concept of two technologies that are evolving in domains that are outside the original technology arena. Reading from the surface, the Cloud represents networks and networking, while Computing represents computer-related resources, applications, and services. The idea of offering networked computers that allow different users to share applications is not a new business model in the IT industry. In this section, we will broaden the business domains that Cloud Computing covers today. Already, the people, applications, processes, hardware, firmware, software, content, SPs, and value-chain vendors that Cloud Computing reaches have already gone beyond the pure technology arena. This section provides evidence for why the existence of numerous versions of Cloud Computing definitions and interpretations are included by customers' and providers' various viewpoints. It is desirable, but may not be practical, to come up with a comprehensive set of standards for a universal capability in the Cloud and to create a single, homogeneous Cloud environment. Therefore, these different perceptions are envisioned to carry on because of the changing business needs required to meet unique values for different clients.

Nevertheless, as the concept and business model of Cloud Computing matures, there are several key principles that must be followed to ensure the Cloud is open and indeed delivers the desired flexibility and agility for satisfying enterprise requirements. It is necessary for this book to clarify the terminologies and give them concise definitions to facilitate more in-depth discussions. This section will breakdown Cloud Computing into the following four areas: the *Cloud itself*, *Cloud Services*, *Cloud Technology*, and the *Cloud Ecosystem*. A concept called *Network-Centric Operations* (NCO) and its potential commonality with Cloud services will be illustrated as well.

1.2.1 The Cloud

The Cloud references a distributed collection of computing resources where the applications can reside anywhere on the accessible networks. In the Cloud, a large

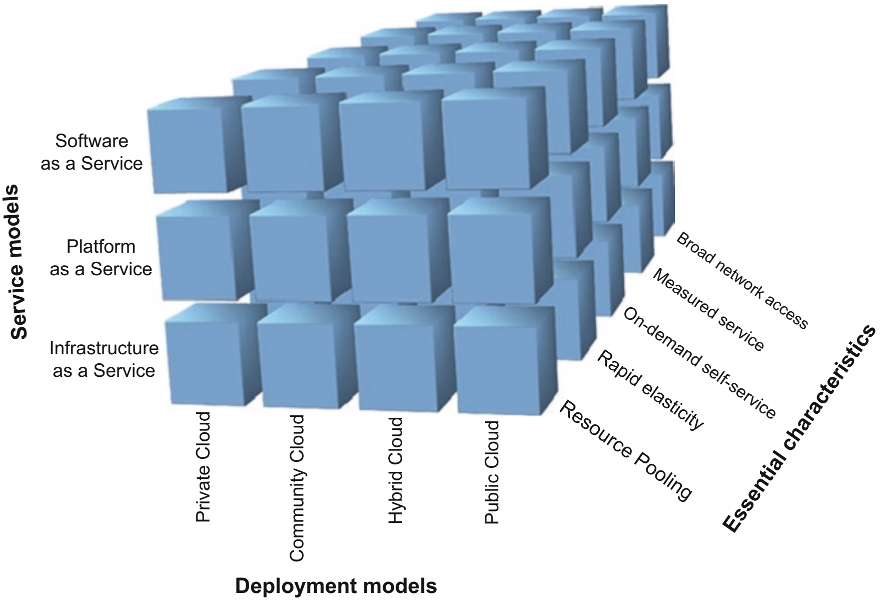


Fig. 1.5 The NIST's model of Cloud computing

pool of accessible virtualized resources such as hardware, development platforms, and ideally services, can be dynamically reconfigured to adjust to a scalable load, with minimal management effort or SP interaction. This pool of resources is typically exploited by a *pay-per-use* model and the guarantees are offered by means of a *Service Level Agreement* (SLA).

In accordance with the definition from the National Institute of Standards and Technology (NIST) Information Technology Laboratory, Cloud Computing actually covers more than just computing technology. As shown in the three dimensional diagram of Fig. 1.5, this Cloud model is composed of five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service), three service models (software, platform, and infrastructure), and four deployment models (Private, Community, Hybrid, and Public Clouds). These will be further discussed in the following sections and chapters [4, 5].

To avoid conflicting definitions with the existing naming convention, the terms "Cloud" and "Cloud Computing" will be used interchangeably throughout this book.

1.2.2 Cloud Services

Cloud services are intentionally presented within a narrow perspective of Cloud applications. They refer to both the applications delivered as services over the Internet and the hardware and systems software found in datacenters. Cloud services take full advantage of the service-oriented paradigm with a focus on the key attributes of statelessness, low coupling, modularity, and semantic interoperability. A Cloud service is comprised of three parts:

- *Basic infrastructure services* that frequently provide remote storage, hosting, firewall services, identity services, backup services, and so forth.
- *Platform software* including various support functions such as standard libraries, storage, portal servers, development tools, and OS.
- *Service-oriented applications* that exist primarily to provide services to end users, but are also accessible by other applications or application platforms.

There are additional features that Cloud services can be used for, depending upon the type of service. Not all Cloud services are required to implement the following features for competitive differentiation. These features include: *ease of operations*, *configurability* (enhance infrastructure, such as security systems, message queuing, and storage tiering), *performance* (processing speed, memory speed, storage access, read and write speeds, latency, bandwidth), *reliability and security* (risk mitigation—is at the heart of the decision over whether or not Public Cloud services are better than private datacenters), and *customer service* (how vendor relationship management will become a key discipline in IT organizations). The exact sets of additional features depend on the specific type of Cloud services.

1.2.3 Cloud Technologies

Cloud technologies provide dynamically scalable and often virtualized resources as services over the Internet. Users need not have knowledge of, expertise in, or control over the technology infrastructure in the Cloud that supports them. Several key elements must be present for Cloud technologies to enable this new Everything-as-a-Service economy model:

- A shared Cloud infrastructure that provides enterprise-grade security, scalability, and *Quality of Service* (QoS).
- A development environment that makes it easy for enterprise application developers who are used to creating small standalone applications to deliver secure, multi-tenant applications that are horizontally scalable to potentially millions of users.
- An operating environment that seamlessly delivers and updates Cloud services without impacting the user experience.
- An easy way to combine multiple Cloud services in order to achieve business or personal tasks.

Cloud technology is a computing paradigm where various computing resources are virtualized as services and allocated dynamically to tailor the users' needs through the connection of Web technology. The Cloud is based on a *Federation of Networks*

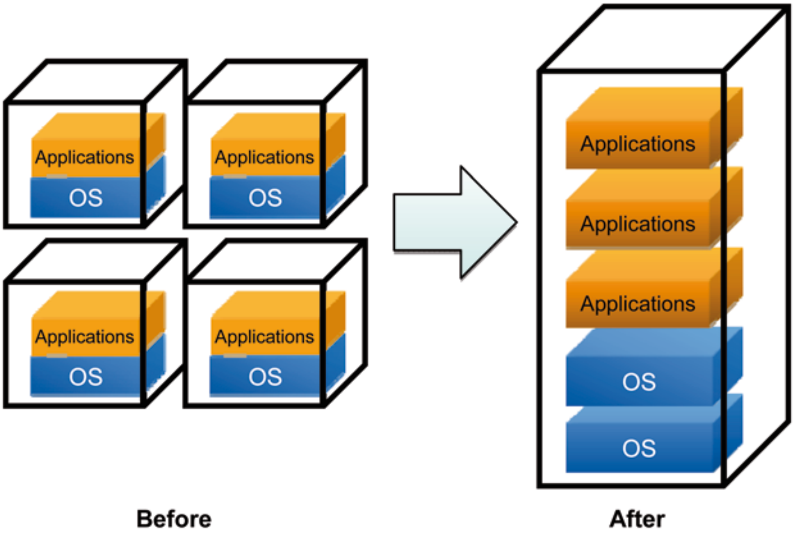


Fig. 1.6 Shared Cloud infrastructure

(FoN) as the platform for computing (i.e., the network of all connected devices). A desktop or mobile device is simply a device that is connected to a network of computers the users are connectively building. The technology that makes Cloud Computing possible is virtualization. The main objective in Cloud Computing is to improve resource utilization by sharing available resources with multiple on-demand needs. Virtualization abstracts the underlying resources such as memory, storage, and network resources so that multiple OS (e.g., Windows, Linux) can run on a single, hardware platform concurrently. This can greatly improve resource utilization. The integration of these open source technologies plays a crucial role in making the idea of Private Clouds attractive to enterprise customers. Figure 1.6 depicts a shared Cloud infrastructure environment [6].

1.2.4 Cloud Ecosystem

The ultimate business goal of an enterprise is to realize a controlled service environment that offers the rapid and flexible provisioning of compute power, storage, software, and security services to meet their mission's demands. It should combine the processes of a best practice with the agility of managed, global infrastructure to make their IT deployment faster, better, cheaper, and safer. This achieves business agility, survivability, sustainability, and security.

The traditional linear value chain for IT services is changing as a result of Cloud service concepts. For instance, transaction costs are reduced as a result of Cloud services' new price and service models. On the other hand, the much lower entry costs

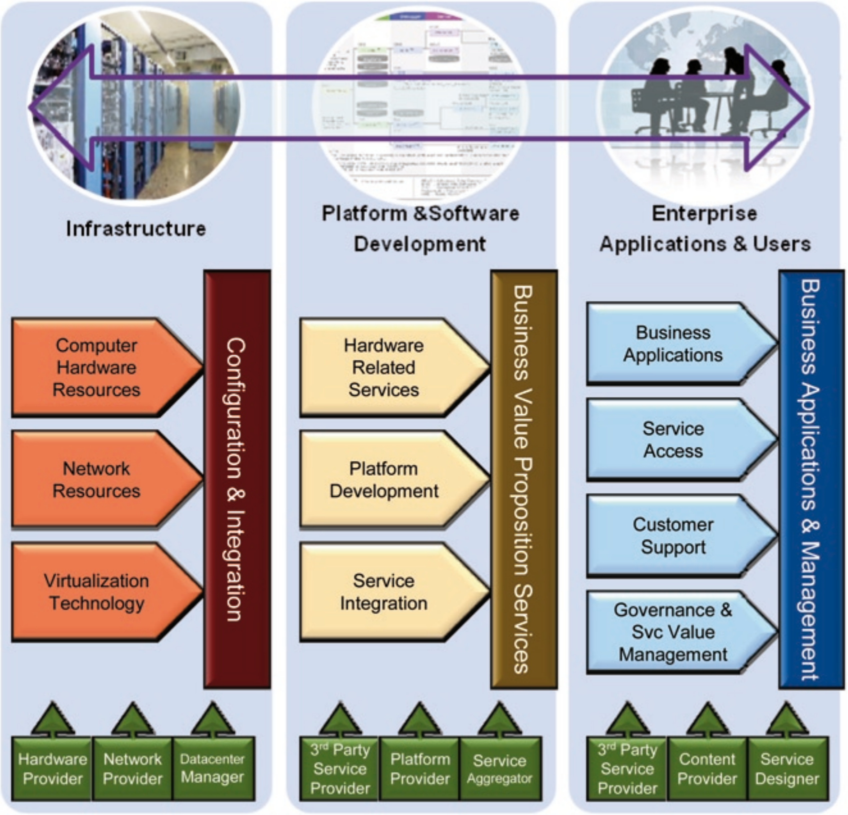

Infrastructure
Platform &Software Development
Enterprise Applications & Users
Computer Hardware Resources
Network Resources
Virtualization Technology
Configuration & Integration
Hardware Related Services
Platform Development
Service Integration
Business Value Proposition Services
Business Applications
Service Access
Customer Support
Governance & Svc Value Management
Business Applications & Management
Hardware Provider
Network Provider
Datacenter Manager
3rd Party Service Provider
Platform Provider
Service Aggregator
3rd Party Service Provider
Content Provider
Service Designer

Fig. 1.7 The Cloud ecosystem

for using a professional IT infrastructure in the Cloud give rise to a large number of small, innovative enterprises that can launch new IT service offerings in the market with minimal capital commitment and flexible operating costs.

A Cloud-based ecosystem for enterprise applications will be attractive for both developer and enterprise customers alike. For developers, Clouds open up a far wider potential audience for their products; for enterprise customers, outsourcing application management to a remote third-party on a scalable, pay-per-use basis, offers far more flexibility combined with a significant reduction in capital expenditure. The Cloud ecosystem is shown in Fig. 1.7. The following list contains the major actors in the ecosystem:

- *Cloud Users*: Cloud users include individuals or organizations who interact with Cloud services to provide services on the Cloud. End users are the consumers of Cloud services. With respect to the former type of users, they may supply information or non-technical services to their target customer communities for profit or non-profit activities (e.g., online legal consultants or blog authors.) These users have an interest to monitor Clouds for where their customers are connected, when most of them visit, when peak traffic times occur, and which infrastructure they use in order to address their services. End users use Web-browsers or thin- and thick-client applications to access and consume services available on the Cloud.

- *Software Developers*: Cloud software developers design and implement distributed, scalable applications through the use of Cloud platforms or development tools. The paradigm of software development is in a service-oriented programming style with an emphasis on asynchronous messaging, using open source interface such as *Extensible Markup Language* (XML) for identifying data types, and an open source library using *Application Program Interfaces* (API). These development tools assist application providers in developing browser-based or thin client applications. These applications reduce the associated costs by seamlessly porting applications across multiple platforms. Additionally, a service-oriented application technique called service orchestration can assist developers in binding their home-grown services with other object-oriented applications to establish federated service bundles.
- *Datacenter Managers*: Datacenter managers are responsible for managing typically large-scale systems using optimized components. Their roles typically cover the facilities department in handling real estate, building maintenance, and space planning for electrical and office environments, as well as the IT department focusing on applications, installation of new devices, and support for users. Traditional IT assets are deployed in dedicated computer rooms using costly rack-mounted components, designed by infrastructure suppliers. Some datacenter managers work for suppliers. These datacenter managers are required to understand IT governance and its implications (e.g., performance, conformance, and responsibility) as part of their managerial roles in the enterprise. They seek Cloud technology to reduce the cost and complexity of their operations through outsourcing or adapting an open Cloud.
- *Datacenters*: The enterprise datacenters are a collection of clusters offering huge amounts of computing power and storage. The increasing complexity of the computing infrastructure demands a more sophisticated means to monitor and automate resource management in this dynamic environment. This results in higher cost and skill requirements to operate their centers. Thus, more enterprises are looking at co-location as a viable solution to relieve their datacenter issues from space and power. Furthermore, using the Cloud, enterprise companies can leverage the advantages of scalability and agility from the new technology, while migrating a certain degree of risk to providers.
- *Service Providers(SPs)*: SPs offer Cloud services by owning or operating a Cloud farm in an open or private manner. They can be viewed as a virtualization of Cloud-related applications that involve the provision of services to customers over the Cloud. These providers make services such as software, platform, and infrastructure resources accessible to the service users through Internet-based interfaces. Some providers use virtual computing environments to enable their customers to develop their custom applications, load them on the Cloud, launch service instances, and manage the deployed services. Other providers aim to

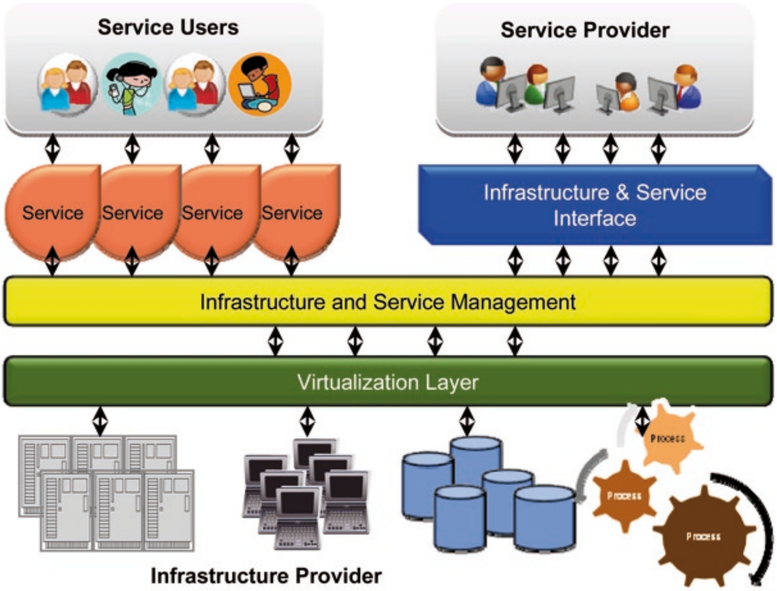


Fig. 1.8 The relationship between service users and provider through the Cloud actors

assist their customers in outsourcing their computing infrastructures, such as hosting services to other carriers. Figure 1.8 shows the relationship between service users and providers through the Cloud actors [7].

- *Cloud Integrators*: Cloud integrators are seen as *the middle men*, bringing order to an enterprise IT environment for its customers. They connect the dots between technology, systems, use cases, and organization. In particular, Cloud integrators assist their customers in planning, optimizing, integrating, and managing their heterogeneous computing environments. For customers who seek the affordability and scalability that Cloud services offer, integrators can help them create, transform, and migrate applications and infrastructure into the Cloud, including the development and deployment of either internal or Hybrid Cloud solutions without compromising security or interoperability. For instance, as the IT outsourcing industry goes through a transformation driven by Cloud Computing, enterprises are increasingly relying on skilled and trusted Cloud integrators as partners in helping them configure the best combination of IT environments suitable for their unique business needs. Examples of integrators include *Value-Added Reseller*s (VARs) and solutions providers.
- *Cloud Aggregators*: Client-Cloud aggregators make money by brokering advertisements. They provide valuable information services for free, and in return, they acquire the intimate personal information needed to better target advertising. These aggregators can make more money if they better target advertisements using this information. For example, aggregators can use consumers' shopping habits, together with local store information, to correlate a closely matched advertisement

list. The more an aggregator knows about the customers, the more it can do in the way of helpful information services. Due to the nature of the business, aggregators may face the challenge of emerging privacy-friendly competitors that perform information integration using client equipment instead of Cloud datacenters. Examples of Cloud aggregators include Microsoft, Google, and Facebook.

- *Cloud Infrastructure Vendors*: The Cloud infrastructure includes hardware and software. The software vendors provide solutions to address the market's needs in virtualization, security, containers, languages, OS, and *User Interfaces* (UIs) (e.g., Web browsers). The hardware suppliers develop specific grids, clusters, servers, routers, gateways, storage media, and racks for the datacenters. These infrastructure vendors develop and deploy optimized software and hardware to drive massive and scalable applications across the Cloud. For instance, replacing individual power supplies with shared power supplies for multiple servers is an example of optimization. Using Cloud technology, enterprises require greater levels of automation in management and bandwidth allocation, where vendors must be able to provide high degrees of availability, performance, and security. Hardware-dependent Clouds will therefore continue to offer some distinct advantages, particularly for high performance and high security applications.
- *Content Providers*: Content providers include traditional media providers such as radio and television networks, as well as enterprises and individuals who have the ability to publish their content on the Cloud. Although business content is still in the desktop-centric paradigm today, content users are far more Web-savvy than before. In the near future, the Cloud will free technology users from the limitations of their desktop, allowing them to share all types of information on the Web. The new business model is driven by two technologies. The first technology enables users to flow through their content in one place without special applications to manipulate the files. The second technology allows users to embed instantly viewable content on the Web. The combination of these two features allows content providers to share and embed all types of information. Moreover, content consumers can view the published information without having to download it and launch it in a desktop application.
- *Third-Party Value-Added Providers*: Value-chain partner integration, also known as partner integration, is essentially a form of Business to Business Integration (B2Bi). This modern value chain encompasses the automated exchange of information between different organizations such as partners, customers, suppliers, distributors, and others. For instance, the creation of full-service platform solutions for an enterprise may require Independent Software Vendors (ISVs) and the IT departments of system integrators to work together to develop and deliver an online application with third-party infrastructure services. High-end, third-party, value-added providers can offer services such as: (1) streamlining the transaction flows between value-chain partners, (2) monitoring partner performance and facilitating real-time decision-making, and (3) collaborating needs to integrate heterogeneous IT systems and business processes across partners. Figure 1.9 portrays the role of third-party providers in between the independent platform provider and various Cloud services [7].

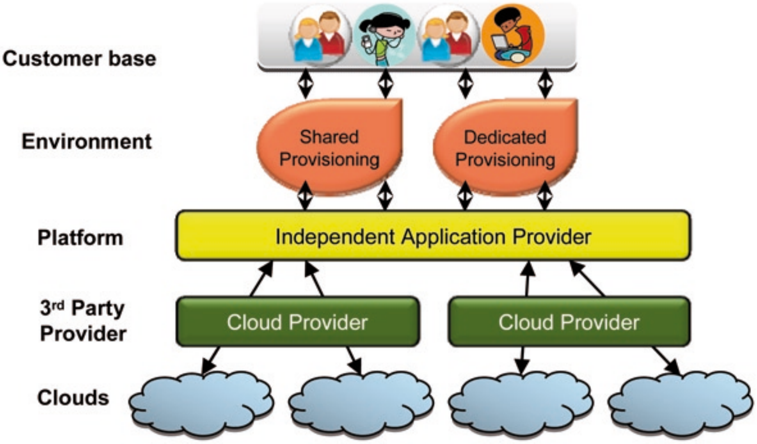


Fig. 1.9 The third party providers in a Cloud environment

- *Service Designers*: By following related industry best practices, such as the ITIL or TM Forum guidelines, service designers model services or offerings for their customers. A service designer can be a contractor of the enterprise or a design team in the enterprise's IT department. Service designers create the lifecycle of a service by utilizing the active service catalog and special order portal to orchestrate various components for formulating a Cloud service. Using the active service catalog, a centralized configuration database can be used to map the relationship between the new service and the required underlying infrastructure; these tools enable effective creation and rapid deployment of new services. The designer can then bundle together several preconfigured services to meet the provider's product portfolio and simplify the ordering process. To achieve optimized business performance, service designers must consider the following features in their design process: (1) services must be able to run in a multi-processor environment with parallel threads; (2) the readiness of cost implication analysis, such as the comparison of clients' usage patterns before and after the pay-per-use model; (3) mitigate solutions for mission-critical applications during service failures; and (4) the user friendliness of the service interface or portal for clients to manage their purchased services.

1.3 History of Cloud and Enterprise Services

To pave the way for future discussions about Cloud services, as well as enterprises' transformation in relation to technical, social, and economical aspects, let us first look at the historical context of Cloud and enterprise services. By examining the forerunners in the Cloud market and the problems they encountered, the results can assist enterprises in addressing challenges in the implementation, adoption, and management of their future Cloud service deployment more effectively. In this section, The concept of NCO from the U.S. Department of Defense (DoD) and its philosophical underpinning with Cloud services will be elaborated.

1.3.1 Initial Establishment

There are five generations of computers: the first generation of vacuum tubes in the 1940s and 1950s, the transistors in the early 1960s, the integrated circuits from the mid-1960s to early 1970s, today's microprocessors, and the future fifth generation, consisting of artificial intelligence and parallel processing. Likewise, the way of conducting computation is also entering a new era, with Cloud Computing being referred to as the fifth generation of computing. This fifth generation is evolving from the first generation of monolithic mainframe computing, client and server computing, *World Wide Web* (WWW) computing, *Service-Oriented Architecture* (SOA) computing, and finally Cloud Computing or service-based computing.

The idea of Cloud Computing originated from the early days of the Internet, where the network was drawn as a Cloud. A Cloud hid its internal process and the complexity of message propagation. From a Cloud user's perspective, there was no need to know where the message went, as long as it successfully entered one end of the Cloud and came out the other end. In its next evolution, when the Web concept was introduced, a user sent a *Uniform Resource Locator* (URL) to the Internet and the requested document could come back from anywhere in the world. There was no need to know where it was stored or who owned it. In the latest trend about utilizing Cloud Computing, it was compared to the electricity network from a century ago. Using this notion, private manufactures stopped producing their own power and plugged into a shared electricity grid, allowing computer users to connect to a Cloud of computing resources to conduct operations and run applications without having to install software or maintain any hardware.

As will be discussed in Sect. 1.4.2, *virtualization* is considered to be one of the key foundations of Cloud Computing. The concept of *Virtual Machines* (VMs) originated from Gerald J. Popek and Robert P. Goldberg's 1974 whitepaper on virtualization requirements [8] and is a means to support a computing environment that does not physically exist, but rather is created within another hosting environment. VMs are the first layer of abstraction in Cloud Computing. They represent a group of physical hardware and/or software and collectively establish an environment that allows different instructions to be executed [9]. Popular VM software includes common languages such as C, C++, Visual Basic, and Java VM. *Virtual Private Networks* (VPN) emerged in the telephony industry for data communications in the 1990s. A VPN service gives its end users the impression that they each consume dedicated channels with customer-centric guaranteed bandwidth. In reality, the virtual service rearranges the shared routers and switches underneath to balance the utilization as necessary.

The first commercial application of Cloud Computing was implemented in the early 1990s using the *Asynchronous Transfer Mode* (ATM) network. It was implemented by General Magic's founders, Bill Atkinson, Andy Hertzfeld, and Marc Porat, right before the Internet became massively popular in 1995. They developed a pre-*Personal Digital Assistant* (PDA) handheld device providing a fairly minimal UI, and a networked computing environment distributing computing resources across many machines in the network.

1.3.2 Early Developments

In the early 21st century, abstraction of services in the form of enterprise software became widespread. However, the understanding and usage of infrastructure and platforms in the Cloud were still limited [10]. Commercial companies, research labs, and universities started to contribute resources to mature and develop Cloud-based software applications.

In 1999, SalesForce.com offered an innovative, on-demand Cloud service for their *Customer Relationship Management* (CRM) solution. This offering was a huge success in the market, offering outstanding flexibility and adaptability for their customers. In the early 2000s, Microsoft extended the concept of *Software as a Service* (SaaS) through the development of its Web services and launched the Azure platform, hosted in its own datacenters. The Azure platform provided an OS and a set of developer services consumable either on premises or over the Internet [11]. In November 2007, Yahoo entered this market with a large-scale supercomputer for the academic research community, called the M45 project [12]. In March 2008, Yahoo and *Computational Research Laboratories* joined forces on Cloud Computing research and established the first large scale Cloud service system of its kind [13]. In July of the same year, Hewlett-Packard (HP), Intel, and Yahoo jointly announced a collaboration project to create a global, multi-datacenter, open source test bed for the advancement of Cloud Computing research and education [14]. In 2007, Google and IBM teamed up to build large datacenters to power a Grid Computing initiative. The goal was to provide a platform to help computer science students at research universities develop Cloud Computing applications hosted by large datacenters [15]. It was an ambitious $30 million, two-year project. Amazon launched *Amazon Web Services* (AWS) in 2006 to provide companies of various sizes on-demand computing power and storage, along with other services a business demands [16, 17]. In November 2008, Amazon announced a $100 million datacenter built in Boardman, Oregon [18].

On the academic front, Professor Ramnath K. Chellappa was the first to discuss the emergence of Cloud Computing driven by electronic commerce, and provided analysis for different roles and intermediaries enabling this framework [19]. Michael Armbrust and others from the University of California at Berkeley argued that the construction and operation of large-scale, commodity-computer datacenters at low-cost locations was the necessary enabler of Cloud Computing. They also identified the corresponding hardware, software, and operational challenges and limitations [20]. Campbell, Roy et al. from HP laboratory proposed a Cloud Computing test-bed called Open Cirrus, which featured research spanning systems, applications, services, open-source development, and datacenters. Open Cirrus was the first Cloud solution that adopted the federation of heterogeneous sites, systems, and application research and datasets. It instantiated the concept of sharing to implement an open stack with non-proprietary APIs for Cloud Computing [21–24].

1.3.3 Recent Major Developments

The computing industry is expanding its scope from software to platforms and infrastructure on the Cloud as virtualization technologies mature. New offerings take advantage of the technology that eliminates the constraints of location and time, allowing enterprises and SPs to now focus on their business values in the Cloud. The section will use some major players in the industry to showcase the recent development in the Cloud industry.

Google provides services such as search, e-mail, maps, office productivity tools (documents, spreadsheets, presentations, databases), social networking, and voice, video, and data services. These services are all delivered over network connections, where users can subscribe at no cost to the basic service or pay for increased levels of service. For example, the Google *App Engine* allows users to run any Web applications on Google's infrastructure. These applications are easy to build, maintain, and scale as the traffic and data storage vary. The Google App Engine supports applications written in several programming languages and has no set-up costs or recurring fees. The Google App Engine's Datastore provides a powerful, distributed data storage service and features a query engine and transactions that grow with the users' data. Unlike a traditional relational database, Datastore supports data objects and entities that have a type and unique set of properties. This enables more effective data queries. Its features are also strongly consistent with users' optimistic concurrency control: a method that allows multiple transactions to be completed without affecting each other, resulting in improved resource usage [25].

Amazon provides an array of remote computing services referred to as AWS. Specially targeted for Websites and client-side applications, the main products from AWS include *Simple Storage Services* (S3) and the *Elastic Compute Cloud* (EC2). The AWS S3 is developed intentionally with a minimal set of features, and is priced depending on usage with no minimum fee. The AWS EC2 allows its customers to have access and control of virtual computers to run their desired applications via a Web services interface. To use Amazon EC2, users simply need to select a pre-configured template image, configure security and network access preferences, choose instance types and an OS, determine multiple application running locations, and pay only for the resources that are actually consumed [26].

Microsoft offers office automation software and platform solutions for the Cloud industry. The Cloud-based office automation capability is called Office Live, which allows for synchronous and asynchronous integration of online Cloud documents with traditional offline desktop-resident versions. The platform solution is called Windows Azure Platform, which offers a flexible and Windows-friendly environment for developers to create their Cloud applications and services. The entire platform solution has three main components: the *Windows Azure* is an OS as a service, the *SQL Azure* is a relational database for the Cloud, and the *Azure AppFabric* is an efficient means to simplify the connection to either Cloud applications or client's on-premises applications [27].

As one of the pioneers of the SaaS model for distributing business software, the SalesForce.com solution is known primarily for two features: applications for sales and CRM services (i.e., sales Cloud and service Cloud respectively), and a Cloud platform for building and running business applications (i.e., Force.com). Force.com enables external developers to create add-on applications that can be integrated with the main SalesForce.com application hosted on the SalesForce.com's infrastructure. The main differentiators offered by SalesForce.com include services that allow their clients access to continually innovating software that is easily personalized, integrated, and deployed. The services provided by SalesForce.com do not provide a predefined, repeatable experience. Instead, they assist customers in building individual experiences through a constant exchange of ideas among the customer community that evolve customers' own solutions alongside other participants' business models [28].

VMware is a leading provider of virtualization software. Each VMware workstation contains a VM software suite that allows one physical machine to run multiple OS concurrently. In addition, VMware's Cloud solution, *vCloud*, promises to deliver a single way to run, manage, and secure applications wherever and whenever they are run. Today, VMware has partnered with hundreds of hosting and Cloud Computing vendors to enable flexible delivery on a common VMware platform, allowing easy transitions between providers. Furthermore, vCloud leverages mature technology, such as VMware *vSphere*, so users can achieve a high-degree of application assurance by using the Cloud-based management and monitoring features [29].

Standard bodies are driven by their different industrial members to create the needed guidance and common specifications to facilitate a variety of solution implementations, as well as catalysts to explore different Cloud options. For example, TM Forum formed the Enterprise Cloud Buyers Council (ECBC), intended to generalize the service procurement and acquisition process. Likewise, the Distributed Management Task Force (DMTF) developed the Open Virtualization Format (OVF) for packaging and distributing software that runs on VMs. The Open Grid Forum (OGF) created the Open Cloud Computing Interface (OCCI) Working Group to define a practical solution to interface with Cloud *Infrastructures as a Service* (IaaS). The Storage Networking Industry Association (SNIA) created the Cloud Storage Technical Working Group to develop SNIA Architecture related to system implementations of Cloud Storage technology. Details of these initiatives and accomplishments will be expounded upon in Chap. 3 [30–33].

1.3.4 Network-Centric Operations

The term *Network-Centricity* has represented different business and operational values in the realm of enterprise services. Efficiency of resource management via network-centricity has widely been regarded as a cornerstone of business assurance. As network technologies develop toward increasingly modularized and streamlined designs, the industry shifts again to a different paradigm—one in which single enterprises/providers are no longer capable of providing comprehensive product sets to satisfy every need of their target customers. In turn, this trend has driven enterprises, SPs, and network operators alike from network-centric (resource-centric) practices, toward service-centric business paradigms that reset the focus on QoS and *Service Level Management* (SLM). In the context of commercial service-provider operations, both network-centric and service-centric practices aim to support horizontal interoperability and efficiency.

The definition of network-centricity in different industries has different perspectives. For example, in the Defense industry, the concept of network-centricity is no longer about telecommunications networks or computer networking. Rather, it refers to an emerging body of organized behaviors pertaining to real-time information management, allowing users and systems to share insights and add value to a shareable knowledge community. Thus, this concept places more emphasis on the context of operations called NCO.

The goal of NCO is to increase cross-silo (different functional divisions at multiple levels) planning, operational intelligence, IT, and customer and management operations. It can be concluded as the following:

- Increasing reach among users and/or customers;
- Increasing the richness of information and expertise that can be applied to supporting operational decisions;
- Increasing agility in rapidly adapting information and IT; and
- Increasing assurance that the right information and resources to do the task will be there when and where required.

NCO provides the ability to coordinate complex missions over diversified operational environments, thus increasing synergy for superior decision-making. The main driver of enterprise information management concerns operational agility, describing the requisite levels of speed, cost-effectiveness, accuracy, and flexibility for organizational prosperity.

NCO has been gaining a lot of momentum and is considered one of the key concepts in the U.S. DoD’s plan for transforming the military. It is a theory that proposes that the application of Information Age concepts to speed communications and increased *Situational Awareness* (SA) through networking improves both the efficiency and effectiveness of military operations. In short, NCO seeks to translate an information advantage, enabled in part by IT, into a competitive warfighting advantage through the robust networking of well-informed, geographically dispersed forces. NCO increases the efficiency and effectiveness of operations

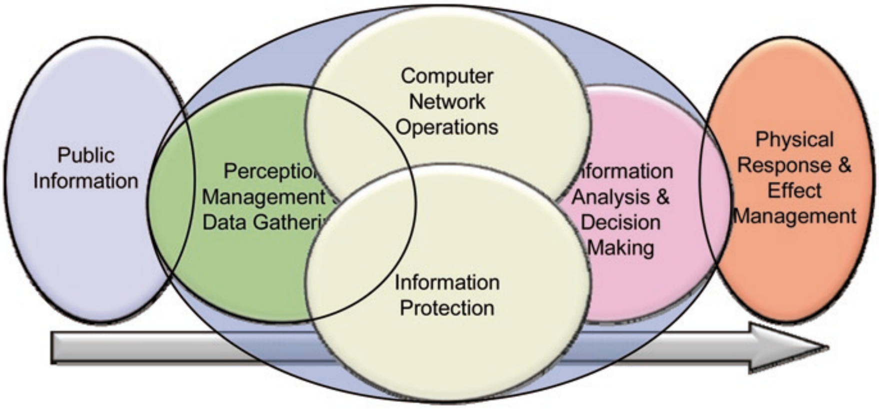

Public Information
Perception Management & Data Gathering
Computer Network Operations
Information Protection
Information Analysis & Decision Making
Physical Response & Effect Management

Fig. 1.10 Logical-to-physical horizontal view of the network-centric information flow

by establishing and facilitating shared SA through networked information management systems. Shared SA enables collaboration and self-synchronization, and enhances sustainability and speed of command. These in turn dramatically increase mission effectiveness. Figure 1.10 depicts a logical-to-physical view of the network-centric information flow. The control information travels from a conceptual space to a physical action space, and monitors information traveling from the physical action space back to the conceptual space. All of the network-centric operational values transform from left-to-right in the order of information, perception, cyber or network operations, information protection, electrical action control, and finally, physical action. Conversely, SA intelligence travels from right to left in Fig. 1.10 [34].

One of the applications of NCO used by the U.S. DoD and their allies in next generation battle applications allows mobilized soldiers to obtain and provide thorough warfare (*situational*) awareness in any place, at any time. NCO possesses collaboration and decentralization in the form of self-synchronizing forces, which can increase and/or improve awareness of the deployed applications. It enables excellent decision making, effective operations, and efficient process transformation. With this new capability, SPs can obtain a better understanding of both the big picture and the local situation than they currently have. When applying NCO concepts to resource management, resource-intensive applications can yield the advantage of moving information instead of moving material or people. These substitutions generate considerable savings in time and resources and therefore can result in increased impact in a given condition.

The transformation from independent systems to a coordinated and integrated *System of Systems* (SoS) is a continuous change process. *Net-Centric enterprise architecture* is an emerging military response to the Information Age and maturing military and technology capabilities. The formal definition of a *Net-Centric EA* is: "a light-weight, massively distributed, horizontally-applied architecture,

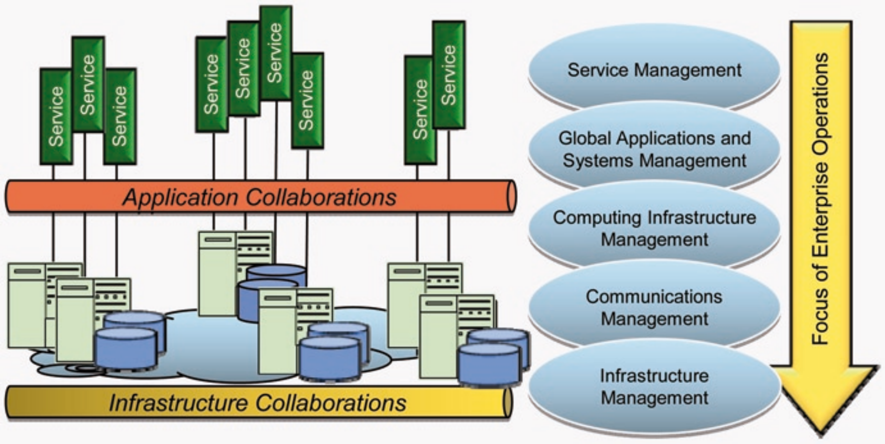

Service
Service
Service
Service
Service
Service
Service
Service
Service
Application Collaborations
Infrastructure Collaborations
Service Management
Global Applications and Systems Management
Computing Infrastructure Management
Communications Management
Infrastructure Management
Focus of Enterprise Operations

Fig. 1.11 Net-centric enterprise management architecture

that distributes components and/or services across an enterprise's information value chain using Internet Technologies and other Network Protocols as the principal mechanism for supporting the distribution and processing of information services" [35].

Cloud Computing implies new ways of providing capabilities and delivering computational resources on demand by use of virtualized resources. Some lessons for the Defense Department already are clear from industry experience with Cloud Computing. It implies far more agility in support of operational missions. More specifically, net-centric transformation uses information as a strategic asset and ensures an interoperable infrastructure, information access and security, and a good *return on investment* (ROI). In order to achieve these goals, two important constructs must be implemented or taken into consideration: *SOA* and *Web 2.0/3.0*.

Figure 1.11 summarizes the fundamentals of the Net-Centric enterprise services. Different infrastructures are bundled together to provide platforms, as different applications are bundled together to provide services. Each of the five layers of the management stack governs the enterprise operations of a specific virtualization level [36].

1.4 Cloud Enablers

Cloud Computing can be considered as a convergence of several key trends and concepts. Among those enablers, SOA enables networked applications to be available on demand, virtualization enables applications to be separated from underlying infrastructure, and Web technology enables content collaboration, as well as facilitates online community interactions. Detail discussion of these three subjects will be given in this section.

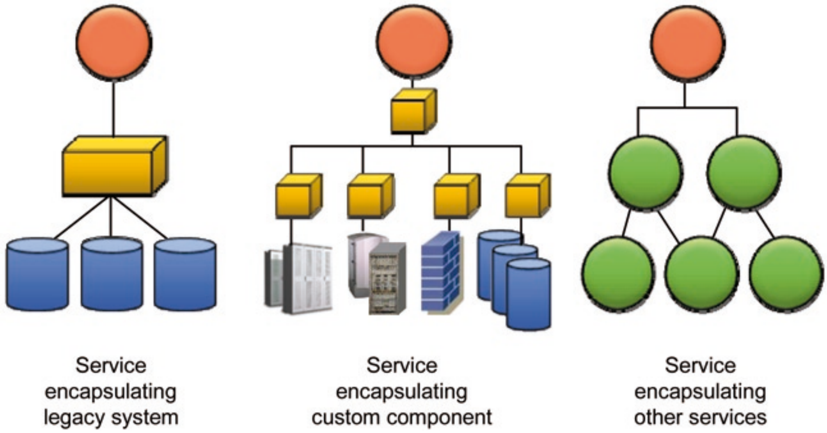


Fig. 1.12 Components of EA

1.4.1 Service Architecture and Abstraction

Abstraction ties into many aspects of service-orientation. On a fundamental level, this principle emphasizes the need to hide as much of the underlying details of a service as possible in order to enable and preserve the prescribed, loosely-coupled relationship. Service Abstraction also plays a significant role in the positioning and design of service compositions. Various forms of metadata come into the picture when assessing appropriate abstraction levels. The extent of abstraction applied can affect service definitions' granularity and thus influence the cost and effort for governing the service.

Service abstraction covers various forms and factors that involve service instances and operations. They can include service metadata, service processes and approaches, systems, and other computing resources. Figure 1.12 depicts the three possible service abstractions. From left to right: the service can encapsulate legacy systems for backwards compatibility; encapsulate custom systems, resources, or processes for resource virtualization; and/or encapsulate other services to construct a bundled service offering. These are all examples of SOAs.

1.4.1.1 Service-Oriented Architecture

The implementation of a SOA can be seen in a layered architecture. Figure 1.13 depicts the vertical slices and horizontal layers that fit into the enterprise service architecture framework and will be used throughout the book. The naming conventions and containing components adopted in this section are purposely altered to accommodate future discussions of Cloud services, therefore they may not be fully identical to the standard terms used in traditional SOAs [1].

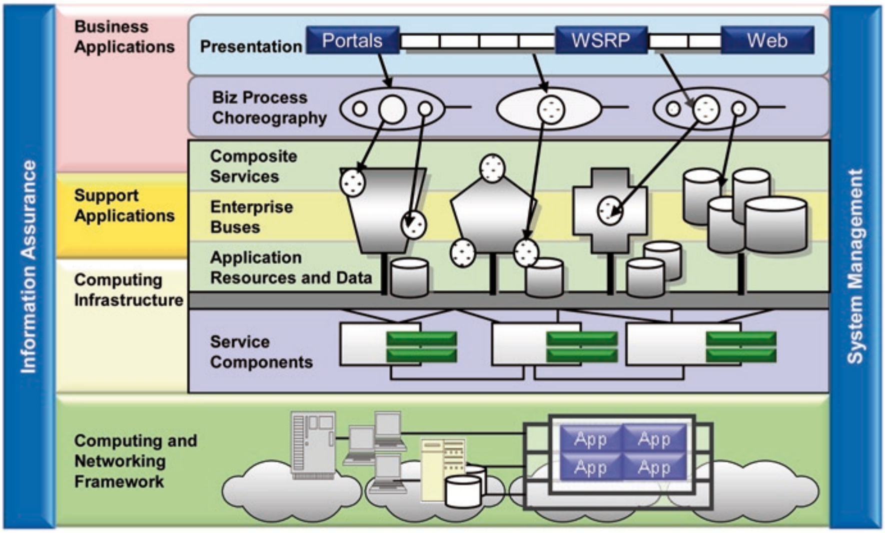


Fig. 1.13 SOA layers in enterprise services

The major SOA functionalities in the enterprise service architecture are as follows:

- *Business Application Layer*: This layer contains two major areas, namely service choreography and business presentations. Services are bundled into a flow through orchestration or choreography, and thus act together as a single application. Each application group supports specific use cases and business processes. The business presentations area bridges the UI to the grouped application in order to establish an end-to-end solution. This can be constructed in the form of a user graphical presentation or an access channel to a service or composition of services. An increasing convergence of standards, such as Web Services for *Remote Portlets Version 2.0* and other technologies, have started to leverage Web services at the application interface or presentation level.
- *Support Application Layer*: Composite services contain control and data flows that coordinate service invocation and data transfers among the different services to accomplish a particular task. A service composition is considered abstract until SPs are discovered and bound. Therefore, service compositions must handle issues relating to service discovery and service dynamics (e.g., self-adaptation). Application resources and data can be dynamically discovered or be statically bound and then invoked, or possibly, choreographed into a composite service. Service resources are exchanged through *Enterprise Service Buses* (ESB).
- *Computing Infrastructure Layer*: These service components are responsible for realizing functionality and maintaining the QoS of exposed services. These special components are a managed, governed set of enterprise assets that are funded at the enterprise or the business unit level. As enterprise-scale assets, they

are responsible for ensuring conformance to SLAs through the application of architectural best practices. This layer typically uses container-based technologies, such as application servers, to implement components, workload management, high-availability, and load balancing.
- *Computing and Networking Framework Layer*: This framework layer consists of package applications (e.g., CRM and Enterprise Resource Management) as well as computing hardware and communication facilities. The composite layered architecture of an SOA can leverage existing systems and integrate them using service-oriented integration techniques.
- *Information Assurance Slice*: This cross-layer function provides the capabilities required to monitor, manage, and maintain the integrity and security of offered services. Through sense-and-respond mechanisms, this background process and tools ensure end-to-end protection at the transaction and session levels.
- *System Management Slice*: This cross-layer function enables the integration of services through a set of capabilities, which cover service planning, insanitation, configuration, monitoring, testing, and reconfiguration. It covers the Web Services Management and other relevant communication and application managements sufficient to support any functionalities specified in the SOA.

The concept of the SOA architectural hierarchy depicted above is widely accepted as a computing paradigm and standardization of parts to realize actual business functions. To illustrate the relationship between standard SOA and the enterprise Cloud services, this model will be adopted throughout the book in order to maintain a consistent theme in different subject discussions, thus it should not be constrained by any SOA-specific descriptions.

1.4.1.2 Service Abstraction

Both SOA and Cloud Computing are service-oriented. To support a truly distributed computing environment, the abstraction of offered services is an essential feature of both solutions. From an implementation's perspective, *Service Abstraction* is considered as one of the eight main design principles of the service-orientation design paradigm.

These eight design principles are:

- *Service abstraction*: Service contracts contain only essential information; information about services is limited to what is published in the contract.
- *Standardized service contract*: Services within the same service inventory are in compliance with the same contract design standards.
- *Service loose coupling*: Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment.
- *Service reusability*: Services contain and express agnostic logic and can be positioned as reusable enterprise resources.
- *Service autonomy*: Services exercise a high level of control over their underlying runtime execution environment.

- *Service statelessness*: Services minimize resource consumption by deferring the management of state information when necessary.
- *Service discover-ability*: Services are supplemented with communicative metadata by which they can be effectively discovered and interpreted.
- *Service compose-ability*: Services are effective composition participants, regardless of the size and complexity of the composition.

This particular principle emphasizes the need to hide as much of the underlying details of a service as possible. Abstraction enables control of the underlying service logic that is exposed to the external world. Ensuring that service instances are designed in a generic fashion can enforce the integrated service to be more flexible in accommodating a large number of potential service requestors simultaneously. Therefore, such services can be better positioned as reusable IT assets.

As seen in the previous section, SOA is an architecture framework and its basic building blocks are functional primitives. By fully understanding the business objectives, information flow, and operational behavior, system integrators can group together different functional primitives to create an abstract layer of services with open service interfaces. For comparison purposes, SOA can be perceived as an IT solution architecture, while Cloud Computing is a way of implementing the architecture. This is because SOA is a more mature IT implementation architecture for IT to address service, data, and process collaborations based on a mesh of software services. While the Cloud concept was established and driven by virtualization technology, and is very similar to the service abstraction of SOA in principle, it has its own development path to address software, platform, and infrastructure needs. Despite the differences, it is unlikely that SOA will be replaced by Cloud Computing. Instead, SOA can feed enterprise service implementation experiences to the Cloud solution architecture, while the evolution of the Cloud will likely complement the SOA service abstraction with more modern virtualization technology. For instance, in order for an enterprise to move to Cloud Computing, functional primitives, software services, and the SOA interfaces have to be defined with a very clear understanding of what platforms, functionalities, and/or software are from the Cloud, and to which degree of complexity will be hidden by the Cloud.

1.4.2 Virtualization

Virtualization is a technology that separates an application from its underlying resources. This technology allows applications to be shared by multiple consumers without location or resource limitations. Platform virtualization makes OS-dependent applications more portable and scalable. Network virtualization facilitates better communication sharing and QoS assurance. Database virtualization improves data integrity and information sharing. Platform virtualization simplifies the development, packaging, and distribution of software images. Datacenter virtualization integrates all the virtualization technologies to provide a comprehensive IT operational environment. All these are key to successful Cloud service implementation.

1.4.2.1 Virtual Platform

Using a *host* software or a control program on an assigned hardware unit, platform virtualization can create a simulated computer environment for many *guest* software instances, as if they are running on a dedicated physical hardware unit. The virtualization host software determines, implements, and enforces hardware access policies for its guest software. Therefore, when running in such a simulated environment, guest software instances do not have any restrictions in accessing physical system resources such as display, keyboard, disk storage, network access, and so forth.

The virtual platform is a perfect solution for dealing with server consolidation, where many small servers are considered to be replaced by one large physical server to save cost and improve utilization. With the traditional method, the OS of each small physical server is impossible to consolidate like physical hardware. Using the virtual platform, individual OS can be converted into a distinct OS instance and can independently exist on a common VM. This is often referred to as the *Physical-to-Virtual* (P2V) transformation.

The benefits of virtual platforms are multifold. IT personnel can centralize the configuration, management, and monitoring of many applications running on a common platform through the virtual platform to improve operational efficiency and cost of space. Likewise, the flexibility of virtual platforms allow new VMs to be added to existing servers without additional hardware purchases. Additionally, errors occurring in guest software will not harm the host system or other guest software instances. Finally, virtual platforms offer great levels of portability, enabling VMs to be relocated to different sized computing resources to achieve better scalability.

1.4.2.2 Virtual Network

Network virtualization is a special application of the platform virtualization technology applied to networking. It is a software-based administrative method consisting of a combination of hardware and software network resources and functionalities. Network virtualization supports multiple simultaneous networks over a shared infrastructure. Each instance is then customized to meet different business needs. In a virtual network, the combined bandwidth is divided into independent and secured virtual channels to serve its targeted user, server, or device. There are two common forms of virtual networks: *protocol-based virtual networks* and *device-based virtual networks*. Examples of protocol-based virtual networks include:

- Virtual Local Area Network (VLAN) is a logical presentation of a local area network (LAN). A physical LAN can be partitioned into a number of VLANs or be grouped to form a VLAN. A VLAN can also be a VPN.
- VPN consists of multiple end-points that communicate with each other using tunnels over a third-party network. A Multipoint VPN is referred to as a network of multiple end-points that are inter-connected by a mesh of tunnels.

- Virtual Private LAN Service (VPLS) is a specific type of multipoint VPN. It allows geographically dispersed end-points to share an Ethernet broadcast domain.

By sharing network resources with other network consumers, network virtualization technology offers an efficient solution to address network utilization that often experiences sudden, large, and unexpected surges in usage. This technology also simplifies the management complexity of networks with centralized planning, fulfillment, and assurance mechanisms.

1.4.2.3 Virtual Database

The concept of a virtual database or federated database is based on a transparent grouping mechanism that accesses and manages heterogeneous physical databases using logical database references. Regardless of whether the data is local or remote, the group of fully-integrated physical databases are interconnected via a computer network. Virtual databases are grouped in a contrastable and federated manner to eliminate the labor, cost and, time required to physically merge these disparate databases. One of the most significant benefits of using the virtual database solution is its ability to be free from physical resource limitations. The uniform front-end UI with data abstraction can enable data users to store and retrieve their information with a single query, even if the constituent databases are heterogeneous.

The implementation of a true virtual database imposes a number of challenges. The federated database system must be able to decompose user queries into meaningful sub-queries that are relevant to the underlying constituent databases. The challenge also comes from the other direction when the returned data needs to be translated into a composite reply. Furthermore, various database management implementations may employ incompatible query languages, thus requiring additional translations or a wrapper mechanism to mediate interactions between them.

1.4.2.4 Virtual Application

Virtual Applications (vApps) imply that software images are executed in a VM. This concept is built upon the maturity of VMs, virtual platforms, and virtual networks, allowing software instances' existence on a virtual infrastructure. Coupled with *Just Enough Operating System* (JeOS), virtual application technology helps developers easily design, implement, deploy, and maintain their server-based applications in a virtual environment.

The development environment of vApps offers many new features to optimize the controllability of virtual application images. For instance, application users can take several simple steps to setup and configure software images, while application vendors can manage the images remotely. This simple yet efficient customer relationship allows application developers to streamline and distribute their products to the market more efficiently and quickly. Because the responsibility of software image *management* is owned by the users and the image is executed in a standardized virtual environment, it frees vendors from potential complex customer support efforts and allows them to focus on their product development. A direct result is the lower cost for software development and support, which in turns makes the products more attractive to customers.

1.4.3 Web Technologies

The term WWW (or the Web) was first introduced in 1990 by Tim Berners-Lee and Robert Cailliau in a proposal to build an interlinked hypertext document system called *WWW*. Berners-Lee had the vision of using the client-server architecture to satisfy *hypertext document* requests by a *browser*. He also explicitly expressed a long-term vision about “the creation of new links and new material by readers” and “the automatic notification of a reader when new material of interest to him/her has become available.” This long-term vision precisely identified the development path of Web development. In this context, the Internet is a global hardware system consisting of interconnected computer networks, whereas the Web is a service running on the Internet [37].

This section will illustrate the brief histories of three Web generations and their special characteristics. Figure 1.14 portrays these three generations, their operational differences, and the user populations in three distinct phases [38].

1.4.3.1 Web 1.0

Web 1.0 is an information portal. It refers to the time before 2001 during the bursting of the dot-com bubble, which was considered a turning point for the Internet. In the Web 1.0 generation, information was owned exclusively by the individuals and organizations who knew the technology. The operational model closely resembled the existing mindset of how information in the form of knowledge or products could be transferred from one entity to many others. The information was mainly kept as read-only content and was presented as static *HyperText Markup Language* (HTML) WebPages. Information users navigated Web pages through linked directories. Providers emphasized the number of views per page, with cost per click as the main business revenue. This is shown in the Web 1.0 picture in Fig. 1.14.

Although servers could be located anywhere in the Internet, SPs mainly used Web technology to reach out to their target clients with one-way media. As a result of this one-way, client-and-server operational model, Web 1.0 lacked flexibility in client interactions and service scalability.

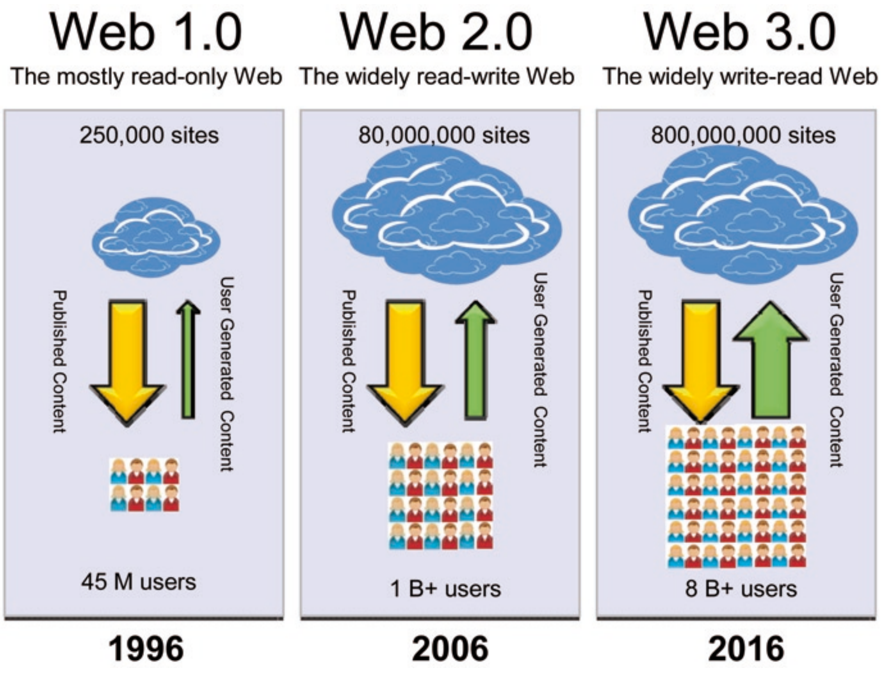


Fig. 1.14 Evolution of web technology

1.4.3.2 Web 2.0

The Web evolution was provoked by technology advances such as high-speed broadband connectivity and improved intelligence of browsers, as well as business model enhancements. The traditional Web 1.0 business model built a great big Website and hoped people would visit it. In Web 2.0, the network was perceived to be a business platform where users could add value on Web pages to customize the content.

The idea of Web 2.0 emphasized and promoted creativity, global information sharing, and collaboration. It originated in a conference brainstorming session between Tim O'Reilly and MediaLive International. The term Web 2.0 became notable after O'Reilly's *O'Reilly Media Web 2.0* conference in 2004. According to O'Reilly, "Web 2.0 is the business revolution in the computer industry caused by the move to the Internet as [a] platform, and an attempt to understand the rules for success on that new platform." Although not much detail was illustrated, the later development of Web 2.0 turned this marketing catch phrase into more realistic service implementations [39].

As depicted in Fig. 1.15, Web 2.0 established a more comprehensive architecture. As shown, it contains the client application, synchronization, business service, integration, and resource tiers. In addition, it also includes development and governance tools that cut through the upper four tiers. Using this architecture, many SPs can work together to offer features to their community-based clients to connect,

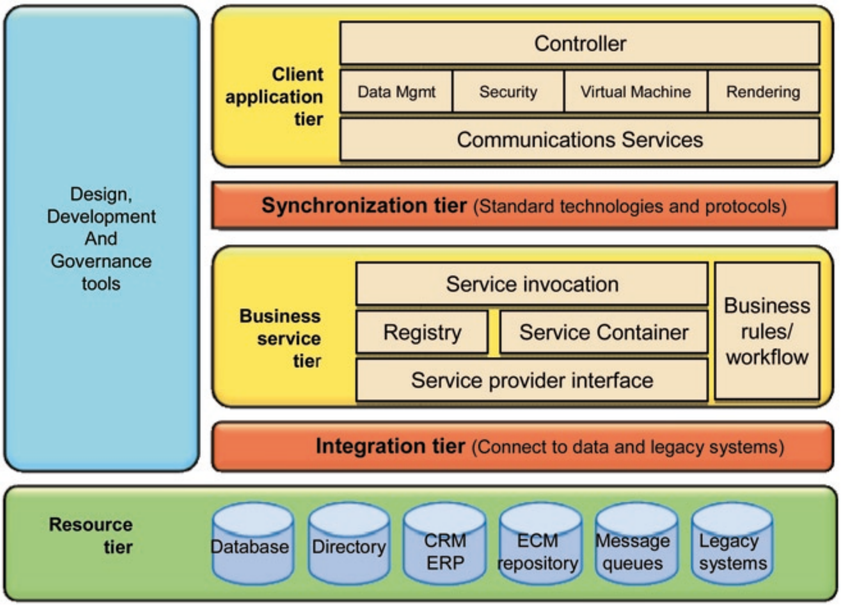


Fig. 1.15 Web as a platform

share, collaborate, and contribute data with improved integration and interaction technologies.

Owning to the philosophy of the Web as a platform for participation and engagement, Web services are components of online functionalities that are developed and associated with integrated online offerings. The concept of virtualization used in Web 2.0 removes the boundary of information and blurs data ownership. On the user front, vendors are exploring a business model where basic applications and services are free, with charges for service support, customization, or association. This model may impact smaller-scaled companies' current online strategy, forcing them to change in order to stay competitive. For software vendors, content creation tools must be made much simpler, user-friendlier, and more interactive in order to encourage more people to participate in information creation [40].

Web 2.0 is particularly important to enterprises. The phrase *Enterprise Web 2.0* refers to implementing Web 2.0 technologies within an enterprise. It has the potential to reshape enterprise software inside, outside, and across the firewall, with increasingly interactive, online applications for two-way flows of business information. The social engagement can potentially expand intra- or inter-enterprise *Community of Interests* (CoIs) for more effective value-chain collaborations.

1.4.3.3 Web 3.0

Although still under development, it is commonly agreed that Web 3.0 will have semantic Web, personalization, intelligent search, and behavioral advertising, among

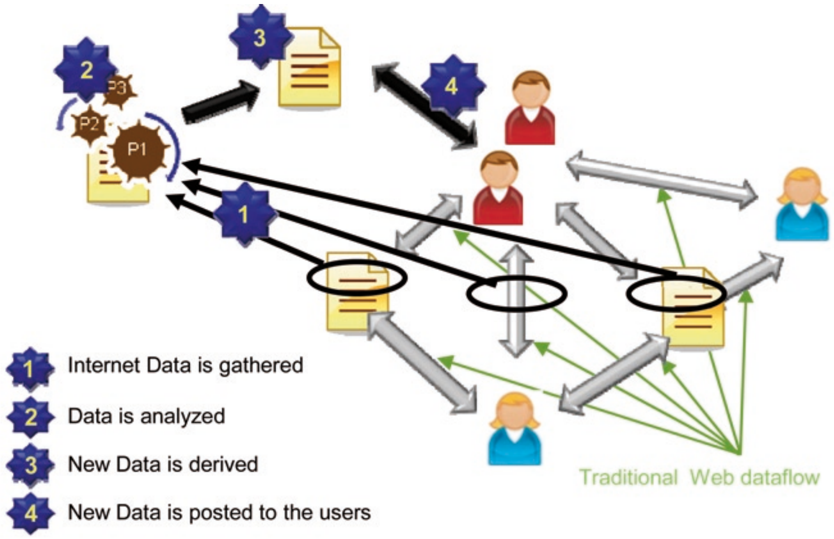


Fig. 1.16 Web as a smart machine

other new features. The term *semantic* implies that the meaning of information and services can be understood by the Web, and thus the Web can provide a sensible reply to the information or service requester.

Web 3.0 distinguishes itself from Web 2.0 by how information is grouped, organized, and provided to consumers. As shown in Fig. 1.16, semantic Web is *a Web of data* which changes the Web (interactions and content) into a language that can be read and categorized by systems rather than humans. A *smart machine* performs automatic information pulling and customizes Web content, including its look and feel in accordance with the preference of the users [41].

This *smart machine* is yet to be finalized due to its ongoing development, however, it is seen as a set of design principles and a variety of enabling technologies. The enabling principles include many defined specifications, including the *Resource Description Framework* (RDF), RDF Schema (RDFS), the *Web Ontology Language* (OWL), and data interchange formats (e.g., RDF/XML, N3, Turtle, N-Triples) that will be further discussed in the following chapters.

1.4.4 Key Cloud Characteristics

Cloud Computing started as a technology evolution, its economic advantages grasped the attention of the traditional IT industry and quickly grew into mainstream business applications, due to many special characteristics that are very relevant to end consumers' usage behaviors. In this section, these characteristics will be shown from different perspectives.

To begin, Cloud services bring enterprises a convergence of modern Web-based technologies and economic developments. From an IT system and architecture perspective, Cloud solutions offer the following special attributes:

- *Service-oriented*: Cloud services can be delivered over the network. The service components are componentized, pluggable, composable, and loosely coupled. Using a standardized framework, the lifecycle management of Cloud services can be greatly simplified.
- *Resource pooling*: Service virtualization facilitates shared resources and costs among a large pool of SCs. It supports a multi-tenancy environment and allows for centralization of infrastructure for lower operational costs.
- *Centralization vs. federation*: Cloud configurations can be deployed to a centralized organization or be distributed among different locations that demand certain degrees of collaboration. Flexibility is established upon a set of Cloud-specific guidance for interoperability.
- *Security*: Cloud technology provides infrastructure-level oversight of security and thus, theoretically can outperform individual security implementations. Although a Cloud can enforce stronger endpoint security and better data protection, the sharing environment can be vulnerable due to its open nature as well.

Cloud-based services provide many new operational and management features that benefit their providers as well as SCs. The implications of these new features are both operational and economical.

- *Low ongoing costs*: Overall cost for maintaining service offerings is reduced because multiple customers share the same resources in the Cloud using virtualization technology.
- *Low barrier to entry*: Because services are componentized, acquisition, installation, and provisioning of service components can be realized on-demand. This can greatly lower the initial capital investment for enterprises who wish to create sizeable, virtual IT environments.
- *Scalability and performance*: Cloud customers can scale their services using high-level management tools. This will help them execute faster and allow for easier acquisitions to create new offerings for their market. As resources can be added anytime, anywhere around the world to keep the assets closer to the end users, Cloud users can provide better scalability to meet their users' demands more quickly.
- *Reliability*: For mission critical applications that require better continuity of operations and disaster recovery, Cloud users can enhance the reliability by offering services from multiple redundant sites.
- *Security*: Cloud services offer better overall security protection due to centralized data and increased security-focused resources. However, the industry needs a common solution to manage loss of control over certain sensitive data.

From a user and consumer perspective, Cloud services offer many evolving new applications, such as ubiquitous network access that is device- and location-independent. Thus, users can access a Cloud application regardless of their locality or what device they are using. The details of these important attributes of the Cloud technology and business paradigm are:

- *Pay-as-you-go model*: Service customers can start small and ramp up as required using the pay-as-you-go model. With the trend of soaring computing resources, enterprises budget their IT costs from fixed, in-house to outsourced Cloud providers. This allows customers to pay for capabilities on a subscription basis. Customers no longer need to engineer their resources for peak loads. Instead, Cloud users can pass this responsibility to their SPs.
- *Device independence*: User facing devices and their supporting frameworks are separated from a layer of service abstraction using virtualization technology. Such a design improves the portability of applications, allowing different vendor products to be executed on the same device. It also improves reliability in case of service exceptions, such as natural disasters or power outages.
- *Location independence*: Cloud services are built upon virtual networks, therefore service users can access their purchased infrastructure, platforms, development environments, software, or hardware from anywhere in the world.

The above list introduces some sample features that Cloud services can bring to their customers. Some of the above features are built upon different technologies and driven by various business needs. These will be examined more closely in the following chapters.

1.5 Enterprise Transformation

Following the definitions of enterprise and EA in Sect. 1.0, this section will touch upon the fundamental framework of enterprise transformation and certain related key considerations. Content from future chapters will fill in the details with respect to people, organizations, processes, and technology domains mentioned earlier.

The business drivers for enterprises to choose the path of transformation can come from different reasons. In other words, enterprises recognize and anticipate business value deficiencies from their existing operations and motivate them to adopt new technologies and business models to remediate these deficiencies. The business value deficiencies that drive transformation can be classified in the following four areas:

- New opportunities for the enterprise, from changes in the business environment to the introduction of a new technology.
- Threats to the enterprise market share due to market or technology changes.
- Successful transformation of competitors prompts recognition of the need to change.
- Business performance degradation triggers the need to change to survive.

By leveraging business and technology opportunities relevant to Cloud Computing, enterprises can take advantage of the transformation to improve the success of

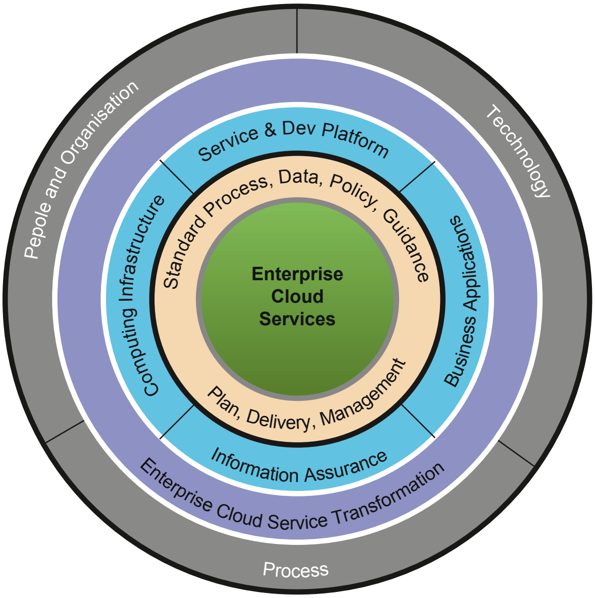

Pepole and Organisation
Tecchnology
Process
Enterprise Cloud Service Transformation
Service & Dev Platform
Computing Infrastructure
Business Applications
Information Assurance
Standard Process, Data, Policy, Guidance
Plan, Delivery, Management
Enterprise Cloud Services

Fig. 1.17 Elements of enterprise transformation

strategic alliances, implement specific strategies with this effort on enterprise datacenters and other channels, or maximize the value of their existing assets. However, regardless of the driving reasons for the transformation, enterprises should keep a clear business strategy in mind to make sure their efforts have executable objectives. These strategies can pursue global markets, such as emerging markets; pursue vertical markets, such as financial or defense; expand new Cloud-based sales channels; or improve their business values by offering integrated services or products. Figure 1.17 depicts the relationship between these four areas and enterprise Cloud services.

1.5.1 People and Organization

Transformation in the context of people and organizations refers to aligning the enterprise business strategy and human resources with the new processes toward a common goal and ensuring that goal is met by focusing on the new business culture. Given clear business drivers identified in the business strategy, enterprises should prepare to adapt external variable changes and be ready to cultivate resources. To implement the changes effectively, leading employees must possess enough knowledge to generate awareness about how they perceive the new organizations' environment. There must also be appropriate guidance for increasing motivation to address challenges and strengthen team performance. The seeding team involved with the early phases of the transformation should start with increasing awareness and skills before establishing a formal force. The executing strategy should include the definition of roles and responsibilities, a skills gap assessment, and skills development plans. There are two primary areas to support the transformation in this category: *leadership and culture development and organizational improvement*.

- *Leadership and culture development*: Enterprises' corporate cultures have a dominating effect on the decision-making process of the transformation. The overall business direction typically dictates an enterprise's operating model and thus influences the direction of executions. Although most large enterprises have certain multiple dimensional matrices that address their line of business, market segment, geography, lines of finance, technology, and operations, the effectiveness of synchronizing these factors mainly depends upon the vision and performance of the leadership. Successful enterprises grow their internal resources to assume leadership, change management, coaching, and implementation roles. The level of the decision maker is very relevant. Leaders in different levels of operations in the enterprise must be capable of engaging large numbers of employees to work in new ways, achieving the speed and scale of change needed for success. Furthermore, enterprises need standard approaches or processes in order to manage the changes. Transformation requires a management culture that can manage migration, quality, results, and accountability throughout the period of changes, and even become a business norm. This in turn requires a project management mindset, including a project governance structure and process and integrated performance measures for executives. The program manager must have enough authority to deliver, and even override, any unsynchronized factors during the planning and execution phases. This will allow the agreed-on objectives and missions to be executed throughout the enterprise.
- *Organizational improvement*: To minimize risk and maximize the impact of the new organizational change, enterprises need to start by identifying the weak link where horizontal and vertical aspects of the organization intersect. In this context, the horizontal aspect implies shared services and the vertical aspect implies silo services. The weak link in this case applies to the adaptation of the Cloud business framework, and thus may not be explicitly identifiable from, or even applicable to, the existing business operations. The organization transformation strategy helps enterprises align their structure, processes, measures, performance management, culture, and people with the desired business objectives. The effort includes restructuring the organizations to shift resources and control core business processes, eliminating old-style functional silos, preparing for new Cloud business models, creating internal and external on-demand business organizations, and enhancing customer requirements. To obtain the most effective results, enterprises have to address their cultural issues by changing the behaviors and attitudes that created them. For continuing improvement of team effectiveness, governance policies and procedures are essential in assuring that results are in alignment with the business and that enterprises' service level commitments to their customers are guaranteed. The efficiency of the new organization and improved CRM will very likely motivate and enable desired behaviors among employees and managers.

1.5.2 Process

Process plays an essential role in enterprise transformation. An effective process strategy must include an integrated portfolio of operational guidance to facilitate integrated service planning, service fulfillment, service assurance, capital deployment, human asset leverage, sourcing, supply chain strategies, sales force effectiveness, marketing readiness, and new technology management. Specific to Cloud services, enterprise process improvements can be made to enhance the provisioning of an enterprise's virtual resources. Using standardized methodologies, service management functionalities, including capacity, configuration, and asset management, can be optimized and consolidated across different operational domains or silos. A cross-domain policy can help enterprises automate their processes and procedures in order to streamline and simplify their operations. An integrated life-cycle management framework with an SLA can improve the consistency of VM deployments, as well as enable comprehensive customer expectation management. Through virtualization of the financial management model, extended orchestration capabilities that incorporate performance metrics can help fuse organizational accountability with customer commitments in the overarching operational objectives of the enterprise. In addition, the following considerations, with respect to the enterprise process, are also key for a successful transformation:

- *Organizational and technological alignment*: Organizational alignment of operations and technology is key for efficient transformations. Through the new organizational arrangement, enterprises must take advantage of new Cloud technology to ensure their operational objectives can meet the business dynamics effectively.
- *Best practices*: In light of the new outsourcing of business operations, enterprises must work with leading vendors and standard bodies to adopt or even create best practices that are most suitable for their business and service offerings. Through a well defined process, all stakeholders in an enterprise's value-chain can work on a clear incentive toward a common interest where their contributions are complementary to a successful shared destiny.
- *Quality compliance*: Service agreements in the form of SLAs and *Operational Level Agreements* (OLAs) can provide the needed transparency in assuring service quality for service customers, as well as operational quality for inter-organizational support.

1.5.3 Technology

Technology strategy defines the role of IT in setting and enabling enterprises' business goals. The strategy should include, but not be limited to, the business case, strategic goals, technology business alignment, service architecture, and IT governance. One of the key principles in the technology domain of a SOE is its ability to address all of the functional areas that support the new technology. This includes the full life cycle of the service delivery related to the new technologies, the management capability to support the delivery, and policy and guidance requirements for the enterprises' industries. For instance, when delivering a virtualized computing resource to mission critical areas, SPs need to consider more stringent security features and regulations. They may also need to incorporate integrated management and performance monitoring tools to more effectively manage the environment. Moreover, including enhanced networking and storage resource management, coupled with a workload management across the production environment, can greatly improve the enterprise services' performance and scalability. The following list shows some examples that may be critical to an enterprise's technology transformation [42]:

- Recognition of relevant standards and technologies
- Evaluation of applicability
- Enterprise architecture
- Use case study and simulation
- Case development
- Sourcing strategy
- Portfolio management analysis
- Architecture engineering
- IT governance
- Application portfolio consolidation
- IT roadmap and directions
- Business aligned and realigned portfolios

In the effort of enterprise transformation, migration and execution are the keys to the success of the project. Migration management is the combination of a culture of flawless execution, scope management, and integration of business, operations, and technology decision making on one hand, and solid techniques in program management, design, component engineering, and vendor management on the other hand. Because the people and organization aspects are more application dependent, for instance the mission goals of gaming providers are different from government agencies, this book will mainly focus on the technology and process aspects that are common across different enterprises. When appropriate, key issues with respect to the corporate culture and organizational challenges will be inserted to relevant sections as case studies in order to help illustrate the solutions proposed in the book.

1.6 General Framework & Book Origination

To ensure an effective enterprise transformation that can justify executable actions to gain business value, plan for adoption, or develop good strategies for implementation, a framework for discussion is needed. Although they are many versions of Cloud architecture available in different industries, there are certain levels of

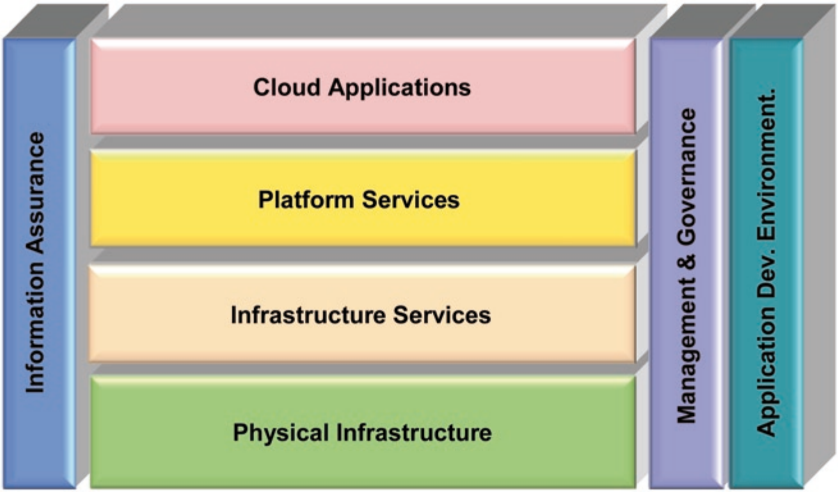


Fig. 1.18 Layered architecture of a Cloud platform

commonality and agreement. In this section, we intend to take advantage of the viewpoints that are commonly agreeable and in the mean time, beneficial for our transformation discussions.

Figure 1.18 shows a horizontal and vertical layered architecture of a Cloud service offering. It provides the relationship of different layers of services, the interface between layers in the service offering, and the supporting management framework. Depending upon the adaptation of different enterprises, there is no hard dependency between each layer. The layer relationship is determined by contractual interfaces between each other. Every layer provides a level of service abstraction from the others, making the horizontal or vertical interactions among layers possible using the plug-and-play method. These seven service layers are:

- *Application Service Layer*: This layer houses applications that are built for a Cloud environment. These applications are exposed to their end users via Web interfaces or Web Services that enable the multi-tenant hosting model.
- *Platform Service Layer*: Cloud platform services provide a set of capabilities exposed as services to assist Cloud users in developing, testing, integrating, and deploying their services. The services in this layer are integrated closely with the Security and Management layers to offer comprehensive enterprise grade products. Availability of platform services may differentiate one Cloud provider from another.
- *Infrastructure Service Layer*: This layer abstracts the platform and above services from the underlying computing, storage, and networking resources. It exposes the upper layers with a set of APIs, allowing service users to access and manage these resource abstractions based on the required scalability and availability specifications.

- *Physical Infrastructure Layer*: This layer houses hardware, firmware, and software resources that support the upper layers of Cloud services. These resources include computers, disk storages, routers, switches, cables, testing devices, monitoring devices, power supplies, antennas, sensors, wires, cables, and so forth. These resources occupy space and require personnel to operate and manage them.
- *Information Assurance Service Layer*: The Cloud Security services are responsible for ensuring token provisioning, identity federation, and claims transformation. These services are built upon open standards such as WS-Security, WS-Trust, WS-Federation, *Security Assertion Markup Language* (SAML) protocols, and OpenID.
- *Management and Governance Service Layer*: The Cloud Management and Governance Services cut across all the layers described above. They provide the data collection, analysis, and reporting functions that allow enterprises and their stakeholders (including Cloud users and service clients) to ensure QoS meets the SLAs, OLAs, or the policies and rules required by industry-specific regulations.
- *Application Development Environment*: A set of tools, functions, and procedures that can assist Cloud users in designing, developing, testing, integrating, and deploying a Cloud-based feature to their clients. It also includes collaboration functionalities for stakeholders in the Cloud ecosystem to participate in Cloud service development for all phases of the service lifecycle. The community-based development approach can produce products that more closely match the clients' needs. Its byproducts can be a form of standard specifications that are reusable in other service development events for the same or different industries.

The business scenarios and market analysis of Cloud services will be in examined Chap. 2. In that chapter, the applications and market size of a Cloud in the general IT industry, the commercial industry, the government and defense industry, and the scientific and education industries are elaborated. In Chap. 3, more detailed architectural-level subjects are expounded on, including key technical issues and relevant industry standards. Challenges for enterprises to adopt Cloud technologies in order to complete the transformation of their core business models and technologies are illustrated in Chap. 4. The chapter offers useful insights into the non-technical and technical issues, preparing enterprise stakeholders to get ready for their cultural and leadership transformations, as well as process and technology changes that were discussed in Sect. 1.5. Chapter 5 discusses networked service management. It offers collaborated management view points for how different Cloud users can make significant contributions to improve their business impacts to their targeted value-chain and CoIs. In Chap. 6, a standards-based, *Policy-Based Management* (PBM) solution is illustrated with special emphasis on cross-Cloud service coordination and negotiation. This revolutionary policy framework, built for space communications, is seen as a practical solution for enterprise policy management. Chapters 7–9 discuss best practices from different industries for service planning, fulfillment, assurance, and billing suitable for enterprise Cloud services. These practices are extracted from various sources and organized to convey an executable transformation path. Special attention is placed upon Cloud security to ensure that proposed solutions are effective and assured. This book will be concluded with a set of clear paths for enterprise strategy and execution guidance, with transformation discussions from software, platform, infrastructure, management, and security perspectives.

References

1. Chang, W.Y.: Network-centric service oriented enterprise. Springer, Netherlands (2007)
2. The Open Group, TOGAF 8, The Open Group Architecture Framework Enterprise Edition, Document Nr. 1911, The Open Group. Dec 2002. URL: http://www.opengroup.org/architecture/togaf/
3. GB921 Business Process Framework Suite, Release 8.1, TM Forum. http://www.tmforum.org/Guidebooks/GB921BusinessProcess/41516/article.html (2010). 11 March 2010
4. Definition of Cloud computing, incorporating NIST and G-Cloud views, Kate. http://www.katescomment.com/definition-of-Cloud-computing-nist-g-cloud/(2010). 24 Feb 2010
5. Cloud computing, NIST. http://csrc.nist.gov/groups/SNS/cloud-computing/
6. Kolke, D.: The future of work: distributed tenant hosting over multi-tenant hosting, ETELOS. http://v3.etelos.com/dm/article.espx?show=12598&cc=1 (2007). June 09 2007
7. Pattern: Cloud Computing, OSA. http://www.opensecurityarchitecture.org/cms/library/patternlandscape/251-pattern-cloud-computing
8. Popek, G.J., Goldberg, R.P.: Formal requirements for virtualizable third generation architectures. Commun. ACM **17**(7), 412–421 (1974)
9. Varian, M.: VM and the VM community, past present, and future, SHARE 89, Sessions 9059–9061. (1977). http://www.princeton.edu/~melinda
10. Markoff, J.: Internet critic takes on microsoft. The New York Times. 9 April 2001
11. Windows Azure platform: Microsoft 2010. http://www.microsoft.com/azure/whatisazure.mspx
12. Yahoo! Reaches for the Stars with M45 Supercomputing Project, Yahoo, 2010. http://research.yahoo.com/node/1884
13. Yahoo! and Computational Research Labratories to Collaborate on Cloud Computing Research,Yahoo. http://research.yahoo.com/node/2039 (2008). 24 March 2008
14. HP, Intel and Yahoo! Create Global Cloud Computing Research Test Bed, Yahoo. http://research.yahoo.com/news/2328 (2008). 29 July 2008
15. Miller, R.: Google, IBM team on data center research, Data Center Knowledge. http://www.datacenterknowledge.com/archives/2007/10/08/google-ibm-team-on-data-center-research (2007). 8 Oct 2007
16. Amazon.com's Two-Pizza Team Rule, Learning API. http://www.learningapi.com/blog/archives/000079.html (2005). 10 Nov 2005
17. What is AWS? Amazon, 2010. http://aws.amazon.com/what-is-aws
18. Miller, R. Amazon building large data center in Oregon, Data Center Knowledge. http://www.datacenterknowledge.com/archives/2008/11/07/amazon-building-large-data-center-in-oregon/ (2008). 7 Nov 2008
19. Chellappa, R. Intermediaries in cloud-computing: A new computing paradigm. INFORMS Dallas, Cluster: Electronic Commerce (1997)
20. Armbrust, M., Fox, A., Griffith, R., Joseph, A,D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Electrical Engineering and Computer Sciences, University of California at Berkeley, Technical Report No. UCB/EECS-2009-28. 10 Feb (2009)

21. Campbell, R., Gupta, I., Heath, M., Ko, S.Y., Kozuch, M., Kunze, M., Kwan, T., Lai, K., Lee, H.Y., Lyons, M., Milojicic, D., O'Hallaron, D., Soh, Y.C.: Open cirrus: cloud computing testbed: federated data centers for open source systems and services research, HP Laboratories, HPL-2009-134. (2009)
22. Allcock, B., Bester, J., Bresnahan, J., Chervenak, A.L., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnel, D., Tuecke, S.: Data management and transfer in high-performance computational grid environments. Parallel Comput. **28**(5), (2002)
23. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system.In: 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 124–131 (2009)
24. Vouk, M.A.: Cloud computing–issues, research and implementations, Information Technology Interfaces (2008)
25. What Is Google App Engine? Google, http://code.google.com/appengine/docs/whatisgoogleappengine.html (2010)
26. Amazon Web Services: Amazon. http://aws.amazon.com/ (2010)
27. Windows Azure platform, Microsoft. http://www.microsoft.com/windowsazure/ (2010)
28. CRM Applications & Software Solutionssalesforce.com. http://www.salesforce.com/crm/ (2010)
29. VMware Virtualization Software. VMware, http://www.vmware.com/ (2010)
30. Enterprise Cloud Buyers Council. TM Forum, http://www.tmforum.org/EnterpriseCloudBuyers/8009/home.html (2010)
31. Enabling Portability & Simplified Deployment of Virtual Appliances: Open Virtualization Format (OVF)—DMTF Standard for packaging and distributing virtual appliances, DMTF, Sept 2008
32. Open Cloud Computing Interface WG (OCCI-WG). Open Grid Forum (OGF). http://www.ogf.org/gf/group_info/view.php?group=occi-wg
33. Cloud Data Management Interface (CDMI). SNIA. http://cdmi.sniacloud.com/ (2010). 12 April 2010
34. Protecting the homeland, report of the defense science board task force, defensive information operations. 2000 Summer study, Vol. 2. March 2001. http://cryptome.sabotage.org/dio/dio.htm (2001)
35. Net-centric. http://en.wikipedia.org/wiki/Net-centric
36. Vietmeyer, R.: Net-Centric Enterprise Services, NCES Engineering DISA. http://www.opengroup.org/gesforum/uploads/40/5530/NCES_Update.ppt
37. Berners-Lee, T., Cailliau, R.: WorldWideWeb: Proposal for a HyperText Project. http://www.w3.org/Proposal (1990). 12 Nov 1990
38. Ciccarelli, D.: Web 2.0 Definition. http://blogs.voices.com/thebiz/2006/09/web_20_definition.html (2006). 19 Sept 2006
39. Stallman, R.: quoted in The Guardian. 29 Sept, 2008
40. Governor, J., Nickull, D., Hinchcliffe, D. Specific patterns of Web 2.0: Chapter 7—Web 2.0 Architectures, O'Reilly. http://oreilly.com/web2/excerpts/9780596514433/specific-patterns-web20.html
41. The future is smart machines. NESTA. http://blogs.nesta.org.uk/innovation/2007/07/the-future-is-s.html (2007). 25 July 2007
42. Schekkerman, J.: Enterprise architecture validation: achieving business-aligned and validated enterprise architectures. Aug 2004, http://www.enterprise-architecture.info/Images/Extended%20Enterprise/Enterprise%20Architecture%20Validation%20Full%20version.pdf

Chapter 2
Cloud Service Business Scenarios and Market Analysis[1]

The introduction of Cloud services is altering the way enterprises build their infrastructure and applications. Lower deployment costs, easier market entry, faster payback on new services, and expected higher ROI will make the Cloud-based environment a top choice for big and small service developers. With this new tool, small companies, even individuals, can leverage a large amount of resources and capabilities with a relatively small investment. This change presents a completely new business model and unprecedented opportunities for small and big corporations to compete in the current IT frontline. It is inevitable that global growth trends in service development will increase the importance of high-leverage application frameworks, enabling more rapid changes to higher-quality services.

The essential characteristics of Cloud services lay with its service delivery model and the related supply-chain business model. Today's service delivery expects service offerings to be available and accessible anytime, anywhere, and by any authorized user or system. Within a supply-chain, many SPs can assemble a collection of bounded services that require a multiple-tier business agreement and contract to facilitate an appropriate supplier and consumer relationship.

Established with a foundation of two bases, a business model (on-demand IT resources) and a set of technologies (massively scalable, highly resilient architectures), it is crucial to address the business cases and service applications of enterprise Cloud services to capture the essence of the drivers identified in the previous chapter. Wide-range Cloud services must be proven by practical use cases illustrated in the following section. The market analysis section will cover both the commercial and government sectors.

2.1 Overview

Cloud technology is changing the way enterprises build their infrastructure and applications. It offers an extraordinary opportunity for enterprises to focus on their core capabilities by outsourcing certain aspects of IT and reducing other IT costs. This new method accelerates provisioning and deployment by transferring them

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_2, © Springer Science+Business Media B.V. 2010

to the Cloud SP via the Internet and making them accessible either from a Web browser or as a Web service.

The domain of Cloud services contains a business model for on-demand IT resources and a set of technologies that addresses massively scalable, highly resilient architectures. With the Cloud-based business model, enterprise users pay for service usage and reduce the overall maintenance effort and user costs. Cloud technology can include server virtualization, Web Security, and Web Services. It allows the enterprises to establish scalable infrastructure dynamically and make their services available transparently without having to deal with IT infrastructure development and management.

The main objectives of this chapter are to illustrate the business scenarios of Cloud services and analyze related domain applications. This chapter will begin by offering a high-level description of Cloud use cases and applications. It will then provide market analysis in the general IT, commercial, and government and defense markets. While market size information can reveal implications of Cloud services, the available data is typically subjective and often comes with many assumptions that potentially delude the main focus. Furthermore, given that Cloud technology is still evolving, the market applications and adaptations will not be easily captured

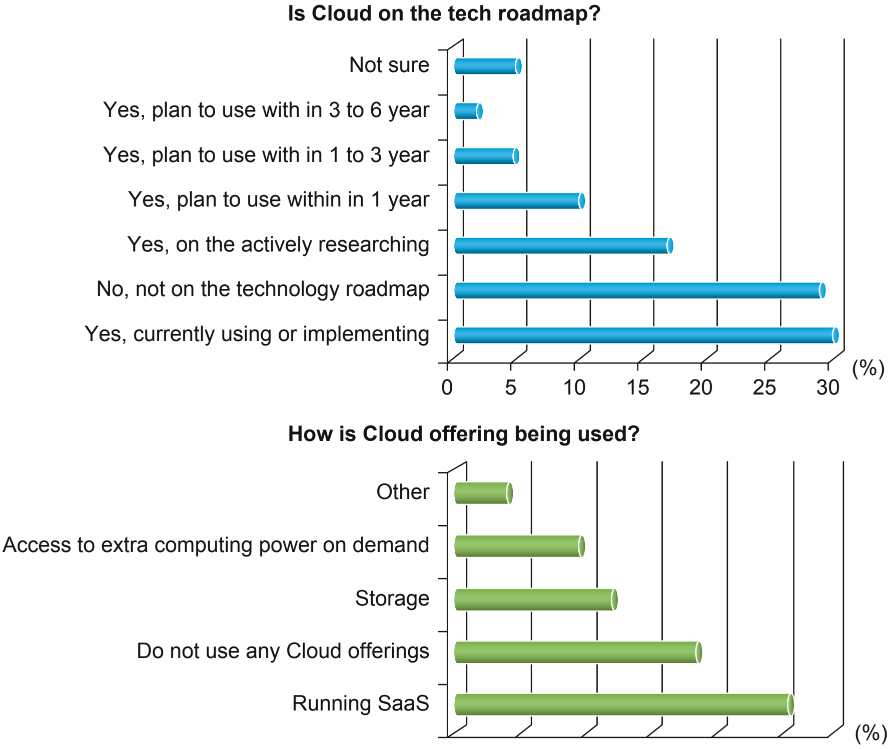


Fig. 2.1 Cloud adoptability survey

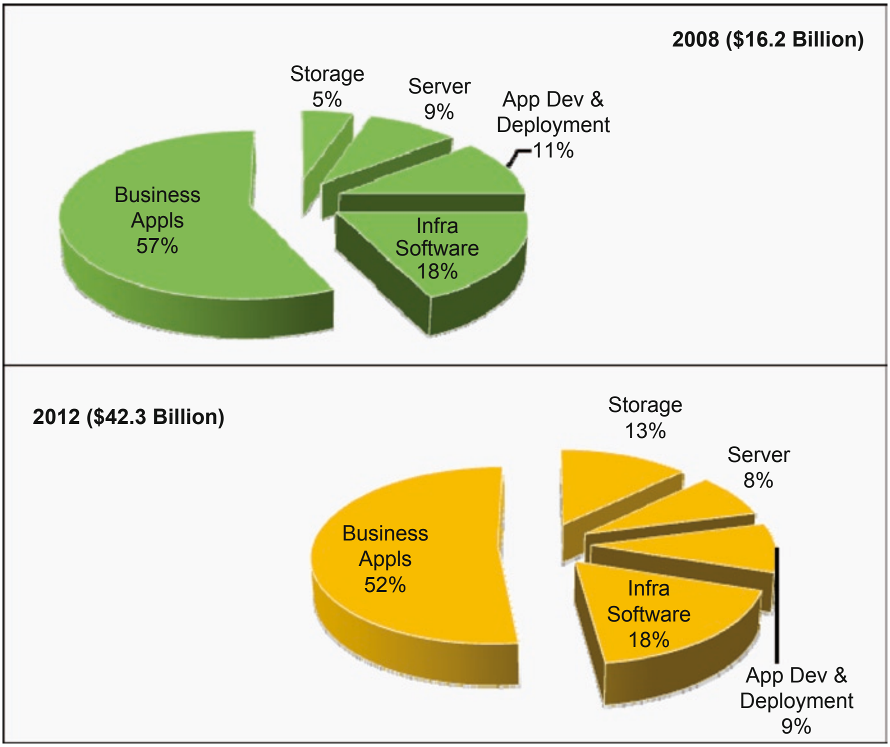


Fig. 2.2 IT Cloud service spending trend

with a quantifiable number. This chapter will focus on discussing why they are important and how they are used in different industries.

According to a recent survey, a significant number of companies indicated that they are using Cloud technology. Out of the available technologies, SaaS is currently dominating the Cloud marketplace (Figs. 2.1 and 2.2) with more than 50% adoption. Today, there are millions of highly distributed computing nodes using Web-based and virtualized services driven by Cloud technology. This fact testifies to the successful model of Cloud services and provides a crucial hint as to where the near future of IT transformation will lie with an increasing emphasis on standardized, federated services versus proprietary, centralized services [1].

2.2 Cloud Use Cases and Applications

Cloud use cases surround the values of enabling convenient, on-demand access to a shared pool of configurable computing resources provided by the Cloud SPs. The pooled resources include networks, servers, storage, applications, and management services that can be rapidly provisioned and released with minimal effort or SP interaction. Furthermore, to provide enterprise users appropriate levels of manageability of the services that are operated by third-party providers, Cloud technology has also evolved with more automatically controlled and optimized resources. This allows enterprise clients to leverage a metering capability at the appropriate abstraction level to monitor, control, and report utilized services. There are four types of Cloud services available right now: *public*, *community*, *private*, *and hybrid*. Each represents a unique business case and is illustrated in the following sections [2].

2.2.1 Public Cloud

The Public Cloud offers open services to the general public and is owned or operated by an organization selling these services. In a Public Cloud, the services are delivered over the Internet via Web applications or Web services. All resources are on a self-serve basis and are normally provisioned dynamically. Services are billed based on utilization by an off-site provider. With existing SOA and Web Services, enterprises can integrate the Public Cloud as an extension of their enterprise IT architecture.

Although the majority of SPs offer their software products to the Public Cloud and have thus gained a lot of traction, some providers have chosen to either provide a system development environment or make their underlying infrastructure available as a service. There are various models from horizontal VMs, to vertical programming models, to horizontal resource allocations. The challenge for the adaptation of each model is how to facilitate mass adoption so that users can use them easily and simply.

There are three scenarios for the Public Cloud configuration. As depicted in Fig. 2.3, the first scenario depicts end users' access to Cloud applications running

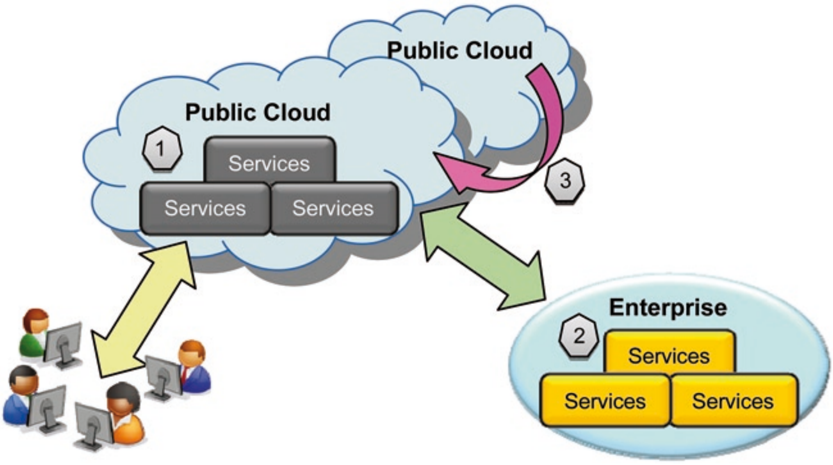


Fig. 2.3 Public Cloud

on the Public Cloud. In the second scenario, enterprise applications are running in the Public Cloud and are accessible by employees and customers. The third scenario depicts a situation where an enterprise switches Cloud providers or works with additional providers.

2.2.2 Community Cloud

The Community Cloud is designed to be shared by several organizations in order to support specific community objectives or shared common interests. For instance, a standard forum assists members in collaborating opinions in mission requirements, policy, and compliance considerations to develop a set of new service specifications. A Community Cloud may encapsulate multiple local and remote resources to appear as a single homogeneous service environment, bridging the ability to utilize these available resources. This type of Cloud may be managed by the organizations themselves or by a third party provider.

The possible integration scenario of a Community Cloud is application dependent. This is because the purpose of a Community Cloud is to facilitate community members' collaboration and/or joint development efforts. The collaboration facility may incorporate full-service standalone applications that are accessible via a Web browser. Community members access Cloud functionalities exposed as a service. It can be a collection of many on-premise business functionalities exposed to the Cloud, data repository, or integrated messaging bus.

Figure 2.4 portrays a Community Cloud where Cloud applications are running in the Public Cloud and interoperating with some partner applications provided by

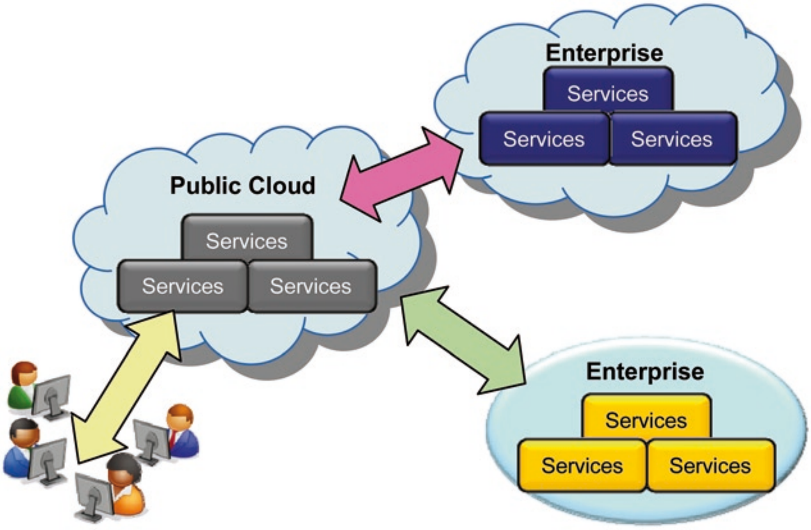


Fig. 2.4 Community Cloud

two enterprises. This emerging hybrid model of on-premise Cloud applications represents a new class of distributed business relationships. For instance, the value-chain network is no longer restricted by a limited number of players. Any potential business contributors can now participate more easily and effectively. However, various issues, such as cross-organizational boundaries and firewalls, are among the key challenges for the Community Cloud. These issues will be discussed in later chapters.

Even though community-based service architecture is a critical revolution of business and service configurations in a enterpriser IT environment, Community Cloud can be implemented by the other three Cloud architectures or configurations. Therefore, there will be no separate discussion of Community-based Cloud architecture from this point on.

2.2.3 Private Cloud

The Private Cloud is built solely for an organization. It is typically hosted inside that organization's firewall and managed either by the organization or a third party. In some instances, Private Clouds can also be outsourced off-premise to reduce operating costs.

The reason for many enterprises to implement Private Clouds (also known as internal Clouds) is to deliver some benefits of Cloud Computing without dealing with concerns such as security, corporate governance, availability, and reliability. For some enterprises, Private Clouds are a stepping stone to external Clouds, particularly for the financial services and defense applications, where future datacenters will look like internal Clouds. However, these enterprises have to buy, build, and manage these Clouds and thus do not benefit from the economic model of Clouds.

When dealing with Private Clouds in house, enterprises need to consider the physical location for the computers, the level of network connectivity, and the increasing cost and quality of electric power. In Fig. 2.5, an enterprise builds a Private Cloud within its private network and makes the services available to its employees and customers via the Public Cloud. Because the service interfaces are through a Web-based portal, this action is transparent between the public and Private Clouds.

2.2.4 Hybrid Cloud

The Hybrid Cloud is a composition of two or more Clouds that can be private, community, or public. These Clouds remain unique entities while becoming a part of a Hybrid Cloud and are bound by an agreeable interface that enables service interoperability. Their association may either be on a continuous basis or mission-oriented for a specified period of time.

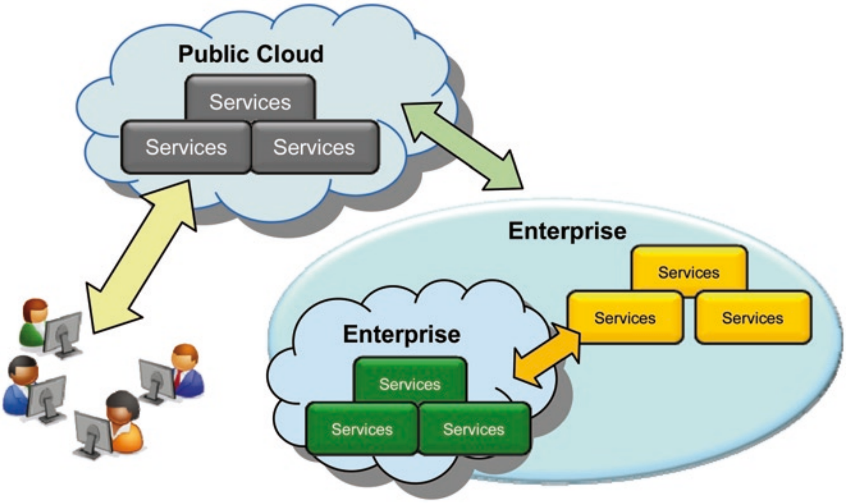


Fig. 2.5 Private Cloud

This configuration is very useful when different applications and associated data cannot exist in silos. Whether it is done to complete different steps of a large business process or leverage collected features to achieve superior business applications, these Clouds are integrated through the coordination of Cloud Aggregators or Cloud brokers. Based on service-oriented relationships, these brokers federate data, applications, user identity, security, and other management features including load-balancing and QoS governance.

A complex scenario of a Hybrid Cloud can consist of multiple internal and/or external providers that perform different business functions. Using Cloud technology, new service functions can be easily plugged into the established service architecture. The type of resources can be logical services or virtualized environments that require physical servers, routers, storage, firewalls, spam filters, or other hardware.

The following figure (Fig. 2.6) portrays the relationship between the enterprise's Private Cloud and two Public Clouds that offer services to the enterprise users and employees. It also shows the data exchanges between the two Public Clouds through the enterprise's VPN—all are using Web-based service interfaces.

2.3 General Information Technologies

One of the key business drivers for enterprises to consume online services is reducing IT expenses and refocusing valuable enterprise resources on enabling core business capabilities. Another important driver is adopting new technologies faster

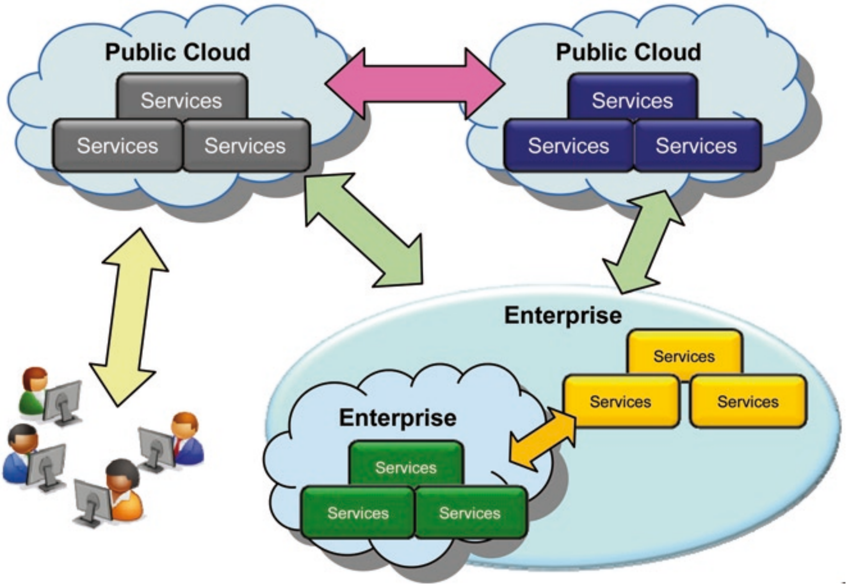


Fig. 2.6 Hybrid Cloud

through third-party vendors without having to risk upfront investment in people and equipment. Cloud providers play an essential role in the processes, as they offer commodity services that are theoretically more predicable, cheaper, and generally better due to their economy of scale. This arrangement allows the provider to pass on the cost savings and efficiency to enterprise customers. As a direct result, adopting Cloud technology strengthens enterprises' core business focuses and ensures their product deliveries can take effect in the market on-time and on-budget. Application characteristics for a Cloud in an enterprise are illustrated in Fig. 2.7. The figure shows some sample business and technology applications as essential candidates for Cloud services [3].

As used in this section, General IT implies that the technology features are common to most enterprise or organization operations. Segment or market-specific discussions, such as financial and telecommunications services, will be elaborated in the following sections.

Let us begin by looking at the characteristics of general IT applications that are potential candidates for the Cloud [4–7]:

- *Non-Core Business Services*: For business functions that are not essential to the enterprise's deliverables, the enterprise can outsource them to external SPs in order to focus on their core business. These functions include Web conferencing, enterprise-content management, and portals that are non-core or non-differentiators.

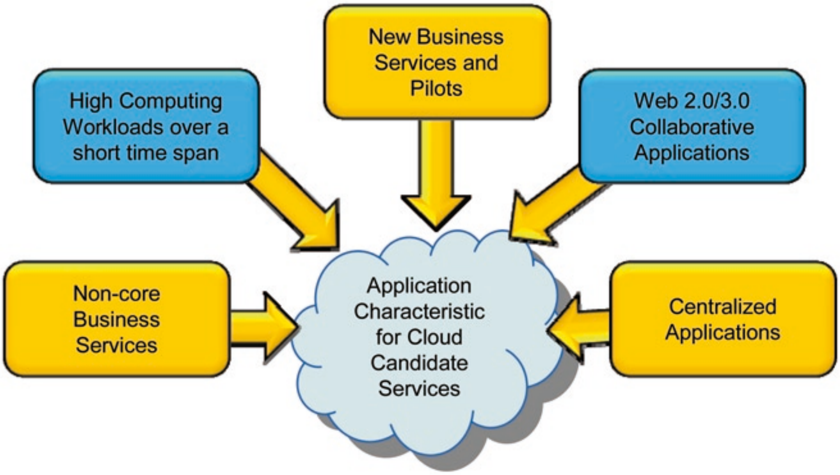

New Business Services and Pilots
High Computing Workloads over a short time span
Web 2.0/3.0 Collaborative Applications
Non-core Business Services
Application Characteristic for Cloud Candidate Services
Centralized Applications

Fig. 2.7 Application characteristics for the Cloud in an enterprise

- *Data Intensive Computing*: The need for an efficient default backup of large data sets, better access to data with a large distributed database, pre-formatted data in large repositories, and indexed large data sets are among the top data management applications that can be handled by external providers.
- *Computing Intensive Activity*: Parallel batch processing means utilizing a large number of computers for a short period of time to accomplish a task. Applications such as symbolic mathematics, business/scientific analytics, image rendering, and 3D animation all require extensive computing resources and can be well served in a networked computing environment.
- *High Computing Workloads Over a Short Time Span*: Applications that do not have uniform workload requirements can experience occasional resource utilization spikes. The need to acquire a permanent infrastructure to support the peak workload is not realistic. A more effective way to manage the need is to contract-out the task and pay for usage.
- *Compute Resource Management*: The techniques to manage many-core resources and multiple-vendor computing units are becoming standardized. Most challenges in dealing with scheduling and dynamic resource provisioning are simplified greatly by the recent evolution of virtualization and automation technologies. The enterprises can now see these efforts as commodity features and feel more comfortable in hiring third party subject matter vendors to manage the resources for them.
- *Storage Architectures and Implementations*: When enterprises use distributed file systems or data-intensive computing applications, there are many features that will dictate the efficiency of the operations, especially when dealing with widely distributed systems, large data, or both. In these events, data caching

frameworks, data-aware scheduling, and cross-center data management can take advantages of virtualization and/or outsourcing to reduce process complexity and increase efficiency.

- *Programming Models and Tools*: For enterprises in the system development business or research organizations that develop new technologies, appropriate programming models and tools are critical for their success. These enterprises typically have to deal with many task computing middleware and applications that involve message exchanges and data storage in integrated parallel programming frameworks. For service-oriented applications, many service orchestrations and collaborations are required in a development community or cross-organizational teams. Virtualization of the development and collaboration tools, or even tools offered by external providers, can greatly simplify the skill-set and eventually the cost and risk to support the business.
- *Delay-Tolerant and Disruption-Tolerant Applications*: Originally designed for Internet-like services across interplanetary distances in support of deep space exploration, this technology is now available to support any operational or performance characteristics where conventional networking approaches are either unworkable or impractical. The applications are suitable for disaster rescue missions or non-battle military applications.
- *Centralized Applications*: Enterprise applications that have cross- enterprise and cross-departmental reach may need to centralize their business operations to improve efficiency. This implies a need for unexpected amounts of computing and storage capacity. Instead of duplicating the effort by creating multiple applications or copies of the same software, these can be consolidated in a virtual networked environment. The new environment can centralize the infrastructure management, offering economics of scale across different departments and enabling more agility to meet dynamic business demands.
- *Web Collaborative Applications*: Enterprises that would like to take advantage of Web 2.0/3.0 technology to generate data that can be exposed to the public, but currently would also like to avoid affecting the existing enterprise infrastructure, can seek solutions from external SPs. Applications such as video sharing, discussion forums, and blogs offered by third party providers can speed up the deployment of Web-based applications to support enterprises' new business initiatives, allowing them to achieve closer communication with customers and partners.
- *Secured Data and Information Management*: Enterprises realize that protecting devices or networks cannot assure data protection. Instead, a comprehensive end-to-end protection is needed to protect the data itself. This can only be done by pushing the data from local and community-based databases to the network. This implies that data should be persistently protected at all times (this includes when the data is at rest as well as in motion), without interruption. The protection must be device-independent and network-independent in or among virtual environments.
- *Automation and On-Demand Services*: To improve the efficiency and effectiveness of system development, enterprises demand developers to adopt new methods in order to focus on new features instead of labor-intensive coding efforts. The new methods must help developers move from heavily customized software

to repeatable assembly service products. The goal is to assist the enterprises in realizing repeatable processes with increased automation and collaboration, or to avoid over or under provisioning.

The aforementioned characteristics will benefit small, medium, and large enterprises as virtualization and automation technologies allow services to meet their needs in cost cutting, risk reductions, and operational efficiency. With appropriate planning and configuration, Cloud service bundles will be capable of bringing a wide array of services and applications to the markets that will impact many existing business paradigms and how users use data. However, not all applications are suitable for running on the Cloud. As will be discussed in Chap. 4, there are some obvious limiting factors, such as data security, potential lock-in with Cloud providers, open and symmetric interfaces, efficiency of data conversion between/outside Clouds, and interoperability with legacy/private applications that can slow down the adaptation of the Cloud. Throughout this book, the authors will tackle these issues and identify possible solutions to remove these barriers.

Before any further technical discussions, let us first look at a simplified business scenario of a Cloud service. In Fig. 2.8, a two-layer Cloud is shown with a basic flow of how a service request will be accepted in the enterprise's datacenter and how the data will be processed and returned to the client [8]:

1. The client sends a service request
2. System management finds the correct resources
3. System provisioning finds the correct resources
4. Computing resources are found and the service request is executed

Results of the service request are sent to the client

In the following sections, the three basic Cloud service scenarios will be illustrated.

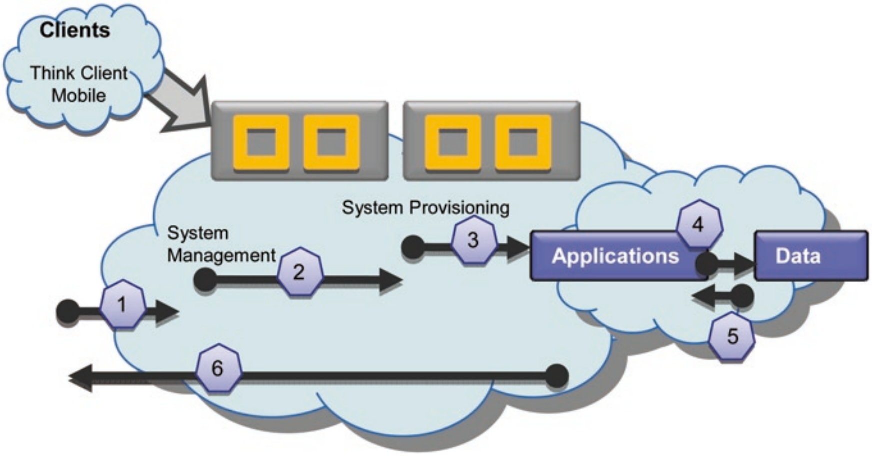


Fig. 2.8 Cloud computing workflow

2.3.1 Software Services

SaaS allows SPs to license software applications to their customers for use as services on demand. The providers can host these applications on their own servers or upload the applications to the consumer devices. These on-demand functions are managed through an SLA process either directly by the SPs or by a third-party provider [9].

As recognized by enterprises, business processes and the data itself are the primary assets. Customer records, workflows, and pricing information are more critical than the application systems. This understanding expedites the adoption of SaaS and drives the development of software systems to become more commodity-based. As a result, expense-reports, Web-based calendaring, applicant screening tools, spreadsheets, and e-mail systems are now more accessible and portable than before.

Compared to *Platform as a Service* (PaaS) and IaaS, SaaS has a relatively more mature business model and technology for Cloud applications. This can be seen in the following two arguments.

First, new applications can now be created from parameters and macros. The availability of this technology allows other vendors to quickly build SaaS applications or establish a support framework atop a common application platform. Many SaaS products today allow for a wide range of customization within a basic set of functions. This includes CRM and *Enterprise Resource Planning* (ERP) applications, email, Web conferencing, digital content creation, Dashboards, and Application Exchanges. SaaS providers can often deliver products that meet their markets' needs more closely than before. Secondly, SaaS has the effect of democratizing software, allowing small and medium businesses to have access to functionalities that were formerly only in the large enterprise domain. For instance, many analytical software tools have been released as SaaS applications and are available on a monthly subscription basis. Some sample SaaS offerings are listed in the following table (Table 2.1):

Table 2.1 Sample SaaS offerings

Category	Applications
Enterprise applications	File backup, sharing, access HRM Finance ERP-Other Business productivity Spend & expense management CRM Marketing applications Business intelligence Business applications-Other Application marketplace Professional services automation (PSA) Large data set analytics

2.3.2 Platform Services

PaaS allows providers to deliver a computing platform and solution stack as a service to customers who have a need to facilitate deployment of applications without the cost and complexity of buying and managing the underlying hardware and software layers. PaaS often includes provisioning a software development platform and providing the facilities required to support the complete life cycle of building and delivering Web applications. It also uses the advantages of distributed development teams working together on the same projects using diversified supporting tools from different sources. Such composite environments enable interactions that are not limited to developers and coders. With this scenario, the entire CoI can participate in the development and provide comments or inputs on any stage of the development cycle. There are two main advantages to this type of service. First, using higher-level programming abstractions for service development, the complexity and dependency of the entire system architecture and UIs can be reduced dramatically. Second, the overall development effort can be more effective as the built-in infrastructure services, such as security, scalability, and failover are now a part of the library. The testing and integrating efforts can be more modularized. Likewise, maintenance or enhancement of the codes will be easier.

The following table (Table 2.2) lists sample PaaS products that are available from different SPs or vendors:

Table 2.2 Sample PaaS offerings

Category	Applications
Enterprise applications	Business process management (BPM) Business framework Workflow management
Web applications	Site hosting Web analytics App/Web server Portal server Web services/SOA tools
User-facing API & management	Mobile application delivery UI framework Content management Billing, payment, and metering Telephone
Software development and testing	Development tools Testing tools Testing environment Deployment tools Application scripting NEW developer sandbox Code performance analytic Application versioning Team collaboration and developer community facilitation

Table 2.2 (continued)

Category	Applications
Messaging	Message queue
Security	Security tools Security portal
Database	Data stores Database Data synchronization Database integration
General platform	Frameworks DNS services Security/Identity management OS Application integration Application/Middleware provisioning

2.3.3 Infrastructure Services

Infrastructure as a Service (IaaS) means the computer infrastructure such as CPU, disk space, servers, software, datacenter space, or network equipment is delivered as a fully outsourced service. This is an evolution of Web hosting and virtual private server offerings. The enterprise customers are typically billed on the amount of resources consumed or occupied.

For networking as a service, SPs offer Web and Web service interfaces for the enterprise customers to expose their application capabilities, such as asset management solutions for other services. For computation resource as a service, SPs offer their customers the ability to scale-up or scale-down computing capabilities on-demand. For instance, customers can schedule batch jobs or background applications in parallel with other enterprise tasks. For storage as a service, enterprise customers can contract the provider to store large amounts of unstructured or structured data, with or without full relational semantics. The Service can also include a messaging service for scalable, reliable, and asynchronous data exchanges.

IaaS enables dynamic acquisition of infrastructure resources, allowing enterprises to aggregate computing assets from a pool of resources on-demand. There are many applications in this type of service, Table 2.3 lists some sample offerings.

2.4 Commercial Markets and Applications

A key business challenge for any commercial corporation is to expand and extend its footprint in both the existing market as well as emerging markets. For larger enterprises, their goals will also include the ability to establish their brands worldwide via superior customer service, and drive growth further through new business services and channels. With help from new IT technologies, information can be exchanged faster and further than before.

Table 2.3 Sample IaaS offerings

Category	Applications
Service management	Cross-systems management Automation/Provision platform Grid management Configuration management Monitoring services
Virtualization	VM application performance monitoring Virtualization platform Hosting
Compute	Compute services
Security	Unified threat management Security compliance Security posture analysis
Storage	Edge storage—content delivery network (CDN) Primary storage Secondary storage Storage compression Backup service
Communications	Load balancers Inside to outside bridging Routing Messaging/Queuing services VLAN networking Firewall

As mentioned in the last section, Cloud services include search engines and SaaS such as ERP, BPM, CRM, and e-commerce applications. BPM, linking, calculation, SOA API integration, and Web page launch syntax are some useful enablers that benefit commercial application developments. For instance, SOA Web services and virtualization improves flexible responses to changing market conditions. Cloud technology also assists business users to feed up-to-date information to business management systems (e.g., PBM and CRM), while allowing business analysts to access the same systems via a standard online syntax without coding.

For business application development, developers can operate an application-testing infrastructure in the Cloud to save time and money compared to traditional test scenarios. Users can participate in earlier development phases and get a transparent view of application performance, reliability, and scalability. Software deployment can be easily accomplished by pushing the debugged code to the target environment in a few simple steps. Scalability and portability from Clouds also offer significant competitive advantages and improvements in productivity.

As the Cloud offers a unified way to link supply chains more efficiently, providers and markets are integrated with speed beyond a single company's control. Global, Cloud-based offerings are expected to reach $150.1 billion by 2013. Much of this growth represents a transformation from traditional IT services to the new Cloud model, as well as substantial new businesses and revenue streams, as will be seen in the following sections. Business-related Cloud services, including advertising, e-commerce, human resources, and payments processing are expected to grow to $46.6 billion. It is anticipated that Cloud-based advertising will continue to reshape and redefine the advertising and media markets over the next few years.

The following sections will demonstrate how the Cloud can provide superior enabling abilities in the areas of marketing, sales, and finance. We will then provide the current state of adoption in both the financial and telecommunications industries [10–12].

2.4.1 Marketing

The marketing process identifies enterprises' products or services and their target customers. It also includes the strategy and execution of sales, communications, business development, and customer relationships in order to grow their market share and increase revenue. In terms of introducing new business services and pilots, enterprises must have the flexibility to launch new business applications, without depending upon upfront IT infrastructure investment. An ideal situation is to obtain platform supports from a reliable outside supplier for new business development. The enterprise can determine later if additional resources will be needed, thus helping the enterprises manage their risks better. Figure 2.9 depicts

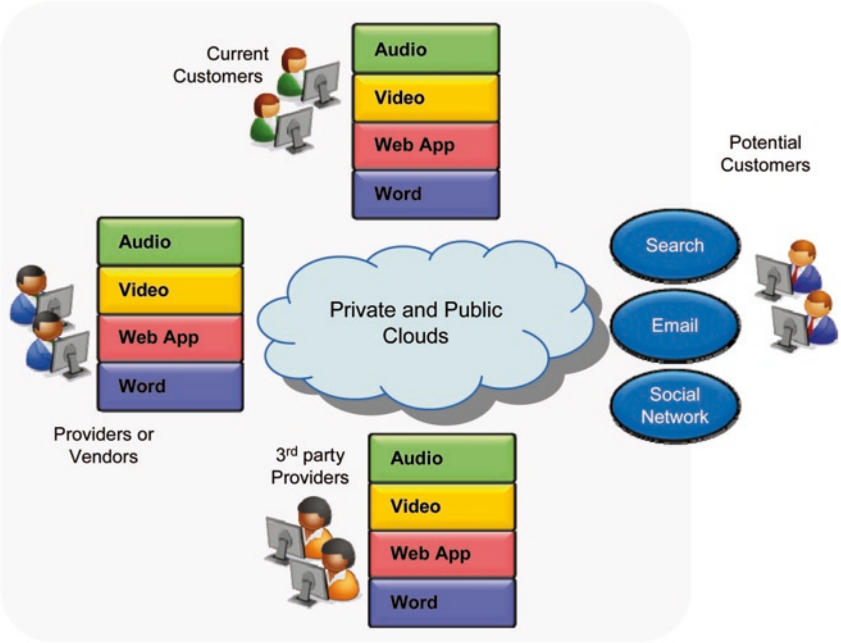


Fig. 2.9 Typical Cloud-based business communications

a typical Cloud-based business communications environment where customers, SPs, and third party providers are communicating through either Private or Public Clouds.

For enterprises that wish to improve branding and customer service, collaborative applications and business tools can be equipped to provide better connections with their customers and service employees. These applications/tools can include internal experts on the products in customer support, sales, product management, research and development, field service, and consulting. For introducing a new CRM system for support services or launching an online marketing campaign, Cloud technologies can effectively improve customer feedback and collaboration with new social-media applications. There are several collaboration tools available today.

- *Interactive Media*: Interactive media includes products such as blogging, social networking, and e-conferencing. Blogging allows voices from the public domain to be heard. It provides a unique vehicle for enterprise customers to get a third party's view on the products and services. Social networking, such as Facebook, allows third-party developers to build applications for sharing trials or demo products. The enterprise CRM systems can then pull a Facebook profile and its friend information into their CRM to profile these potential clients, allowing the enterprises to create a more personalized online community. As for e-conferencing tools, products such as Cisco's WebEx allow enterprises to provide interactive Webinars through a phone bridge or computer to provide Internet-based marketing campaigns. Once prospects are indentified, enterprises can communicate with them using targeted, personalized messages which are more effective in catching their attention. Furthermore, email can be coordinated with other channels, allowing recipients to choose how they want to interact with the enterprise. Other interactive media channels include mobile, print, and social networking.
- *Passive Media*: Enterprises are increasingly using Websites as their faces for improving their visibility to Internet search engines. By improving the content of their Web pages, enterprises can draw traffic to their site without having to purchase advertising from marketing specialists. In the Website, the enterprise can incorporate a product FAQ, blogs, videos, and trials to capitalize on potential customers searching for references. A product or service FAQ can also be used to assist customers' problem solving to lessen calls to the customer support centers.
- *Management Tools*: It is important to measure contribution and quantify the value of an enterprise's investments. This helps enterprises capture a picture of the type of people who are most interested in certain products at certain locations. The Cloud framework offers a unified environment for enterprises to integrate management tools more effectively. It assists enterprises in realizing sophisticated automated systems, permitting consistently accurate and cost-effective analysis of market situations, and helping the enterprise respond to this intelligence more precisely.

2.4.2 Sales

The quantity of investments and quality of leads typically helps enterprises achieve higher quality products and services and larger sales. The key is the economics of the sales process. Besides opening new storefronts or channels, an enterprise can increase its existing service and channels mix to improve the customer experience. Using Web technology, enterprises can deploy trial services and channels quickly and review initial progress through pilot launches. All these can be accomplished without having to commit to extensive upfront investments. For a Cloud SP, the good news is that products are reasonably precise versus the large application footprint of traditional enterprise products, therefore it is easier to close deals because the products are sold per module structure. However, the bad news is that the average selling price is considerably lower.

Enterprises are always investigating better ways to improve their investments by leveraging their existing sales force. This is because they have to justify the development of a new sales force, where the short-term returns are much less than the amount of investments needed to fund the creation of the new channel. This is particularly important in technology products. In such an event, enterprise clients can benefit from better and cheaper features, such as security, upgrades, or performance management if the providing enterprises choose to outsource these functionalities. In addition, if the providers can leverage their existing investments in IT and reduce licensing, hosting, and maintenance costs by outsourcing, the enterprises can further gain more benefits in economics of scale. Leveraging these capabilities allows an enterprise to focus on its core business and build innovative applications for its product line and customer service.

As the new value chain relationship brings suppliers closer to their clients, traditional channel partners, such as VARs, must work harder to create niche values for their existing customers. For instance, they can use Cloud technologies to streamline their existing process and broaden their client base by creating discrete expertise in special areas to maintain and even grow revenue.

The value of a Cloud in the area of sales compensation is also very noticeable. Cloud technologies enable enterprises to report sales credits regularly, transparently, and consistently for their compensations. An accurate compensation is always the best driver to motivate a salesperson. In addition to the transparent dispute process, an enterprise can tie sales results with a predictable schedule and objectives to further drive and achieve goals. It can constantly evaluate the sales performance data and make rational changes to the compensation plans.

2.4.3 Finance

The transformation of Cloud-based services represents not only a change in marketing and sales, but also in accounting, finance, and business operations.

SLA plays an important role in modern services and products. It is not only a passive vehicle for SPs to warrantee their services, but is also an instrument to manage contracts throughout distributed, value-chain relationships and service ecosystems. Although the SLA concept has been available for business management in many industries for quite awhile, this mature technology has not yet found a major uptake in broad finance applications. Fundamentally, SLA management includes negotiation, implementation, execution, and assessment. In the IT industry, the service criteria normally pertain to metrics in availability, security, problems, change, and performance. It is in the interests of both suppliers and consumers to create and operate SLAs that demand minimum human interaction to govern. Using the standard interface and automation, Cloud technologies offer outstanding means for enterprises to realize the true values of SLA management.

From a financial analyst perspective, business intelligence can now be gathered in a more unified way through a Cloud where market information, management team activities, and competitive analysis can be correlated by the same framework. For instance, *Monthly Recurring Revenue* (MRR), Quarterly contracts, Annual contracts, 12-month pre-payments, and so forth can be correlated with the churn or renewal statistics. All this can be fed to CRM infrastructure processes for potential enhancement or changes.

2.4.4 Financial Industry

The IT departments of financial institutions recognize the potential efficiency benefits from Cloud technologies, especially those in capital markets where technology is a key driver. However, there are business drivers and risks this industry cannot afford to ignore, including [13]:

- In an era of tighter budgets, financial SPs are looking for ways to cut their IT budget while satisfying increased regulations and rising fraud. How can the Cloud model provide a clear formula to measure ROI without obfuscating the calculation of risk and cost?
- In light of the person-to-person payments model, can a Cloud provide a business framework to facilitate this new application? In particular, how much freedom should the clients possess without risking the integrity of providers' systems?
- To guarantee the "always-on" banking service, how should financial institutions adopt Cloud paradigms and deploy the right features with the right priority? For instance, will availability and accessibility through a Public Cloud be an acceptable option?
- With regulation of the financial services industry on the rise, how many of them are solvable by Cloud technologies? Has the Cloud industry developed enough mature standards and specifications to support this trend?
- When the financial industry shifts more services to the Cloud, what new security requirements should be added to the existing standards? For instance, will the Cloud model concentrate everybody's data in the hands of a powerful few?

Obviously, not all questions can be answered today because of a lack of clear technological infrastructure to address these business applications. Furthermore, security concerns and wider market confusion continue to inhibit the speed of adaptation. Nevertheless, most financial institutions are taking the steps to be on the cutting edge of person-to-person payments, embracing innovations from alternative vendors, tightening security vulnerabilities, and serving up real-time data that will allow their customers to do banking anytime, anywhere. In fact, surveys show that the majority of financial SPs have initiated many back-office, IT-focused, transformation projects, evidenced by their active participation in Public and Private Cloud deployments.

As for the regulation aspect, it is expected that changes will be made dramatically as the U.S. Securities and Exchange Commission (SEC), the Commodity Futures Trading Commission (CFTC), and other regulatory bodies review certain Cloud environments. Once these regulatory bodies can certify that technologies and associated specifications are secure enough for customer data, the financial services industry will be able to take the whole advantage of the Cloud to improve their technology, process, and management. In the mean time, the Cloud industry must continue to make the definition and relevance of Cloud services far clearer to banks and insurers, or articulate the benefits more meaningfully to the wider financial services community.

In the following chapters, the authors will address many of these issues from a technology perspective.

2.4.5 Telecommunications Industry

As the current provider of network services, the telecommunications industry has natural advantages in adopting Cloud-based technologies. This includes network-based platforms provided as a service from a datacenter. By adopting the Cloud concept into their operations, the telecommunications industry can immediately increase the value of their networks in multiple ways and create new business roles with more potential revenue. Per a report published by Telecom Trends International, the expected market size will generate $45.5 billion in revenue by 2015, mainly in the domain of providing access to computing resources over the Internet. This will reduce the need for deploying and maintaining expensive call centers.

For example, current *Operations Support Systems* (OSS) and *Business Support Systems* (BSS) used in the telecommunications industry are highly componentized. They can be integrated well with Cloud services, yielding advantages in processing and performance for managing IT resources. As telecommunications operators dominate the majority of public networking assets, they have a relatively stronger position to influence network traffic and utilization and thus, transport revenues. In addition to their existing operational framework for scalable services, telecommunications operators can easily claim an end-to-end model in the Cloud service value chain, with improved QoS for user-to-application experiences. With this advantage,

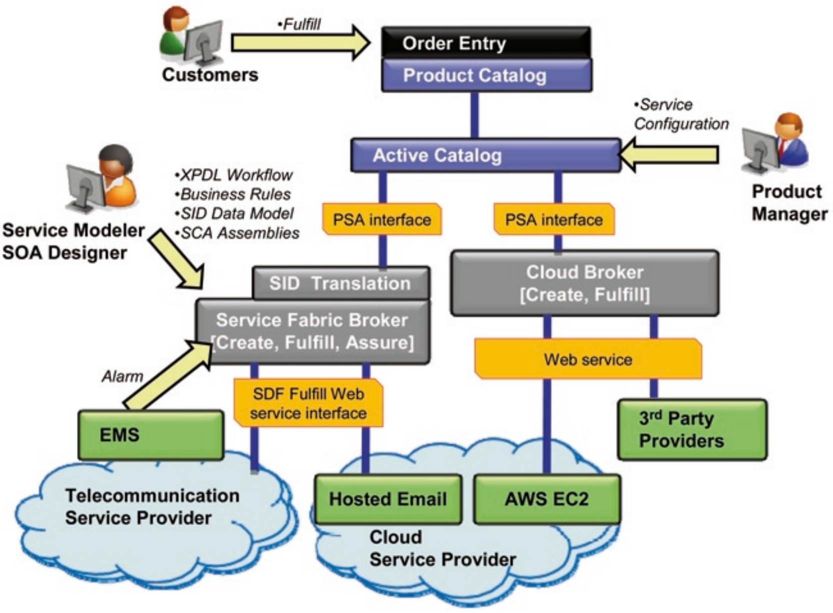


Fig. 2.10 TM Forum Cloud catalyst

network operators or telecommunications providers have an opportunity to extract two revenue streams. One stream charges end users based on the levels of service quality, the other stream charges Cloud-based providers for their network service quality. This network-based approach to service assurance can also be extended to the software revenue market by offering QoS on software development and deployment applications.

Figure 2.10 portrays a TM Forum Cloud Catalyst where telecommunications SPs are playing a key role in enabling service creation, fulfillment, and assurance [14–17].

Although the above scenario is very compelling, the actuality is that telecommunications equipment today still does not meet expectations in offering smart and efficient resource allocation that is transparent to the OSS. The network operator must leverage natural advantages with technology, such as Cloud technology, and continue to improve efficiently integrating their network with storage and computing assets. Using Cloud technology, telecommunications providers can gain an immediate competitive edge in optimizing their internal operational costs by making their current networks and platforms virtual. For instance, PaaS and IaaS technologies offer high elasticity to provision the providers' network infrastructure and allow them to add service capacities on demand, expediting time-to-market for new services.

Cloud technology not only delivers business value to the service customers, but also increases and extends their sustainability. Telecommunications SPs and vendors are moving very aggressively to integrate and adopt this new trend. For instance, the TM Forum established the ECBC to remove operational, management, and technology barriers of commercial Cloud services. Their objective is to bring transparency and efficiency to the relationship between buyers and sellers by lowering the gating factors for adopting Cloud services. Vendors and operators such as Alcatel-Lucent, Amdocs, AT&T, BT, CA, Cisco, EMC, HP, IBM, Microsoft, Nokia Siemens Networks, Telecom Italia, and Telstra are among the first wave of participants. Additionally, industry organizations including DMTF (resource management) and itSMF (service delivery) are also initial members.

Other international standard bodies are also embracing this new technology trend. For instance, the Telecommunication Standardization Sector (ITU-T) coordinates standards for telecommunications on behalf of the International Telecommunication Union (ITU). Their first meeting of the ITU-T link Focus Group (FG) on Cloud Computing took place in June 2010 in Geneva. The mission of this meeting was to define a roadmap to guide further developments of standards in ITU-T to address the benefits (vision and value proposition) of Cloud Computing from telecommunication perspectives. This FG will define use cases, service models, reference models, and requirements in support of Cloud Computing and apply Cloud Computing to the telecommunication industry for both fixed and mobile services.

2.5 US Government and Defense

The U.S. Government recognizes the value of information technology and management. It established an open government platform that enables efficient and effective services across the federal government to protect and serve the public. Cloud technologies contribute to the enhanced functionality for the federal workforce to improve interoperability, feedback, collaboration, and the dissemination of information. By using commercially available Cloud technologies, offered government services can be more cost-effective and can provide a better QoS.

From an information management perspective, standardized government data can improve information sharing throughout the government and with the public. Government data is disseminated in accessible formats that are based on a shared architecture, making information more findable, understandable, relevant, and useful, while also ensuring a positive customer experience. Information assurance policies can help different government departments implement a transparent, accountable, and efficient government that balances openness with the need to maintain privacy and security.

Virtualization technology provides a highly scalable IT infrastructure for use by the federal workforce to enable rapid delivery of new capabilities at a reduced cost. Using the Cloud collaborative technology, different departments can enhance the sharing of information between federal agencies and with other governments and the public.

A recent study by Market Research Media forecasts that the U.S. Government spending on Cloud Computing will enter the next phase of explosive growth at about 40% CAGR in 2010 for the next six years, passing $7 billion by 2015 [18–21].

2.5.1 Federal Chief Information Officers Council

Cloud Computing plays a key role in the U.S. President's initiative to modernize IT by identifying enterprise-wide common services and solutions and by adopting a new Cloud Computing business model. The Federal Chief Information Officer (CIO) Council, under the guidance of the Office of Management and Budget (OMB) and the Federal CIO, established the Cloud Computing Initiative to fulfill the U.S. President's objectives for Cloud Computing.

The CIO Council serves as the principal interagency forum for improving practices in the design, modernization, use, sharing, and performance of Federal Government agency information resources. The Council's role includes developing recommendations for IT management policies, procedures, and standards; identifying opportunities to share information resources; and assessing and addressing the needs of the Federal Government's IT workforce.

Currently, the CIO Council has the following committees: *Architecture* and *Infrastructure*, *Best Practices*, *Information Security and Identity Management*, *IT Workforce*, and *Privacy*. The CIO Council also has working groups focusing on Data.gov, Cloud Computing, and IT Capital Planning.

- *Federal* EA: This architecture is a management practice to maximize the contribution of a federal government agency's resources, IT investments, and system development activities to achieve its performance goals. It describes relationships from strategic goals and objectives through investments to measurable performance improvements for the entire enterprise or a portion (or segment) of the enterprise. The architecture helps the federal government organize and clarify the relationships between agency strategic goals, investments, business solutions, and measurable performance improvements. As illustrated in Fig. 2.11, enterprise, segment, and solution architecture provide different business perspectives by varying the level of detail and addressing related but distinct concerns. This figure shows segments are across multiple agencies. They can be leveraged within an agency, across several agencies, or across the entire federal government. As for an individual agency, it contains both core mission area segments and business service segments. In the center of the figure, enterprise services cross-cut services spanning multiple segments [22].
- *Apps.gov*: This is the first Cloud Computing mall launched by the U.S. White House for government agencies to quickly browse and purchase Cloud-based IT services for productivity, collaboration, and efficiency. Traditionally, the Federal Government often buys IT through numerous, fragmented, suboptimal

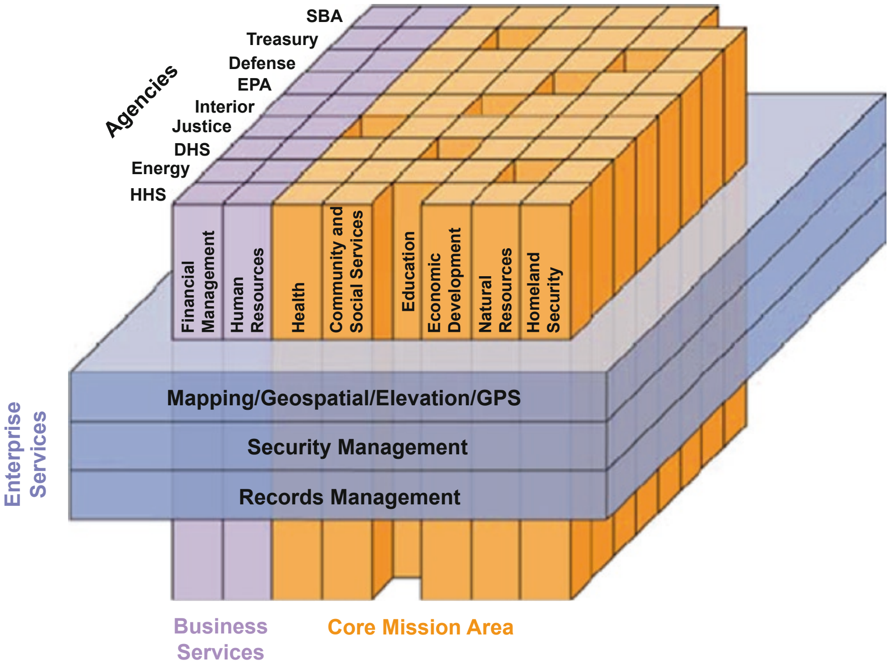


Fig. 2.11 Federal enterprise architecture

purchases that are limited in scope. Moving the majority of routine Federal purchase card transactions to these online Federal eMalls can achieve significant savings. Specifically, its visibility to view and analyze purchase data across the Government can help policy makers effectively develop strategic sourcing policies. Currently, Apps.gov offers the following four tiers of Cloud services: (1) *Business Apps* such as analytical, business processes, CRM, tracking and monitoring tools, business intelligence, and so forth; (2) *Productivity Apps* such as word processing and spreadsheets as well as collaboration, document management, and project management; (3) *Cloud IT Services* such as solutions for storage, Webhosting, and VMs all hosted in the Cloud; and (4) *Social Media Apps* such as text, audio, video, podcasts, and other Web 2.0/3.0 multimedia communications.

- *Data.gov*: Based on the U.S. Open Government Initiative and developed by the Federal CIO Council, Data.gov is an interagency Federal initiative and is hosted by the General Services Administration (GSA). This Cloud service enables the public to participate in government by providing downloadable Federal datasets for developers to build applications, conduct analyses, and perform research. It includes searchable data catalogs that provide access to data in three ways. The *Raw Data Catalog* provides an instant view and download of platform-independent, machine readable data (e.g., XML, *Comma-Separated Values* (CSV), *Keyhole Markup Language* (KMZ/KML), or shape file formats) and links to a metadata page specific to the respective dataset. The *Tools Catalog* provides application-driven access such as widgets, data mining and extraction tools, applications, and other services to Federal data through hyperlinks. The *Geodata Catalog* features a geodata catalog called *GeoOneStop* that includes trusted, authoritative, and Federal geospatial data. This catalog includes links and metadata pages to download the datasets. It also includes links to more detailed Federal Geographic Data Committee (FGDC) metadata information. Data.gov increases the ability of the public to easily find, download, and use datasets that are generated and held by the Federal Government. As a result, new software applications providing useful services to the citizens have been rapidly developed for the public by the private sector.

- *IT Dashboard*: Federal IT spending of nearly $80 billion a year requires continuous improvements in oversight. Agency CIOs are responsible for evaluating and updating select data on a monthly basis. Responding to the need, the IT Dashboard was developed to display data received from agency reports to the OMB, including general information on over 7000 Federal IT investments and detailed data for nearly 800 of those investments that agencies classify as major. The performance data used to track the 800 major IT investments is based on milestone information displayed in agency reports to the OMB called Exhibit 300s. The IT Dashboard provides the public with an online window into the details of Federal IT investments and provides users with the ability to track the progress of investments over time. It increases the visibility of agencies' IT spending, promotes accountability, and helps managers identify and eliminate redundancies.

2.5.2 General Services Administration (GSA)

The GSA is participating in the Federal Cloud Computing Initiative and is responsible for coordinating its activities with respect to the Initiative via its Cloud Computing Program Management Office (CC PMO). GSA and the CC PMO are focused on implementing projects for planning, acquiring, deploying, and utilizing Cloud Computing solutions for the Federal Government that increase operational efficiencies, optimize common services and solutions across organizational boundaries, and enable transparent, collaborative, and participatory government. The overall objective is to create a more agile Federal enterprise, where services can be provisioned and reused on demand to meet business needs.

2.5.3 National Business Center (NBC)

The Department of the Interior's National Business Center (NBC) plans on bringing the benefits of Cloud Computing to NBC's business services clients and datacenter

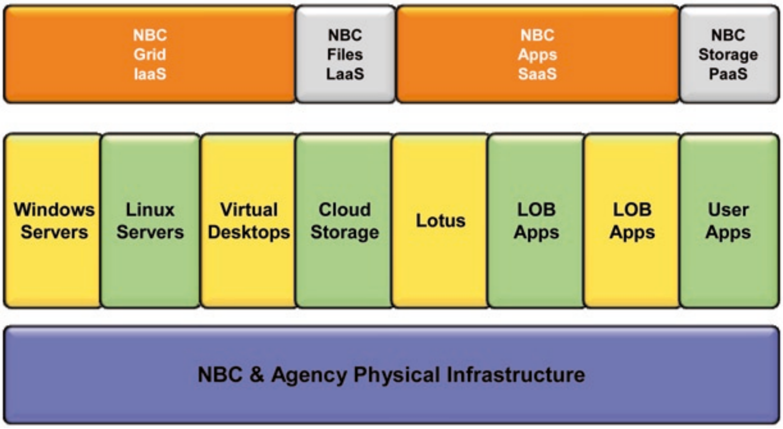


Fig. 2.12 NBC's Cloud architecture

hosting clients through advancements to the highly efficient NBC shared infrastructure. As shown in Fig. 2.12, five Cloud services are offered today. They are [23]:

- *NBC Grid*: This IaaS offering allows the end-user provisioning of a variety of types of servers and OS through a single customer portal. It provides technology-agnostic server hosting with a variety of pricing models, including metered and pre-paid, based on the customer's usage of *Random-Access Memory* (RAM) or CPU per hour.
- *NBC Files*: This Cloud storage offering allows burstable storage capacity on a metered, pay-per-gigabyte price model. Its usage and status can be monitored via a unified customer portal. These capabilities can be leveraged for application storage and content delivery, or as a backup platform.
- *NBC Stage*: This PaaS offering allows software developers to build applications with a highly scalable capacity, while staying within the bounds of the federal Government's IT regulations and standards.
- *NBC Apps*: This offering is a Cloud-based application marketplace offering the following three types of application: (1) general purpose applications including messaging, collaboration, and Web 2.0/3.0 tools like wikis and blogs, (2) acquisition SaaS consists of an on-demand version of *Electronic Servicing Environment* (ESE), and (3) the HR *Line of Business* (LoB) SaaS offering Onboarding, *Learning Management System* (LMS), Performance and Competency Management, and Time and Attendance Packages.
- *NBC Hybrid Cloud*: This Hybrid Cloud offering allows customers to combine NBCGrid and NBCFiles with their existing infrastructure, creating front ends to complex Web applications and burstable storage and server capacity in concert with existing NBC or client physical infrastructure.

In addition to the Cloud services provided above, NBC also assists its agencies in determining the financial benefits of migrating to the Cloud, identifying which

Cloud services should be used for best business value, and how they should be integrated with their current systems. NBC also help its agencies assess their existing applications to identify the right applications, architecture, and operations plans to migrate to the Cloud. One example is helping agencies devise a strategy to maintain data privacy and protection standards.

2.5.4 National Institute of Standards and Technology

The NIST promotes the effective and secure adoption of Cloud technology within government and industry by providing technical guidance and standards. It acts as catalysts to help service, software, and hardware industries formulate their own standards. The current scope covers Cloud architectures, security, and deployment strategies for the federal government. The NIST is also participating in a group that will coordinate Cloud standards across Standard Development Organizations (SDOs). Figure 2.13 depicts NIST's view of the Cloud Computing Hierarchy, including different deployment models, delivery models, essential characteristics, and foundational elements and enablers [24–25].

As per the NIST's current road map, it sees the need for IaaS standards that should include *Virtual Machine Image* (VMI) distribution, VM provisioning and control, Inter-Cloud VM exchange, persistent storage, VM SLAs, and secure VM

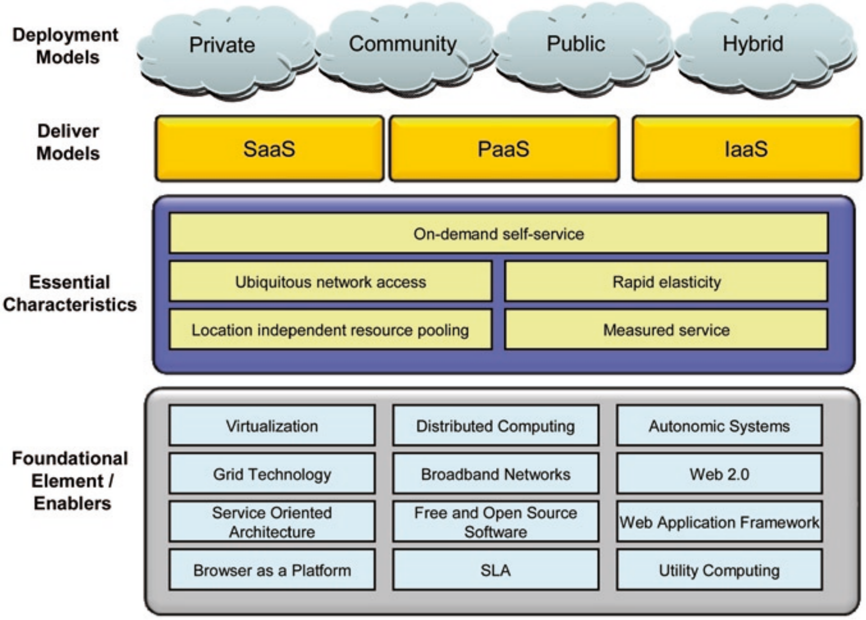


Fig. 2.13 NIST Cloud computing hierarchy

configuration. Although many VM applications exist in different private implementations, the NIST is looking for a unified specification for better interoperability. As for the PaaS standards, the NIST proposes the interest in improved programming languages and APIs for Cloud-specific service implementations. To support future SaaS standard implementations, the NIST recognizes the need for SaaS-specific authentication/authorization, data schemas for data import and export, and other application-specific standards and guidance. For cross layer integration, the NIST also indentifies the areas of interest in *Identity and Access Management* (IAM), data encryption, key management, *Records and Information Management* (RIM), and E-discovery.

In addition to the above federal-level government initiatives, many state-level government agencies are also actively introducing Clouds to their internal operations as well as services to their state residents. A survey conducted during the first two weeks of April 2010 by the nonprofit Public Technology Institute (PTI) studied 93 local government IT executives and found that 45% of local governments are using some form of Cloud Computing for applications or services. It also revealed that an additional 19% of local governments plan to implement some form of Cloud Computing within the next 12 months.

Among the local governments that have begun implementing Cloud services, the City of Los Angeles is one of the nation's first deployments of Cloud Computing. It chose Google's enterprise solution to turn the city's email infrastructure over to Google Apps Premier Edition in November 2009. Likewise, the Utah Department of Technology Services (DTS) is transforming its statewide IT infrastructure to a Private Cloud in order to achieve IT consolidation, virtualization, and SOA. The Virginia Information Technologies Agency scheduled a multi-year state-wide IT consolidation, targeting more than 90 agencies by using SaaS and SOA for enterprise applications and agency developed solutions. The Michigan Department of Information Technology issued SOA standards aiming to construct a new datacenter, state Cloud, and advanced virtualization of state agency servers. The Indiana Office of Technology consolidated five datacenters into one and reduced its server count by one third via virtualization technology. It also uses multiple SaaS platforms for incident reporting, newsletters, delivery tracking, and live chat assistance.

2.5.5 The U.S. Department of Defense

The *Global Information Grid* (GIG) is a central Network-Centric capability of the U.S. DoD. It represents global IT capabilities across all branches of service for the entire department. The GIG essentially consists of a set of Services that provide the underpinnings for providing the right information at the right place at the right time. Every capability, from security to messaging to management, is represented as a Service. The DoD integrated network services environment is illustrated in Fig. 2.14 [26].

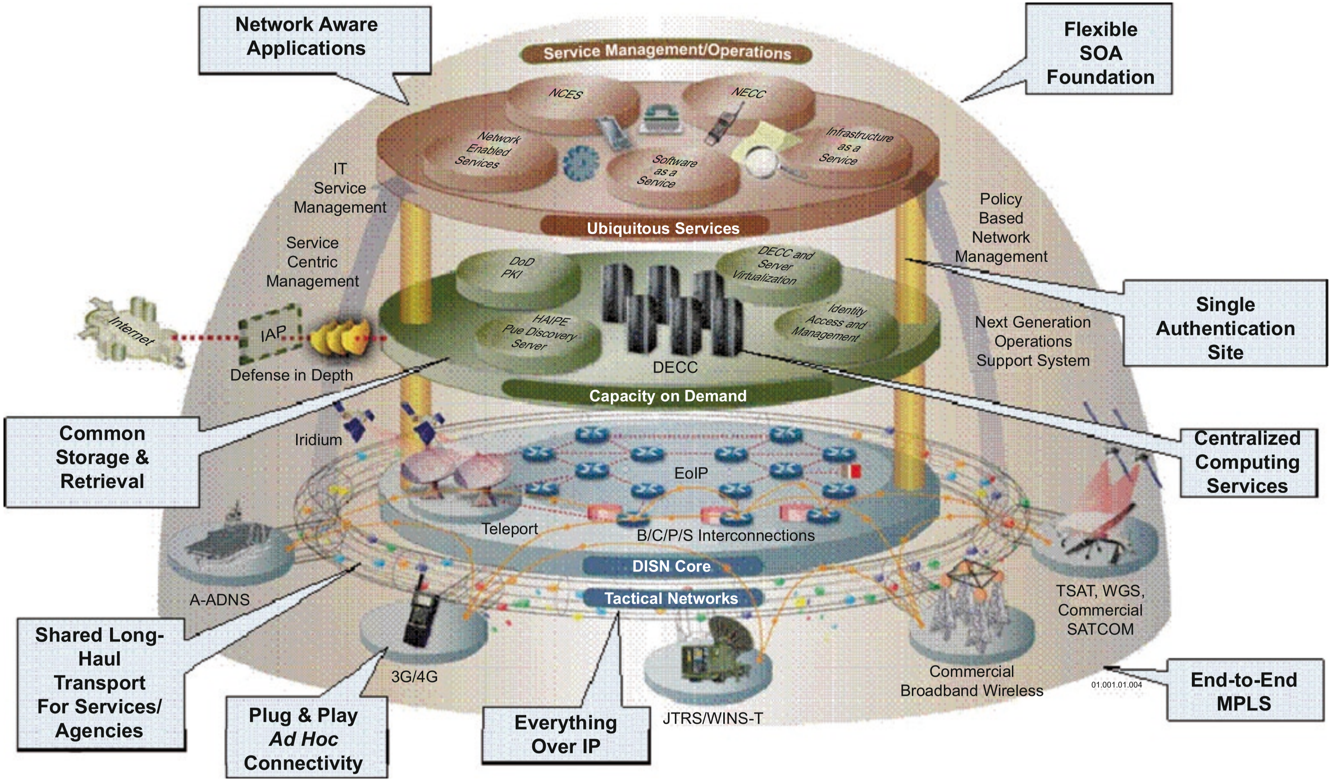

Network Aware Applications
Service Management/Operations
Flexible SOA Foundation
NCES
NECC
Network Enabled Services
Software as a Service
Infrastructure as a Service
IT Service Management
Ubiquitous Services
Policy Based Network Management
Service Centric Management
DoD PKI
DECC and Server Virtualization
Single Authentication Site
Internet
IAP
HAIPE Pue Discovery Server
Identity Access and Management
Next Generation Operations Support System
Defense in Depth
DECC
Capacity on Demand
Common Storage & Retrieval
Iridium
Centralized Computing Services
EoIP
Teleport
B/C/P/S Interconnections
DISN Core
Tactical Networks
A-ADNS
TSAT, WGS, Commercial SATCOM
Shared Long-Haul Transport For Services/ Agencies
3G/4G
Commercial Broadband Wireless
End-to-End MPLS
01.001.01.004
Plug & Play Ad Hoc Connectivity
Everything Over IP
JTRS/WINS-T

Fig. 2.14 *DoD* integrated network services

Net-Centricity focuses on effective information sharing in a complex environment. It also distills the urgency and importance of the military context because information itself proffers a new set of weapons, and even new battlefields. As a result, Net-Centricity focuses not only on leveraging shared IT capabilities to gain an advantage on opponents with traditional tactics, it also covers protecting or even launching information-based attacks.

The Defense Information Systems Agency (DISA) is a U.S. DoD combat support agency with the goal of providing real-time IT and communications support to the government, the military Services, and the Combatant Commands. As Cloud technologies are seen by the U.S. DoD as an obvious way to address enterprise-level information challenges, DISA is moving quickly to adopt Cloud technologies to process large data on networks more rapidly while realizing budgetary efficiency. The leading drivers include capital budget limitations, data and content storage, support of operational spikes, global application lifecycle management, and software development collaboration [26–31].

Before discussing Cloud services, let us first look at the concept of NCO and Network-Centric Enterprise Services:

- *Network-Centric Operations*: As a strategic military asset, information has always been a part of warfare. The core challenge of the U.S. DoD is managing who has information, how to share it, and how to rely upon it to make decisions. In a military operational context, it is the decision of *Command and Control* (C2). Due to this need, a strategic program called Network-Centric Warfare (a.k.a. Net-Centricity) was established during the late 1990s in response to the rise of the Internet. The idea of Network-Centric Warfare has gone through many phases, aiming to improve cooperation across the different branches of the department. Net-Centricity centers on supporting the military's C2 capabilities for true NCO. There are three dimensions to this information management. *The right information*: commanders on the battlefield need all relevant and reliable information from different forces, different locations, and different branches of the service. *In the right place*: commanders might call upon forces from hundreds of miles away, on land, at sea, in the air, or in space. *At the right time*: knowing where the opponents are right now is far more valuable then where they were an hour or a day ago. An extended concept of C2 was cited by the US Navy as the next generation C2 solution, which is called the *Command and Control of Command and Control* (C2C2). It is created to enhance the collaboration and cooperation among different C2 systems to strengthen the cohesive information power. A C2C2 system includes: theater sensing/intelligence, network/cyber architecture, commanders' decision aids that compile transmitted data into useful information, and network protection. Cloud technologies, when engineered correctly, make dramatic, positive changes to the mission assurance posture of the federal enterprise. Cloud technologies can enable stronger end-point security and better data protection. They also enable the use of thin clients and the many security benefits they provide [9, 32].
- *Network-Centric Enterprise Services (NCES)*: The NCES Program offers capabilities for members of CoIs to interact with each other through a SOA approach.

SOA provides essential best practices to facilitate a broad, architectural approach to achieving agile information sharing in complex organizations. It acts as a service infrastructure that enables NCO to drive collaboration among people and systems, allowing users to get more information, more quickly. The goal of this service is to provide unprecedented visibility to the value of information so decision makers can achieve superior decisions strategically and tactically. For example, NCES distributes services such as security applications over a network and combines and reuses these applications to create business applications that communicate and coordinate efficiently with each other. NCES offers four *Core Enterprise Services* (CES) They are: the *Service-Oriented Architecture Foundation* (SOAF), *Collaboration*, *Content Discovery & Delivery* (CD&D), and the *Portal*. Additionally, an important application of NCES is the *Command and Control Framework* (NECC). NECC provides the commander or warfighter with the data and information needed to make timely, effective, and informed decisions in a net-centric environment [33].

Today, NCO or NCES are not yet fully integrated with Cloud technologies. However, DISA has invested in the following three initiatives to provide true Cloud services to DoD agencies and members of CoIs. These initiatives are the GIG Content Delivery Service, the Rapid Access Computing Environment, and Forge.mil. Figure 2.15 shows the relationship between these three offerings and their mapping to the Cloud layer [27].

- *GIG Content Delivery Service* (GCDS): As aforementioned in the NCES, the CD&D services are essential in the DoD infrastructure to provide common specifications to expose, search, retrieve, and deliver information across the enterprise. Alternatively, the *NCES* Content Discovery focuses on Enterprise Searches (including Centralized Search and Federated Search) and Enterprise

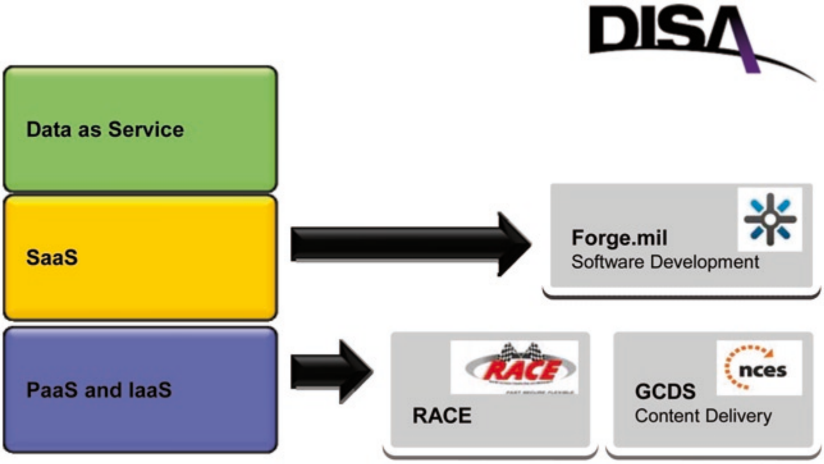


Fig. 2.15 DISA Cloud services portfolio

Catalogs. Content producers have the capability to make available and advertise their information products to content users across many CoIs. The *NCES Content Delivery*, on the other hand, supports the efficient delivery of mission critical information products to the warfighter and first responder sometimes over slow, limited, or even non-existent communications paths in scheduled or unanticipated situations. It offers two content delivery capabilities, namely the *Enterprise File Delivery* (EFD) and GCDS. EFD provides a multi-platform, peer-to-peer means to forward stage content and synchronize file directories across terrestrial networks (the *Non-classified Internet Protocol Router Network* (NIPRNet) and the *Secret Internet Protocol Router Network* (SIPRNet)) via satellites (*Global Broadcast Service*). GCDS is a DISA commercially managed solution designed to improve delivery of Web content to users on NIPRNet and SIPRNet via standard Web protocols (i.e., the *Hypertext Transfer Protocol* (HTTP) and the *Hypertext Transfer Protocol Secure* (HTTPS)). Using Cloud technology, GCDS can rapidly provide reliable and secure content and applications on-demand that account for IA and secure delivery of application data to geographically dispersed user communities more effectively. GCDS also demonstrates the scalability, reliability, controllability, and performance from the Cloud to efficiently obtain and distribute applications and content to end users regardless of network conditions [34–36].

- *Rapid Access Computing Environment (RACE)*: DISA's RACE provides quick-turnaround computing solutions to DoD customers with highly standardized computing platforms quickly, inexpensively, and securely. Its goal is to deploy new applications to military personnel more rapidly. Also based on commercial Cloud technology, DISA RACE provides a user, self-service provisioning portal, which allows DoD users to provision the software bundle using LAMP technologies—Linux, Apache HTTP Server, MySQL, and Hypertext Preprocessor (PHP), Python or Perl—or Windows servers within the production environment within 24 h. Today, RACE uses VMware running on HP blade servers. Users can choose Microsoft Windows or Red Hat Linux operating environments, and can configure their virtual servers with up to four CPUs, 8 GB of memory, and up to a terabyte of storage in 10 G increments. DISA says it has cut the acquisition time for a new server from six months to 24 h with RACE. RACE uses the same method of SLA inside the RACE environment (similar to the regular computing environment) and claims to achieve 99.999% availability at all times. RACE provides availability and performance of any DISA applications such as payroll, financial systems, and logistics systems. Hundreds of military applications including C2 systems, convoy control systems, and satellite programs have been developed and tested on its user-provisioned virtual servers. DISA also applies the same information assurance process to its Cloud-based applications that it applies to applications that run on traditional computing platforms. RACE initially featured the rapid delivery of Test & Development environments. Its latest release enables DoD users to use self-service provision operating environments within the highly secured Defense Enterprise Computing Center's (DECC) production environment. With its rapidly accessible and scalable computing infrastructure, RACE uses virtualization and the nearly unlimited capability of Cloud Computing to offer Defense Department customers PaaS/IaaS in test and production environments. This is the first of its kind for DoD [33, 37].

- *Forge.mil*: In April 2009, DISA established an open source/government source software lifecycle development cycle patterned after the open source community's SourceForge.net. It is a family of services provided to the U.S. military, DoD government civilians, and DoD contractors to support the DoD's technology development community. The goal of this program is to enable the DoD to improve software development efficiency and to drive collaborative dynamics that help quickly deliver better software to support net-centric operations and warfare. There are five services in this program:

  – *SoftwareForge*: It is a collaborative environment for shared development of open source and DoD community source software amongst distributed developers. It features a free public code repository/library, finds pre-existing source code, manages project lifecycles for public projects, shares new code with others, and collaborates with other DoD projects. The tools available in SoftwareForge include: *software version control*, *bug tracking, requirements management,* and *release packaging*, along with *collaboration tools* such as wikis, discussion forums, and document repositories.
  – *ProjectForge*: Hosted in a DECC, this is a SaaS version of SoftwareForge for private-access projects, supporting both unclassified and classified development efforts. It offers on-demand application development and lifecycle management tools for managing project lifecycles, team efforts, and collaboration with team members.
  – *TestForge*: Adopting common test and evaluation criteria with on-demand standard testing tools and methods, this service can eliminate duplicative testing and improve dependability.
  – *CertificationForge*: This service enforces development guidance and process management through this certification service to ensure large programs can be developed, fielded, and operated more efficiently and effectively.
  – *StandardsForge*: This service will drive collaborative IT standards development [27].

2.6 Scientific, Educational, and Others

Modern science is generating and using datasets that are increasing exponentially in both complexity and size. The amount of computing resources to perform appropriate levels of analysis, archival, and sharing becomes a grand challenge. From an application perspective, these challenges involve a broad range of technologies [38]:

- *High-Performance Computing (HPC)* is compute-intensive and typically contains high-performance I/O systems, wide-area networking, and parallel file systems in dynamic environments.

- *High-Throughput Computing (HTC)* focuses on using many computing resources over long periods of time.
- *Many-Task Computing* bridges the gap between HPC and HTC. It focuses on using many resources over short periods of time.
- *Data-Intensive Computing* focuses on data distribution and harnessing data locality by scheduling computations close to the data.

For research groups, Cloud technology provides convenient access to reliable, high performance clusters and storage without having to purchase and maintain sophisticated hardware. For developers, virtualization allows scientific software to be optimized and pre-installed on machine images and effectively controls the computing and storage resources. For instance, the National Science Foundation (NSF) established a computing infrastructure known as Science Gateways (a.k.a. hubs). These hubs offer scientists collaborative Websites with Web 2.0/3.0 technology for many scientific programs, such as large scale modeling and simulation. The following sections will show some key Cloud implementations from scientific, educational, and international applications.

2.6.1 US Department of Energy (DOE) and Magellan

In accordance with the American Recovery and Reinvestment Act, the US Department of Energy (DOE) takes a lead role in examining Cloud technology for its performance in cost-effective and energy-efficient applications for scientists. The DOE is exploring the Cloud concept with its federal partners to identify opportunities to provide better service at lower costs through Cloud services. The goal is to assess its impact for accelerating discoveries in a variety of disciplines, including analysis of scientific data sets in biology, climate change, and physics. These include protein structure analysis, power grid simulations, image processing for materials structure analysis, and nanophotonics and nanoparticle analysis. Because of the nature of this program, it is named Magellan in honor of the Portuguese explorer who led the first effort to sail around the globe [39].

The DOE is funding the project with $32 million, with the money divided equally between the Argonne Leadership Computing Facility (ALCF) in Illinois and the National Energy Research Scientific Computing Center (NERSC) in California. Both centers will install similar mid-range computing hardware, but will offer different computing environments. At NERSC, the program measures a broad spectrum of the performance of DOE science workload from its 3000 science users. Monitoring software is used to analyze what kinds of science applications are performing better in a Cloud environment. A current list of Cloud environments initiatives are as follows:

- To measure the comparative performance of scientific applications in a Cloud environment versus similar applications running on the current departmental cluster or supercomputing environment.

- To provide fast random access storage for data intensive applications. This environment uses flash storage for its substantially increased bandwidth, *I/O operation rate* (IOPS), and decreased latency.
- To investigate and test applications that can be ported to Cloud Computing models, such as Hadoop (MAP/Reduce).
- To provide science communities easy access to applications, databases, or automated workflows through a set of servers and software called "science gateways."
- The Private Cloud, consisting of 1440 Intel Nehalem quad-core processors (5760 cores total), will offer alternative models for access to computing resources based on research groups and time periods.
- To facilitate rapid information exchanges and enable scientists to use available computing resources regardless of location. These two centers will be linked by a 100 Gbit/s network, developed by DOE's Esnet.
- To maintain control over the user authorization process while using Cloud services, the program also explores hybrid solutions that have the ability to manage these two centers.

2.6.2 NASA Nebula

The NASA Ames Research Center in Silicon Valley initiated a Cloud Computing pilot called Nebula (Fig. 2.16). As a Hybrid Cloud, Nebula enhances NASA's

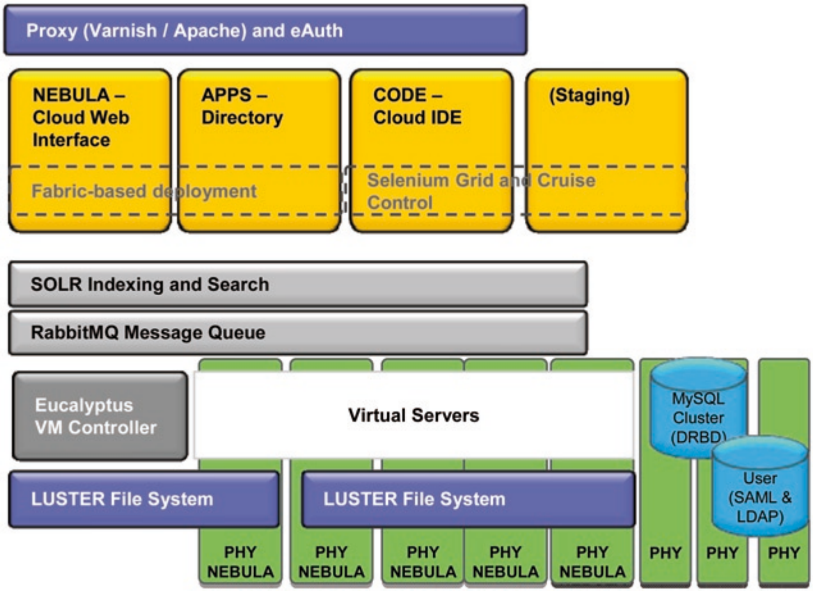


Fig. 2.16 NASA Nebula services

ability to collaborate with external researchers by providing consistent tool sets and high-speed data connections. Built from the ground up, Nebula is a collaborative mega-system created by thousands who seek improved operability with open-source technology [40].

The fully-integrated nature of the Nebula components provide for extremely rapid development of policy-compliant and secure Web applications. It also fosters and encourages code reuse and improves the coherence and cohesiveness of NASA's collaborative Web applications. Today, for instance, astronomy enthusiasts are informally working with NASA scientists by uploading high resolution photographs to get a better view of the Moon using the LCROSS participation site.

Nebula uses rack-dense, two *Rack Unit* (RU) servers with 12 Serial Advanced Technology Attachment (SATA) drives in a *Redundant Array of Independent Disks* (RAID) 6 configuration. The current configuration uses 1TB drives, for non-blocking access to 10TB of usable storage per server. In the current infrastructure, Nebula provides users with 4 CPU cores and 5TB of usable storage, per rack unit, at non-blocking network speeds. Three classes of storage (distinct hardware configurations) are offered for different applications:

- *Ephemeral*: VMs use ephemeral storage to run, but the information is on a local disk and is not saved by default. Nebula uses hot-swappable commodity drives in a hardware RAID configuration. This allows up to three drives to fail before data loss occurs.
- *Persistent Block Device*: Nebula uses the *Internet Small Computer System Interface* (iSCSI) to provide a persistent network storage block device. It provides highly-reliable and permanent storage, and decouples the storage from the connected server as a single point-of-failure.
- *Object Store*: To ease the storage of petabytes of data and billions of files, Nebula uses open-source implementations of object stores and adds custom code in the *access control layer* (ACL) and potentially the API layer. The compute layer is EC2, so right now S3 is being considered for compatibility with the popular Amazon Cloud technology.

The *Ames Internet Exchange* (AIX), which hosts the Cloud, was formerly called "Mae West," one of the original nodes of the Internet. It is still a major peering location for Tier 1 ISPs, as well as home of the "E" root name servers. In the local network, Nebula is built upon a converged 10Gig-E switching fabric. Each customer provisions a VPN within Nebula. Access to this private virtual network is provided over a dedicated VPN interface. For external connectivity, Nebula connects to CENIC and Internet2 at 10GigE connections. The Nebula is under development as the first IPv6-powered computing Cloud.

Eucalyptus is an API-compatible, open-source clone of the Amazon AWS Cloud platform. This provides NASA researchers with the simplest possible approach to access IaaS. All AWS-compatible tools will work "out-of-the-box" or with minor customization. The virtual server images within Nebula can easily be run on EC2 by outside partners, collaborators, or independent researchers.

Nebula is currently being used for education and public outreach, for collaboration and public input, and also for mission support. When completed, Nebula will offer cost-effective (1) IaaS for an evolution of Web hosting and virtual private server offerings; (2) PaaS for facilitating the deployment and installations of applications; and (3) SaaS for managing workflows, terms of service, and several levels of basic policy compliance, security, and software assurance of users who desire to utilize the underlying Nebula components.

2.6.3 Education

In the traditional education system, teachers convey knowledge to their students through the means of textbooks and homework. For students, the textbooks direct them and provide the information for them to learn. The students are eventually evaluated on whether or not they have learned the subject material by taking tests. From a teacher's perspective, textbooks and assisting (field/lab) materials provide teachers with activities and assessments that enable the teachers to deliver the lessons to their students and assess their recollection of the information. Teaching materials, including textbooks and lab materials, are organized and presented in accordance with academic standards and require students to comprehend their data in order to satisfy necessary examinations [41, 42].

Based on the current paradigm, most students and teachers are concerned more about the availability of teaching materials, tools, or other related resources and less concerned about where these resources are located or who is delivering them. Cloud technology can potentially make the accessibility of these materials very easy by simply allowing the teachers or students to request appropriate services from the Web. The majority of students and teachers are already familiar with Public Clouds, or consumer-based Cloud services such as those offered by Amazon, Google, Adobe, Expedia, or Facebook. Therefore, instead of going to the library to research subjects or going to school laboratories to work on homework, students can choose internet libraries, Web community sites, software applications, and the server capacity they need. They can also schedule server capacity requests to repeat for the entire semester or as needed.

Cloud technology introduces a new way for students and teachers to exchange ideas and communicate. It is so powerful that books, while still relevant, would just become a part of the way of thinking about learning. The advantages of Cloud technology for the education environment can be summarized as the following:

- *New Phase of Textbooks*: Textbooks create generations of passive and dependent learners. With Cloud technology, the knowledge exchange is no longer limited by the interaction between teachers and students. Financially, the cost of textbooks could be reduced dramatically if the books were digitalized and used only for the time they were needed. The textbooks could even be customized for every student based on their abilities and interests. Moreover, by using digital means, material can be kept up to date.

- *Cost Reduction of Learning Tools*: In addition to textbooks, students also have to spend hundreds of dollars on computer software in order to complete their assignments and prevent their computer from failing on them. With Cloud technology, students can use desired applications without the necessity of purchasing the software or worrying about upgrades. For educational institutions, or any type of organization, they will no longer have to purchase expensive software for an individual or a limited small number of employees or students.
- *Gain Experience from Real Environments*: For advanced studies, Cloud technology has made it easy to create realistic assignments. For instance, when studying about managing redundancy for scalability and high availability of a computing environment, students can actually work hands-on with all the resources from SPs without building or managing a datacenter. Furthermore, they can interact with a real environment and gain life experiences, where load balancers or Web server front ends are no longer needed in the classroom.
- *Scale Sizable Project*: For engineering or research projects, students often have to simulate various environments. In these applications, horizontal scalability is a critical design goal and the Cloud service allows students to change the size of their environments on demand. Therefore, if the project would have taken 100 local servers, for instance, instead of waiting for the school to release such a huge amount of resources, the students can acquire them in a few minutes and can release them once the assignment is over.
- *Simplify Courseware Management*: Courseware management can be simplified by virtualization technology. The teacher can compile a VM image containing a complete software stack and reference materials for a course. Each student or team can then deploy that image on their own server instance and instantly gain access to identical resources. Depending on the level of the curriculum, students can be granted appropriate privileges to manage their instances. Changes to the course's focus or damage to student instances can be easily recovered by re-instantiating the image from the teacher's server. During the development, students and teachers can actively communicate with each other via question boards, blogs, and other collaboration tools using commercial Cloud services.

In addition to the improvement of education-related medias, systems, and managements discussed above, many universities are assisting in researching advanced subjects of Cloud Computing and Cloud technologies. For example, the NSF funded Yale University, Massachusetts Institute of Technology (MIT), and University of Wisconsin at Madison for cluster-based, large-scale data analysis through the NSF's Cluster Exploratory (CLuE) program. NSF also funded Boston University's *Colocation Games* (CGs), a general framework for modeling, analyzing, and facilitating the interactions between stakeholders in Cloud environments. The research initiatives, such as the University of Massachusetts' Amherst Center for Intelligent Information Retrieval (CIIR), the University of Virginia's Feedback-Controlled Management of Virtualized Resources project, the Virginia Tech and NC State University's

Hybrid Opportunistic Computing for Green Clouds, and Wayne State University's automated configuration processes of virtualized machines and Cloud applications, are also funded by NSF.

Other than these NSF sponsored projects, many researchers in different universities are also proactively contributing to these domain subjects. For instance, the Institute for Genome Sciences at the University of Maryland School of Medicine developed the *Cloud Virtual Resource* (CloVR) to provide a new community resource for sequence analysis in environmental and biomedical research. The University of Santa Barbara's Massive Graphs in Clusters (MAGIC) project focused on developing software infrastructure that can efficiently answer queries on extremely large graph datasets. These are among many ongoing projects that are sponsored by government agencies or private enterprises.

2.6.4 Other International Organizations

In addition to the U.S. organizations and government agencies that have adopted Cloud technology as mentioned above, many international institutes and governments also see the value of Clouds and are in the process of establishing their efforts to install this technology within their organizations. Below are some good examples [43, 44].

The UK Government's CIO established a private Government Cloud Computing infrastructure called *G-Cloud*. The program includes IaaS, PaaS, and SaaS. The business objective is to enable public bodies to host their *Information and Communication Technology* (ICT) systems from a secure, resilient, and cost-effective service environment. Services are available from multiple suppliers, which will allow public sector bodies to switch suppliers quicker and cheaper than was previously possible. Using G-Cloud, the UK Government increases the efficiency of shared services and helps government systems broaden the use of ERP systems across central and local governments. The *Government Applications Store* provides greater visibility of applications that can be shared across the public sector. Examples include electronic document and records management, banking, vetting, and so forth. The G-Cloud is a key enabler of the £3.2 billion savings per year outlined in the *Operational Efficiency Programme* as it provides the access point for ICT services, applications, and assets. From a SP's perspective, this program rationalizes the government ICT estate to increase overall capability and security, reduce costs, and accelerate deployment speeds.

The Canadian Government's CTO of Public Works Government Services presented a paper on Cloud Computing and the Canadian Environment. This paper advocated the Canadian Government's advantage as a prime location for the construction of large energy efficient datacenters. This is mainly due to its geographical characteristics, cooler temperatures, low-density population, IT expertise, quality construction standards, legislative framework (including the Privacy Act and the Personal Information Protection and Electronic Document Act), and low-cost green energy. The Government of Canada could also engages with provincial, territorial, and municipal counterparts in defining Canada's Cloud Computing position through a comprehensive Canadian Cloud Computing Strategy [45].

Japan's *Ministry of Internal Affairs and Communications* (MIC) outlines the *Digital Japan Creation Project* (ICT Hatoyama Plan) with the aim of actively introducing new technologies to create an innovative electronic government to help boost Japan's economy. This includes a nation-wide Cloud Computing infrastructure tentatively called the *Kasumigaseki Cloud*. Proposed to be completed by 2015, the *Kasumigaseki Cloud* enables various ministries to collaborate to integrate and consolidate hardware and create platforms for shared functions. *Green Cloud Datacenters* are designed to support this Cloud. They reduce energy consumption by being located in cold regions, utilize wind and solar power, and employ low-loss direct current. The facilities will use tunnels and other underground sites with strong earthquake resistance and stable temperatures. As for the Cloud service, the *National Digital Archive* will be developed to provide the highest degree of access to digitized government documents such as books and scholarly articles, cultural property information, geographic and time space information, statistical information, and other high demand information. Figure 2.17 portrays a high-level view of this Cloud network [46].

The Seventh Framework Programme (FP7) bundles all research-related EU initiatives together under a common roof. The broad objectives of the FP7 have been grouped into four categories: *cooperation*, *ideas*, *people*, *and capacities*. The FP7 is funding several projects on Cloud Computing and has also compiled a group of experts to outline the future direction of Cloud Computing research. Figure 2.18 portrays the FP7 Service and Software Architecture, Infrastructure, and Engineering Objective. Although it does not completely follow either SOA or Cloud hierarchy, this map lays out compatible layers of services based on the scopes of interest and the associated project titles [47].

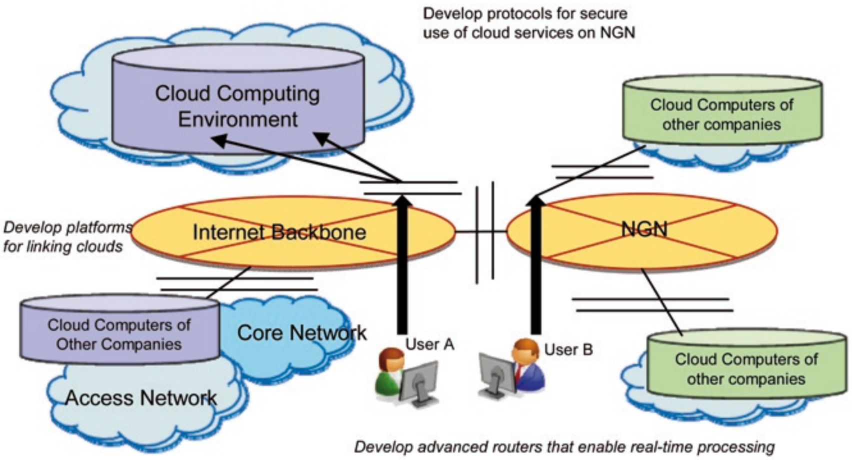

Develop protocols for secure use of cloud services on NGN
Cloud Computing Environment
Cloud Computers of other companies
Develop platforms for linking clouds
Internet Backbone
NGN
Cloud Computers of Other Companies
Core Network
User A
User B
Cloud Computers of other companies
Access Network
Develop advanced routers that enable real-time processing

Fig. 2.17 The Cloud in digital Japan creation project

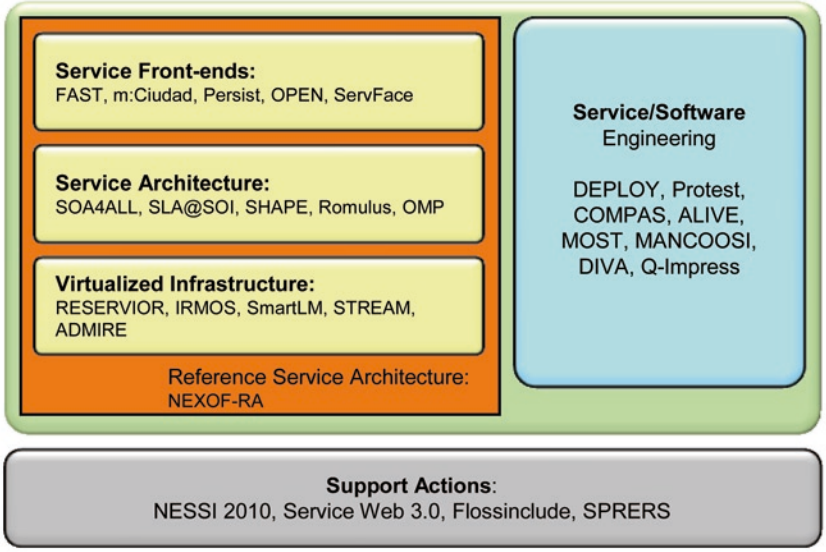

Service Front-ends:
FAST, m:Ciudad, Persist, OPEN, ServFace
Service Architecture:
SOA4ALL, SLA@SOI, SHAPE, Romulus, OMP
Virtualized Infrastructure:
RESERVIOR, IRMOS, SmartLM, STREAM, ADMIRE
Reference Service Architecture:
NEXOF-RA
Service/Software
Engineering
DEPLOY, Protest, COMPAS, ALIVE, MOST, MANCOOSI, DIVA, Q-Impress
Support Actions:
NESSI 2010, Service Web 3.0, Flossinclude, SPRERS

Fig. 2.18 Software architecture, infrastructure and engineering

Some Cloud and SOA-related projects are listed below by category to show examples of current research directions [47]:

1. *Service Front-ends*: *ServFace* is an extended SOA concept, providing service annotations for correspondent UI compositions.
2. *Engineering*: *Q-IMPRESS* provides quality impact predictions for evolving service-oriented and quality sensitive software, such as industrial production control and telecommunications.
3. *Service Architectures*: *SOA4All* abstracts from software and treats billions of resources as services in a SOA via advanced Web technology.
4. *Virtualized Architectures*: The *Resources and Services Virtualization without Barriers Project* (RESERVOIR) aims to develop technologies to support a service-based online economy, where resources and services are transparently provisioned and managed. It introduces an ICT infrastructure for the reliable and effective delivery of services as utilities.
5. *Support Actions*: *Service Web 3.0* captures revolutionary changes at all levels of computing from the hardware through the middleware and infrastructure to applications and intelligence.
6. *Virtualized Architectures*: SmartLM is a grid-friendly software licensing solution for location-independent application execution. It provides a generic and flexible licensing virtualization technology for new service-oriented business models across organization boundaries.

Many Asian governments also actively promote Cloud services as a part of their e-government offerings. For instance, the Yellow River Delta Cloud Computing

Center in Dongying and the Cloud services Factory in Wuxi are two examples in China. The Government Information Technology Service (GITS) of Thailand established a Private Cloud for use by Thai government agencies. The Ministry of Economic Development (MED) of New Zealand is developing the business.govt.nz Web portal, which aims to provide network assistance to small and medium enterprises and their consultants and advisors [48].

2.7 Conclusion

Cloud services are in the early stages of a long-term evolution from traditional IT to computing as a utility service. Through automation and virtualization technologies, enterprises can now abstract the complexity of accessing vast amounts of networked resources and information. A large, growing number of vendors are creating solutions to simplify the ability to exploit services, platforms, and infrastructure in the Cloud.

Using the new technology, SPs transform huge, centrally managed and operated datacenters to simple “pay by use” business models. The commercial and government IT-related industries are pressing full speed ahead in adopting this scalable, distributed IT and management methodology. This can potentially enable their employees and customers to access applications, computer resources, and information as needed, without having to acquire, support, or maintain the underlying hardware or software.

Throughout this chapter, we have seen the four use cases of Cloud services, namely Public, Community, Private, and Hybrid Clouds, and their applications in various industries. As mentioned previously, our market analysis focuses more on the business drivers and implementations of the technology transformation, and thus pays less attention to the statistical numbers of their market share or enterprise expenditure distributions. Although rough estimates are derived from major research institutes and concluded in different market sections, the authors believe these numbers are changing rapidly based on the current momentum of Cloud technology’s maturing process. Any per-industry prediction will look misleading, as many major SPs are cross-industrial (e.g., BT in telecommunication and Amazon in retail) due to their virtualization services. As a result, the boundaries of Clouds are blurring and the accuracy of market size estimates will be quite different from the actual allocations. This is especially true when enterprises are heavily used in Public Cloud technology.

The authors believe a major driving force of the technology transformation is not solely from the commercial industry, as many government agencies are in the process of introducing Cloud technology. These changes will greatly impact how enterprises use and manage data. Furthermore, as the development environments are more accessible to everyone, it is expected that substantial growth of these types of applications for the public will take place soon. The driving forces and potential means to realizing these changes will be discussed in the following chapters.

References

1. Rotman, D.: Privacy in the Cloud—Impact to Auditors, KPMG. Nov 2009. http://www.isaca-sv.org/PrivacyintheCloud.pdf
2. http://csrc.nist.gov/groups/SNS/Cloud-computing/index.html
3. Lakshmanan, G., Pande, M.: How the Cloud stretches the SOA scope, MSDN architecture center. Arch J, http://msdn.microsoft.com/en-us/architecture/aa699420.aspx
4. Boos,C.: Cloud computing needs automation, Chris Boos on automation. http://www.hcboos.net/2008/05/Cloud-computing-needs-automation/ (2008). 21 May 2008
5. Newman, A.: Automation tools to drive Cloud computing journey, serverwatch. http://www.serverwatch.com/virtualization/article.php/3828816/Automation-Tools-to-Drive-Cloud-Computing-Journey.htm (2009). 8 July 2008
6. Cerf, V., Burleigh, S., Hooke, A., Durst, R., Scott, K., Fall, K., Weiss, H.: Delay-tolerant networking architecture, IETF. April 2007. http://www.ietf.org/rfc/rfc4838.txt
7. Jander, M.: Of space travel, new protocols & Dr. Cerf, internetevolution. http://www.internetevolution.com/author.asp?section_id=625&piddl_msgorder=asc&doc_id=182277 (2009). 24 Sept 2009
8. Wayner, P.: Cloud versus cloud: A guided tour of Amazon, Google, AppNexus, and GoGrid, InfoWorld. http://www.infoworld.com/d/Cloud-computing/Cloud-versus-Cloud-guided-tour-amazon-google-appnexus-and-gogrid-122 (2008). 21 July 2008
9. Van Buskirk, S.: Luncheon & Keynote address, TechNet Asia-Pacific 2009. http://www.afcea.org/events/asiapacific/09/intro.asp (2009). 2–5 Nov 2009
10. Albanesius,C.: Gartner: Cloud computing to jump 21% this year. PCMAG. http://www.pcmag.com/article2/0,2817,2343925,00.asp (2009). 26 March 2009
11. Chong, F., Miguel, A., Hogg, J., Homann, U., Zwiefel, B., Garber, D., Joseph, J., Zimmerman, S., Kaufman, S. Design considerations for S+S and Cloud computing, MSDN architecture center. Arch J. Sept 2009. http://msdn.microsoft.com/en-us/architecture/aa699439.aspx
12. Worldwide Cloud computing market shares, strategies, and forecasts, 2009–2015, WinterGreen Research, Inc., July 2009, p. 712. http://www.researchandmarkets.com/reports/1057978
13. 10 trends that will shape the future of financial institutions, Ulitzer. http://www.wikinvest.com/wikinvest/api.php?action=viewNews&aid=959665&page=Industry%3AFinancial_Services&comments=0&format=html (2010). 22 Feb 2010
14. Engebretson, J.: FCC provides additional broadband plan details, connectedplanetonline. http://connectedplanetonline.com/residential_services/news/broadband-plan-details-0222/index.html (2010). 22 Feb 2010
15. Cloud computing to generate $45.5 billion in revenue by 2015, says Telecom Trends International, Telecom Trends International, Inc. http://www.telecomtrends.net/Press%20release-CLOUD%20COMPUTING.pdf
16. TM Forum rallies industry giants to create ecosystem to accelerate Cloud services adoption, TM Forum. http://www.tmforum.org/TMForumPressReleases/TMForumRalliesIndustry/40561/article.html (2010). 4 March 2010
17. Service model catalyst getting the provider to Cloud 9, TM Form. April 2010. http://www.tmforum.org/TMFBoothPresentations/2413/home.html
18. U.S. Federal Cloud computing market forecast 2010–2015, PRLog. http://www.prlog.org/10240263-us-federal-cloud-computing-market-forecast-2010-2015.html (2009). 20 May 2009
19. U.S. Government Launches Cloud Computing Mall, Market Research Media. http://www.marketresearchmedia.com/2009/09/16/u-s-government-launches-Cloud-computing-mall/ (2009). 16 Sept 2009
20. U.S. Federal Cloud computing market forecast 2010–2015. Mark Res Media. http://www.marketresearchmedia.com/2009/05/20/us-federal-Cloud-computing-market-forecast-2010-2015/ (2009). 20 May 2009

21. Niemann, B.: Cloud computing: Informal presentation to the enterprise architecture WG. http://federalCloudcomputing.wik.is/@api/deki/files/94/=BrandNiemannCloudComputing-forEAWG08272009.ppt (2009). 27 Aug 2009
22. FEA practice guidance, federal enterprise architecture program. Nov 2007, http://www.whitehouse.gov/omb/assets/fea_docs/FEA_Practice_Guidance_Nov_2007.pdf
23. NBC's Federal Cloud playbook, NBC. Aug 2009. http://Cloud.nbc.gov/PDF/NBC%20Cloud%20White%20Paper%20Final%20(Web%20Res).pdf
24. http://ornot.files.wordpress.com/2009/08/Cloud-computing-paradigm-chart1.jpg
25. Cloud computing, NIST. csrc.nist.gov/groups/SNS/cloud-computing/
26. Sienkiewicz, H.J.: Cloud computing: An operational perspective. www.cio.gov/Documents/Cloud_Com_Operational_View_Sienkiewicz.ppt (2009). 27 Feb 2009
27. Sienkiewicz, H.J.: Cloud computing: A perspective, Defense Information Systems Agency. Sept 2009, http://www.au.af.mil/au/awc/awcgate/disa/Cloud_computing_and_saas.ppt
28. Gourley, B.: Cloud computing and net centric operations. http://ctovision.com/wp-content/uploads/2009/01/Cloud_Computing_and_Net_Centric_Operations-3.pdf (2008). 30 Dec 2008
29. Bloomberg, J.: Net-centricity: SOA in battle, ZapThink. http://www.zapthink.com/2009/08/19/net-centricity-soa-in-battle/ (2009). 19 Aug 2009
30. Cloud computing pay-per-use for on-demand scalability. http://www.grid.org.il/
31. Jackson, K.: Department of defense Cloud advances presentation at Cloud expo. http://soa.sys-con.com/node/1242690 (2010). 11 Jan 2010
32. Ackerman, R.K.: TechNet Asia-Pacific 2009 Day 4—SIGNAL's Online Show Daily, SIGNAL Online. http://www.afcea.org/signal/articles/templates/Signal_Article_Template.asp?articleid=2119&zoneid=276 (2009). 5 Nov 2009
33. Contract and buyers guides: 8 avenues to smart buying, GCN. http://gcn.com/microsites/reports/dod-security-buyers-guide/enabling-sharing.aspx
34. http://www.disa.mil/nces/product_lines/gcds.html
35. http://www.disa.mil/nces/product_lines/content_discovery.html
36. www.disa.mil/nces/product_lines/gcds.html
37. DoD embraces Cloud computing. http://www.defensemarket.com/?p=67 (2009). 21 Oct 2009
38. Science Cloud 2010 Workshop, ScienceCloud. http://dsl.cs.uchicago.edu/ScienceCloud2010/
39. DOE to explore scientific cloud computing at Argonne, Lawrence Berkeley National Laboratories, Berkeley Lab. http://newscenter.lbl.gov/press-releases/2009/10/14/scientific-Cloud-computing/ (2009). 14 Oct 2009
40. Nebula Cloud computing platform. http://nebula.nasa.gov/services/
41. Cloud computing in education, UC Berkely iNews, http://inews.berkeley.edu/articles/Spring2009/Cloud-computing (2009). Publication Date: 17 March 2009
42. Harrison, D.: Is Cloud computing a credible solution for education? Camp Technol. http://campustechnology.com/articles/2009/11/12/is-Cloud-computing-a-credible-solution-for-education.aspx (2009). 12 Nov 2009
43. Cloud book. http://www.Cloudbook.net/ukCloud-gov
44. The Government Cloud (G-Cloud). http://www.cabinetoffice.gov.uk/cio/ict/ict_strands/g_Cloud.aspx
45. Global Government Cloud Computing Roundtable. Cloud Computing and the Canadian Environment. http://www.scribd.com/doc/20818613/Cloud-Computing-and-the-Canadian-Environment
46. Digital Japan Creation Project (ICT Hatoyama Plan): Outline, Ministry of Internal Affairs and Communications. http://www.soumu.go.jp/main_sosiki/joho_tsusin/eng/Releases/Topics/pdf/090406_1.pdf (2009). 17 March 2009
47. Information & Communication Technologies. Projects & Clusters, Cordis. http://cordis.europa.eu/fp7/ict/ssai/projects_en.html
48. Wyld, D. C.: The Cloudy future of government IT: Cloud computing and the public sector around the world, Int. J. Web Semant. Technol. (IJWesT) **1**(1), (Jan 2010)

Chapter 3
Cloud Service Architecture and Related Standards$_{3,2}$

Many enterprises plan to migrate their IT infrastructures to Cloud-based infrastructures through a phased approach. With the existing enterprise systems having arcane and inconsistent interfaces, the implementations have a tendency to develop into more complicated process flows, consisting of many subsystem interfaces to accommodate existing processes. In some cases, enterprise IT systems need to duplicate some functions to maintain consistency of business information so that enterprises can make sound financial decisions. These issues, however, are not the intent of this book. Instead, our approach is to look at the Cloud service architecture as a clean sheet scenario, peeling off issues and challenges layer by layer to reveal relevant, ultimate solutions.

While on-demand service is an outgrowth of timesharing, virtualization, and datacenters, ther Cloud service architecture is now a benchmark of new IT development. Through real or virtual agents, new generation SLAs are likely to offer a rich range of services by following mature, standardized guidance. From a user perspective, mainstream consumers will aggressively try to decrease the cost of their computing devices and be more receptive to having their client machines run free or open-source applications than the consumers currently do. Software market cycles will soon shorten due to the ease of accessibility to Cloud development platforms. Rather than the glacial pace of multi-year upgrade cycles in the current IT industry, multiple releases per year will soon become the norm. This will be rapidly accelerated even more by the development of abstracting hardware and software from the OS and software from software. All these attributes, with respect to decoupled, distributed, and mash-able "fabrics," will impact the fundamental architecture of enterprise Cloud services.

Technologically, the Cloud is a culmination of standards and technologies that have come together to form a new type of business operation. This chapter provides a view into architectural considerations and standards as they affect common architectural domains, such as enterprise, software, and infrastructure architecture. To make informed decisions and take full advantage of the potential benefits of adopting a Cloud service model, IT architects and decision makers must weigh the business drivers and technical requirements against the economic, regulatory, political, and financial landscapes surrounding the company. Industry standards that enforce

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_3, © Springer Science+Business Media B.V. 2010

the engagement of their partners and customers for business improvements will be an essential factor for their success. These include the standards of management and operation, applications, clients, platforms, services, storage, and more.

3.1 Overview

Based on the general considerations and visions for Cloud technology and its applications, Clouds should be uniquely identifiable so that they can be individually managed even when under federations of Clouds or when combined with other Clouds. From customers' perspective, users view the Cloud differently depending on their role within the organization. This will be necessary to distinguish and harmonize Cloud business and infrastructure policies in force. This chapter aims to systematically examine the different infrastructures and enterprise services. Figure 3.1 depicts the general infrastructure of the Cloud, which includes integration, services, and management. As the figure indicates, services are usually composed of software applications, platform services, infrastructure services, and physical infrastructure. As seen on the right side of the diagram, management aspects typically include service management (service fulfillment, service provisioning, service assurance, etc.), customer services, and information assurance.

From management's perspective, the following three characteristics are essential to any enterprise Cloud [1]:

1. Configurations are dynamic and automated (or semi-automated) in varying and unpredictable ways, and possibly even include event-driven conditions.

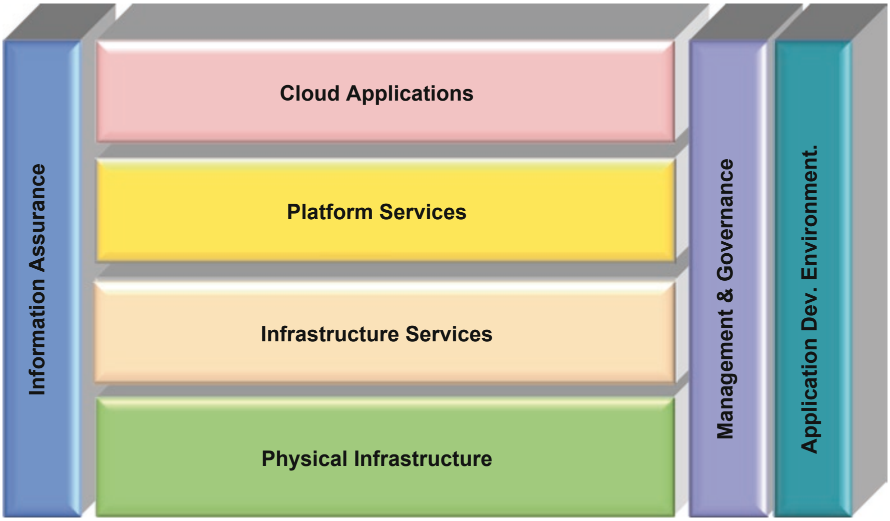


Fig. 3.1 General Cloud infrastructure

2. Systems management technologies are scalable so that they are manageable in aggregate conditions (e.g., integration of business constraints with infrastructure constraints).
   (a) A Cloud is dynamically provisioned and able to optimize its own construction and resource consumption over time.
   (b) A Cloud is able to recover from routine and extraordinary events.
   (c) A Cloud is aware of the context in which it is used, thus the Cloud's contents dynamically behave accordingly (e.g., if Clouds are combined and composited, necessary types of policies will have to be harmonized across Cloud boundaries). Application platforms today are unaware of their usage context, however business functionality in next generation platforms will have to be managed with context in mind.
3. A Cloud is secure and has the necessary information assurance capabilities.

Cloud Computing has numerous, well-known predecessors and technologies, including utility computing, Grid Computing, virtualization, hypervisors, etc. As shown in Chap. 1, one technological concept that does not always enter the Cloud conversation, but definitely should, is SOA. SOA has played a role in enabling Cloud environments to become what they are today, and will also play a significant role in the evolution of Cloud technologies.

In many ways, Cloud Computing can be seen as an extension of SOA past applications and into application and physical infrastructure. As enterprises and Cloud providers look to provide Cloud solutions, their basic goal will be to enable the enterprise IT infrastructure as a service.

The lessons learned in integrating and providing enterprise applications as discrete services should also be applied as the infrastructure layers are organized and provided as a service. The application and physical infrastructure, much like applications in SOA, must be discoverable, manageable, and governable. Ideally, much like with SOA, open standards will evolve that dictate how the services are discovered, consumed, managed, and governed. These standards sum up the entire lifecycle of a Cloud solution [1].

Figure 3.2 captures the idea of the three-layered Cloud service approach, and shows how each of those layers are essentially offering services to an overall SOA.

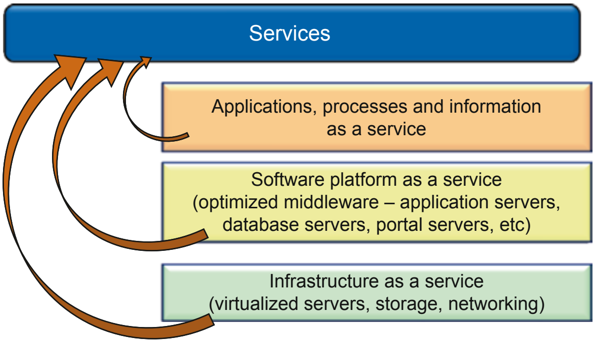

Services
Applications, processes and information as a service
Software platform as a service (optimized middleware – application servers, database servers, portal servers, etc)
Infrastructure as a service (virtualized servers, storage, networking)

Fig. 3.2 Enterprise Cloud services

In some cases, the services in the bottom two layers are presented as part of a SOA, but the important part is the recognition of the service-based approach to all layers of the Cloud.

Cloud Computing is poised to be a significant player in the technology industry now and in the foreseeable future. In its ultimate form, it will provide the means for IT to be delivered to consumers as a service. Products and service offerings in the Cloud space continue to grow and underscore the fact that this is where things are heading. The following sections will offer a closer look at Cloud service architecture, survey some of the most related standards, and offer solutions that are moving Cloud technologies from an idea to bottom-line returns for enterprises.

As mentioned previously, commercial network architecture is typically designed with a number of horizontal network layers, each with a distinctly unique purpose. The connectivity services are typically separated from end-user services. The convergence of networks and IT, driven by Web technologies, has forced digital services into distributed computing environments. Customers are demanding SLAs at the level of distributed applications, rather than at the level of standalone products. As a result, a sophisticated mesh of revenue models directs commercial flows across the value network. The challenge to the SP is how to manage and operate the set of services and infrastructure effectively. This task involves people, processes, and systems. In the new world of distributed value chains, enterprises and providers rely upon proven and well-adopted industry standards with clearly defined procurement specifications to be agreed to with equipment and enterprises.

The majority of operational problems stem from the underlying business processes, systems, and data. In order to be competitive in this global economy, one has to react to change and bring its products to market faster and better than the competition. A holistic, service-oriented EA model is the enterprise model of choice to meet the new challenges in this evolving global economy. By leveraging and extending

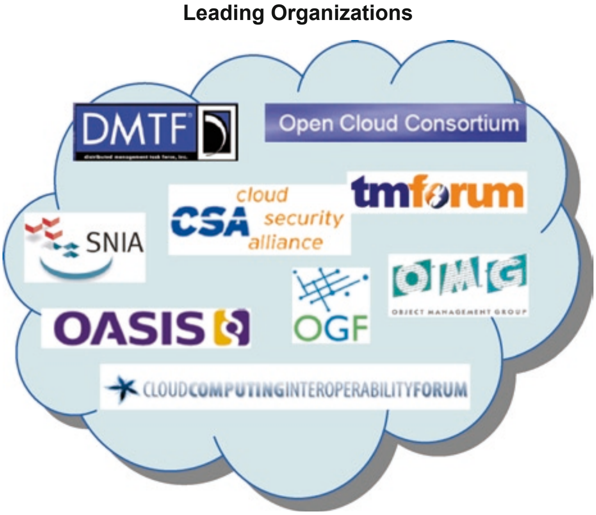


Fig. 3.3 Sample industry standards and forums

industry standards and best practices, enterprises can ensure and improve interoperability, manageability, performance, scalability, and supporting service modeling and interfacing (e.g., standardized service contracts, service loose coupling, service abstraction, service reusability, service autonomy, service discoverability, service composites, etc.).

The later sections of this chapter will survey a wide range of well-known and well-documented industry standards, shown in Fig. 3.3, for a number of Cloud Computing relevant areas [2].

3.2 Types of Cloud Services

Cloud Computing solutions come in multiple forms: *public*, *hybrid*, *community*, *and private*. First, let us take a look at the layers of the Cloud. Figure 3.4 is a distillation of what most agree to be the three principle components of a Cloud model. This figure accurately reflects the proportions of IT mass as it relates to cost, physical space requirements, maintenance, administration, management oversight, and obsolescence. Further, these layers not only represent Cloud anatomy, they also represent IT anatomy in general.

3.2.1 *Software as a Service*

SaaS is perhaps the most familiar to everyday Web users. The application services layer host applications that fit the SaaS model. These are applications that run in a Cloud and are provided on demand as services to users. Sometimes the services

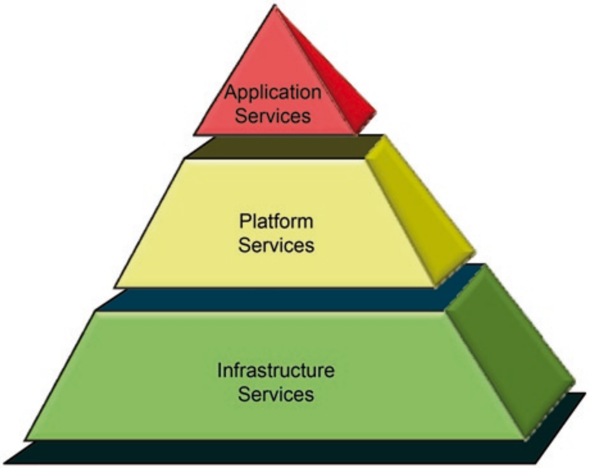


Fig. 3.4 Types of Cloud services

are free and providers generate revenue from things like Web ads. Other times, application providers generate revenue directly from the usage of the service. This top layer of the Cloud is deeply embedded into our daily lives such as filing taxes online using Turbo Tax, check our emails using Gmail or Yahoo Mail, or keep up with appointments using Google Calendar. These are just a couple of examples of these types of applications. There are literally thousands of SaaS applications, and the number grows daily thanks to Web 2.0/3.0 technologies. Perhaps not quite as apparent to the public at large is that there are many applications in the application services layer that are directed to the enterprise community. There are hosted software offerings available that handle payroll processing, HRM, collaboration, CRM, business partner relationship management, and more. Popular examples of these offerings include IBM Lotus Live, IBM Lotus Sametime, Unyte, Salesforce.com, Sugar CRM, and WebEx. In all cases, applications delivered via the SaaS model benefit consumers by relieving them from installing and maintaining the software, and can be used through licensing models that support *pay-per-use* concepts [1, 3].

SaaS helps enterprises improve the efficiency of existing client-server applications, allowing services to be more effective over the Internet. It also expands the scope of existing web applications, whether it focuses on business-to-business or business-to-consumer applications. In order to employ SaaS, the enterprise has to first understand the complexities of delivering SaaS in a multi-customer environment. As organizations continue to adopt outsourced models for automating critical business processes, SaaS is becoming more attractive for many different types of SPs as well as ISVs. Under this model, software features can be easily enabled or disabled by customers or users based on a specific industry, work environment, or other criteria.

Through this single-source approach, SPs reduce internal operating costs and help lower the total cost of ownership for customers. Implementation time is shortened and greater user acceptance is achieved. Figure 3.5 depicts SaaS in a Cloud Computing infrastructure. Some of the challenges of implementing SaaS include [4–6] the following:

- *Multi-tenant deployment*: Multi-tenant platforms use common resources and a single instance of both the object code of an application as well as the underlying database to support multiple customers simultaneously. Current Web 2.0/3.0 deployments utilize the multi-tenant deployment, in which applications aim to facilitate collaboration and sharing between users. Very few standards have been established for multi-tenant application delivery or the operational governance to ensure isolation among customers. Questions may surface regarding the suitability of the SaaS model for mission-critical applications. Several solutions to the issues associated with multi-tenant deployment exist, namely having separate databases per customer, a shared database but separate schemas, or a shared database and shared schemas.
- *Scalability*: Given that SaaS applications are delivered via the Internet, the major challenges that apply to scalability are performance and load management. The

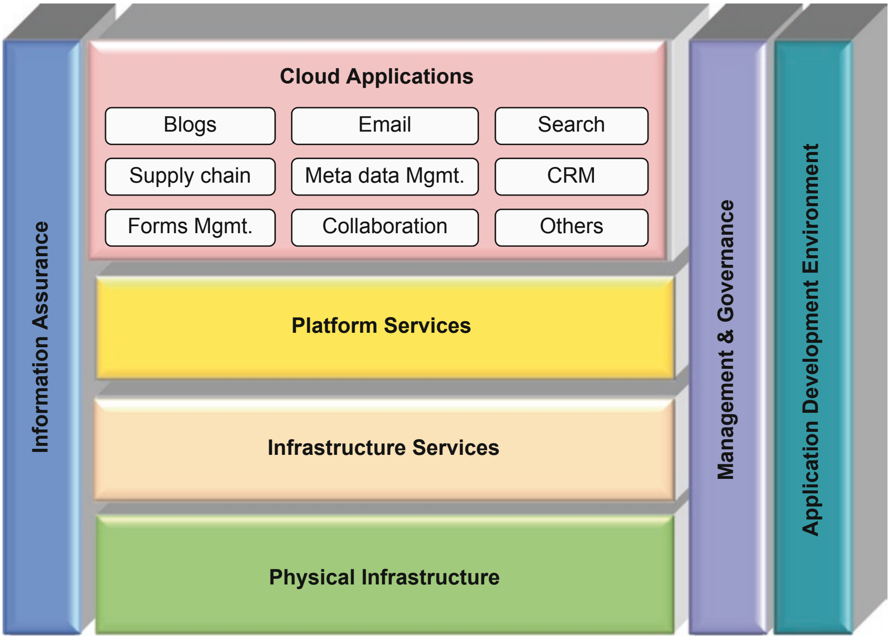


Fig. 3.5 Cloud computing infrastructure—SaaS

designs of an application's architecture, database schema, network connectivity, available bandwidth, etc., are all effecting and complex factors of the deployment.

- *Reliability*: Reliability is the level of accuracy in which an application provides its intended services, usually dictated by user documentation or application specifications. In addition, reliability is about providing correct results and handling error detection and recovery in order to avoid failures.
- *Usability*: The trend in application development is migrating towards a more dynamic user experience. Many SaaS providers are leveraging *Asynchronous JavaScript and XML* (AJAX) to improve the overall user experience.
- *Data Security*: Data security means ensuring that data is guarded from corruption and that access to data is controlled. The very nature of SaaS poses security challenges. In order to detect and prevent intrusion, adequate strong encryption, authentication, and auditing must be a part of the application design to restrict access to private and confidential data.
- *Auditing*: Auditing involves two aspects: *auditing of security and auditing of information*. Security audits are when a third party validates a managed SP's security profile. Information audits refer to a subsystem that monitors actions to, from, and within an application.
- *Data ownership*: Data protection and ownership are probably the most difficult challenges of SaaS. The difficulty arises when the data owning party and the data safeguarding party are not the same. In addition to safeguarding data and

information, there must be a restoration procedure and corresponding disaster recovery plan in place.

- *Integration*: Integration refers to the process of combining different applications so that they work together to run smoothly as one application. As mentioned in Chap. 1, SOA is usually the approach of choice when it comes to many integration strategies.

The common application layer services provide semantic conversion between associated application processes. Examples of common application services of general interest include the virtual file, virtual terminal, and job transfer and manipulation protocols. These topics are discussed in further detail in Chap. 5.

3.2.2 Platform as a Service

Cloud computing has also evolved to include platforms for building and running custom applications, a concept known as PaaS. PaaS applications are also referred to as on-demand, web-based, or SaaS solutions. In the PaaS layer, application infrastructure emerges as a set of services. This includes but is not limited to *middleware as a service*, *messaging as a service*, *integration as a service*, *information as a service*, *connectivity as a service*, and so on. The services here are intended to support applications. These applications might run in the Cloud, or they might run in a more traditional enterprise datacenter. In order to achieve the scalability required within a Cloud, the different services offered here are often virtualized. Examples of offerings in this part of the Cloud include IBM WebSphere Application Server virtual images, AWS, Boomi, Cast Iron, and the Google App Engine. Platform services enable consumers to be sure that their applications are equipped to meet the needs of users by providing application infrastructure based on demand [7].

Traditionally, building and running on-premise applications has always been complex, expensive, and risky. Each application required hardware, an OS, a database, middleware, Web servers, and other software. Once the stack was assembled, a team of developers had to navigate complex programming models such as J2EE and .NET. A team of network, database, and system management experts had to be present to keep everything up and running. Inevitably, a business requirement would necessitate a change to the application, which would then kick off a lengthy development, test, and redeployment cycle. To make matters worse, large companies often need specialized facilities to house their datacenters. Enormous amounts of electricity are usually needed to power the servers as well as the systems to keep them cool. Finally, a failover site is also needed to mirror the datacenter so information can be replicated in case of a disaster. Figure 3.6 depicts PaaS in a Cloud computing infrastructure.

Just as Amazon.com, eBay, Google, Microsoft, iTunes, YouTube, etc. made it possible to access new capabilities and new markets through a web browser, PaaS offers a faster, more cost-effective model for application development and delivery.

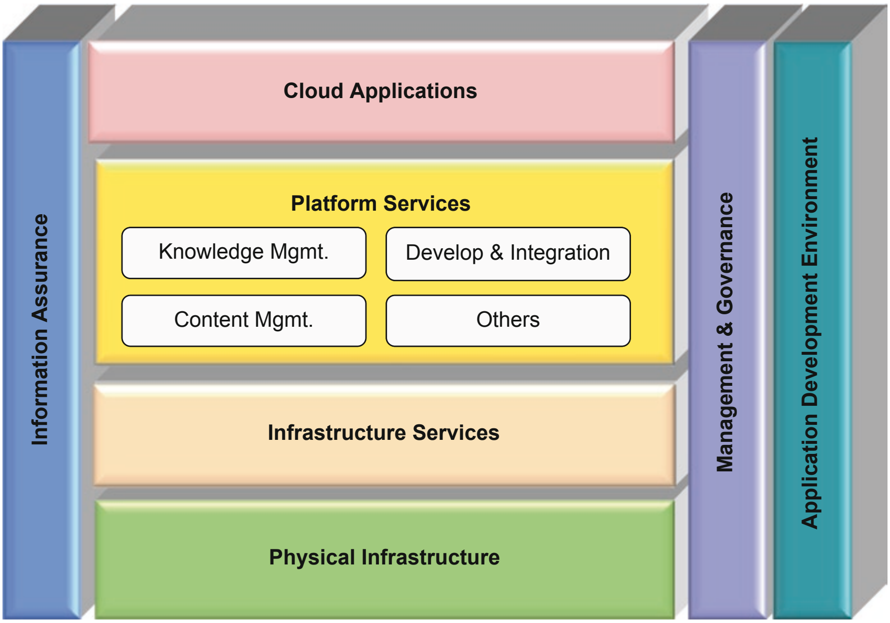


Fig. 3.6 Cloud computing infrastructure—PaaS

PaaS provides all the infrastructure needed to run applications over the Internet. It is delivered in the same way as a utility like electricity or water. Users simply "plug in" and take what they need without worrying about the complexity behind the scenes. And like a utility, PaaS is based on a metered or subscription model, so users only pay for what they use. With PaaS, ISVs and enterprise IT departments can focus on innovation instead of complex infrastructure. By leveraging PaaS, enterprises can redirect a significant portion of their budgets from simply keeping the business running as usual to creating new and innovative applications that provide real business value. PaaS is driving a new era of mass innovation. Finally, developers can access unlimited computing power; anyone with an Internet connection can build powerful applications and easily deploy them to users wherever they are located.

An enterprise should select the platform based on its existing system landscape and skill sets, the types of applications the enterprise offers, the service delivery standards the enterprise offers, and the associated costs. Generally speaking, there are four types of platforms [8]:

- *Social application platforms*: Platforms like Facebook provide APIs so third parties can write new application functionalities that are made available to all users.
- *Web application platforms*: Platforms like Google provide APIs and functionalities for developers to build Web applications that leverage its mapping, calendar, and spreadsheets, plus YouTube and other services.

- *Business application platforms*: Platforms like Force.com provide application infrastructure specifically geared toward transactional business applications such as database, integration, workflow, and UI services. For companies unwilling to compromise on scalability, reliability, and security, Force.com is the clear choice for a flexible platform that manages critical business processes.
- *Raw computing platforms*: Platforms like AWS provide storage, processor, and bandwidth as a service. Developers can upload their traditional software stack and run their applications on the Amazon infrastructure.

According to some industry experts, more PaaS choices are available besides the do-it-yourself option, such as managed hosting, where a provider runs the infrastructure, hosts applications, and may offer SaaS-specific services. Another example is the Cloud *Integrated Development Environments (IDEs),* where applications are built using the provider's on-demand tools and collaborative development environment. In addition, many pioneer enterprises are now extending SaaS beyond single-point applications for specific needs such as sales enablement or partner relationship management. The trend is taking a broader advantage of PaaS by migrating traditional datacenter operations to less-expensive, web-centric computing environments.

3.2.3 Infrastructure as a Service/Hardware as a Service

IaaS or *Hardware as a Service* (HaaS) forms the bottom layer of the Cloud. A set of physical assets such as servers, network devices, and storage disks is offered as provisioned services to consumers. The services here support the application infrastructure—regardless of whether that infrastructure is being provided via a Cloud—and many more consumers. As with platform services, virtualization is often used to provide on-demand rationing of resources. Examples of infrastructure services include IBM BlueHouse, VMWare, Amazon EC2, Microsoft Azure Platform, Sun ParaScale Cloud Storage, and more. Infrastructure services address the problem of properly equipping datacenters by assuring computing power when needed. In addition, due to the fact that virtualization techniques are commonly employed in this layer, cost savings brought about by more efficient resource utilization can be realized [9, 10].

IaaS, sometimes referred to as HaaS, is another provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers, and networking components. The SP owns the equipment and is responsible for housing, running, and maintaining it. The client typically pays on a per-use basis. Characteristics and components of IaaS include the utility computing service and billing model, automation of administrative tasks, dynamic scaling, desktop virtualization, policy-based services, and Internet connectivity.

IaaS allows enterprises to scale their IT capacity up or down on command without any capital expenditure; allows data to be safely backed up and restored in

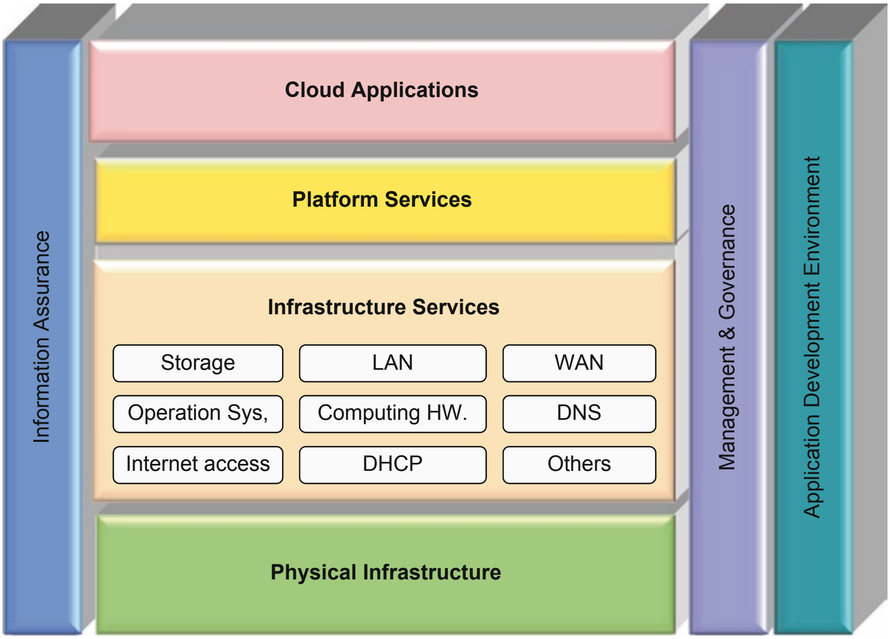


Fig. 3.7 Cloud computing infrastructure—IaaS/HaaS

hours; and allows free, highly skilled IT staff to work on value-added tasks such as development and planning, instead of chasing bugs and installing patches ad infinitum. As a result, enterprises can significantly improve their personal market value and build a path to fulfilling a more strategic corporate role than ever before [11–13]. Figure 3.7 depicts IaaS/HaaS in a Cloud Computing infrastructure.

IaaS is enabled by a new business concept based on virtualizing the IT environment. Fundamentally, IaaS provides IT resources (processing power, storage, datacenter space, services, compliance, etc.) on-demand, enabling IT to bill these services as a variable fixed cost. The interest in IaaS can be attributed to significant increases in IT-enabled business models, such as e-commerce, Web 2.0/3.0 and SaaS, which drive demand, and by advances in technology that enable it, including virtualization, utility computing, and datacenter automation. These capabilities may enable many enterprises to better their service offerings and business efficiency. To others, it may sound like a nightmare in which they lose control of their IT environment as the computing tasks are offloaded to an outside supplier. As a result, IaaS can be viewed as a useful and enabling strategic weapon in the IT arsenal for the following reasons [14, 15]:

- *IT professionals as large-scope strategic leaders rather than micromanagers*: It is predicated by Forrester Research that "there will be more than two billion PCs in use by 2015 at a 12.3% compound annual growth rate." With that kind of explosive growth in the computer sector, it is clear that the IT administrator's scope of responsibility is going to change dramatically. The idea of a one-server-to-one-administrator model is gone. The days of logging into a single box to run patches, tweak the registry, or change permissions are gone. An ideal IT administrator/operator is someone who understands the big picture, who can grasp the importance of the 200 or 2,000 computers in use at an enterprise and how they all operate together, and can manage them as a fleet. This is precisely the IaaS model. It might sound like a significant downsizing when IT administrators can manage five to 10 times the number of devices they are managing today, however, the real equation is the availability of trained staff in today's IT market and how to best utilize their talents. What enterprises urgently need is to recruit, train, and retain IT professionals that are able to think in bigger scopes rather than micromanaging a few stations. The IT professionals of the future will need to understand how to manage hundreds or even thousands of devices. IaaS does not take away responsibility, but adds a strategic dimension to IT operations, making managers more marketable because they are now accustomed to working at a higher level.

- *IT systems must be aligned and support the business*: The purpose of IT is to conduct business more efficiently and effectively. Thus, ideally, IT would be aligned with and able to support the core business strategy of an enterprise. Today's businesses realize that IT is not just a tool to help, but is a critical part of day-to-day operations and is frequently instrumental in delivering the end product. With e-commerce, business-to-business portals, IP phone systems, and even e-mail, today's applications are fully integrated into the business, so it is critical that they behave the way the business does. With IaaS and its variable but predictable costs, it is relatively easy for any enterprise to manage spending on a monthly instead of annual basis. IaaS enables a whole new and more transparent way of accounting for IT, making precise usage and costs transparent down to the resource level (e.g., blade servers, OS, storage, etc.). This creates a closer link between what the business unit spends and the "services" it receives. Once that link is established, IT can begin to change business unit behavior to prioritize costs/benefits.
- *Choice of implementing IaaS in-house*: IaaS is both a structural concept and a mindset. Thus, it can be potentially implemented internally. It does not have to come from an outside SP. If implemented internally, the IT department can charge-back its services proportionally to the parts of the enterprise that have the heaviest users. Another new solution enabled by IaaS puts the IT budget inside other departments' budgets. In this scenario, the enterprise gives the departments dollars to spend versus IT having to carry and justify those expenses throughout the year.
- *Ability of dynamic expansion*: From a business growth standpoint, an enterprise has to be ready to expand without spending big bucks on IT resources until it is absolutely necessary. It is difficult to estimate the capital or even set it aside when the number of new customers, or whether there will be any, is unknown. Even if the money is accounted for in the budget and the enterprise does acquire new business, there is still the need to rapidly provision that environment. One of

the misconceptions about IaaS is that when an enterprise decides to use this kind of outsourcing, it is a permanent decision. IaaS is the perfect solution for an enterprise to outsource its infrastructure until the enterprise has the ability to build its own capabilities. The enterprise can have entire environments up and running in days, sometimes hours, instead of weeks. After landing a new customer, an enterprise should send it to an IaaS provider. Once the budget approval is done, the enterprise still has the choice of bringing it in-house.

- *Flexibility and scalability*: In business, opportunity implies change, and change can always be a challenge. IaaS enables rapid change because it lets companies add or remove infrastructure and services on-demand. While rapid change can impact stability, with IaaS, an enterprise can add horsepower up to 60–80% of the existing IT environment that is already stable, while gaining more control over the 20–40% that is in chaos. Imagine if an enterprise has to grow an infrastructure 10–20% in 30 days. If it decides to use a SP to help, the enterprise is not permanently stuck in that mode nor have they set a precedent for the future. One of the ideas behind IaaS is that not only can one scale up quickly, but also scale down or scale out. Furthermore, IaaS is generally delivered in addition to a utility computing platform. As long as there is a platform like VMware for virtualization, it will look identical to one's own infrastructure.
- *Datacenter automation as an integral part of IaaS*: System administrators in today's datacenters typically manage only 10–20 specific hosts or devices because they fall under the administrator's area of responsibility and/or expertise. However, as mentioned in reason 1, the administrators will soon need to manage a large number of devices and stations as a fleet, due to new technologies such as virtualization and high-density computing. Datacenter automation tools like *Opsware* (acquired by HP in 2007) and BladeLogic are a large part of the IaaS model because they enable a single administrator to manage potentially hundreds of devices. These tools provide templates and policies for configuration, patch management, and security compliance. An administrator can configure a single template based on best practices or corporate policy and apply it to several hundred machines. A delta report will show all of the devices that need attention. Built-in automated remediation lets the administrator select all of the devices and apply a single change or group of changes at once. In addition, these tools can group devices based on PBM as well as exceptions.
- *Enabling easy regulatory compliance in a federated environment*: Within a federated Cloud Computing environment, all the enterprises and SPs will have to obey the same set of regulatory rules. Using the underlying features of IaaS, compliance becomes easier. Pre-compliant VMs can be kept in a library, providing a head start when a new application environment has to be deployed. Instead of installing the server from scratch, one can deploy a copy of a pre-configured (and even pre-compliant) VM. Many organizations maintain a stockpile of prebuilt VMs in a library for this purpose. It also dramatically improves the provisioning time. After the servers are online, using datacenter automation templates, one can keep the machines in compliance and even monitor their compliance and patch-level status in a dashboard.

- *Minimization of effects and drawbacks of unexpected events*: From machine malfunctions and failures to natural disasters, there are hundreds of ways to lose data and only a few really good ways to recover it. So far, IaaS is considered the best way. Data can be backed up automatically in real-time to a strategic network of datacenters that serve as mirrored storage and backup sites. Multiple backup servers on a single physical server are possible due to virtualization, which greatly reduces the hardware and operating costs. IaaS providers generally offer these as backup "targets." Because VMs are bootable, instead of performing a bare metal restore or reinstall, all that is needed is simply to boot up the VMs, which significantly reduces the recovery time. The VMs also contain all of the precious custom configuration information that is so often lost or under-documented. At last, the use of a virtual approach also reduces the issues of hardware compatibility, as long as the VMs run on VMware and as long as VMware is installed on the recovery hardware.

3.3 Holistic Enterprise Architecture and Cloud Services

To achieve business modularity maturity, organizations must fundamentally shift the way they model target EA. This means a change from vertical enterprise pillars, such as process and information, product and production, IT and infrastructure, and people and organization, to a horizontal approach. Figure 3.8 shows a holistic enterprise layered perspective verses the traditional enterprise pillar perspective.

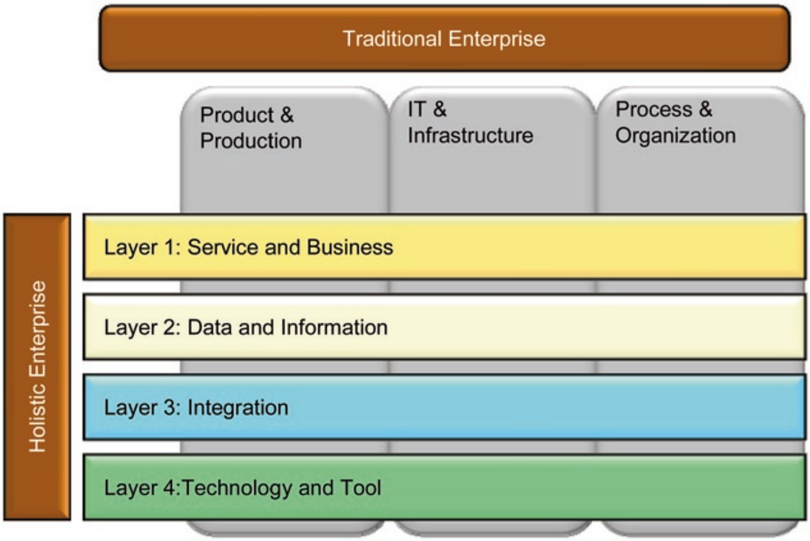


Fig. 3.8 Holistic enterprise architecture

Chapter 5 discusses the management and technical aspects of transforming enterprises to integrate Cloud application, platform, or infrastructure services with enterprise services. In this section, we will elaborate on the holistic EA and examine the service offerings Cloud technologies provide each layer of the architecture.

3.3.1 Service and Business Layer

The *Service and Business Layer* provides vision and strategic guidance, in addition to the fundamental business organization of an enterprise. The four main components of the business layer include: *business strategy*, *organization and roles*, *value network*, and the *process model*. The business layer provides the means to manage the lifecycle of business objectives and instills these objectives across the various enterprise domains. This is done by steering the establishment of the enterprise-specific process model. Furthermore, this layer also defines the roles and organizational models that take into account the extended enterprise context and integrates non-tangible concepts, such as the topology of decision making, authorization to perform pre-determined business activities, and permission to manipulate enterprise business objects. These concepts are essential to an enterprise because they govern the overall business objective and set the requirements for enterprise-wide security implementations. When integrating an enterprise with the Cloud, the Service and Business Layer often uses the Cloud SaaS. The details of this integration are in Chap. 5.

3.3.2 Data and Information Layer

How data and information are stored, communicated, and interpreted, is the vital foundation of any business. They can be created, updated, and deleted only by those who are authorized to perform such manipulations within and outside of the enterprise and value network. The key function of the *Data and Information Layer* is to ensure accessibility and accuracy, and to avoid redundancy and unstructured information and data, thus optimizing effectiveness and efficiency. From a business perspective, three main components of this layer include: *semantic information definition*, *a logical data model*, *and a physical information exchange model*.

The Data and Information Layer introduces the important concept of sharing information and data among various enterprise domains. This is absolutely key in attaining the ability to accommodate a changing business environment.

In the Cloud environment, application-dependent transaction data and processes may generate data redundancy and weaken business operation efficiency and flexibility. To overcome this challenge, the Data and Information Layer must endorse the concept of enterprise-wide or cross-Cloud-wide information and data. The sharing of information and data requires two perspectives: a semantic definition of information and a data model. The semantic definition, also interpreted as the business object glossary, ensures the common understanding across different enterprise or Cloud domains. The data model perspective enables and implements information exchanges between distributed systems.

One solution to address the challenges of both the logical data model and the physical information exchange model is to create a common data model, such as the *Common Information Model* (CIM) that is developed by the DMTF and is discussed in Chap. 6. The common data model is shared and owned by a lifecycle that is managed by business processes, breaks traditional technological dependencies, and ensures the desired loose coupling of information and data with applications as well as with businesses. In addition, this common data model approach liberates data management from its traditional dependence on applications, and promotes a shared information exchange above the proprietary application data models. Another benefit of this approach is that it shifts the information and data ownership to a more business-driven lifecycle management. This makes sense, since business processes (i.e., the layer above) manage the lifecycle of business objects instead of the Technology and Tool layer. When integrating an enterprise with the Cloud, the Data and Information Layer often uses the Cloud PaaS. The details of this integration are illusatated in Chap. 5.

3.3.3 Integration Layer

The two main components of the *Integration Layer* include: *process and information integration and enterprise application integration*. This layer operates the loose coupling among the logical business, the Data and Information Layer, and the Technology and Tool Layer. The main purpose is to promote the interoperability of components by conforming to standards, reducing the dependency on technologies, and ensuring scalability and responsiveness to change. It is important to note that this layer is particularly crucial to SOA rules and design principles. It is the core layer that federates different technologies and Cloud applications serving enterprise businesses.

3.3.4 Technology and Tool Layer

The *Technology and Tool Layer* can be viewed as the physical layer of the holistic enterprise perspective and includes two main components: *application* and the *technological platform*. A major component in Cloud computing is tooling. In many ways, this might be the most critical to the success of a Cloud solution. There is significant technology present in the marketplace to deliver Cloud solutions, however, these technologies are often difficult to deliver due to a lack of comprehensive, understandable tooling.

Consider the application services layer in the Cloud. Tooling in this layer could provide an environment that assists with Cloud application development, and could provide the means to package and deploy the application to a Cloud infrastructure. There are already many tools that fit this description, but the problem is that they are nearly always tied to the Cloud provider's infrastructure. Open standards are key to getting the most power and flexibility from tooling. Developers cannot afford to incur the costs of learning new tools every time they switch Cloud infrastructures; further, development shops cannot continually incur the cost of rewriting applications because they switch Cloud infrastructures. For this reason, tooling should aid application development, packaging, and deployment in a way that makes the finished project portable across multiple Cloud infrastructures.

Tooling also has a very clear role in the infrastructure services layer. Building out the infrastructure for a Cloud is not a trivial process. All of the physical assets for a Cloud provider, whether that provider is internal or external, need to be considered, such that the right physical resources are allocated to the Cloud. Tools in this space should help companies visualize their IT assets so that no resources are left out of consideration for the Cloud. However, it will not be enough to provide a visualization of the assets to the Cloud constructor. The tooling in this space should offer some bit of intelligence toward the creation of the Cloud. In the past, IT administrators have had a tough job of trying to match expected demands to physical resources. This has led to the problem of under-utilization of resources. This issue is a huge catalyst for the Cloud. Tools guide users through the physical makeup of the Cloud based on the expected demand characteristics of the system. When integrating an enterprise with the Cloud, the Technology and Tool Layer often uses the Cloud IaaS/HaaS. The details of this integration are in Chap. 5.

3.4 Enterprise Architecture and Cloud Transformations

Cloud Computing has dramatically changed how business applications are built and run. At its core, Cloud services eliminate the costs and complexity of evaluating, buying, configuring, and managing all the hardware and software needed for enterprise applications. Instead, these applications are delivered as a service over the Internet. In this section, we will describe the architectures necessary to transform enterprises to take advantage of Cloud services [16].

3.4.1 Enterprise Architecture Styles

Architecture styles define the following:

- Families of software systems in terms of patterns for characterizing how architecture components interact
- The types of architecture components that can exist in the architectures of those styles and constraints on how they may be combined
- How components may be combined together for deployment

- How units of work are managed, e.g., if they are transactional (n-phase commit)
- How functionalities that components provision may be composited into higher order functionalities, and how they can be exposed for use by human beings or other systems

There are essentially two types of architecture styles [17]: *SOA* or *non-SOA*. The *SOA* style is inherently *top-down* and emphasizes decomposition to the functional level but not lower. It is *service-oriented* rather than application-oriented, factors out policy as a first class architecture component that can be used to govern transparent performance of service-related tasks, and emphasizes the ability to adapt performance to user/business needs without having to consider the intricacies of architecture workings. Implementation of an SOA results in better architecture layering and factoring, and better interfaces that become more business than data oriented. Policy becomes more explicit and is exposed in a way that makes it easier to change it as necessary. Service orientation guides the implementation, making it more feasible to integrate and interoperate using a commodity infrastructure rather than using complex and inflexible application integration middleware.

On the other hand, the *non-SOA* (in contrast with the SOA) style is inherently *bottom-up* and takes much more of an infrastructural point of view (or *inside-out*) as a starting point, building up to a business functional layer. Application platforms constructed using *client-server*, *object-oriented*, and *tier* architecture styles are those to which the non-SOA approach is applied. This is because they form the basis of enterprise application architectures today, and because architectures of these types have limitations that require transformations to its counterpart SOA platforms.

As a rule of thumb, integrating businesses at functional levels is simpler than at lower technology layers, where implementations might vary widely. Hence, this section emphasizes decomposition to the functional level—which often is dictated by standards within a market, regulatory constraints on that market, or even accounting (e.g., Accounts Payable/Accounts Receivable/General Ledger (AP/AR/GL)) practices.

Architecture style will be critical to orchestrating services and enabling operability between thousands of collaborating businesses. Interoperability must be realized through the implementation of an architecture that integrates at a business functional level rather than a data level. Taking an SOA point of view requires a system architect or service designer to separate concerns from the start. Application platforms should be distributed from the beginning, rather than be made so after the fact by attaching some distribution layer to them. Enterprises must understand how they have permitted business security and access control models to be built into their architectures and how, now that technology innovations enable them to challenge these limits. The enterprises must remove them from their computing platforms to realize business agility goals demanded by new generation architectures. Technologies enterprises have used in the past can be useful to them in the future. Success in implementing a SOA is less a function of technology than it is of a business and technology architecture vision that forces business and technology architects to view business capabilities from a global, outside-in and top down perspective.

3.4.2 Architecture Transformation

IT teams in enterprises today are under pressure to transform their existing or legacy, non-SOA, application platforms to SOA that effectively leverage the capabilities afforded through the use of Cloud and service grid technologies [17]. In this section, we will explore strategies for implementing architecture transformations from non-SOA (*bottom-up*) to SOA (*top-down*) and issues likely to be encountered in the process.

How to construct an SOA that meets modern IT computing requirements has been a topic of debate, in particular, what is the best approach to leverage past investments in infrastructure, software development, and third party software products. Furthermore, funding and how long it will take to accomplish this are other aspects being discussed. There exists a number of difficult topics that IT leaders in today's enterprises want to see addressed by Cloud and service Grid Computing, such as the following:

- Datacenter management
- Architecture transformation and evolution (evolving current architectures or beginning from scratch)
- Policy-based management of IT platforms

This section will address the architectural challenges and potential solutions. Using policy-based management mechanism to solve these challenges will be discussed in Chap. 6.

3.4.2.1 Transforming Existing Architectures

In the past, IT architectures took aim at the enterprise as their endpoint. Driven by new complex and dynamic business relationships, enterprises must be able to support architectures that can support entire ecosystems and, in so doing, enable these architectures to scale downward to an EA as well as upward and outward. As it has already become the critical center of business operations today, IT leaders have to continue to chase cost and margin optimization. They also have no choice but to carefully set and navigate a course to renovate and replace their existing practices and technologies with new thinking so that product lines and services that companies offer today can remain relevant through significant market transitions.

Clouds, service grids, and SOA style are technologies that will be fundamental to successful enterprise transformations. There are near term objectives, like the need for cost and resource efficiency or IT application portfolio management, that justify the use of these technologies to re-architect and modernize IT platforms and optimize the way enterprises currently deploy them. However, there are longer term business imperatives as well, like the need for a company to be agile in combining their capabilities with those of their partners by creating a distributed platform.

It is through its relevance to enterprises' existing portfolios of critical applications that real uptake will ensue, and enterprises will begin to realize the true potential of the utility model that Cloud technologies offer. Cloud technology offerings today are mostly suitable to host EAs. While these offerings provide clear benefits to enterprises by providing capabilities complementary to what they have, the fact that they can help to elastically scale EAs should not be understood to also mean that simply scaling in this way will meet modern IT computing requirements. The architecture requirements of large platforms, like social networks, are radically different from the requirements of a healthcare platform, which has geographically and corporately distributed care providers, medical devices, patients, insurance providers, etc. The requirements for these two platforms (i.e., social networks vs. healthcare platforms) are very different from those that provision straight-through processing services common in the financial services industry. Clouds will have to accommodate differences in architecture requirements like those implied here, as well as those relating to characteristics that will be discussed subsequently.

It is enticing to think that one could implement an SOA simply by wrapping an existing non-SOA application platform with Web service technologies to service-enable it. The reality may not be so simple. Technically, it is possible to wrap an non-SOA platform with Web service technologies and then evolve the non-SOA architecture to a SOA one as budget and other resources allow. Although a non-SOA might be possible to access application functionality using Web services, using the wrapper alone can not yield the benefits of a full SOA implementation. Compensation for non-SOA limits may even be more costly than taking an alternative approach.

3.4.2.2 Addressing Architecture Layering and Partitioning

Before laying out the plans of transforming an architecture, let us first clarify a set of architectural characteristics. These characteristics should not increase the size of the management team or other costs, should allow the system to be quickly adaptable to new technologies integrated to it, and should make the system extensible from within the enterprise out to the broader ecosystem and vice versa.

It is important that Cloud services does not realize the goals of autonomic computing as they are defined currently, though combining the characteristics of existing Clouds gets closer to this goal. This fact does not diminish their value for optimizing deployments of applications in place today. Not every Cloud needs to be autonomic, but there are benefits along each path regardless. In addition, implementing architecture features on the applications management drivers path will lead to optimizing costs of operating and maintaining automating systems management and the infrastructure and business functionalities that currently run a business, resulting in more efficient datacenter management.

Evolving an architecture toward Cloud service management and SOA capabilities can help enterprises expand their IT systems beyond enterprise boundaries.

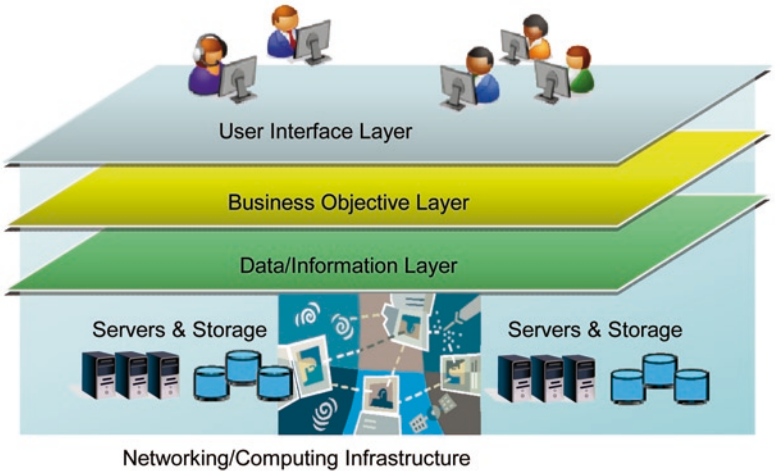


Fig. 3.9 Web-layered architecture

This supports implementation of more flexible partner networks and value chains, but can also scale to serve virtual organizations. The first step of transitioning from one architecture style to another is to align and relate to the web application layering wherever possible. From a layered perspective, a web application is usually described with a graphic of a 3-tiered architecture, shown in Fig. 3.9, that includes a *User Interface Layer*, a *Business Objective Layer,* and a *Data/Information Layer*. The User Interface Layer is usually implemented using a web server and scripting languages. The Business Objective Layer is where all business logic programmed in various programming languages can be used to code libraries of specific, broken-down business functions. The Data Layer is where code that manipulates basic data structures goes and is usually constructed using object and/or relational database technologies. Finally, all three layers are deployed on a server configured with an OS and network infrastructure, enabling an application user to access web applications from browsers. Note that these layers correspond to the layers of the holistic EA discussed in Sect. 3.3 above.

As aforementioned, the first step in transforming an architecture is to align what an enterprise already has with the layered model, so that cross-layer violations are eliminated. An example of such an alignment could remove database specifics and business logic from the User Interface Layer. Assuming layering violations are addressed, it makes sense to introduce a service API between the User Interface Layer and the Business Objective Layer. Note that the service layer, composed of a number of services, is a means of accessing lower level functionalities. The concerns of one architecture layer do not and should not become or complicate the concerns at other levels.

Proceeding from cleaning up layering architecture violations, another important task is to clean up partitioning violations. Partitioning refers to the "componentizing" or "modularizing" of business functionalities such that a component in one business functional domain accesses functionality in another domain through a single interface. Ensuring that common interfaces are used to access business functionalities in other modules eliminates the use of private knowledge to access business functionalities in other domain spaces. Partitioning also may be referred to as factoring. Just like the previous step, the next phase of transformation focuses on partitioning functionalities in the database so that, for example, side effects of inserting data into the database in an area supporting one business domain does not also publish or otherwise impact the database supporting other business domains.

3.4.2.3 Benefits of Transformations

Transforming a non-SOA to a SOA can be a lengthy process, as it is a function of existing system complexity, size, and age. Thus, it poses the question of whether or not it is worth the trouble. Clearly, it is possible to transition an architecture to become a well organized platform that is centrally hosted or hosted in a service grid or even many service grids. As a result, services and their supporting business objectives and data functionalities can be replaced easily with an alternative service implementation without negatively impacting other areas of the architecture, provided that functionality in one service domain is accessed by another service domain only through the service interface. Such a capability is required in order to simplify management of an application portfolio implemented on such an architecture, as well as distribute and federate service implementations.

When performing an architecture transformation, some of the common questions include whether or not it is necessary that all architecture components be entirely transformed; whether or not the queue-based middleware in the old architecture should be replaced; and whether or not all the old applications should be replaced with custom applications that have appropriate policy extension points. There is no fixed answer to these concerns. Certainly, it is possible to replace enterprise application integration technologies with commodity or open source technologies, simplify them, or maybe even eliminate them in some cases. It is unlikely that middleware supporting reliable messaging and long-lived business transactions between business partners needs to be totally replaced or removed from a SOA. However, its use can be couched in ways that eliminate tight coupling between partners and commingling of business policies. This is done with an integration functionality that makes partner integration difficult to change as policies change or as partner networks expand.

Cloud solutions can form the basis of an application portfolio management strategy that can be used to address tactical short term needs (e.g., interoperability within a business community of practice using the Cloud to provision community resources), and can address longer term needs to optimize the application portfolio and possibly re-architect it for the following reasons:

- Cloud vendors usually offer the capability to construct customized virtualized images that can contain software for which a enterprise has licenses. Hosting current infrastructure in a Cloud provides an isolated area in which an enterprise and/or partners could interoperate using existing technologies.

- Cloud vendors usually offer an application functionality that can replace existing capabilities (e.g., CRM systems). Incorporating this functionality into an existing application portfolio leads to an incremental re-architecture of application integrations using newer technologies and techniques, which, in turn, should result in service-oriented interfaces that can become foundational to the future state. An incremental move toward a re-architected platform host, using Cloud technologies, may prove to be the only way to mitigate risks of architectural transformation while keeping an enterprise business running.
- Cloud APIs, together with the concepts of distribution, federation, and services that are baked in, provide a foundation on which to implement loosely coupled, SOAs and can logically lead to better architecture. Standardized interfaces, loose architecture couplings, and standardized deployment environments and methods can increase reuse potential by making it easier to compose new services with existing services.
- Clouds provide a means to deal with heterogeneity. Initially, heterogeneity is dealt with through management layers. Better architecture, as noted above, further enhances this as heterogeneity is encapsulated beneath standardized and service oriented APIs. Once heterogeneity is contained, a portfolio optimization/ modernization strategy can be put into place and be implemented.

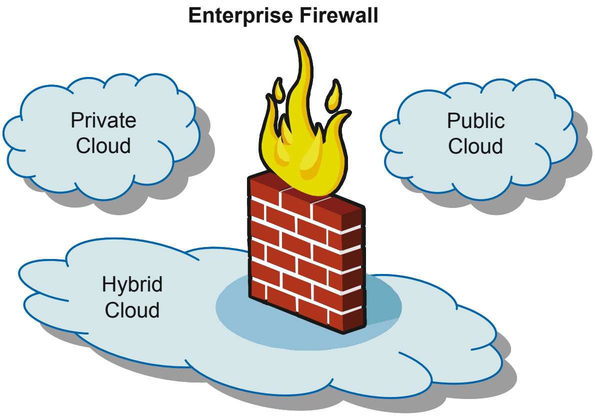
3.5 Cloud Architectures and Vendor Implementations

Cloud Computing instantiations are based on the following core components and technical characteristics:

- An *architecture style* (or styles) that should be used when implementing Cloud-based services
- An *external user and access control management* that enables roles and related responsibilities that serve as interface definitions that control access to and orchestrate across business functionalities
- An *interaction container* that encapsulates the infrastructure services and policy management necessary to provision interactions
- An *externalized policy management engine* that ensures that interactions conform to regulatory, business partner, and infrastructure policy constraints
- The *utility computing capabilities* necessary to manage and scale Cloud-oriented platforms.

From a Cloud solution perspective, the deployment can take one of three forms: public, community, private, and hybrid. Figure 3.10 depicts the basic concepts of these three forms. The following sections will examine the three architecture forms as they relate to an enterprise consumer of the Cloud.

Fig. 3.10 Three forms of Cloud computing architecture



3.5.1 Public Cloud

Public Clouds are Cloud services provided by a third party (e.g., vendors or service providers). As Fig. 3.10 indicates, they exist beyond the company firewall and are fully hosted and managed by the Cloud provider. Among the three Cloud types (public, private, community, and hybrid), the Public Cloud is probably the most well-known and mature in its offerings thus far. An example of a Public Cloud is the Amazon EC2 infrastructure, which provides a Public Cloud infrastructure that hosts Amazon Machine Image instances that deliver capabilities to users [1, 18].

Accessibility and affordability are two of the key characteristics that have led to the popularity of the Public Cloud. More specifically, Public Clouds attempt to provide consumers with hassle-free IT elements. Whether it is software, application infrastructure, or physical infrastructure, the Cloud provider takes on the responsibilities of installation, management, provisioning, and maintenance. Consumers are only charged for the resources they use, so under-utilization is eliminated.

However, the Public Cloud does pose a certain degree of inconvenience, in a "convention over configuration" way. Public Cloud services are usually delivered with the idea of accommodating the most common use cases. Configuration options are usually a smaller subset than what they would be if the resources were controlled directly by consumers. Since consumers have little control over the infrastructure, processes requiring tight security and regulatory compliance require careful arrangements when using Public Clouds. These arrangements are discussed in Chap. 9.

To understand how an enterprise can leverage Public Cloud Computing solutions, let us consider two important view points. First, enterprises consume applications that are provided in the Public Cloud. This might be an application designed to process employee payroll data, or it might be a CRM system. By utilizing software delivered in this way, an enterprise can remove the burden of installing and maintaining the application on private datacenters. Another benefit is the cost savings associated with license fees, since most Cloud providers charge based on consumption. Second, enterprises utilize Cloud-based hosting solutions to deliver applications to consumers. By doing so, companies are freed from the maintenance and upkeep of production systems, since the Cloud provider is responsible for providing infrastructure resources to meet the demands users place on the application. This model also provides for an increase in the ubiquity of an enterprise's services, since solutions delivered by way of a Public Cloud can be accessed at any time from any machine with a viable network connection.

Regardless of the scenario, a common theme is the bottom line value to a business. Public Clouds can help enterprises reduce costs associated with owning software and datacenter infrastructure components. Less directly, Public Cloud usage can deliver value by enabling enterprises to respond quickly to changes in demand for their services, enabling the services to reach new markets and enabling valuable human resources to concentrate on delivering business innovation, rather than simply delivering the technological infrastructure that supports the business.

From an application provider perspective, this suite of tools lets users meet, discuss, collaborate, and innovate all by leveraging Cloud-provided services. The tools also help organizations implement solutions that leverage Public Cloud offerings in order to deliver the sought after Cloud value.

Finally, a popular Cloud services pricing structure is the *pay-per-use* structure, as discussed in Chap. 5. To achieve this, Cloud resource usage must be tracked and reported. These reports should be able to provide statistics about Cloud usage that support chargeback in the enterprise. For each user, retrieve information about their VM usage and CPU, memory, and IP utilization rates can be viewed or downloaded into a spreadsheet.

3.5.2 Private Cloud

A Cloud's type is usually defined in terms of where the physical resources and data reside. Private Clouds are Cloud services provided within the enterprise. Private Clouds exist within an enterprise firewall, all of the computing resources and services that make up the Cloud are protected by that firewall [1].

Private Clouds offer many of the same benefits that Public Clouds do with one major difference: the enterprise is in charge of setting up and maintaining the Cloud. Private Cloud solutions deliver many of the same benefits as their public counterparts, such as cost reduction, business agility, and enhanced innovation. The main difference is that the enterprise maintains full control over and responsibility for the Cloud. In addition, finer-grained control over the various resources making up the Cloud gives a company all available configuration options. Private Clouds are ideal when the type of work being done is not practical for a Public Cloud, due to security and regulatory concerns.

Although a Private Cloud does not free the enterprise from the responsibility of procuring and maintaining computing resources, there are many reasons why enterprises choose Private Cloud solutions over Public Clouds:

- *Security and compliance regulations*: Private Clouds usually need more stringent control and oversight with respect to how and where data is stored than is typically provided by a Public Cloud service.
- *Capabilities that cannot be achieved in a Public Cloud*: An enterprise might require a very specific vendor technology, or might need availability guarantees not achievable by Public Cloud usage.
- *Private Cloud as financial property*: If an enterprise is heavily invested in its existing datacenter, it makes sense to optimize the utilization of those resources rather than pay for Public Cloud services. Even companies without such cost investments often see price advantages to on-premise solutions, as the flexibility of off-premise solutions could come at a premium.

One example of a Private Cloud is AWS' new service offering—the *Virtual Private Cloud* (VPC), as shown in Fig. 3.11 [19, 20]. Targeted at customers with existing IT investments, the VPC service provides a way for companies to create a logically separated set of EC2 instances and a secure VPN connection to their own networks.

Generally, VPC requires three elements: a *VPC instance*, an *IP Security* (IPSec) *VPN gateway*, and a *block of IP addresses* provided by the customer. The VPC ad-

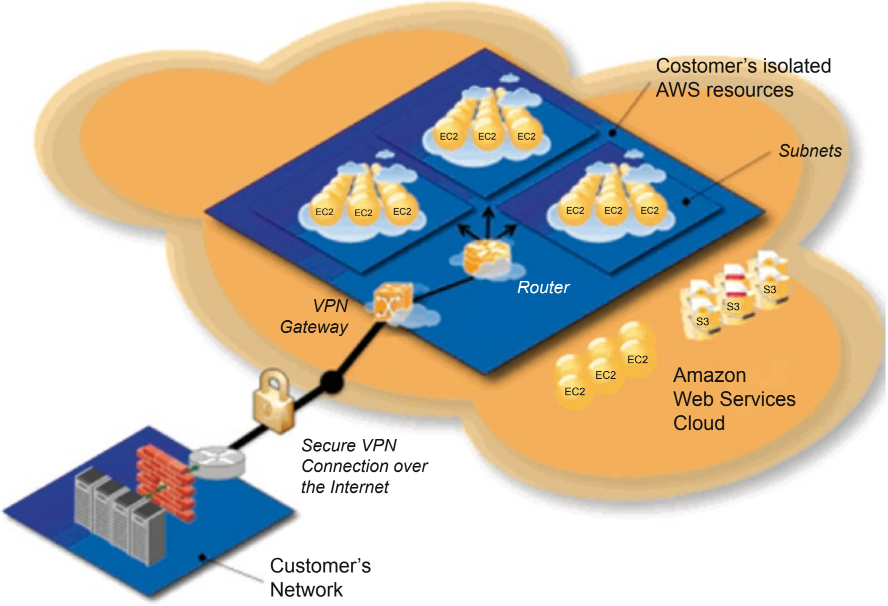


Fig. 3.11 AWS virtual private Cloud

dresses can be divided up into subnets to further partition traffic. All Internet-bound traffic is routed through the customer's network and outbound security systems before reaching the public network.

Private Clouds accelerate the adoption of Cloud services if the enterprises can access a form of the Cloud that would give them the best of both worlds. This includes the flexibility and cost-effectiveness of accessing a virtually infinite pool of resources without owning it, while being able to integrate those resources into their existing datacenter environments so they could continue to leverage existing investments in their management and control infrastructure. Amazon VPC allows customers to seamlessly extend their IT infrastructure into the Cloud while maintaining the levels of isolation required for their enterprise management tools to do their work.

The Private Cloud solution potentially addresses the ever increasing costs of server and middleware management and administration in several ways. Private Clouds provide tools to build consistent, repeatable application server deployments. These deployments are optimized for virtualized environments, enabling enterprises to reduce administrative costs and leverage the benefits of server consolidation that come from such environments. In addition, the Private Cloud solution provides the flexibility to shape and tune the configurations that it dispenses. Thus, the easy integration capabilities can provide enterprises with seamless, E2E workflows that can significantly improve IT efficiency and agility even further. On the other hand, the difficulty and cost of establishing an internal Cloud can sometimes be prohibitive, and the cost of continual operation of the Cloud might exceed the cost of using a Public Cloud.

In conclusion, Private Clouds offer enterprises many of the same benefits as their public counterparts, and because of the familiarity with existing resources, Private Clouds can even provide an easier on-ramp to Cloud Computing. It provides a means to create virtualized, repeatable deployments that include everything from the OS to custom user scripts and applications. Once in the Cloud, the virtual systems can be utilized just like standard Application Server deployments.

3.5.3 Hybrid Cloud

Hybrid Clouds are a combination of public and Private Clouds. These Clouds are typically created by the enterprise and management responsibilities are split between the enterprise and the Public Cloud provider. As the name suggests, a Hybrid Cloud leverages services that are in both the public and private space [1].

Hybrid Clouds are the suitable solution when an enterprise needs to employ the services of both a public and a Private Cloud. In this sense, an enterprise can outline the goals and needs of services, and obtain them from the public or Private Cloud as appropriate. A well-constructed Hybrid Cloud can service secure, mission-critical processes, such as billing and receiving customer payments, as well as those that are secondary to the business, such as employee payroll processing.

The major drawback to this type of architecture form is the difficulty in effectively creating, managing, and governing such a solution. Services from different sources must be obtained and provisioned as if they originated from a single location, and interactions between private and public components can make the implementation even more complicated. Since this is a relatively new architectural concept in Cloud Computing, best practices and tools about this pattern continue to emerge, and there could be a general reluctance to adopt this model until more is known.

Private Clouds should not be confused with Hybrid Clouds. A Hybrid Cloud uses both external (under the control of a vendor) and internal (under the control of the enterprise) capabilities to meet the needs of an application system. A Private Cloud lets the enterprise choose and control the use of,both types of resources.

3.6 Cloud Related Standards and Forums

In this section, a sample of standards and forums that are related to Cloud Computing and infrastructure will be discussed. As appropriate, the Cloud service (SaaS, PaaS, Iaas/HaaS) to which the standards are most applicable will be associated. Cloud Computing and infrastructure uses many more standards than what is listed in this section, and the use of these additional standards is discussed where appropriate throughout the book.

3.6.1 Open Grid Forum

OGF [21] is a community-initiated forum of individual researchers and practitioners working on distributed computing, or “grid” technologies. OGF is committed to driving the rapid evolution and adoption of applied distributed computing. Applied distributed computing is critical to developing new, innovative, and scalable applications and infrastructures that are essential to productivity in the enterprise and within the science community. OGF accomplishes its work through open forums that build the community, explore trends, and share best practices, and consolidates these best practices into standards. Figure 3.12 depicts a positioning of some Cloud standards as proposed by OGF.

OGF maintains a comprehensive repository of informational, historical, and experimental documents on various topics such as Web Services Agreement Specification (WS-Agreement) [22], GLUE Schema [23], Lightweight Directory Access Protocol (LDAP) [24], Storage Resource Manager Interface (SRM) [25], Data Movement Interface (DMI) [26], GridFTP [27], OVF Specification [28], Job Submission Description Language (JSDL) [29], Basic Execution Service (BES) [30], and Usage Record (UR) [31].

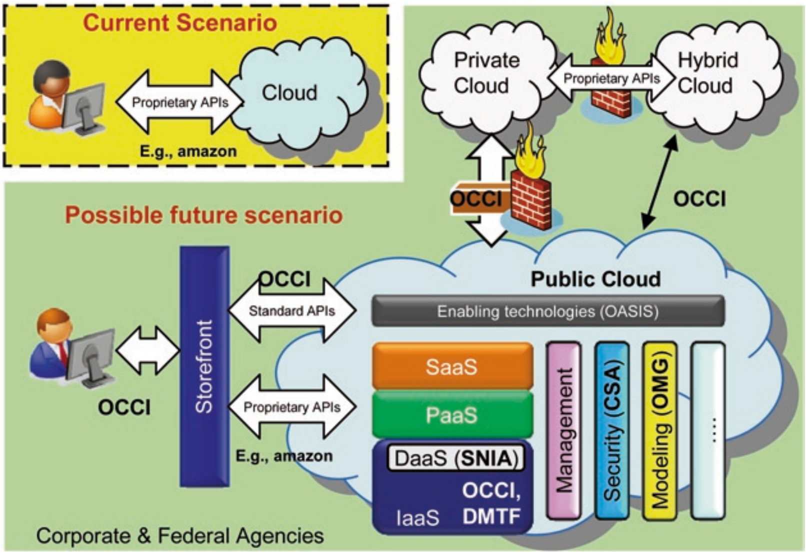


Fig. 3.12 Possible positioning of some Cloud standards (courtesy: Dr. Craig A. Lee (OGF) brief to the CCWG on 21 Sep 2009)

3.6.2 Open Virtualization Format

The OVF Specification describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of software to be run in VMs. The key properties of the format are as follows:

- *Optimized for distribution*: OVF supports content verification and integrity checking based on industry-standard *Public Key Infrastructure* (PKI), and provides a basic scheme for managing software licensing.
- *Optimized for a simple and automated user experience*: OVF supports validation of the entire package and each VM or metadata component of the OVF during the installation phases of the VM lifecycle management process. It also includes relevant, user-readable, descriptive information that a virtualization platform can use to streamline the installation experience.
- *Supports both single VM and multiple-VM configurations*: OVF supports both standard single VM packages and packages containing complex, multi-tier services consisting of multiple interdependent VMs.
- *Portable VM packaging*: OVF is virtualization platform neutral, yet also enables platform-specific enhancements to be captured. It supports the full range of virtual hard disk formats used for hypervisors today, and is extensible, which allows it to accommodate formats that may arise in the future. VM properties are captured concisely and accurately.

- *Vendor and platform independent*: OVF does not rely on the use of a specific host platform, virtualization platform, or guest OS.
- *Extensible*: OVF is immediately useful and extensible. It is designed to be extended as the industry moves forward with virtual appliance technology. It also supports and permits the encoding of vendor-specific metadata to support specific vertical markets.
- *Localizable*: OVF supports user-visible descriptions in multiple locales, and supports localization of the interactive processes during installation of an appliance. This capability allows a single packaged appliance to serve multiple market opportunities.
- *Open standard*: OVF has risen from the collaboration of key vendors in the industry, and is developed in an accepted industry forum as a future standard for portable VMs.

Virtualization and OVF are further discussed in Chap. 5. OVF is often used in the IaaS layer.

3.6.3 HTTP

The HTTP [32] is a protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol that can be used for many tasks beyond its use for hypertext, such as naming servers and distributed object management systems, through an extension of its request methods, error codes, and headers. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred. HTTP has been in use by the WWW global information initiative since 1990. This specification defines the protocol referred to as "HTTP/1.1," and is an update to the Internet Engineering Task Force (IETF) RFC 2068. The HTTP protocol is a request/response protocol. A client sends a request to the server in the form of a request method, URI, or protocol version, followed by a *Multipurpose Internet Mail Extensions* (MIME)-like message containing request modifiers, client information, and possible body content over a connection with a server. The server responds with a status line, including the message's protocol version and a success or error code. This is followed by a MIME-like message from the server containing server information, entity meta-information, and possible entity-body content. HTTP is a protocol with the lightness and speed necessary for a distributed collaborative hypermedia information system. It is a generic, stateless, object-oriented protocol, which may be used for many similar tasks such as naming servers and distributing object-oriented systems, by extending the commands or "methods" used. A feature of HTTP is the negotiation of data representation, allowing systems to be built independently of the development of new advanced representations. As discussed in Chap. 5, HTTP is often used in the IaaS and SaaS layers.

3.6.4 XML and JSON

XML [33] is widely used for the representation of the arbitrary data structure of Web services. XML is a text format derived from the *Standard Generalized Markup Language* (SGML). Compared to SGML, XML is simple. HTML, by comparison, is even simpler. However, XML is designed to transport and store data, not to display data like HTML.

The most recent buzz regarding XML is around its new role as an interchangeable data serialization format. XML provides two advantages as a data representation language:

- XML is text-based
- XML is position-independent

Together, these encourage a higher level of application-independence than other data-interchange formats. Unfortunately, XML is not well suited to data-interchange, much as a wrench is not well-suited to hammering nails. It carries a lot of baggage and does not match the data model of most programming languages. When most programmers saw XML for the first time, they were shocked at how ugly and inefficient it was. It turns out that that first reaction was the correct one. There is another text notation that has all of the advantages of XML, but is much better suited to data-interchange. That notation is *JavaScript Object Notation* (JSON) [34].

JSON is a lightweight data-interchange format. It is easy for humans to read and write and is also easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language-independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, structure, dictionary, hash table, keyed list, or associative array
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence

XML and JSON are used in the PaaS and SaaS layers.

3.6.5 AJAX

AJAX [35] is a group of interrelated Web development techniques used on the client-side to create interactive Web applications. It is based on JavaScript and HTTP requests. AJAX is not a new programming language, but a new way to use existing standards. AJAX is the art of trading data with a Web server, and changing parts of a Web page, without reloading the whole page.

With AJAX, Web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. The use of AJAX techniques has led to an increase in interactive or dynamic interfaces on Web pages. Despite the name, the use of JavaScript and XML is not actually required, nor do the requests need to be asynchronous. In addition, AJAX.org Platform is a pure javascript application framework for creating real-time collaborative applications that run in the browser. AJAX.org Platform radically changes the way people write applications, more details can be found on AJAX.org. AJAX is used in the SaaS layer.

3.6.6 HTML5

HTML5 [36] is being developed as the next major revision of HTML, the core markup language of the Web. HTML5 is the proposed next standard for HTML 4.01, XHTML 1.0 and DOM Level 2 HTML. It aims to reduce the need for proprietary plug-in-based *Rich Internet Application* (RIA) technologies such as Adobe Flash, Microsoft Silverlight, and Sun JavaFX. The ideas behind HTML5 were pioneered in 2004 by the *Web Hypertext Application Technology Working Group* (WHATWG). HTML5 incorporates Web Forms 2.0, another WHATWG specification. The HTML5 specification was adopted as the starting point of the work of the new HTML working group of the *World Wide Web Consortium* (W3C) in 2007. HTML 5 contains several features that address the challenge of building Web applications that work while offline. This document highlights these features (SQL and offline application caching APIs, as well as online/offline events, status, and the localStorage API) from HTML 5 and provides brief tutorials on how these features might be used to create Web applications that work offline.

In this 5th version of HTML, new features are introduced to help Web application authors, new elements are introduced based on research into prevailing authoring practices, and special attention has been given to defining clear conformance criteria for user agents in an effort to improve interoperability.

Users of typical online Web applications are only able to use the applications while they have a connection to the Internet. When they go offline, they can no longer check their e-mail, browse their calendar appointments, or prepare presentations with their online tools. Meanwhile, native applications provide those features: e-mail clients cache folders locally, calendars store their events locally, and presentation packages store their data files locally. In addition, while offline, users are dependent on their HTTP cache to obtain the application, since they cannot contact the server to get the latest copy. HTML5 is used in the SaaS layer.

3.6.7 Web Syndication

Web syndication [37, 38] is a form of syndication in which Website material is made available to multiple sites. Most commonly, Web syndication refers to making Web feeds available from a site in order to provide other people with a summary of the Website's recently added content, such as the latest news or forum posts. The term can also be used to describe other kinds of licensing Website content so that other Websites can use it.

In addition to freely distributed material, some broadcasters and others use similar methods for the controlled placement of proprietary content on multiple partnering Internet destinations. In addition to Web feeds, commercial syndicators may use other methods to distribute their content such as Reuters, Associated Press, and All Headline News. Such commercial Web syndication borrows its business models from syndication in other media, such as Print, radio, and television. Primarily, syndication arose in those other media so that content creators could reach a wider audience. Generally, commercial Web syndication can be categorized in three ways: *business models*, *types of content*, or *methods for selecting distribution partners*.

Commercial Web syndication involves partnerships between content producers and distribution outlets. There are different structures of partnership agreements. One such structure is licensing content, in which distribution partners pay a fee to the content creators for the right to publish the content. Another structure is ad-supported content, in which publishers share revenues derived from advertising on syndicated content with that content's producer. A third structure is free or barter syndication, in which no currency changes hands between publishers and content producers. This requires the content producers to generate revenue from another source, such as embedded advertising or subscriptions. Alternatively, they could distribute content without remuneration. Typically, those who create and distribute content for free are promotional entities, vanity publishers, or government entities.

Types of content syndicated include *Really Simple Syndication* (RSS) or Atom feeds and full content. RSS is a syndication format that was developed by Netscape in 1999 and became very popular for aggregating updates to blogs and news sites. RSS also stands for "Rich Site Summary" and "RDF Site Summary." Atom is a XML-based syndication format that is used to publish headlines of the latest updates on blogs and Websites for retrieval by users and other sites. Based on RSS 2.0, Atom was turned over to the IETF for standardization. Most news aggregators support Atom along with the traditional RSS formats. With RSS feeds, headlines, and summaries, sometimes a modified version of the original content is displayed on users' feed readers. With full content, the entire content, which might be text, audio, video, applications/widgets or user-generated content, appears unaltered on the publisher's site.

There are two methods for selecting distribution partners. The content creator can hand-pick syndication partners based on specific criteria, such as the size or quality of their audiences. Alternatively, the content creator can allow publisher sites or users to "opt in" to carrying the content through an automated system. Some of these automated "content marketplace" systems involve careful screening of potential publishers by the content creator to ensure that the material does not end up in an inappropriate environment.

Just as syndication is a source of profit for TV and radio producers, it also functions to maximize profit for Internet content producers. As the Internet increases in size, it has become increasingly difficult for content producers to aggregate a sufficiently large audience to support the creation of high-quality content. Syndication enables content creators to amortize the cost of producing content by licensing it across multiple publishers or by maximizing distribution of advertising-supported content. However, a potential drawback for content creators is that they can lose control over the presentation of their content when they syndicate it to other parties.

Distribution partners benefit by receiving content either at a discounted price, or for free. One potential drawback for publishers, however, is that because the content is duplicated on other publisher sites, they cannot have the content exclusively. For users, the fact that syndication enables the production and maintenance of content allows them to find and consume content on the Internet. One potential drawback for them is that they may run into duplicate content, which could be annoying.

JavaScript is typically a useful tool for syndicating content to other Websites. Using JavaScript has the following benefits:

- *Simple implementation given the target Website*: Just add one line of HTML to the target page. It works for any Web server environment and does not need server-side technologies such as PHP, Perl, Python, or Java.
- *Real-time content update*: When updating the content on the site, changes are immediately reflected on syndication sites. With cached solutions such as RSS, there is typically a self-imposed delay of up to an hour.
- *Full control over the content*: The publisher has control over how the content is presented, or can allow partners to customize the presentation.
- *Easy viewer management*: Publishers can log information about the end-users who see their syndicated content. They can log each user that fetches their JavaScript file and then compare those results to the number of click-throughs they receive for their syndicated content.

However, using JavaScript for Web syndication also has some inherent weaknesses:

- *Dependence on the users' browsers*: If JavaScript is not enabled, the content will not appear. People with disabilities will not have access to the content, and search engines will not index the content.
- *Inefficiency*: The content must be loaded from a central location for every user. This might lead to bandwidth problems when serving the content.

As will be discussed in Chap. 5, Web syndication is appropriate for the SaaS layer.

3.6.8 XMPP

The *Extensible Messaging and Presence Protocol* (XMPP) [39] is an open technology for real-time communication, which powers a wide range of applications including *Instant Messaging* (IM), presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data.

Although the core technology behind XMPP is stable, the XMPP community continues to define various XMPP extensions through an open standards process run by the XMPP Standards Foundation. There is also an active community of open-source and commercial developers, who produce a wide variety of XMPP-based software. Users are not "locked in" when using XMPP technologies. Since mid-2001, the XMPP Standards Foundation (formerly the Jabber Software Foundation) has documented and managed the Jabber/XMPP protocols through an open standards process focused on the discussion and advancement of XEPs. Such specifications define XMPP extensions and are not to be considered part of XMPP, which is only the core specifications produced by the IETF. XMPP is used in the SaaS layer.

3.6.9 REST

Representational State Transfer (REST) is a style of architecture based on a set of principles that describe how networked resources are defined and addressed. REST is a term coined by Roy Fielding in his Ph.D. dissertation [40] to describe an architecture style of networked systems. Use of REST APIs is further discussed in Chap. 5.

The design rationale behind Web architecture can be described as an architectural style consisting of the set of constraints applied to elements within the architecture. By examining the impact of each constraint as it is added to the evolving style, the properties induced by the Web's constraints can be easily identified. Additional constraints can then be applied to form a new architectural style that better reflects the desired properties of a modern Web architecture. This section provides a general overview of REST by walking through the process of deriving it as an architectural style. Later sections will describe in more detail the specific constraints that compose the REST style.

An application or architecture considered RESTful or REST-style is characterized by:

- State and functionality are divided into distributed resources
- Every resource is uniquely addressable using a uniform and minimal set of commands (typically using HTTP commands of GET, POST, PUT, or DELETE over the Internet)
- The protocol is client/server, stateless, layered, and supports caching

The motivation for REST was to capture the characteristics of the Web that made it successful. Subsequently, these characteristics are being used to guide the evolution of the Web. REST is an architectural style, not a standard. There is not, nor will be, a REST specification. On the other hand, REST does use standards such as HTTP, URL, XML, HTML, GIF, JPEG, etc., for resource representations. REST-style Cloud configuration management is discussed in detail in Chap. 7. As discussed in Chap. 5, REST is used in the SaaS, PaaS, and IaaS layers.

3.6.10 Security and Data Privacy Standards

Security and data privacy standards can be broken down into a few different categories:

- IAM
  - *Identification Management* (IdM), federation SAML, *WS-Federation*, *Liberty Identity Federation Framework* (ID-FF))
  - Strong authentication standards (*HMAC-based One Time Password* (HOTP), *OATH Challenge Response Algorithms* (OCRA), *Time-based One Time Password* (TOTP))
  - Entitlement management (*eXtensible Access Control Markup Language* (XACML))
- Data Encryption (at-rest, in-flight), Key Management
  - Public Key Infrastructure or PKI,
  - *Public Key Cryptography Standards* (PKCS),
  - *Provisioning of Symmetric Keys* (KEYPROV),
  - *Enterprise Key Management Infrastructure* (EKMI)
- RIM-International Organization for Standards ISO15489
- E-discovery (*Electronic Discovery Reference Model* (EDRM))

The detailed discussion of security and privacy issues will be deferred to Chap. 9. As an example however, in the following three sections, we examine three particular protocols in more detail: OAuth, OpenID, and *Secure Sockets Layer* (SSL)/ *Transport Layer Security* (TLS).

3.6.10.1 OAuth

The OAuth protocol [41] enables Websites, applications, and consumers to access protected resources from a Web service via an API, without requiring users to disclose their SP credentials to the consumers. More generally, OAuth creates a freely-implementable and generic methodology for API authentication. OAuth does not require a specific UI or interaction pattern, nor does it specify how SPs authenticate users, making the protocol ideally suited for cases where authentication credentials are unavailable to the consumer, such as with OpenID. The discussion on OpenID will be deferred to the next section. OAuth aims to unify the experience and implementation of delegated Web service authentication into a single, community-driven protocol. OAuth builds on existing protocols and best practices that have been independently implemented by various Websites. Open standards, supported by large and small providers alike, promote a consistent and trusted experience for both application developers and users of those applications.

The fundamental benefit of OAuth is that it allows users to share their private resources (photos, videos, contact list, bank accounts) stored on one site with another site without having to hand out their username and password. There are many reasons why one should not share private credentials. For example, giving an email account password to a social network site so it can look up associated friends is the same thing as going to dinner and giving the ATM card and PIN code to the waiter when it is time to pay. Any restaurant asking for a PIN code will go out of business, but when it comes to the Web, users put themselves at risk sharing the same private information. This is a good metaphor for OAuth from a user's perspective. Instead of giving the ATM card and PIN code, the card can double as a credit card with a signature authorization. Just like a username and password provide full access to the user's resources, the ATM card and PIN code provide the user with great control over bank accounts, much more than just charging goods. However, when the user replaces the PIN code with a signature, the card becomes very limited and can only be used for limited access.

Unlike OpenID, where users must do something first (i.e., get an OpenID identity they can use to sign-into sites), OAuth is completely transparent to users. In many cases, end-users will not know anything about OAuth, what it is or how it works. The user experience will be specific to the implementation of both the site requesting access and the one storing the resources, and will be adjusted to the device being used (Web browser, mobile phone, PDA, set-top box).

Users generally do not care about protocols and standards, they care about better experience with enhanced privacy and security. This is exactly what OAuth sets to achieve. With Web services on the rise, people expect their services to work together in order to accomplish something new. Instead of using a single site for all their online needs, users use one site for their photos, another for videos, another for email, and so on. No one site can do everything the best. In order to enable this kind of integration, sites need to access user resources from other sites, and these are often protected such as private family photos, work documents, bank records, etc.

3.6.10.2 OpenID

OpenID [42] is the fast, easy, and secure way to sign in to Websites. OpenID is an open, decentralized standard for authenticating users. It can be used for access control, allowing users to log on to different services with the same digital identity where these services trust the authentication body. OpenID replaces the common login process that uses a login-name and password, by allowing a user to log in once and gain access to the resources of multiple software systems. The term OpenID can also refer to an ID used in the standard.

An OpenID is in the form of a unique URL, and is authenticated by the user's OpenID provider, i.e., the entity hosting their OpenID URL. The OpenID protocol does not rely on a central authority to authenticate a user's identity. Since neither the OpenID protocol nor Websites requiring identification may mandate a specific type of authentication, non-standard forms of authentication can be used, such as smart cards, biometrics, or ordinary passwords. In summary, OpenID has the following benefits:

- *Accelerate the sign-up process*: Most Websites ask for an extended, repetitive amount of information in order to use their application. OpenID accelerates that process by allowing users to sign in to Websites with a single click. Basic profile information (such as name, birth date, and location) can be stored through a user's OpenID and used to pre-populate registration forms, so the user spends more time engaging with a Website and less time filling out registration pages.
- *Easy maintenance of a single set of usernames/passwords*: Most Web users struggle to remember the multiple username and password combinations required to sign-in to each of their favorite Websites, and the password recovery process can be tedious. Alternatively, using the same password for all Websites poses a security risk. With OpenID, a user can use a single, existing account (from providers like Google, Yahoo, AOL, or blogs) to sign in to thousands of Websites without ever needing to create another username or password. OpenID is the safer and easier method to joining new sites.
- *No single control over online identity*: OpenID is a decentralized standard, meaning it is not controlled by any one Website or service provider. Users control how much personal information they choose to share with Websites that accept OpenIDs, and multiple OpenIDs can be used for different Websites or purposes. If a user's email (*Google*, *Yahoo*, *AOL*), photo stream (*Flickr*), or blog (*Blogger*, *Wordpress*, *LiveJournal*) serves as his/her primary online presence, OpenID allows the user to use that portable identity across the Web.
- *Minimize password security risks*: Many Web users deploy the same password across multiple Websites. Since traditional passwords are not centrally administered, if a security compromise occurs at any Website a user uses, a hacker could gain access to his/her password across multiple sites. With OpenID, passwords are never shared with any Websites, and if a compromise does occur, users can simply change the password for their OpenID, thus immediately preventing a hacker from gaining access to their accounts at any Websites they visit.

Finally, because the focus of most OpenID providers (such as Google, Yahoo, and AOL) is in identity management, they can be more thorough about protecting users' online identities. Most Website operators are less likely to be as dedicated to protecting users' identities as the OpenID providers, whose focus is on securely hosting user identities.

3.6.10.3 SSL/TLS

One problem when administering a network is securing data that is sent between applications across an un-trusted network. TLS/SSL is typically used to authenticate servers and clients and then used to encrypt messages between the authenticated parties [43].

The TLS protocol, the SSL protocol, versions 2.0 and 3.0, and the *Private Communications Transport* (PCT) protocol are based on public key cryptography. A Security Channel (Schannel) authentication protocol suite provides these protocols.

All Schannel protocols use a client/server model. In the authentication process, a TLS/SSL client sends a message to a TLS/SSL server, and the server responds with the information that the server needs to authenticate itself. The client and server perform an additional exchange of session keys, and the authentication dialog ends. When authentication is completed, SSL-secured communication can begin between the server and the client using symmetric encryption keys that are established during the authentication process. For servers to authenticate to clients, TLS/SSL does not require server keys to be stored on domain controllers or in a database, such as the Microsoft Active Directory service. Clients confirm the validity of a server's credentials with a trusted root certification authority's certificate. Therefore, unless user authentication is required by the server, users do not need to establish accounts before they create a secure connection with a server.

In addition, SSL version 3, documented in an IETF draft, provides one of the most commonly available security mechanisms on the Internet. Developed by Netscape, SSL is used extensively by Web browsers to provide secure connections for transferring credit card numbers and other sensitive data. An SSL-protected HTTP transfer uses port 443 (instead of HTTP's normal port 80), and is identified with a special URL method. When an SSL session is established, the server begins by announcing a public key to the client. No encryption is in use initially, so both parties (and any eavesdropper) can read this key, however the client can now transmit information to the server in a way that no one else can decode. The client generates 46 bytes of random data, forms them into a single very large number, encrypts them with the server's public key, and sends the result to the server. Only the server, with its private key, can decode the information to determine the 46 original bytes. This shared secret is now used to generate a set of conventional cipher keys to encrypt the rest of the session.

3.7 Enterprise Transformation Implications

One of the goals of this book is to guide readers through occurring changes from infrastructure, developer, and end user perspectives that signal the demise of the full-featured server OS and the virtual server. Virtualization, and the large scale, multi-tenant operations model known as Cloud computing, enable IT professionals to rethink the packaging, delivery, and operation of software functionalities in extremely disruptive and beneficial ways [44].

There are some fundamental questions often asked: (1) how will Cloud computing affect the future of IT; (2) how will the role of IT and the roles within IT change as a result of the changing landscape of the technology it administers; and (3) what new applications and resulting markets are enabled by this fundamental shift in concept [45]?

First and foremost, software packaging will be application focused, not server focused. Traditionally, the focus of distributed system deployment has been the server, not the application. In the highly customized world of IT systems development before virtualization and the Cloud, servers were acquired, software was installed upon the servers in very specific ways, and the entire package was managed and monitored largely from the perspective of the server. For example, the focus is on what processes are running, how much CPU is being used, etc. As OS functionality begins to get wrapped into application containers, or moved onto the hardware circuitry itself, the focus shifts. The packaging begins to be defined in terms of application architecture, with monitoring happening from the perspective of software services and interfaces rather than the server itself. These packages can then be moved around within datacenters, or even among them, and the focus of management will remain on the application. That is not to say that no one will be watching the hardware; infrastructure operations will always be a key function within datacenters. However, outside of the datacenter operations team, it will matter much less.

Enterprise IT will have greater influence on solution architectures and force them to align better with what is offered from the Cloud. Whether or not the Cloud will stifle differentiation in software systems is debatable. As end users select SaaS applications to run core pieces of their business, meet integration and operations needs from the Cloud, and generally move from systems providers to SPs, the need to reduce customization will be strong. This will reduce costs and strengthen system survivability in the face of constant feature changes on the underlying application system.

In addition, the altered relationship between software and hardware due to the Cloud will result in new organizational structures within the IT department. Traditionally, when it comes to IT operations, specifically datacenter operations, administrative groups divide up along server, storage, and network lines of the client-server application architectures. However, this is a prime example of a time when applications were tightly coupled to the hardware on which they were deployed. This particular type of static deployment model requires particular expertise in customizing technologies in pursuit of meeting specific service-level goals. When software deployment is decoupled from the underlying hardware, it begins to allow for a re-evaluation of these operational roles. Currently, a lot of enterprises are already in a transition in this respect, with increasing reliance on roles like virtualization administrators and operations specialists to fulfill the changing trend [44, 46].

Furthermore, the changing landscape of software development platforms will result in new philosophies of software architecture, deployment, and operations. As mentioned earlier, applications instead of servers will become the focus, particularly agile applications ranging from web applications to data processing to core business systems, and will become more relevant in large-scale systems development. Thus, agility and project management will be the two major changes. Agility is measured in terms of the frequency and speed in which features and fixes are released from a SP's perspective. From an enterprise developer's perspective, agility is measured in how rapidly features and fixes iterate over the write-build-test cycle. Agile programming and project management methods make a ton of sense in the Cloud, as do service-oriented approaches to software and systems architecture.

Lastly, there will be a significant reduction in the demand for tactical systems administrators. More specifically, tactical system administrators in the traditional sense, i.e., who grabs a trouble ticket from the top of the queue, takes care of the request, closes the ticket, and repeats the cycle, will reduce in numbers significantly. The reason for this is automation. A lot of tasks, such as provisioning, failure detection/notification or even recovery, scaling, and some aspects of infrastructure management, are highly automatable. However, in certain situations, especially in the case of Private Clouds, tactical systems administrators are still needed. They are primarily needed to monitor the overall performance of applications running in the Cloud on both internal and external resources, as well as the performance of the Cloud providers themselves.

A common mistake made by many enterprises is to look for magic bullets that solve budget, agility, or performance problems. Several options can be considered such as (1) try to move all legacy infrastructure into a Cloud model at once; (2) put an ultimatum in place that demands that all new work be done in the Cloud, or (3) experiment with "baby Clouds," small, noncritical projects that can prove both capability and economy, thus rationalizing a steady expansion into more critical application domains. For many enterprises, the third approach is more practical because it allows adopting enterprises to see what is available, and adopt those services at their own pace. In addition, Clouds are not defined by who runs them, but by the services they provide. For enterprises who use Cloud services for mission critical applications such as marketing and R&D support systems, they will always run their own infrastructure for some workloads and some data sets [47].

Successful enterprise transformation relies on effective Cloud service and customer management. In today's ever-changing global economy, Cloud SPs have to respond to both the customer's increased demands for superior customer service and to stiffer competition. Providers might have to expand their markets beyond their self-contained boundaries and broaden their business relationships. Enterprises now face very different regulatory environments and their business strategies and approaches to competition are quite distinct, nevertheless they share several common characteristics such as the following:

- Remain heavily dependent upon effective management of information and communications networks to stay competitive
- Adopt a service management approach to the way they run their businesses and their networks
- Move to an end-to-end process management approach developed from the customers' point of view
- Aautomate customer care, service, and network management processes
- Integrate new OSS/BSS with legacy systems
- Focus on data services offerings
- Focus on total service performance, including customer satisfaction
- Integrate current technology and new technologies
- Emphasize more of a buy rather than a build approach that integrates systems from multiple suppliers

A number of commercial industry standards and best practices are able to find suitable solutions in addressing the challenges listed above. The well-accepted and well-adopted standards and best practices may give enterprises the tools they need to deliver a more productive environment and efficient management infrastructure. Generally speaking, there are four emerging categories of Cloud Computing standards:

1. *Meta-element association*: Defining "distributed and non-deterministic computing" from the Cloud and SOA perspective
2. *Governance*: Integrating service governance and Cloud governance domains
3. *SLAs*: Establishing agreements between Cloud service offering consumers and providers
4. *SOA, events, and agents*: Defining communication among and within Clouds between services enabled in these Clouds

New Generation Operations Systems and Software (NGOSS) is the TM Forum's next generation OSS initiative. It is envisioned as a comprehensive, integrated framework for developing, procuring, and deploying operational and business support systems and software. It encompasses a toolkit of industry-agreed frameworks, specifications, and guidelines that cover key business and technical areas; and aims to deliver measurable improvements in development and software integration environments. The elements of NGOSS fit together to provide an end-to-end framework for OSS/BSS development integration and operations. Elements of NGOSS may be used as an end-to-end framework, as part of a comprehensive methodology. In addition, the TM Forum Cloud services program is addressing some of the Cloud issues such as security, portability, and reliability from the traditional telecommunications and wider enterprise perspective.

3.7.1 Information Framework

While people spend a fair amount of time looking over the horizon at what the next industry-changing phenomenon will be and what will impact enterprises' businesses, it is easy to forget that it is the simple things enterprises have to get right in order to survive in the current and future marketplace. Simply put, although enterprises should be thinking about the emerging issues of Cloud Computing, challenges of streamlining processes, improving data integrity, and increasing customer experience are still the foundations of an enterprise's success.

Although information models are challenging and complex conceptual models, they serve as the bridge between business entities/domains in order to fuel an effective and efficient enterprise. An information architecture forms one of the cornerstones upon which a successful enterprise thrives.

NGOSS's enterprise-wide information framework, or *Shared Information and Data* Model (SID), provides more than a comprehensive CIM for the complete activities of an enterprise. It also provides a common language for software developers and integrators to use in describing management information, which in turn allows easier and more effective integration across OSS/BSS software applications provided by multiple vendors. More importantly, it provides the concepts and principles needed to define a shared information model, the elements or entities of the model, business-oriented models, as well as design-oriented models and sequence diagrams to provide a system view of the information and data.

3.7.2 Process Framework

NGOSS's business process framework, commonly known as the *Enhanced Telecom Operations Map* (eTOM), defines several major business processes within and external to the enterprise. It is responsible for the framework and the common language of business processes. It can be used to catalog existing processes within a SP, act as a framework for defining scope of a software-based solution, or simply enable clearer lines of communication between a SP and a system integrator.

eTOM represents task-centric services that are modeled to encapsulate process logic or use case steps. It ties together the grouped logic or steps as a specific activity automated by the service logic. The purpose of the NGOSS eTOM framework is to continue to set a vision for the industry to compete successfully through the implementation of business process driven approaches to managing the enterprise. Approaches include ensuring integration among all vital enterprise support systems concerned with service delivery and support. The focus of the eTOM framework is on the business processes used by enterprises, the linkages between these processes, the identification of interfaces, and the use of customer, service, resource, supplier/partner, and other information by multiple processes. Bringing the ITIL and TM Forum standards into alignment will most definitely strengthen the practicality of the models for more generic business needs in SOEs.

3.7.3 Service Level Management

Service performance encompasses both technical performance factors as well as the more subjective customer satisfaction. A SP, by offering various performance levels for a given service, has the capability to balance the level of performance offered against price and customer expectation. SLAs provide SPs with the opportunity to diversify their customers, build stronger long-term relationships and brand image, and maintain and grow their market share. However, the growing complexity of global services brings together a myriad of services, suppliers, and technologies, all with potentially different service requirements. A SLA is an element of a formal, negotiated contract, which documents the common understanding of all aspects of the service and the roles and responsibilities of both parties from service ordering to service termination. A SLA can include many aspects of a service, such as performance objectives, customer care procedures, billing arrangements, etc. Naturally, managing SLAs is also a multi-aspect task.

TM Forum has published tremendous amounts of work on SLA management and SLM. The documents include four volumes: overview, concepts and principles, applications and examples, and enterprise and applications. These four volumes are based on a common concept, with each volume concentrating on specific topics. The objective of this series is to assist the two parties (i.e., the end customer and the SP) in developing, provisioning, and managing SLAs by providing a practical view of the fundamental issues.

3.8 Conclusion

Constructing a lean and tight IT model requires significant improvements in enterprises' data quality and information integrity. Tremendous investments in application platforms in the past ten years have resulted in heterogeneous, best-of-breed application systems that have proven hard and costly to integrate within enterprise boundaries.

In the Cloud environment, the traditional computing platforms, i.e., physical desktops, laptops, and servers, will transform to a virtual computer and disappear behind a "networked Cloud." Computing services on the other hand, will be delivered in a highly scalable and elastic fashion. All together, the need for outlaying capital resources for computing power will be greatly reduced. Note that the Internet technologies and techniques that enable this conceptual transition will extend to the underlying hardware, storage, and applications. A new challenge of standardization in the Cloud environment is thus presented. There have been many efforts in standardizing and organizing different aspects of Cloud technologies, such as what was summarized in this Chapter. Although many efforts have been initiated, domain specifications for areas such as data interoperability, protocols, and processes for inter-Cloud collaboration and Cloud balancing, remain premature and thus require further investigation and development.

References

1. Amrhein, D., Quint, S.: Cloud computing for the enterprise. Part 1: Capturing the Cloud, IBM DeveloperWorks. April 2009. http://www.ibm.com/developerworks/websphere/techjournal/0904_amrhein/0904_amrhein.html
2. Cloud-standards.org. April 2010. http://cloud-standards.org/wiki/index.php?title=Main_Page
3. SaaS.com. http://www.saas.com/inside.jsp?type=solutions&page=SolutionsOverview
4. Fishteyn, D.: Deploying Software as a Service (SaaS), white paper, SaaS.com. http://www.saas.com/homepage/pdf/SaaS.com_Whitepaper_PartI.pdf
5. SaaS-Sourcing—Why software as a service is dominating industries: TrendPOV. http://www.trendpov.com/node/1884

6. Oestreich, K.: Building a real-world iaaS cloud foundation. Cloud Comput. J. http://cloud-computing.sys-con.com/node/976449 (2009). 26 May 2009
7. What is PaaS? Salesforce.com. http://www.salesforce.com/paas/
8. Types of PaaS Solutions. Salesforce.com. http://www.salesforce.com/paas/paas-solutions/
9. Black, N.: What is SaaS? Understanding the concepts of Cloud computing. Feb 2009. http://blog.firmex.com/what-is-saas-concepts-cloud-computing
10. The Top 20 SaaS and Cloud Computing Buzzwords. SearchCRM.com. http://searchcrm.techtarget.com/feature/The-top-20-SaaS-and-cloud-computing-buzzwords
11. O'Day, P.: IaaS —Infrastructure as a service a threat or a weapon? Cloud Comput. J. http://soa.sys-con.com/node/439721 (2007). 11 Oct 2007
12. O'Day, P.: IaaS, A.: Web 2.0 allows user to bypass IT department, Bill St. Arnaud Blogspot. Oct 2007. http://billstarnaud.blogspot.com/2007_10_01_archive.html
13. Dornan, A.: Web 2.0 Aallows uUser to bBypass IT dDepartment, Bill St. Arnaud Blogspot., October 2007. http://billstarnaud.blogspot.com/2007_10_01_archive.html
14. Yates, S.: Worldwide PC adoption forecast, 2007 to 2015, Forrester Research. June 2007. http://www.forrester.com/rb/Research/worldwide_pc_adoption_forecast,_2007_to_2015/q/id/42496/t/2
15. Infrastructure as a service—Leveraging Cloud computing as a strategic weapon, Bluelock.com. Issue 16, June 2010. http://www.busmanagement.com/article/Infrastructure-as-a-Service–Levaraging-Cloud-Computing-as-a-Strategic-Weapon
16. Winans, T.B., Brown, J.S.: Moving information technology platforms to the Clouds, Deloitte, May 2009
17. Winans, T.B., Brown, J.S.: Cloud computing: A collection of working papers, Deloitte, May 2009
18. Amrhein, D.: Cloud computing for enterprise. Part 2: WebSphere sMash and DB2 express-C on the amazon EC2 Public Cloud, IBM DeveloperWorks. May 2009. http://www.ibm.com/developerworks/websphere/techjournal/0905_amrhein/0905_amrhein.html
19. Amazon Virtual Private Cloud, Amazon. http://aws.amazon.com/vpc/ (2010)
20. Urquhart, J.: Putting Amazon's spot pricing in perspective, cnet news. Dec 2009. http://news.cnet.com/wisdom-of-clouds/?keyword=Amazon+Web+Services
21. Open Grid Forum. http://www.ogf.org/ (2010)
22. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web Services Agreement Specification (WS-Agreement), Grid Resource Allocation Agreement Protocol (GRAAP) WG. http://www.ogf.org/documents/GFD.107.pdf
23. Andreozzi, S., Burke, S., Ehm, F., Field, L., Galang, G., Konya, B., Litmaath, M., Millar, P., Navarro, J.P.: GLUE Specification v. 2.0, GLUE Working Group. http://www.ogf.org/documents/GFD.147.pdf
24. Zeilenga, K.: OpenLDAP Foundation, Lightweight Directory Access Protocol (LDAP): Technical specification road map, Network Working Group. http://tools.ietf.org/html/rfc4510
25. Sim, A., Shoshani, A., Badino, P., Barring, O., Baud, J.-P., Corso, E., De Witt, S., Donno, F., Gu, J., Haddox-Schatz, M., Hess, B., Jensen, J., Kowalski, A., Litmaath, M., Magnoni, L., Perelmutov, T., Petravick, D., Watson, C.: The storage resource manager interface specification Version 2.2, Grid Storage Resource Management, http://www.ogf.org/documents/GFD.129.pdf
26. Antonioletti, M., Drescher, M., Luniewski, A., Newhouse, S., Madduri, R.: OGSA-DMI Functional Specification 1.0, GWD-R-P.134. http://www.ogf.org/documents/GFD.134.pdf
27. Allcock, W.: GridFTP: Protocol extensions to FTP for the Grid, GWD-R. http://www.ogf.org/documents/GFD.20.pdf
28. Open Virtualization Format Specification. http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf
29. Anjomshoaa, A., Brisard, F., Drescher, M., Fellows, D., Ly, A., McGough, S., Pulsipher, D., Savva, A.: Job Submission Description Language (JSDL) Specification, Version 1.0, Job

Submission Description Language (JSDL) Specification. http://www.ogf.org/documents/GFD.136.pdf

30. Foster, I., Grimshaw, A., Lane, P., Lee, W., Morgan, M., Newhouse, S., Pickles, S., Pulsipher, D., Smith, C., Theimer, M.: OGSA Basic Execution Service Version 1.0, GFD-R.108. http://www.ogf.org/documents/GFD.108.pdf
31. Mach, R., Lepro-Metz, R., Jackson, S.: Usage record—format recommendation, GFD-R-P.098. http://www.ogf.org/documents/GFD.98.pdf
32. World Wide Web Consortium (W3C). HTTP Specifications and Drafts. http://www.w3.org/Protocols/Specs.html (2002)
33. XML, Wikipedia. http://en.wikipedia.org/wiki/XML (2010)
34. JavaScript Object Notation. http://www.json.org/xml.html
35. http://www.ajax.org/#home
36. HTML5, Wikipedia. http://en.wikipedia.org/wiki/HTML5 (2010)
37. Web syndication, Wikipedia. http://en.wikipedia.org/wiki/Web_syndication (2010)
38. Riggott, M., Sutherland, J.: Layman's guide to Web syndication, Mercurytide company. http://www.mercurytide.co.uk/news/article/web-syndication/ (2006). 15 Dec 2006
39. Extensible Messaging and Presence Protocol (XMPP), XMPP Standards Foundation. http://xmpp.org/about/ (2004)
40. Fielding, R.T.: Architectural styles and the design of network-based software architectures. Ph.D. dissertation, Information and Computer Science, University of California, Irvine (2000)
41. OAuth Core 1.0 Revision A, OAuth. http://oauth.net/core/1.0a/ (2009). 24 Jun 2009
42. OpenID, Wikipedia. http://en.wikipedia.org/wiki/OpenID (2010)
43. What is TLS/SSL? Microsoft TechNet. http://technet.microsoft.com/en-us/library/cc784450(WS.10).aspx (2003). 28 Mar 2003
44. Urquhart, J.: Cloud computing and the big rethink: Part 5: CNET News. http://news.cnet.com/8301-19413_3-10377531-240.html (2009). 19 Oct 2009
45. Urquhart, J.: James Hamilton on Cloud economies of scale: CNET News. http://news.cnet.com/wisdom-of-clouds/?categoryId=9798870 (2010). 28 Apr 2010
46. Urquhart, J.: Does Cloud computing need LAMP? CNET News. http://news.cnet.com/wisdom-of-clouds/?categoryId=10093856 (2010). 23 May 2010
47. Business Process Framework concepts and Principles, TMF GB921, Release 9.0. TM Forum. 18 Aug 2010

Chapter 4
Challenges of Enterprise Cloud Services$_1$

In traditional IT organizations, the operator has complete control of and visibility into their service offering and infrastructure. All components are accessible and can be measured by the enterprise's set of well known tools. Whether complex or simple, all components are used to analyze the measured metrics and tune their systems to their optimal performance. However, an enterprise no longer has control of or visibility into the components of the service when using Cloud services. Without this visibility, the service-level warrantee is no longer straightforward. Additionally, attempts at isolating problems between an enterprise and its vendor has become more commonplace and deal with more complex issues. Thus, the relationship between Cloud vendors and enterprises must evolve.

Another challenge of enterprise Cloud management is its limitation in obtaining the correct level of visibility into the Cloud infrastructure's configuration and operational parameters. These parameters can include the transaction-ID, instance-ID, application type, image-ID, security level, location, DNS information, etc. However, when this information is not standardized, it is difficult for the operator to understand an exact scenario. This impacts other parts of the service attributes. As types of services offered over a Cloud become diversified, enterprises may require hundreds of instances and thousands of metrics to be monitored. A set of management frameworks or process agreements beyond vendor-specific views becomes crucial to make this Cloud manageable.

Extending from the discussion of standards in Chap. 3, this chapter highlights the need of such guidance in specific areas, namely infrastructure, platforms, software, management and operation, and security. The authors argue that deploying a set of unified, multi-tier management frameworks, either under the control of agents on monitored instances or as active checks from the management server, is the first step in ensuring service integrity and quality. At the lower level of this tiered framework, the monitoring information must be augmented with vendors' information. This approach allows the management systems to process vendor data with a mediated view and allows the SA systems to show much richer and more timely information. Therefore, dynamic changes in the Cloud can be recognized and handled by the system in real-time [1].

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_4, © Springer Science+Business Media B.V. 2010

4.1 Overview

In the IT industry, many enterprises have moved away from centralized, mainframe-based applications to distributed computing models that are based predominantly on service-oriented and Internet-oriented architectures. Existing applications and IT resources that are designed according to the principles of service orientation provide a solid foundation for the adoption or integration of Cloud-based frameworks.

In the previous discussions, we revealed the business values, service architecture, fundamental technologies, and operational considerations of enterprise services. With the Cloud service framework, enterprises have the ability to scale quickly to meet changing user demands. With Cloud services, one has the benefit of separating applications from physical resources or using external assets to handle peak loads.

However, not all enterprises are ready to take the opportunity for their technology transformation. Reasons for this can vary. Some are due to enterprises' existing business restrictions. For instance, the existing processes and data are tightly coupled and many points of integration are not well defined. Others are due to a dependency on legacy systems, where their internal core architecture requires a major effort to upgrade or depends on a proprietary interface. For any of the above reasons, the transformation project becomes less attractive for these enterprises.

The underlying technologies associated with Cloud services can be a part of an innovative approach for creating a more dynamic enterprise. This is feasible because applications and the services they support are no longer locked to a fixed, underlying infrastructure. As virtualization and SOA permeate the enterprise, loosely coupled services running on an agile, scalable infrastructure should in theory make every enterprise a node in the Cloud. With these new capabilities, enterprises can adjust quickly to change. Like any other revolution, Cloud Computing is the result of a technological process and business model transition. Let's first review the driving factors of Cloud services in Fig. 4.1 [2]. Seven elements are categorized in three value domains, namely economic, architectural, and strategic. The economic values are enabled by the pay-as-you-go, pay-as-you-grow models, including no CAPEX. The architectural values are driven by a simple, abstract environment for development. The strategic values are gained because the enterprise can focus on their core business and leave the rest to someone else [3].

The driving forces of enterprises' Cloud adaptation mentioned above can be briefly concluded as following:

- The virtualization technology and market's fast development
- The hardware's fast development, like CPU and network devices
- The wideband network's fast development
- The fast increase of corporate IT infrastructure requirements
- The fast change and time-to-market requirements of Internet applications
- The economic crisis forcing companies to cut costs

As a developing technology and new business paradigm, concerns about the risk associated with conducting the transformation is understandable. For instance, there

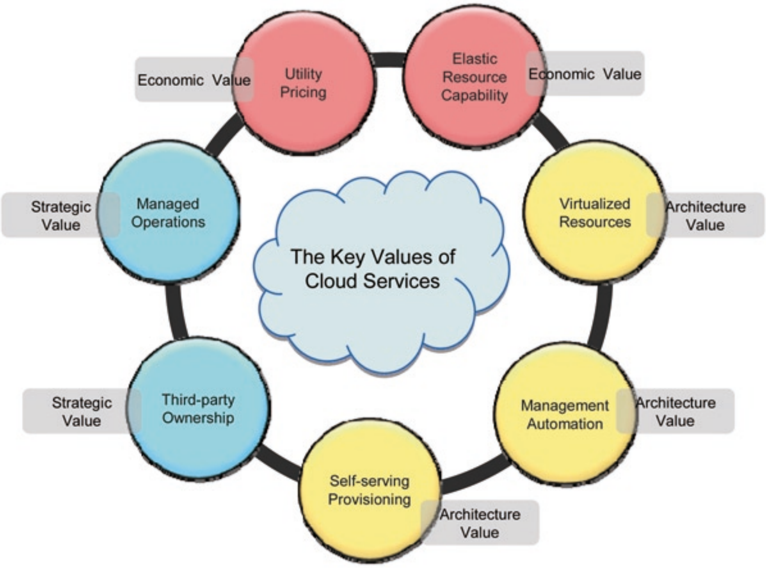


Fig. 4.1 The seven elements of Cloud service value

is currently a lack of enthusiasm on the part of many IT organizations to embrace external Clouds due to the risk attributed to their internal data asymmetry.

An *IDC*'s analysis is shown in Fig. 4.2. Many IT and management organizations are trying to mitigate their risks by identifying technology and process gaps in the

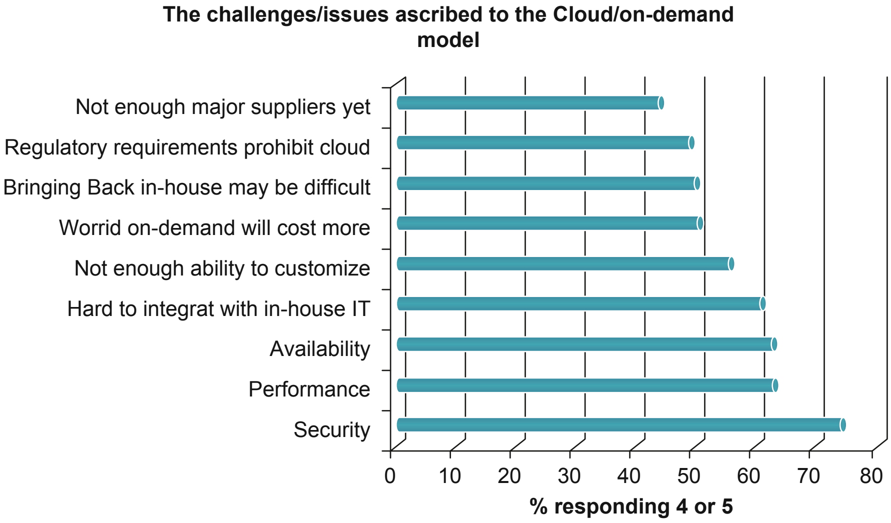


Fig. 4.2 The challenges and issues most concerning enterprises

Table 4.1 Greatest concerns surrounding Cloud adaptation

Security	45%
Integration with existing systems	26%
Loss of control over data	26%
Availability concerns	25%
Performance issues	24%
IT Governance issues	19%
Regulatory/compliance concerns	19%
Dissatisfaction with vendor offering/pricing	12%
Ability to bring systems back in-house	11%
Lack of customization opportunities	11%
Measuring ROI	11%
Not sure	7%
Other	6%

hope of finding tangible answers to these problems. Polls from various sources conclude similar concerns and issues.

Table 4.1 highlights a report from *CIO Research* with respect to enterprises' concerns of Cloud adaptation. Both reports conclude that security, performance, and integration are among the top issues that most concern enterprises [4].

It should be noted that the answers are not common when comparing the customer and the vendor viewpoints from the analysis results. This is mainly due to the fact that the vendor's business drivers value the return of QoS and SLA while the customers value the service experiences. Samples of concerns are listed below with more in-depth discussions taking place in the following sections.

Customer's Perspective:

- Data Security
  - Many customers do not trust "the Cloud" with their data
  - Data must be locally retained for regulatory reasons
- Latency
  - The Cloud can be many milliseconds away
  - Not suitable for real-time applications
- Application Availability
  - Cannot switch from existing legacy applications
  - Equivalent Cloud applications do not exist

Vendor's Perspective:

- SLAs
  - What if something goes wrong?
  - What is the true cost of providing SLAs?
- Latency
  - SaaS/PaaS models are challenging
  - Much lower upfront revenue

- Application Availability
  - Customers want open/standard APIs
  - Need to continuously add value

Often times, the enterprise IT department focuses too much on functionalities only at the infrastructure-level. By doing so, the technical staff loses sight of management and operational perspectives. Ignorance about the business practices and financial policies sometimes causes more damage than the technical solution fixes. Throughout this chapter, we will examine the challenges and gaps that an enterprise encounters and address how an enterprise can successfully transition to the Cloud-based paradigm. In the remaining chapters, guidance and recommendations for installing a standardized mechanism for a truly practical and profitable model will be proposed. Upon having this guidance in place, enterprises can then effectively use the following five steps to develop their corresponding solutions [5]:

1. Identify the rationale for adopting the Cloud service model for an enterprise from business, cultural, and value perspectives. Understand the data, services, processes, and the Cloud resources in the enterprise that can support the transformation. Assess the risk and compliance requirements applicable to the internal systems.
2. Develop a risk assessment mechanism associated with different levels of risk and make it part of the system development lifecycle. The assessment should include candidate data, candidate services, and candidate processes for the transformation effort.
3. Create a governance strategy and security strategy. Bind the candidate services to the identified data and processes. Relocate the services, processes, and information as needed in order to satisfy the defined business strategy.
4. Implement security, governance, functional operations, and system requirements.
5. Assess the potential Cloud SPs for their risk management practices. With the requirements in hand, the transformation project managers can have their risk assessments mapped against a particular Cloud offering and can decide whether or not that service is appropriate for the enterprise.

In the follow sections, the authors will layout challenges and issues from both technical and non-technical perspectives to assist IT managers in appreciating potential risks during project planning and execution.

4.2 Non-Technical Challenges

Understanding the implementations of Cloud-related technology and processes and internal company maturity can guide enterprises in determining how and when to leverage Cloud services to support core, as well as non-core, business capabilities. While technical issues may seem explicitly quantifiable on the surface, it is equally critical for non-technical issues to be resolved. Some of the significant non-technical hurdles to the adoption of Cloud Computing services by large enterprises are financial, operational, and organizational issues.

4.2.1 Financial

Conventional IT organizations have to deal with internal customers as well as IT SPs on different planes, namely data, control, and management. The first effort required in moving to a Cloud environment can look twice as costly as in-house implementation because the IT department needs to handle changes to both internal clients and external suppliers. By simple observation, the effort seems to work to the advantage of small and medium-sized companies over large enterprises. In fact, Cloud offerings are most attractive to small and medium-sized companies due to their flexibility and on-demand cost structure. For this reason most current customers of Clouds are small businesses.

4.2.2 Enterprise Scalability

Knowing that the ownership of Cloud services is not always cheaper, especially in the initial stages, the enterprises should assess the benefit of the investment in regards to the duration of service versus the ROI. This includes sunk cost in storage systems, people, network, and so forth. Each has their own financial implication. These must be well calculated before any action is taken for the transformation.

Cost variability is an important aspect of Cloud Computing. When one considers cost transparency, scalability, and variability, a new challenge and opportunity for organizations arise. When an enterprise is dealing with temporary spikes in computing loads, rather than move an entire infrastructure out of their datacenter, an external Cloud presents a preferred addition to the current infrastructure. For other events, a Cloud provides an attractive option to mirror an IT environment as a warm-backup. However, the on-going monthly fees to Cloud vendors versus the up-front implementation fees and hardware purchases for a client server may not look as optimized in a five-year analysis. Therefore, the executives of an enterprise need to articulate the financial values from different perspectives in order to justify the need for the transformation effort. The key questions are, what are the trade-offs and which benefits are important to the consumer?

- Will the enterprise gain any financial advantages when their developers only need to be concerned with the high level Cloud-based API over their backend processes? What will the financial implications, or perhaps business implications, be when all of the infrastructure specialists who architect, deploy and maintain servers, and maintain uptime and business continuity are eliminated?

- As more services come online, will there be sufficient technical and process knowledge in-house to justify the financial impact for choosing the right services for the on-going business development? Will new business opportunities benefit from the newly added Cloud services? Or will these new services cause more confusion to potential consumers?
- It is proven that Cloud technology can assist an enterprise in testing new ideas for a small application quickly and inexpensively. The new application can then be scaled as needed for different trial markets. However, the benefit of effectiveness (not efficiency) depends more on doing a lot of small applications that meet a business goal than on a larger application. Thus, the question is how to define the optimized size of applications for an enterprise that satisfies this profile and argument?

There are many factors involved in cost justification. This also goes for how long the investment will last before the servers go out of warranty.

Looking from a Cloud vendor's perspective, although standard bodies are promoting open frameworks that allow Clouds to be interoperable with different entities seamlessly, there is always a need to customize certain features for different clients' needs. That being said, the massive capital investments Cloud providers have or will make in their datacenters by highly qualified personnel will not generate revenue if their customers leave. Therefore, it is expected that the service customers may incur switching and migrating costs to compensate for the provider's investment. In the end, performing this migration into or out of a Cloud will not be inexpensive—either software must be purchased or services paid for, creating a well-bounded financial decision.

4.2.2.1 Software Licensing

License management and virtualization are big issues for large enterprises. Managing packaged software may not be as easy as adding up software packages in a personal computer, especially when different software packages are used across many functional organizations. The integration of software packages and the calculation of licensing costs is one of the unavoidable financial challenges of an enterprise.

In today's IT service departments, the administrators are responsible for ensuring the compliance of licensing agreements with their vendors and monitoring the usage of purchased services or tools to maximize the investment. They may come across some surprising observations, including people in the organization using software that the IT department never knew they had or paying for licenses that they never use. The transformation to Cloud Computing should theoretically remove these problems because the usage is controlled by the Cloud providers. However, the savings involved may not be entirely predictable if Cloud vendors use old models of software licensing that are wholly incompatible with the new service paradigm.

- *CPU based*: In most cases, software running on the Cloud is variable. Both the IP department and the user may not know how many CPUs are utilized in the measurable period due to the nature of Cloud services. If an application needs more CPUs, the Cloud service will, in theory, make an acquisition without asking permission from the end user. This flexibility may introduce a variety of changes that may frustrate and confuse customers.

- *Instance based*: Using virtualization as a horizontal scalability method can be a troublesome issue when licensed instances need to spin up on multiple computing units to meet higher demands. Legally, the enterprise should license more instances than are currently needed in order to satisfy future demand. Statistics show licensing costs can increase by nearly 20% when moving to a virtual architecture. Figure 4.3 shows the difference of functional distribution and cost ratios in a sample case [6].
- *Named Users*: Some licensing fees are calculated based on concurrent users. In fact, many applications use this model, with vendors tightly controlling access to the software based on the number of licensed users. However, the purpose of an elastic environment is the ability to scale up on-demand. Therefore, the client should acquire licenses for as many potential users as possible, even if the service is rarely accessed.

These old models of licensing structures based on CPUs, instances, or named users simply do not work in the on-demand, elastic world of Cloud Computing and virtualization. Over-provisioning is one solution, but that is costly and defeats the benefits of reduced operations and capital expenses by a Cloud environment.

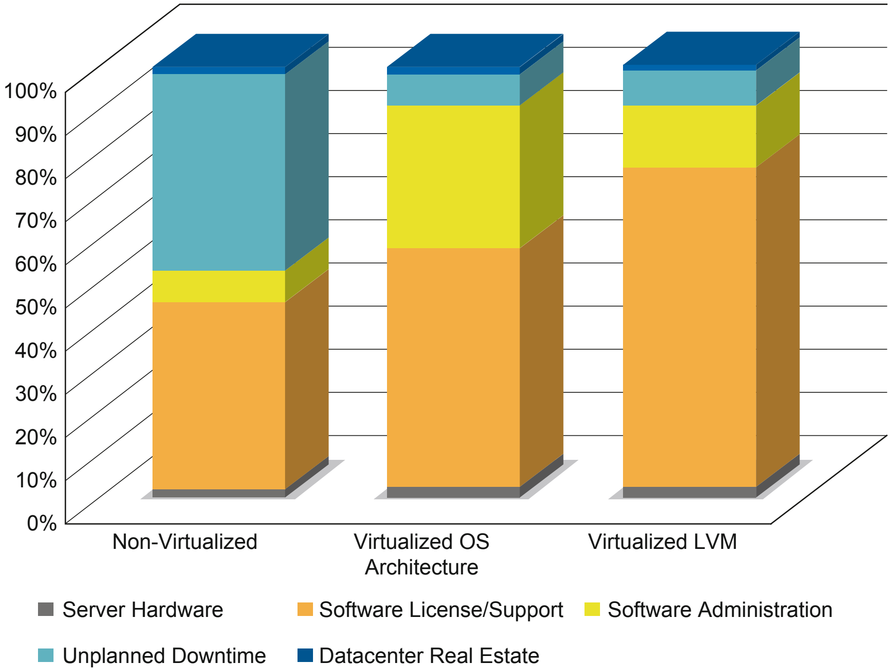


Fig. 4.3 The cost of ownership

4.2.3 Business Operations

Business operations covers the consumption and non-technical management of IT services. It includes, but is not limited to, how an enterprise deals with semantic management, security procedures, and transaction processes. Using Cloud services, an enterprise can initiate an Internet-based business on its systems, add extra virtual resources when needed, and remove these resources entirely when there is no demand. This elastic feature enables different business models, including pay-as-you-go subscriptions for computing resources or IT management functions, which allows enterprises to scale up or down based on their operational needs. Although the benefits are obvious, enterprises should evaluate their potential Cloud providers with similar validation patterns that they use today for their new and existing datacenter resources. This way, making decisions can be executed based on rational and adoptable processes in the organization to avoid any potential oversight by a whole new technology. These considerations should include the following four subjects [7]:

- *Deliver strategic value in addition to measurable cost-savings*: The initial success of Cloud offerings has been driven by their commodity prices. Going forward, Cloud vendors must deliver values other than low-pricing to stay competitive. This will force enterprises to consider how to position their future IT strategies from a business operations perspective.
- *Move core business operations to the Cloud*: Although some major enterprises have announced their migration plans to move their core business operations to a Cloud, clear use cases for leveraging Cloud services as part of their competitive edge are still vague. It is questionable, in some cases, whether or not Cloud services can indeed offer trustworthy operations support that matches current enterprise business practices. Furthermore, when an enterprise's competitors also use the same set of Cloud providers, it is difficult for the enterprise to claim a distinguished position.
- *Address complexities of business integration*: The complexity of an application process framework normally increases when deployed technologies evolve over a long period of time. This implies that the transformation to Cloud technology would be more difficult where highly customized applications or home-grown applications are in place, especially when organizations have their own unique requirements for functionalities, performance, and/or security. As the array of Cloud-based methods expands, the demand for integration tools and services will soar.
- *Match employee skills*: Possessing the required skills to manage the new Cloud technology while maintaining existing internal business processes imposes another dimension of challenges to an enterprise's transformation plan. It may be difficult to bring all existing technical personnel up to speed on advanced subjects of Cloud Computing with respect to architecture, implementation, and operation. On the other hand, it is equally challenging for an enterprise to recruit Cloud *Subject Matter Experts* (SMEs) from the market to assist with the transformation

project. This is due in part to the commitment and reeducation of the new recruits to comprehend existing business models and processes. Both cases involve risks from new and existing employee's openness to the new arrangement as well as availability of operational insights needed for the transformation.

4.2.4 Organizational

An enterprise should fully understand the organizational implications of maintaining an IT investment in-house versus buying it as a service from external providers. The IT managers as well as the business process stakeholders have to look at the short-term costs and long-term gains of the transformation effort. Service levels offered by different providers are critical for the enterprise to analyze the QoS regarding uptime, response time, and performance, with corresponding benchmarks to existing practices. Despite the extra cost to the enterprise, it is always beneficial to implement a proof-of-concept environment or prototype process to get the organizations through the initial learning process and provide proof points as to the feasibility of adopting Cloud technologies. A couple of issues should be looked at during the exercise:

- *Resolve intensifying channel conflicts*: A growing number of system and service vendors have launched or expanded their channel programs in order to extend their reach into new segments of the Cloud market. Many vendors are contending with a rising number of disputes between their direct sales teams and channel partners. More of these conflicts are expected to arise, becoming the vendors' internal, as well as their customer's, problem. From a service customer's perspective, this trend may cause more confusion, as the providers not only need to straighten out their technical issues but also need to lay down a clear supply chain relationship.
- *Distribute business levels*: For enterprises that have spent a decent amount of investment on their own storage and security systems, they will have a tough time justifying the decision to migrate to a Cloud environment. Similarly, existing software systems for cross-organizational applications may encounter challenges for a simple switch-over to a Cloud environment if their implementations are not fully open and convertible.
- *Escalating security threats and business reliability*: As previously mentioned, enterprises are generally worried about lost or stolen data. Most still see Cloud Computing as an unreliable security threat and thus listed security as their primary concern, followed by performance and reliability. As Cloud Computing services gain greater attention and acceptance, they will become a bigger target for hackers. The challenge will be to safeguard enterprise data from threats of external attacks outside the firewalls as well as internal attacks from applications that run on the same computing units. Some of these problems are beyond IT operations and may not be solved by technical solutions. For this reason, Cloud providers must work closely with their clients to obtain official certifications of

their security practices from independent third parties specific to the industry they are serving.

4.3 Software Services Perspective

The adaptation of Cloud-based software services is very straightforward when there is a considerably flexible approach to phasing-in or relating to other applications. In other words, treating a Cloud transformation no differently than converting a set of approaches, each with its own examples and capabilities, to a new vendor can be easy. However, the biggest challenge in SaaS adaptation may be the fact that there is no standard or single architectural method in place. Figure 4.4 shows software services in the Cloud architecture.

4.3.1 User Data

This section will focus on the relationships of data with enterprise software systems. It will also address issues of software service frameworks that enterprises should consider before their Cloud transformations.

From a service client's perspective, it is important not only to have access to the data but to also have comprehensive access to the services that process the data. Figure 4.5 portrays a sample data management flow depicting how current enterprise data can be converted to Cloud compliant presentation. Bear in mind, without an

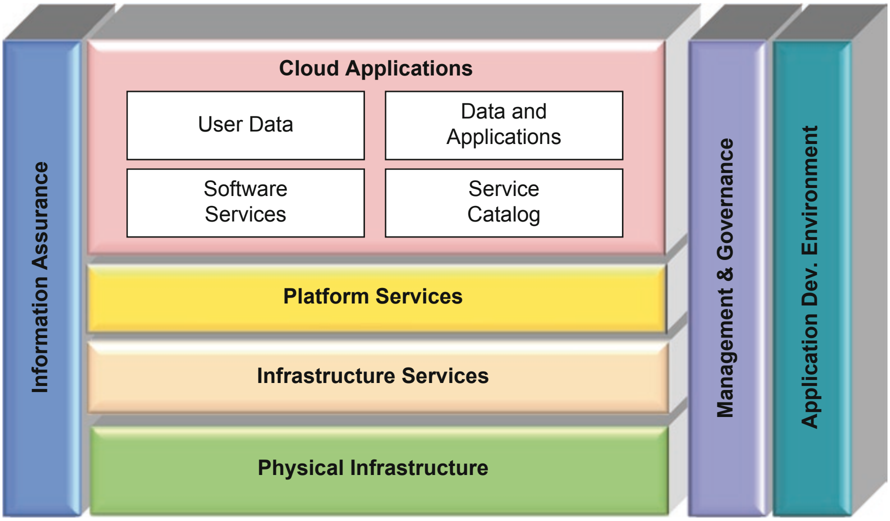


Fig. 4.4 Cloud applications in the Cloud architecture

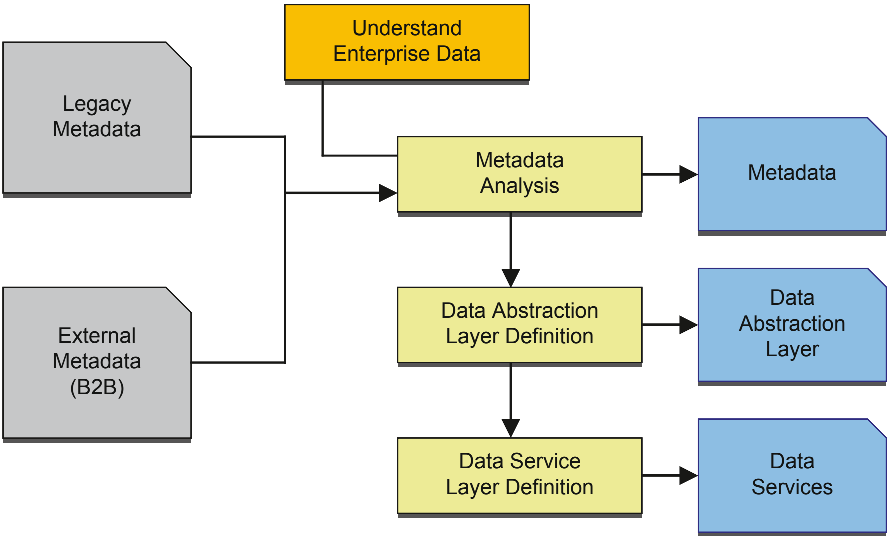


Fig. 4.5 The data management

open means to engage the applications that process the data, the IT shop will need to write extensive code to implement such functionalities. There are more challenges besides accessing and processing data. The following subsections are organized in a way to assist enterprises in thinking strategically about integration before tactical actions are taken. In Sect. 4.6.1, more data and security challenges and issues will be illustrated more closely.

4.3.1.1 Accessibility

Managing data in an internal or external Cloud requires data security and privacy, including the control of data movement. It also includes managing data storage and the resources for data processing. For users who need to access certain resources in the Cloud, the accessibility feature needs to incorporate access agreements such as acceptable use or conflict of interest. In some implementations, end user signatures are required to confirm their commitment to the policies. With such agreements as guidance, the client organization can invoke mechanisms to detect vulnerable code or protocols in firewalls, servers, or mobile devices and distribute appropriate patches to the target systems or devices as necessary. Therefore, the security of accessibility for both the end users and the Cloud can be guaranteed. Furthermore, the Cloud itself needs to be protected from any user with malicious intent that may attempt to gain access to information or shut down a service. For this reason, the Cloud should incorporate a *Denial of Service* (DOS) protection with improved infrastructure that contains more bandwidth and better computational power to filter and identify attacks to the Cloud [8].

In addition to these traditional capabilities of access protection, the Cloud architecture introduces new attributes to IT managers that did not previously exist. The following attributes challenge the software architects for how the new system should be designed in light of data protection, synchronization, and mobility.

- *Where is my Data?* As data can be managed through a third party provider, one of the top concerns from a client's perspective is the exact location the data is kept. To that extent, the client might not even know if their data is stored next to a competitor's database. Although it may not be feasible for certain vendors, it is always helpful for the client to ensure their providers are committed to storing and processing data in specific jurisdictions. For sensitive data, a contractual commitment in the form of an SLA to assure local privacy requirements can add comfort to the user community.
- *Who has access to my data?* In a large enterprise, appropriate permission policies to access corporate data are always an issue, especially when the company has a complex organizational structure. In a scenario where data is accessible by many employees and value-chain players for inter- and intra- company transactions, a sensible and effective security policy may not be straightforward and thus could potentially lead to data leakages. More discussion on this subject can be found in Sect. 4.6.
- *Is my data safe?* During the course of action while a piece of datum is accessed, delivered, or stored, the datum can be exposed to different levels of safety risk. Different attentions and solutions are required to assure appropriate safety. For instance, stored data risk data confidentiality and integrity and transit data risk entering and exiting a Cloud through devices controlled by unknown and unsafe owners. Furthermore, there is no single, universally bullet-proof encoding mechanism to secure data in action.

4.3.2 Data and Applications

The challenges of data and applications in a Cloud environment are mainly surrounded by efficiency and effectiveness requirements where the networked stored data can potentially be spread across different locations.

4.3.3 Integrity

The new application adaptation and system integration effort pose integrity risks. This issue is related to application packages, thus data should be looked at from both vertical and horizontal perspectives.

Horizontally, the value of transforming existing enterprise applications to Cloud-based infrastructure arises from the need for more efficient application delivery and operations. When more applications from different vendors are connected, the measure of success relies upon how well these functionalities are exchangeable and connectable within a Cloud environment. Although vendors claim to have open architecture and flexible contract-based relationships with their offerings, the transaction and interaction details between applications is not as transparent. When customization efforts are invested to integrate these applications with Cloud technology, can the corporate data and processes hold their integrity without suffering the efficiency and effectiveness of the original applications?

From a vertical integration perspective, Cloud providers must integrate and coordinate different levels of technologies and processes based on the client's business needs. When dealing with the client's communications needs, for instance, one way of assuring integrity is to implement the SSL or TLS. SSL and TLS assure that sessions are not being altered by others. At the Network Layer, the network can be secured by using the *Secure Internet Protocol* (IPsec). The SPs should be comfortable in dealing with different technologies and management domains sufficient enough to comprehend the associated issues. They also need to create enough tangible solutions to integrate those features into their SaaS SLAs with quantifiable and sensible metrics for business and operational liability.

4.3.3.1 Portability

Portability is the ability for an enterprise application to change data or services from or to different SPs with limited proprietary interfaces. In today's enterprise applications, if the development or porting effort to bring a system to a Cloud environment requires a lot of changes in the enterprise, bringing that system back in-house will be difficult and expensive as well. There are three aspects to this topic:

- *Outsource to a Cloud provider*: This concern is also called "long-term viability." Ideally, a Cloud SP will never go away or get acquired by another company. In most events, the data will remain available even after such an instance. However, enterprise customers should prepare a mitigation strategy to minimize potential impact to their business operation if such an event does occur. Additional agreements to protect and convert their sensitive data to a standard form can add another layer of protection to this risk.
- *Bring services back in-house*: If an enterprise decides to move their application back in-house, the effort requires moving the data and processes into a non-Cloud architecture. This can help measure the degree of portability of the original transformation project. If an enterprise uses Cloud services as a transition project to develop their in-house solution, a well-planned strategy can save time and money during and after the replacement implementation.
- *Move to another service provider*: Similarly, if an enterprise decides to move their applications to another provider, interoperability and migration policies are among the most critical issues concerning portability. These issues will remain challenging for both the clients and vendors mainly because there is a lack of standards to facilitate interoperability. In the later chapters, we will see some intermediate solutions proposed to ease the cost and effort associated with portability.

4.3.3.2 Interoperability

Data interoperability and application interoperability are two common subjects in system integration. They deal with issues such as semantic interoperability for data to be defined and stored on one Cloud versus another. At the application level, an enterprise needs to consider the notions of transformation and translation so that data can appear native when it arrives at its destination. Interoperability is not solely about transporting data between different forms. A broader definition should also include data governance and data security as part of the integration effort [9].

It is important that both data and applications expose standard interfaces. Enterprises need the flexibility to create new solutions with an open framework to assure interoperability for data and applications regardless of where they reside. For a large enterprise that commits to integrate Cloud applications with their legacy systems, they must secure the applications because the mixed functions move around the Cloud and the legacy systems during and after the transformation. It is extremely important that the Cloud providers support recognized interoperability standards with security considerations so the enterprise can combine any Cloud provider's capabilities into their solutions safely. Cloud standards bodies, such as the Open Cloud Consortium, are investigating interoperability standards to facilitate data interoperability between Clouds. Although cited in many open forums, secured data access interfaces and data governance are among the unsolved concerns of many enterprises.

As mentioned earlier, there is always a danger that sensitive data could fall into the wrong hands during inter- or intra- Cloud data exchanges. In the case of direct system interaction between Clouds, the price of performance efficiency comes with the risk of data leakages. Without a security-focused intermediate system, data leakages could potentially occur either as a result of the operator having unnecessarily high privileges assigned or by accidental or intentional misuse of their given privileges. Therefore, before a standard is defined, there is always an option to use intermediate data exchange systems to gate Cloud interactions in order to avoid major alterations to the existing applications. Such a system can shield internal data and logic representation from other systems by focusing only on external interfaces and security issues. This solution, however, is one of the most expensive alternatives with heavy performance and cross-boundary policy implications.

4.3.3.3 Software Services

Cloud applications and services that fall under this category are targeted at end users. The providers deliver business functionalities to meet specific business needs such as CRM or application development and testing.

One of the interesting observations from many reports indicates that most business applications, regardless of how they are delivered, are almost never used out of the box without some form of customization. This is especially true in enterprise content management solutions. In fact, the Cloud providers may intentionally demand customization as part of their solution to add differentiators to the offering or increase service revenue.

4.3.3.4 Agility

The agility of SaaS represents the degree and ease of configuration and customization of its applications. Higher agility enables application users to quickly become competitive with their newly added features as differentiators. It improves an enterprise's efficiency in running customized business systems for their end users. Because the service vendors are able to optimize their services for all customers at one time, the agility of SaaS advances efficiency and allows the Cloud providers to pay off their investments faster than in a single tenancy environment. Multi-tenancy makes the economics work for both the business user and the SP. The concept of multi-tenancy will be discussed in more detail in Sect. 4.7.1.2.

Knowing that agility is one of the best ways to accommodate applications in scale, the challenge of an enterprise is how to assess an appropriate size (to be sure they are large enough) of their IT assets to take advantage of Cloud technology. First, let us review the two technical drivers of agility:

- *Virtualization* allows the abstraction of computing, s torage, and networking resources from underlying infrastructure. It shields the users from the knowledge of the underlying resources and thus reduces the required skill level to operate these applications.
- *Automation* eliminates the need for human intervention in common, repeatable tasks and decisions. With automation, users can focus on business aspects of the applications rather than worry about IT-related operations.

When considering the effort it takes for virtualization and automation features to support the customization of a SaaS application, an enterprise should consider whether or not the business applications should adopt full featured service offerings from the selected vendors. Even if all vendors promote the idea of pay-as-you-use technology, for a business owner, the answer may not always be affirmative. Enterprises are currently looking into the least-common-denominator systems' infrastructure to enable customization. Thus, a full scale application architecture revision may not take place immediately due to this and other practical reasons such as cost and skill-set restrictions. In this case, the level of agility will be diminished by the degree of interoperability.

4.3.3.5 Flexibility

Software service flexibility implies the level of freedom that a SP's environment allows for users to customize or extend the Cloud platform for their needs. This includes building new business applications that leverage customer data. There are two aspects to the challenge of flexibility:

- *Software Upgrade*: For applications that are not service-oriented, packaged software may not be able to simply move to a SaaS model with minimum effort. Assumptions made by most enterprises about software environments are not necessarily true in an SaaS deployment. Many packaged enterprise applications cannot simply be moved to a Cloud environment without adversely affecting the rest of the enterprise's business or IT ecosystem.

- *Software Scaling*: Some Cloud vendors take into consideration the need to burst beyond licensed limits without impacting service, aiming to solve the problems described in Sect. 4.2.1.2. The software scaling mechanism allows providers to charge back to the customer later, based on what resources or number of users were actually serviced. There are other options, such as flat-fee or transaction-based utility services using on-demand models, that are charged monthly based on how many users or requests were served regardless of the licensing utilization. For complex business operations with licensing costs that vary in a large pricing range, a unified model will not be sufficient. In this case, a comprehensive policy should be installed to maximize the returns for both the client and the suppliers of Cloud software. Guidance for such a policy is not generally available in a standard form and thus is open for argument.

4.3.3.6 Adoptability

SaaS adoptability measures how easy it is to migrate an enterprise's application to a different environment conveniently. Each of the major Cloud providers imposes an architecture that is dissimilar to the common architectures found in most enterprise applications. For instance, AWS offers a rather flexible infrastructure by provisioning an "empty" image for users to store anything in. However, applications cannot be easily moved in or out of this infrastructure due to its idiosyncratic storage framework. That means migration is not as easy. Other providers' offerings have different levels of weaknesses. Samples of three major players are listed and illustrated briefly below [10]:

- *Salesforce's Force.com*: This is a development platform tied to a proprietary architecture deeply integrated with salesforce.com infrastructure and not very compatible with regular enterprise applications. Enterprises have to leverage the force architecture by creating their own add-ons. As a result, enterprises are justifying the gains of locking-in to this proprietary solution versus the effort to develop fresh applications to address their needs.
- *Google's App*: This is a set of application services written in the Python programming language. Google oftentimes makes decisions based on technological superiority despite evidence that it retards adoption. Python may be able to deliver values in Google's environment more efficiently, but by no means is it the most popular scripting language around. Enterprises' adoptation of such new methods will likely require employee's reeducation on this newer technology, or will require enterprises to seek out qualified experts in the field.
- *Microsoft's Azure*: This is a .NET-based architecture that offers services based on the existing Microsoft development framework. In order to create a market differentiator, this product does not offer regular SQL *Relational Database*

Management System (RDBMS) storage. As a result, adopting applications will require a different database application architecture, thus preventing the existing enterprise applications to migrate to this environment easily. The lack of simple migration will dissuade many Microsoft users from exploring Azure.

Any Cloud-based architecture differing from the established enterprise application architecture does not necessarily imply deficiency or difficult migration. The degree of difficulty in migrating from an existing application actually depends on which target Cloud offering an enterprise chooses to migrate to. Even in the absence of an automated tool, there is the potential for SPs to perform migration services efficiently and inexpensively. It is up to the enterprises' business and technical stakeholders to make a rational justification.

4.4 Platform Services Perspective

Developers are always in search of a better software platform to improve their development projects and this trend will continue as long as there are existing opportunities. Enterprises' internal IT management ultimately has to choose strategy of competing with Public Clouds, providing services that embrace the Cloud technology, or both. From a developer's perspective, the key to enticing software developers to make their decisions on the migration plan relies on the development experience itself. This experience includes productivity, flexibility, types and strength of services, and so on [11].

This section provides a summary of technical challenges that a business should consider during the design or adoption of a PaaS solution. Figure 4.6 portrays the relationship between the PaaS with the rest of the Cloud infrastructure.

4.4.1 Data and Information

Data and information are two essential elements of an enterprise's Cloud transformation project. They hold the ultimate metrics for whether or not the project can be successful. Therefore, the IT managers and application users must examine these elements before other management and infrastructure issues are discussed.

4.4.1.1 Information Management

Information management in PaaS includes the structure, management, storage, and distribution of data that is used by applications and services. Traditionally, enterprise applications used relational data models and relational DBMS to enforce data

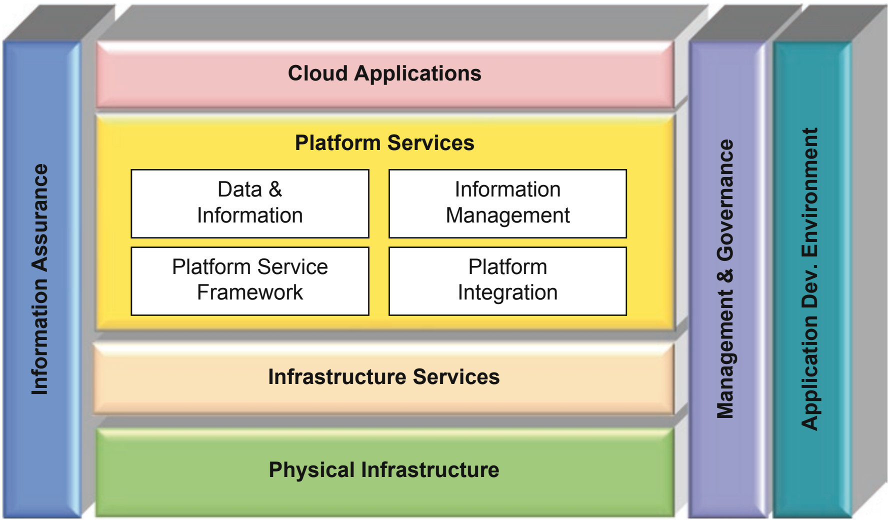


Fig. 4.6 Platform services in the Cloud architecture

consistency, transactional reliability, and increased throughput. These management systems use the *Atomicity, Consistency, Integrity, and Durability* (ACID) principles as the measure to support a reliable database design. These principles are being challenged by the new Cloud technology due to the notion of Data as a Service. Ubiquitous access to data can now be offered independently of the platform that hosts the source data. Designing a service that will run in a Cloud forces the SP to consider requirements that are related to multi-tenancy (more in Sect. 4.7.1.2) applications. This requires alternative schema designs that must be flexible, secure, and versioned. The management challenges can be concluded in the two main issues below:

- *Change to a non-relational management paradigm*: Systems are increasingly processing semi-structured or unstructured data (such as documents and media) that are not well-suited to structured relational data models and thus require generalized data models, such as name-value stores and entity stores instead. Some providers adopt these models in their offerings in the hope of providing greater flexibility to tenant-specific schema changes, but leave the management of data redundancy and possible inconsistencies to the application. This results in a more complicated data management system and pushes unnecessary management complexity back to the enterprise developers.
- *Update partitioning strategies*: Partitioning strategies must be able to support the new application that is scaling out of the underlying databases. The enterprise must be ready to support much greater volumes of transactions and manage much greater volumes of data than before. This is usually implemented by functional segmentation or horizontal partitioning. Such changes impact the current

schema designs and data partitioning strategies with a high potential of performance implication. To improve the performance of such changes, some vendors are moving away from the ACID principles and moving toward the *Basically Available, Soft State, Eventually Consistent* (BASE) model in the hopes that the new scheme can work with the decoupling logical partitioning method more effectively [12].

Both of the above efforts challenge the portability between the traditional enterprise information management mindset and the Cloud-based enterprise information management offerings. Additionally, this method requires the information management system to possess the ability to verify and ensure the integrity of the data during Data at Rest, as well as the privacy and security during Data in Progress.

4.4.2 Platform Service Framework

The framework of PaaS concerns overall application design, development, testing, and deployment. It should also cover the architecture, tools, and management related to PaaS. The following subsections will focus on the key challenges that appear to be the most critical to the enterprise users and the SPs.

4.4.2.1 Scalability

PaaS can improve software development by scaling the software environment through the elasticity of resources. For example, a developer can get extra storage space as an on-demand resource instead of placing a work order and waiting for several days for the permission. The Cloud also helps developers create multiple versions of evaluation environments for their applications. Moreover, a tester may acquire extra VMs to either generate test data or perform data analysis in order to shorten the software assurance schedule.

From a management perspective, software monitoring can be done by monitoring API calls for server requests across Cloud domains. Although Cloud vendors' open systems can facilitate better monitoring, this issue ultimately rests with the developers and clients on how much effort will be needed and where the check points should be installed. With Cloud services as an external function to the enterprise applications, the monitoring function no longer has a purely technical focus. The applications will need to expand their functions in order to deal with business implications such as SLA compliance. The other key feature in the management aspect is "auto-scaling." Many providers claim to be elastic but this really means that their offerings only have the potential to be elastic. These services will not automatically scale as the application becomes heavily loaded. This in turn will require the developer to reconfigure the system based on their expected scale and thus puts the burden back on the enterprise's resource management strategy.

4.4.2.2 Portability

Virtualization is one of the contributing factors of enforcing enterprise software portability. However, this cannot be simply driven by the software development platform. Instead, it is enabled by the software developer. To elaborate the impact, let us briefly look at the lifecycle of a software service from a developer's perspective:

- Determine the functional requirement, design the architecture of a software service, and identify the required Cloud resources.
- Write and test the software on a single machine and name specific instances of objects that act as services for the rest of the application.
- Activate the named instances of the servers on the target network once the software is executed satisfactorily on one system. Push the service on various evaluation servers and test the application.
- Create a permanent partitioning map of the application, distribute the services to the production servers, and make the service available to the Cloud community once the application is tested satisfactorily.

Although the above flow may seem rather typical and straightforward for an enterprise to port its business applications to a Cloud platform, non-technical professionals continue to rely on their technical counterparts to determine how a Cloud transformation can solve a particular problem. Applications such as CRM, custom Web applications, or even open-source data processing systems almost always rely on special knowledge and skills to create, compose, integrate, configure, or distribute software services to meet related business needs. The enterprise executives must assess the ROI and risks to determine if the knowledge and skills should be eliminated after the transformation. Should these enterprise assets be a part of the package that ought to be outsourced to a third-party provider?

Technically, one significant drawback of Cloud Computing is its limitations with regards to certain hardware (processor) architectures when dealing with the scalability issue. The hardware limitations can potentially prevent developed systems from being deployed to different classes of the computing environments. Although this is in the process of changing, it is still a barrier that developers and Cloud experts need to overcome at this moment.

4.4.2.3 Tool Availability

The industry needs to provide more development tools that are Cloud-focused given the level of movement from recent enterprises' IT trends. To expedite the adaptation of Cloud technology, vendors should extend their successful open development languages in the Cloud or create innovative new approaches to Cloud development. Additionally, the offered platform should tighten up the development and testing experience of PaaS to make the software development process flow as seamlessly as possible.

A new concept of the dynamic service catalog (see Chap. 3) that can provide real-time resource statuses and information should be part of the platform. Such a function can provide application developers with up-to-date service awareness for better utilization of Cloud resources.

In addition, it is always beneficial for the Cloud vendors and SPs to develop a *Physical to Cloud* (P2C) migration tool. If these tools can translate services to several different Cloud architectures, or facilitate translation between non-Cloud and Cloud environments, it will ease the portability concerns for the prospect enterprises.

4.4.3 Platform Integration

Ultimately, a PaaS-enabled application must be integrated with existing enterprise applications or other new Cloud-based applications from other providers. This section aims to address some key technical challenges in this domain.

4.4.3.1 Level of Virtualization

It is always a challenge for enterprises to balance technology revolutions with related business development between newer methods and existing solutions. For instance, if the application developer is getting sufficient functionalities directly from their development platform, what is the need for advanced services from the Cloud? If the platform is capable of hiding the computing infrastructure to distribute today's application components, then why use something from third party providers to accomplish the virtualization? Additionally, there are other concerns enterprises should address in the transformation plan including the following:

- *The level of control corresponds to the level of virtualization*: Hiding functional and resource details from the application users is a strong driver for virtualization, however doing so tends to cover up some technical insights that IT managers still wish to know. This may prevent the IT department from troubleshooting cross-vendor system defects or integration gaps, although many vendors have different degrees of APIs to handle such events.
- *The alignment with the current business objectives*: Virtualization can also discount the thoroughness of some key corporate practices, such as audit processes and governance of the core business. Although standard bodies are proactively trying to define guidance and specifications to comprehend this challenge, there is not yet a set of standards close to meeting these expectations.

4.4.3.2 Limitations

Most enterprise applications are connected to other applications and form complex systems that are interconnected through a variety of service functionalities such as data, functions, and presentations. Enterprises use a variety of integration techniques that may result in a tightly coupled service environment, which prevents easy separation and replacement by off-premises capabilities. In such cases, the transformation of Cloud services requires an enterprise to either establish work-around functionalities within its subsystems or install a bridge function between legacy applications and services that can be hosted locally or off-premises. Although this may seem like a solvable interface issue, the data layer integration posts another dimension of challenges to already complex considerations. For instance, if an enterprise allows off-premises applications to use the same data as on-premises applications, the application provider must consider a variety of factors, including where the master data should reside. For data that is read-only, data replication and synchronization will include both technical and process issues. For data that is both readable and writeable, data coordination and multiple-thread locking mechanisms are among the top technology mechanisms that are employed. Additionally, limitations to the PaaS level integration can also include the following areas of challenges [5]:

- *Service contracts and transactions*: For enterprises that are already service-oriented, business services can migrate to the Cloud architecture more easily. However, when the applications involve complex legacy processes with human-driven workflows and cannot be easily partitioned into a service-contract paradigm, the only viable option is to support a hybrid operation mode that allows the workflow to span both online and offline scenarios. Moreover, traditional transaction management with atomic approaches might no longer be possible. This may require an enterprise to examine alternative models that can ensure data consistency.
- *Applications development*: Applications that use a common service directory might be able to update the location and binding requirements of destination services within the service directory. The clients can then reconfigure themselves dynamically to be relocated off-premises. However, when clients need to interact with services that have multiple layers of contracts, service-virtualization techniques must be included to mitigate the problem. In this case, extra caution should be taken to ensure that the intermediary intercepts and transformation requests are compatible or transparent to the new destination services. This cross-layer SLA challenge concerning management and processes will be addressed in the later sections.

When enterprises transform their applications into Cloud architectures and become more dependent on services from multiple SPs, existing centralized transaction-based technologies must also be upgraded to the Web 3.0 framework [13].

4.5 Infrastructure Services Perspective

IaaS advances the adaptation of virtualization technologies and also matures the operational methodology that will replace traditional IT services, such as storage and network virtualization, with a new form of resource virtualization. The control software and VM present a uniform API and hardware abstraction layer for applications to access resources such as CPU, memory, storage, and networking. Through the Web contract interface, the service bundle that deals with network I/O or storage I/O is now directly available to the enterprise application. This allows multiple applications to share the same physical systems in a multi-tenant environment, managed by their related management software.

Although many infrastructure technology issues impeding enterprises' efforts for moving to a service-oriented model have been largely solved, some old issues are now magnified by the loosely-coupled relationship. Several non-technical issues, such as billing and chargeback, and manageability and security challenges will be illustrated in later sections. The following section focuses mainly on subjects related to the infrastructure itself and performance [14].

4.5.1 General Infrastructure

The consistent behavior of the management and execution scheme for a homogeneous architecture is a common principle for both the physical as well as the virtual IT infrastructure. To assure the applicability of these principles to Cloud infrastructures, vendors have chosen to hide their service diversity at the hardware layer, software layer, and/or virtual container to allow for a consistent interface across different types of Cloud middleware and hardware. Virtual containers, for instance, are created to be less like hardware abstractions and more like service delivery abstractions. With various forms of virtualization technology, enterprises have a variety of alternatives, each with associated challenges, for their infrastructure investment strategy. Figure 4.7 portrays this concept in an IT-centric view. Here virtualization helps the traditional IT resource to be detached from its service. Through the Cloud

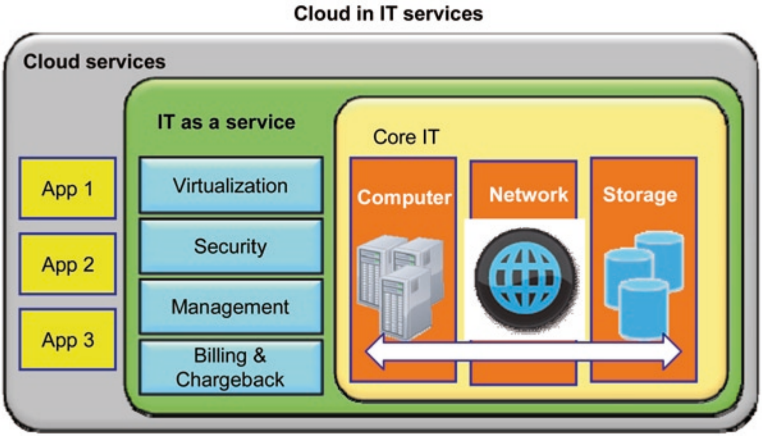


Fig. 4.7 The Cloud services in an IT environment

service tier, these physical resources are further separated from the enterprise applications shown at the right, bottom layer.

4.5.1.1 Automation and Commoditization

When an enterprise demands an organization to be dynamic for meeting unexpected market changes, combining virtualization technology with a dynamically expandable computing infrastructure service can facilitate an effective technology solution. However, the maturity level of these two Cloud functionalities posts some considerations for the IT managers:

- *Automation*: When integrating IaaS as a part of the virtualized instances, IT architects can organize the virtual instances into atomic units to localize and isolate potential service failures. In the event of a service failure, this design allows atomic units to be swapped for speedy service recovery and to avoid further damages to other service units. However, the design of how to effectively group these atomic units and how to efficiently define the cross-unit interactions for optimized service can be a big challenge to both the SP and the client. Management coordination among different units for guaranteed end-to-end QoS can also be a complex issue.
- *Commoditization*: Commoditization speeds up the development of a new generation of appliances that are more specialized and powerful in order to promote automation through large and geographically dispersed resources. New levels of load and mobility require more network capacity, automation, and management. This in turn requires more understanding of the packaging scheme with regards to its cost and operational implications. In the area of service delivery, load balancers are commoditized and specialized to support cross-layer coordination, although they can lack the needed features to address cross-vendor gaps.

Server-virtualization technology helps reduce the server-hardware footprint in an enterprise. By using IaaS, an enterprise can derive immediate infrastructure cost savings by replicating virtual server instances to run on the Cloud infrastructure as needed. While Cloud infrastructure-related services can bring many benefits that were not previously available to enterprises, the IT architects must continue to weigh broader design considerations, such as availability, scalability, security, reliability, and manageability.

4.5.1.2 Network Capacity and Mobility

Today's network infrastructure contains millions of specialized servers connected by complex and growing networks. They consume huge amounts of energy, from electricity to the human capital required to keep the business going. As Cloud vendors take over these resources and consolidate them into a centralized environment, the complexity and efforts are more challenging than ever. In a narrower view concerning only the network capacity and mobility, the problems can be seen as follows:

- *Cross-technology control and management coordination*: One of the most important implications of the new network infrastructure is the network control software that now resides in the service and now dictates that the network will actually terminate inside the server. As newer networks are built on meshes of more powerful servers that in turn connect to other even more powerful networks, the communications and data load must be managed by a new generation of systems that can deliver unprecedented levels of automation and management. Specialization will shift from the hardware in the core of the network to yet-to-be-seen hardware automation and a more intelligent network management scheme.
- *Border conditions and inter-Cloud interoperability with existing networks*: Many existing applications requiring very high bandwidth or very low latency and jitter may prefer to use a Layer 2 connection. As many Cloud providers' mainstream product lines are based on Layer 3 networking, the inter-Cloud or intra-Cloud interoperability will be an issue. Such cross-layer interconnection can present technical challenges, especially in the areas of control and management coordination that the Cloud providers are not ready to deal with.
- *Cross-technology QoS and SLA coordination*: As most Public Clouds clearly made large investments in Layer 3 connectivity and thus might be understandably reluctant to consider alternatives, it seems the enterprises interested in staying with their Layer 2 networks will have to absorb the majority of the transformation costs. This will include cross-layer QoS assurance and restructuring of the SLA management—a brand new area for both the providers and the clients.

4.5.1.3 Data Movement and Integrity

When dealing with data movement and integrity, data synchronization, migration, segregation, and recovery are among the most important subjects.

Data synchronization has two fundamental drivers for the growing adoption of Cloud integration services. Firstly, growing adoption of Cloud-based applications and platforms leads to even greater data fragmentation challenges in enterprises. The enterprises must be able to ensure the data integrity in a computing environment where data and applications are spread across different services or vendors. Secondly, departmental line-of-business purchases and implementations of service applications have led to a need for easy-to-use, self-service integration solutions that non-technical users can manage while IT organizations remain in control.

Data migration adds two flavors to the application: direct and staged. Direct data migration refers to moving information from one data source and data schema to another and translating the differences in semantics from the source to the target system. Staged data migration refers to a temporary location where the data from the source system or systems is replicated in order to support more complex and valuable data integration operations. While the IT department may be interested in supporting data management between front-office and back-office systems, IT organizations are typically tasked with broader data integration and data quality requirements across all enterprise systems.

Data segregation can also be another major issue. Because in the Cloud the data is typically in a shared environment alongside data from other customers, encryption may be effective but has no guarantee for true security. The Cloud provider is obligated to provide evidence that their encryption schemes are well designed and certified by experienced specialists. Furthermore, the encryption must not post noticeable performance burdens to the applications. In the event of an encryption accident, the effected enterprise data is totally unusable and vulnerable and thus must be assessed carefully.

With respect to service recovery, a Cloud provider must be clear on the consequences of the enterprise data and service in case of a disaster. Service offerings that do not replicate client data and application infrastructure across multiple sites are vulnerable to a total loss. When processes are in place, an enterprise should also verify its feasibility and the time needed to achieve full recovery.

4.5.1.4 Bug in Large-Scale Distributed Systems

When applications are deployed across a Hybrid Cloud infrastructure, it can be difficult to debug application failures that occur because of infrastructure malfunction. This is a tough challenge especially for vendors who have a responsibility to maintain high volumes of interconnected complex applications. Traditional network-monitoring and tracing tools might cease to work across the boundaries of enterprise and service-provider firewalls. An enterprise must ensure its Cloud-infrastructure providers can provide diagnostic tools with enough functionality to help inspect Cloud-infrastructure flows. The challenge of troubleshooting system malfunctions are the following [15–17]:

- Software reliability in large-scale systems can be impacted by problems such as data corruption and deadlocks in parallel processing. Because data movements in parallel programs typically follow certain patterns, it is possible to extract data movement in order to check the violations of these invariants. These violations indicate potential bugs such as data races and memory corruption. Although it is theoretically possible to identify these bugs, a mature commercial product is yet to be release.
- Bugs manifested during one instance may not be triggered during another because of various nondeterministic events. Such non-determinism can be caused by different process execution orders, thread interleaving, signal delivery timing, I/O events, and so forth. Therefore, it is difficult to reproduce bugs and thus renders a significant challenge for detecting and locating these problems.

- Some software bugs can only be triggered in very large-scale systems, thus it is very difficult to duplicate the problems in a scaled-down environment. As a result, it may cause a huge waste of resources if developers cannot avoid occupying the full scale system for manual debugging. This will be an issue to deal with for solution architects from both the client IT and providers organizations.

4.5.2 Service Performance

Infrastructure service performance includes the availability, efficiency, and effectiveness of application services. Because a number of higher-level application services might be running on an outsourced Cloud infrastructure, any performance degradation at one service-provider infrastructure could affect more than one business function. This could mean, in the worse case, loss of business productivity or revenue in multiple areas. Therefore, enterprises should know whether or not their infrastructure-service providers can help mitigate such risks. Alternatively, an enterprise might use secondary infrastructure-service providers as a warm backup to accommodate service failure at the primary provider.

4.5.2.1 Availability and Reliability

The availability and reliability of service offerings is as much a business issue as it is a technology issue. Acceptance of service levels and corresponding price differences continue to improve as the business model of IaaS matures. There are two main factors that are involved in the calculation of availability: *Mean Time Between Failure* (MTBF) and *Mean Time To Repair* (MTTR). MTBF is obtained from the data sheets of the equipment. MTTR is the average time to fix and restore the resource in order to be put back into service. MTTR is based on the degree to which the system will be monitored by operators. Nowadays, the hardware technology allows internal and Cloud SPs to design solutions that match nearly any set of availability and reliability requirements at a specific price level. Each level of availability service costs significantly more than the previous level. Therefore, it is merely a price match effort instead of a technical issue.

As the number of IaaS providers increases, the flexibility of contracts in the form of SLA and price competition will evolve to provide service levels and price points that IaaS clients require. Currently, many of the service-level offerings are provided with a “one size fits all” mentality. These providers typically have limited levels of offerings without the possibility for modifying these SLAs. It also leaves no flexibility for many businesses that need to meet specific service levels for their users.

Moving forward, Cloud SPs should provide a dynamic service-level hierarchy along with a rational SLA management system to match the dynamic infrastructure and dynamic systems they committed to deliver.

4.5.2.2 QoS Governance

QoS as a widely deployed performance benchmark in many enterprise infrastructure services also plays a critical role as a Cloud service differentiator in dealing with service-oriented distributed systems. Virtualization of resources sets forth new challenges to be investigated within QoS and presents opportunities to apply the knowledge from a service-oriented paradigm.

Being a contracted performance objective, QoS is specified by consumers as a service requirement their providers agree to maintain and sustain in their operations. The Cloud providers need to consider and accommodate different QoS parameters for individual consumers as negotiated in specific SLAs. To achieve this, Cloud providers can no longer continue to deploy traditional system-centric resources that ignore incentives specified in the SLA. Instead, they should adopt the market-oriented resource scheme to regulate the supply and demand at market equilibrium, with an active two-way SLA to incorporate feedback in terms of economic incentives for both Cloud consumers and providers. This can promote proactive QoS-based resource allocation mechanisms that differentiate service requests based on actual utilization. Figure 4.8 [18] portrays a sample market-oriented Cloud architecture. This includes the functions of QoS negotiation between users and providers to establish SLAs, mechanisms and algorithms for allocation of virtual resources to meet SLAs, risk management associated with the violation of SLAs, and interaction protocols for interoperability between different Cloud providers.

4.6 Security Challenges

As mentioned in Sect. 4.1, one of the most significant technical hurdles for Cloud benefits to be crossed is security. Security encompasses many different things, including the policies on access control, identity management, monitoring, detection and forensics, encryption, patch management, privileged virtual environment, and protection of the actual virtual infrastructure. In the past, security has implied perimeter security, ensuring that no unauthorized access is allowed from the outside. In a virtual world, with virtual IT services, a physical perimeter no longer exists. Therefore, businesses must assume that all transferred data may potentially be intercepted. The information assurance mentioned earlier is an enforced idea that ensures security policies can safely travel with the data, be at rest, or be in progress across mixed physical and virtual environments.

However, virtualization security solutions today primarily focus on protecting the virtual OS, the virtual networks, or the control and management software itself—with the focus mainly on protecting infrastructure and perimeters rather than data. While protecting virtual infrastructure is important, data protection is actually much more important to enterprises.

Many answers to the protection of infrastructure and data can be traced to organizational, technical, and management issues. At the heart of the problem is the real

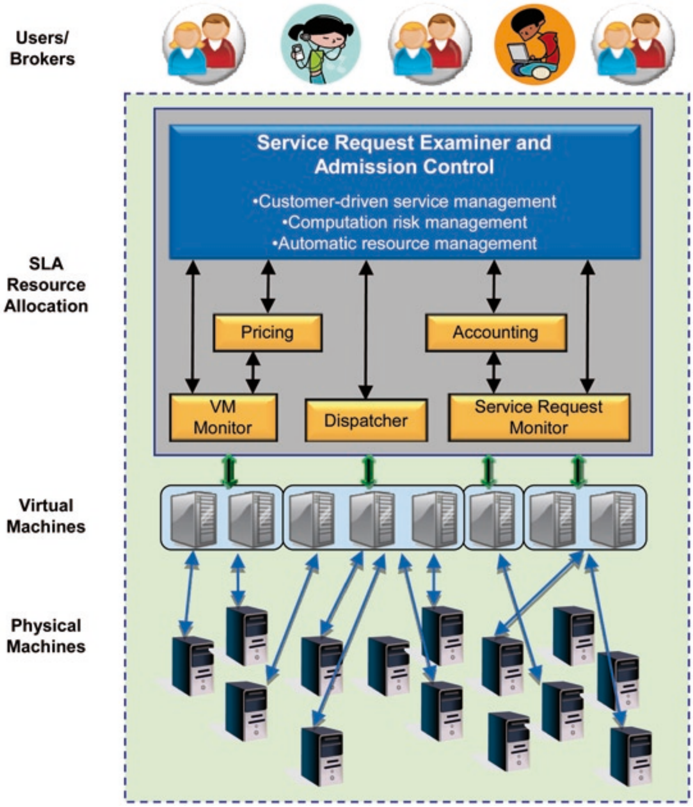


Fig. 4.8 High-level market-oriented Cloud architecture

nature of information security and its goals within an enterprise's business objectives. Before diving into the technical details, let us first look at the security layers and their associated functionalities, portrayed in Fig. 4.9 and listed below [19, 20]:

Each layer performs different functions to support the *Software-Platform-Infrastructure* (SPI) model and is also influenced by its peer applications as well as its vertical suppliers or consumers based on their business requirements. Samples of application for each layer are listed below:

- *Applications: Systems/Software Development Life Cycle* (SDLC), Binary Analysis, Scanners, WebApp, Firewalls, Transactional Security
- *Information: Data Loss Prevention*, *Content Monitoring and Filtering* (CMF), Database Activity, Monitoring, Encryption
- *Management: Governance, Risk Management, Compliance* (GRC), IAM, Value Analysis/Value Management, Patch Management, Configuration Management, Monitoring
- *Network: Network Intrusion Prevention and Detection Service* (NIDS/NIPS), Firewalls, *Deep packet inspection* (DPI), Anti- *Distributed Denial of Service* (DDoS), QoS, *DNS Security Extensions* (DNSSEC), OAuth

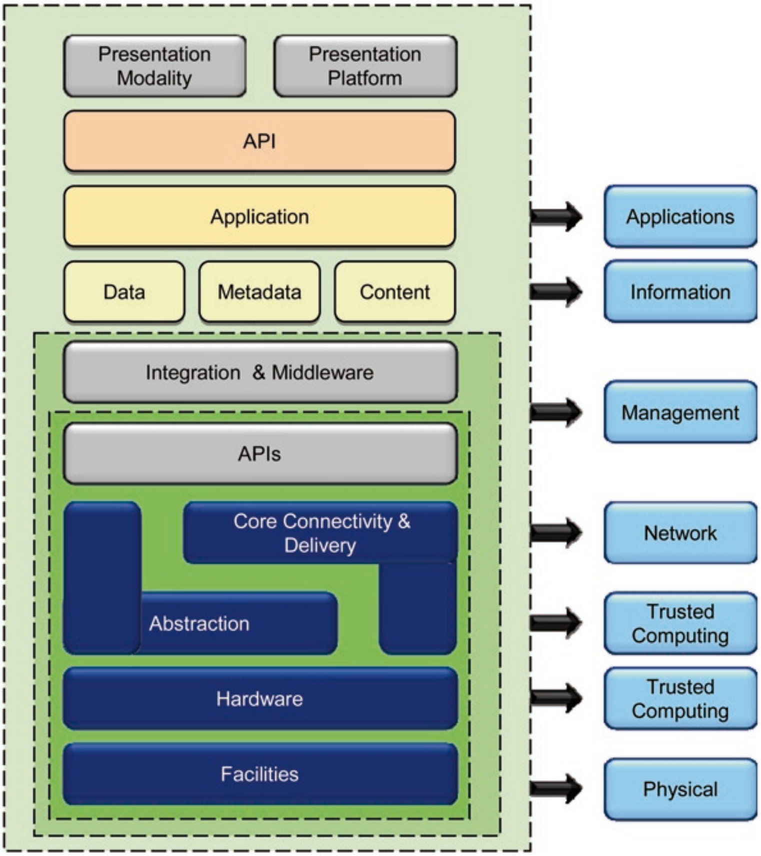


Fig. 4.9 Security and the Software-Platform-Infrastructure model

- *Trusted Computing*: Hardware & Software *Running Object Table* (ROT) & API's
- *Compute and Storage: Host-based Firewalls, Host Intrusion Detection and Prevention System* (HIDS/HIPS), Integrity & File/Log Management, Encryption, Masking
- *Physical*: Physical Plant Security, *Closed Circuit Television* (CCTV), Guards

Although most of the security layers will not be addressed individually in this section, the principles for achieving a high degree of information assurance are identical. Let us first start with data security.

4.6.1 Data

With the advent of virtualization, physical devices are being replaced by dynamic, on-demand virtual "devices,". Networks are being virtualized and applications are treated as atomic service units. The only remaining "constant" object is the data itself. Data generally has a longer lifetime and is thus more vulnerable for attacks than the virtual environment, which is created and brought down based on business and operational requirements. Regardless of which device the data is on or which network the data travels, continuous protection of the data is the only way to ensure the information assurance is ubiquitous and scalable.

Data fragmentation and dispersal held by an unbiased Cloud is a new way to protect data. In fact, shifting public data to an external Cloud can potentially reduce the exposure of the internal sensitive data. In the following subsections, we will cover some key challenges related to enterprise data.

4.6.1.1 Ownership

Most enterprises are uncomfortable with the idea of storing their data and applications on systems they do not control. Migrating workloads to a shared infrastructure increases the potential for unauthorized access and exposure. To reassure their current and potential customers, Cloud providers must provide a high degree of transparency into their offerings with consistent policy and mechanisms around authentication, identity management, compliance, and access technologies [21].

Data ownership in a large and complex enterprise can be a dangerous concept, and it should be the enterprise itself which owns the data. Feedback from enterprises' polls indicates that one of the biggest challenges related to data is the role of security in an enterprise. A National Cyber Security Alliance study found that over 73% of business users believed they had nothing to do with ensuring the security of data and that their IT departments should deal with it. Unfortunately, many IT departments do not have the appropriate authority to enforce user-related security. This lack of ownership hinders the implementation of a solid security solution [19].

The definition of privilege implies a person who "owns" the data and can create, read, write, or change it. There are several levels of privilege which an end user may have over business data and the definitions. These are categorized in Table 4.2. The following describes these privileges in detail.

- *See Data*: Limits the visibility to read data. This can avoid duplication and potential synchronization problems.
- *Change Data*: Limitation on who can change what data and when. It is important to incorporate version control of the data.
- *Change Data of a Business Object/Instance*: This reflects the responsibilities of different functional departments and their associated business decisions.
- *Change Foreign-Key Data about a Business Object/Instance*: Reassignment of a customer to a particular business organization. This must be governed by business policy. These changes are rare but can trigger data synchronization issues.
- *Create a New Instance of a Business Object/Instance*: Instantiate a new entry with updated attributes, descriptions, or conditions.

Table 4.2 Forms of power over data

Privileges	Example	Freq. of occurrence	Appl. impact
See data	Grant authority to view data	Common	Low
Change data	Grant, update, or insert authority	Common	Low
Change incidental attributes	Change a zip code	Common	Low
Change foreign key data about instances	Change a departmental assignment	Rare	Substantial
Create new instances of object	Create a new sales district record	Often	Minimal
Name an instance	Decide how to name a district	Rare	Downstream
Create new business objects	Create a new way to group dealers	Rare	Severe
Change definitions of objects	Mutate a district into something else	Rare	Dangerous
Change definitions of attributes	Change dealer type to credit rating	Rare	Dangerous

- *Declare a Given Instance of a Business Object*: The authority to give a name to a customer or a product. Generally limited to those who have the necessary skills and knowledge about the consequences.
- *Create a New Business Object*: This is a very rare event when a business executive needs to declare a new business object. Business analysis is needed to assess the cost and impact.
- *Change the Definition of a Business Object*: Also a major change that may impact the business applications. The enterprise needs to understand the consequences.
- *Change the Attribute Definition of a Business Object*: The change may impact the nature of the business application and thus should be monitored closely.

With all the security and synchronization issues considered, a better policy is to be as liberal as possible in granting access to data, and make it so easy to get to the data that no end user would need to rekey the data into his own personal version. Anyone who has any influence over the content or structure of the data can assume a "stewardship" responsibility to assure the quality of the data. Stewardship implies an accountability to external authorities, such as other departments, in order to make sure the interests of others are being considered. This will tighten the interdependency and integrity of an enterprise's organizational functions. The stewardship can be classified as one of the following three categories:

- *Quality Stewardship*: A shared role between an IT analyst and a key end user with the power to ensure data quality.
- *Definition Stewardship*: The responsibility to clearly define what the data means or to consciously make decisions about evolving that definition.
- *Access Stewardship*: The ability to permit or prevent data access.

These stewardships must be systemic in nature, thus decisions can be incorporated into the system implementation. Enterprises must include qualified SMEs in both technology and business to create these regulations.

4.6.1.2 States

The primary risk of data loss is the misuse or unauthorized disclosure of confidential data. Different aspects of the data lifecycle present different relative risks. In a Cloud application, the IT managers do not have physical control over a system. Therefore, the enforcement of security principals must rely on some other means to restrict access to information. Encryption of information has come to be the most important way to restrict access to meaningful information, even when access to a physical system cannot be controlled. Thus, encryption becomes a critical component of security when IT services are delivered via the Cloud. To illustrate the protection of data, the industry typically divides data into the following three states [22, 23]:

- *Data at Rest*: This is the state when data/files are on computers and/or storage devices, e.g., USB flash drives. In this state, the relative risk level is low. Because the data can be at risk through loss or theft of laptops or backup drives, the potential risk of data leakage is "one-to-unknown." Data residing on the server is presumably only accessed by authorized users. The data can be regarded as not secure if (a) access to the memory is not rigorously controlled, (b) regardless of how the process terminates, the data can be retrieved from any location other than the original at rest state, (c) the storage device does not have enforced strong keys/passwords, or (d) the storage device allows the user to store passwords on the media.
- *Data in Transit (Data in Motion)*: This is the state when data is transferred via networks, mobile telephones, wireless microphones, wireless intercom systems, or Bluetooth devices. At this state, the relative risk level is high. The potential risk of data leakage can occur from "one-to-many," as one authorized user could leak confidential data to many unauthorized users. Protecting Data in Transit is probably one of the easiest tasks among the three, with the exception that the origin and destination nodes must assure protection of Data in Progress.
- *Data in Progress (Data in Use)*: This is the state when all data is not in the rest state. In this state, the relative risk level is medium. The potential risk of data leakage can occur from "one to one." In this case, data on only one particular node, e.g., in a network, could be stolen. This data can be regarded as not secure if (a) access to the memory is not rigorously controlled, or (b) regardless of how the process terminates, the data can be retrieved from any location other than the original at rest state.

Virtually all of the above vulnerabilities can potentially be exploited regardless of whether it is SaaS, IaaS, or PaaS. It is the clients' and providers' joint responsibility to protect data in these three states.

4.6.1.3 Anonymity

Anonymity helps enterprises manage the relationship of data between providers and customers. This technique protects data security even if the data is owned and managed by the customers. The theory is that if data is made anonymous at the source and controlled by the customer, the customer trusts the provider who made their data anonymous. In return, the customer will reveal their data attributes, allowing the provider to create other personalized profiles such as advertising campaigns. This benefits the providers with protection from legal action, personalized advertising, and segmentation. It also provides benefits to the customers like anonymous data, personalized services, and so forth.

The basic idea of k-anonymity protection can be illustrated by a real-life example. For instance, a data holder that has a privately held collection of person-specific data wishes to share a version of the data with marketing researchers. How can the data holder release a practical, useful version of the private data, while guaranteeing that the individuals who are the subjects of the data will not be identified? The answer lies in the released version providing k-anonymity protection, making the information for each person indistinguishable from other k-1 individuals in the release. Figure 4.10 shows that the data included the name, address, ZIP code, birth date, and gender of each entry in the mailing list. This information can be linked by ZIP code, birth date, and gender to the department's employee information, thereby linking employee number, pay grade, and eligible vacation to particularly named individuals.

While anonymity seems like a workable solution, as users become creators of data in Cloud services, and especially in mobile applications, making data anonymous becomes a problem. In the method mentioned above, data is made anonymous by removing explicit identifiers such as name, address, and telephone number. The data may look anonymous enough by itself, however, when co-related with a dataset from other sources (e.g., facebook or social community profiles) people may potentially be uniquely identified [24, 25].

A practical solution should contain a rigid policy that is controlled by the user with the ability to manage all data, not just location or date of birth as in the example above. So far, a tool that is capable of supporting data anonymity for all levels of users to manage their data is yet to be seen.

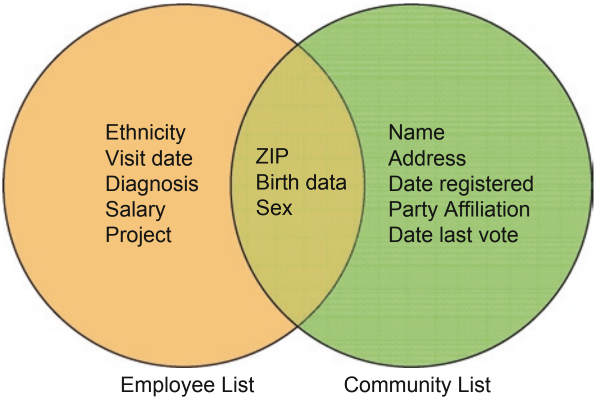


Fig. 4.10 Linking to re-identify data

4.6.2 Secured Access

Sensitive data processed outside the enterprise brings an inherent level of risk. This is because outsourced services may bypass the physical, logical, and personnel controls IT departments have over in-house programs. When an enterprise moves their mission-critical applications and data onto Cloud-based platforms, they must ensure that they can maintain the same level of access assurance as the current or previous internal applications. The enterprise IT managers and application owners should be aware of the systems as well as the people who manage the data. The providers should prepare to supply specific information on the hiring and oversight of privileged administrators and the security regarding control over their access.

In the following chapters, we will lay down the best practices for managing user identity and access across Cloud-based environments, such as who is responsible for managing identities, how to ensure the right access is available, and what is the proper mix of preventative and detective controls to secure a Cloud environment. Let us start by looking at the two common access technologies [20, 26, 27].

4.6.2.1 Two-Factor Authentication

Lacking the appropriate control over the network that provides connectivity to Cloud storage or Cloud Computing resources means all data sent could potentially be intercepted and even altered. As a result, sensitive information, such as login IDs and passwords, can be stolen. The Cloud service login process that provides strong *Two-Factor Authentication* (2FA) and complies with industry policies and guidelines can enforce a secured access.

In addition to the traditional authentication factors, such as login IDs and passwords, 2FA requires the addition of a second factor: the addition of something the user has or something the user is. By using a single set of 2FA credentials, an enterprise can increase the level of protection for corporate applications and data in the Cloud by providing fast and convenient token or token-less authentication. Underneath this method, the SAML is used to support the 2FA implementation, thus users can log on to a Cloud system using their existing two-factor, token-based or token-less credentials. Once logged on securely, the Cloud portal then allows easy *Single Sign-On* (discussed in the next section) to each Cloud service, without requiring further authentication.

While 2FA is becoming the de-facto standard for remote access to server-based business applications, most Cloud solutions still only provide authentication with static passwords that can be easily compromised. It is up to enterprises to emphasize their needs and pressure the vendors to provide an integrated solution.

4.6.2.2 Single Sign-On

Potential security threats range from service disruptions that are Internet hacks, to the risk of proprietary business logic (in application code and trade-secret content) being discovered and stolen. The practice of a secure-by-design and security-review process becomes more crucial for delivering applications to run on Cloud Computing platforms.

Single Sign-On (SSO) is a property of access control of multiple, related, but independent software systems. With this property, a user logs on once and gains access to all systems without being prompted to log on at each item. Single sign-off is the reverse property whereby a single action of signing out terminates access to multiple software systems.

Integrating SSO with existing enterprise identities is a key requirement and priority of many enterprises that adopt Cloud services. SSO provides convenience and better application experiences to end users and can reduce security issues that arise from having to manage multiple security credentials. Rationalizing and consolidating multiple identity systems within an enterprise is usually the first step in meeting the SSO challenge. New identity-federation technology can also improve the portability of existing user credentials and permissions and should definitely be a key part of the SSO strategy with Cloud SPs.

When a security credential is stolen, it implies that many systems in the enterprise will be in danger of attack. The enterprise should have a well-designed procedure to quickly disable the account to minimize potential damages.

4.6.3 Data Governance

Data governance is a set of processes that ensure data quality, business processes, and risks can be continuously managed and improved throughout an enterprise. It ensures that data can be trusted and that people can be made accountable for any adverse event that happens because of low data quality.

Data governance is a quality control discipline for assessing, managing, using, improving, monitoring, maintaining, and protecting organizational information. It is a system of decision rights and accountabilities for information-related processes to be executed according to agreed-upon models. The models describe who can take what actions, with what information, when, under what circumstances, and using what methods. They also describe an evolutionary process for an enterprise, altering the company’s way of thinking, and setting up the processes to handle information so that it may be utilized by the entire enterprise. As enterprises move to a service-oriented paradigm, their customers are ultimately responsible for the security and integrity of their own data, even when it is held by a SP. Meanwhile, SPs are subjected to external audits and security certifications.

For different industries (e.g., financial services and telecommunication services), enterprises must comply with many regulations. By moving the data into the Cloud, an enterprise will lose some capabilities to govern their own data and will rely on the SPs to guarantee the safety of their data. Additionally, there are more challenges on data governance:

- There is currently a lack of a universal policy language that governs the appropriate protection needed to enforce security upon servers, laptops, removable

media, and so forth. The policy must be embedded in the data itself and be understood by every device.
- Enterprises may have problems obtaining support for investigations, especially when certain data is owned by other departments or a supply-chain partner stored in different Cloud vendors.
- In a complex service environment where vendor solutions are integrated with home-grown systems, it will be a challenge to enforce indirect administrator accountability.
- There are potential technical issues in examining data quality due to proprietary implementations.
- There can be complications in obtaining needed log files from the transit and final devices that support the business operations.
- Due to a lack of standards, QoS metrics of data governance from different Cloud providers may need further translation or correlation in order to obtain a cross-departmental view.

Although the above challenges can be mitigated, the Cloud customer's inability to respond to audit findings may be one of the biggest challenges. The reason could be due to a lack of experience in understanding the impacts of the findings or the absence of an existing process to solve quality problems.

4.6.3.1 Information Lifecycle Management

Information Lifecycle Management (ILM) is the practice of applying policies to assure effective management of information throughout its useful life. RIM professionals have been using this practice for a long time to manage information in the form of paper, microfilm, negatives, photographs, audio or video recordings, and other assets. The operational aspects of ILM include backup and data protection; disaster recovery, restore, and restart; archiving and long-term retention; data replication; and day-to-day processes and procedures necessary to manage a storage architecture.

As Cloud SPs take over many data management roles from enterprises, they should also incorporate the following procedures into their services:

- *Creation and Receipt*: Create data from a member of an organization at varying levels or create an information recipient from an external source. The format includes correspondence, forms, reports, drawings, computer input and output, and other sources.
- *Distribution*: Send the internal and external information to others.
- *Use*: Generate business decisions, document further actions, or serve other purposes after information is distributed internally.
- *Maintenance*: Process filing, retrieval, and transfers. Filing is the process of arranging information in a predetermined sequence and creating a system to manage it. Transferring information refers to the process of responding to requests, retrieving information from files, and providing access to authorized users.

- *Disposition*: Handle information that is less frequently accessed or has met its assigned retention periods. Retention periods should consider the potential historic, intrinsic, or enduring value of the information. This may include ensuring that others cannot obtain access to outdated or obsolete information.

Although very impotent, there is no guidance to track these problems and no one is able to track them except for the SPs. Therefore, it is up to the SPs to provide guarantees that customer data is safe and access to data is restricted and protected.

4.6.4 Data Leakage

Sensitive data includes credit card, social security, or bank account numbers of customers or employees. It can also be extended to include intellectual property or competitive information. There are many ways this data can be at risk and the drivers to prevent data leakages can be listed as the following:

- Growing cases of data and IP leaks. These create risks for personal or corporate sensitive data.
- Regulatory mandates to protect private and personal information. For example, an enterprise's corporate Website can lose employees or private customer records due to phishing.
- *Health Insurance Portability and Accountability Act* (HIPAA), *Gramm-Leach-Bliley Act* (GLBA), *Sarbanes-Oxley Act* (SOX), *Payment Card Industry* (PCI), *Family Education Rights and Privacy Act* (FERBA) demand compliance to a more stringent security guidance.
- Internal policies enterprises can customize to meet their special security needs. For instance, a software development company may be more concerned about its software source code than other information.

Data Leak Protection (DLP) is also referred to as data loss protection or prevention, anti-data leakage, insider-threat protection, or outbound content management. It monitors, documents, and often prevents sensitive information from leaving an enterprise without authorization. The capability can dynamically apply to the desired type and levels of control at different data states. Functionally, a DLP includes the following features [28, 29]:

- Perform packet inspection on outbound network communication including e-mail, IM, *File Transfer Protocol* (FTP), the HTTP, and other TCP/IP protocols.
- Track complete sessions with full understanding of application semantics.
- Detect and filter content based on policy-based rules.
- Use linguistic analysis techniques to monitor and match data patterns.

Most DLP products work by scanning Data in Motion for e-mail, IM, or removable media that leave an enterprise. Some products also scan Data at Rest for information in data stores. This helps enterprises get a handle on all the sensitive data they own. They discover data on the network, end points, e-mail gateways, and file shares, and use predefined policies to prevent data from leaving the enterprise by blocking networks, ports on laptops, or applications. Administrators have the option to alert the data owners when sensitive data is leaving the company, or take action to block or quarantine identified data.

Although these solutions discover and classify data by their importance, most of the solutions cannot persistently encrypt or protect that discovered data with appropriate access controls or classification policies. Since these solutions do not take preventative or protective action on data, the gap prevents them from fulfilling a complete cycle of protection.

4.6.4.1 Lack of Smart Data with Embedded Policies

As mentioned in Chaps. 2 and 3, information assurance provides more comprehensive protection over DLP solutions. This allows for persistent care of sensitive data with multiple levels of policies for encryption, access control, and classification. Its capability to embed policies with the data itself ensures protection of Data at Rest on various devices or in motion across the network. By focusing on the data and its associated policy, information assurance reduces the multiple integration points required by DLP and can achieve better leakage prevention.

Information assurance is a well defined concept in the defense industry with many practical implementations for both military and cyber applications. While the principles are identical, the nature of business data and its relationship with commercial applications have not yet been fully explored. To ensure that enterprises can adequately protect their data, they need to know how to make the data objects smarter by using metadata tags to carry security policies. After that, the data can be empowered to protect, replicate, or even delete itself as required, allowing data to communicate its vital characteristics to the devices it passes through or to other data objects throughout its lifecycle.

4.6.5 Security Framework

The basics of information assurance exist today, and they are very well suited to commercial and federal data protection scenarios. It is particularly designed to answer the protection requirements of regulatory compliance. In the following section, we will review some characteristics that may slow the development of such solutions [19].

4.6.5.1 Lack of Transparent Solutions

A useful and workable information assurance solution needs to be transparent. Enterprise users should not have to modify their work habits or change their business practices to take full advantage of the security solution. This is because most users will reject changes imposed on their familiar work patterns and may choose to bypass new security provisions that are not user friendly.

Additionally, any new security system should not require the enterprise's current software applications or computer platforms to be upgraded as part of the security deployment. The introduced solution must be capable of working with any device, on any platform, without requiring special patches or programming. In an ideal case, if the solution can be deployed as an independent service offering, most of the issues mentioned above can be eliminated. However, it is unclear how the enterprise security policy can be invoked to interact with all the enterprise data without having to be co-located with the data storage and applications. Beyond that, most IT environments today still support legacy systems to some extent. The case for an IT manager to justify major overhauls solely for security upgrades may not be strong enough.

4.6.5.2 Insufficient User Provisioning

User provisioning is a key feature of enterprises' user identity and security management. In the enterprise Cloud transformation plan, it must consider how its enterprise users and their associated security policy are provisioned by their Cloud SPs. For instance, when organizational roles for a user are changed, the corresponding identity management processes should be invoked to ensure the user's permissions are adjusted accordingly within and across the Cloud. Similarly, when a user leaves an enterprise, access to the enterprise Cloud should also be deactivated. The user provisioning activities for Cloud services must be automated as much as possible to reduce errors from manual provisioning. An effective solution can prevent loss of employee productivity that is due to service-access issues.

In today's network security solutions, most of the products are not properly architected to keep up with the complexity of an enterprise's internal business and organizational structures. Because of the deficiency in offering sufficient in-depth user provisioning automatically, many enterprise customers are worried about their vulnerability to attack.

4.7 Operational and Management Challenges

IT management deals with end-to-end lifecycle management of applications and services to accomplish its business objectives. The lifecycle management includes planning, implementing, operating, and supporting an IT portfolio that consists of the hardware, network, infrastructure, software, and services that support day-to-day business operations. Leveraging the Cloud platforms for enterprise business applications assumes that the services deployed are in a controlled environment with appropriate SLM, resource management, service provisioning, security and trust models, and monitoring [13].

Enterprises need a consistent view across all areas both on-premises and in Cloud-based environments. This includes managing the asset provisioning as well as the QoS that enterprises receive from the SP. Existing IT-management frameworks are still relevant. The operation and management department should consider the impact that arises as they integrate external operation processes, personnel, and tools into the existing IT practices.

Additionally, an enterprise must be familiar with the key functional capabilities needed for service management. They include:

- Defining policies to guide project implementation and operation procedures.
- Putting processes in place to systematize execution.
- Identifying organizational roles with clearly defined accountabilities
- Developing services with operations-friendly implementation best practices by including proper instrumentation.
- Implementing and maintaining the tools that automate IT-management operations.
- Monitoring the health and availability of Cloud applications and services.
- Collecting metrics and reporting on service usage, performance, and billing.
- Enabling automated provisioning of services and updating service configurations.

IT management in the Cloud service world must continue to embrace the end-to-end strategy of planning, delivering, and operating the IT capabilities that are needed to support their business operations. In the following sections, the discussion will be organized based on planning, fulfillment, and assurance. The billing aspect was previously addressed in an earlier section of this chapter.

4.7.1 Strategy and Service Planning

Enterprises must consider the business impact to their operational roles and responsibilities when dealing with their transformation to a Cloud-based service architecture. Business continuity, liability, and employee and customer satisfaction are all key concerns that must be addressed. Enterprises must establish clear and reliable business relationships with their Cloud SPs.

Additionally, enterprises should verify if the application-performance information and operation service interfaces from their SPs can be consumed by standard off-the-shelf IT monitoring solutions. Above and beyond the technical and non-technical challenges, they also need to consider the following issues:

- How easy is it to integrate with existing in-house OSS?
- How difficult is it to migrate back to an in-house OSS? Is it even possible?
- Does the system have enough customization capabilities to suit my needs?
- Will on-demand cost more? What is the sweet-spot to consider when weighing the Cloud versus in-house?
- Are there any regulatory requirements on the enterprise's industry that may prevent me from using the Cloud?

4.7.1.1 Expertise to Plan for Cloud Technology

Services that are outsourced to a Cloud provider are now maintained by administrators and operators who are not employees of the enterprise. Traditional IT roles and accountabilities might need to be collapsed into a group of service-provider's roles that are contractually responsible for the duties that are specified in an SLA. Legally enforceable liability clauses should be clearly defined to mitigate any negative result from a provider's poor performance. Similarly, IT-management processes for resolving user issues and technical problems are now handled by the SP. Establishing clear escalation procedures and integrating effective communication channels into the end user-support process of the enterprise are vital for minimizing service disruptions.

Before finalizing a Cloud vendor, enterprises should perform due diligence thoroughly to exam the SLAs (Sects. 4.7.2.2 and 4.7.3.1) for improving the understanding of what is guaranteed and what is not. With respect to the level of performance, the majority of Cloud technologies will always incur some service latency, possibly making the services slower than an application that runs in the enterprise's local datacenter. When third-party vendors are building enterprise services on top of the Cloud, enterprises should make sure applications can scale and perform well.

For a PaaS customer, the enterprise architects need to plan equipment and resources for customizing Cloud services to make them more relevant and tailored to their businesses. Proper man power should also be included to match the allocated applications resources. Although some resources will be outsourced, the planners should also consider the availability of the physical hardware and software components that need to be ensured for realizing the benefits of Cloud Computing. For the planning team, they should possess wider technical fluency and expertise in the selected Cloud Computing platforms, which tend to emphasize technologies such as open source or newer Web-style programming languages. Additionally, current application models will have to be modified and updated to fit into the new Cloud Computing models.

As a Cloud is administrated by SLAs that allow applications to be distributed among multiple servers (some may even involve other Cloud SPs), the enterprise should be equipped with appropriate SMEs who have insightful technical and business knowledge and can appreciate the advantages and implications of different options.

4.7.1.2 Multiple Tenancy Impacts

Traditionally, enterprise architects have simply focused on the design of the application and underlying data storage, as they are the only consumer of the application. When moving to a Cloud environment, this assumption is no longer valid. Today, an enterprise must consider multi-tenancy and different approaches for scaling out their enterprise services.

As discussed previously, *multiple tenancy* means a Cloud permits multiple clients to use the same resources at the same time, without them knowing it. To avoid

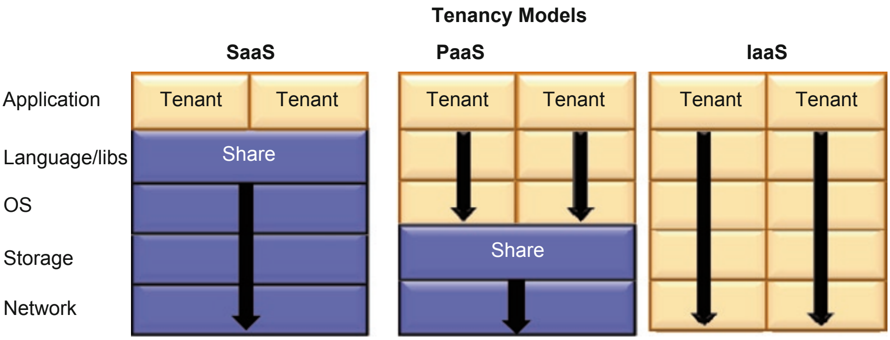


Fig. 4.11 Multiple tenancy models

potential conflicts of interest among customers in such a service environment, certain limitations and risks must be fully understood. Firstly, let us look at the relationships of multiple tenancy. In Fig. 4.11 [30], three service categories are listed to illustrate how multiple tenancy environments interact with different levels of Cloud resources. For SaaS, the environment starts sharing its resources from the software development languages and library. For PaaS, the sharing point starts with storage. Lastly, the IaaS uses different supporting resources throughout.

Although the problems associated with multiple tenancy take place in different service categories, the phenomenon and impacts have many commonalities. For instance, rogue applications or customers consuming too many resources can cause unexpected QoS degradation. Tenants can become active on-demand without informing other clients on the same premise. These resource and performance issues must be well planned and evaluated during the service planning phase.

As Cloud services can be introduced in many forms, planners must analyze issues from different levels of services. For instance, today's CPU design is based on a single OS that can run multiple applications. Most OS are not designed to adjust their footprint to the needs of being wrapped into software frameworks that hide any division between the application and the system they run on. This created the need for the additional functionality to both manage resources and manage applications.

As a result, additional features and packages to entice Cloud software developers, system administrators, or even consumers in the multi-tenancy environment are changing. The planning team is now obligated to incorporate issues such as how Cloud Computing will affect software architectures. IT operations professionals need to understand that their roles and responsibilities are changing. Operational and management principles, such as separation of concerns, must be taken into consideration to allow the security model of an application to change as well.

Traditional IT change management can be further complicated if the service is required to run at or very close to 100% availability, which provides little room for upgrades. Rolling updates or service upgrades that use update domains requires careful planning. This is in addition to the requirement that demands that SPs sup-

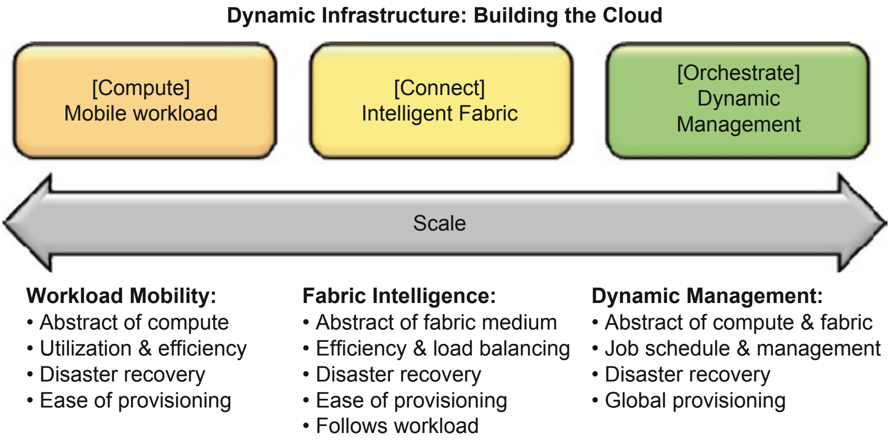


Fig. 4.12 Dynamic infrastructure to build the Cloud

port highly-available services. The planner should ensure that Cloud providers (especially external providers) have the ability to apply the right patches, workarounds, and access restrictions, and are able to isolate systems in a secure way. The planning team should consider if the patches are applicable to all the clients on that resource. If the changes do not introduce other impacts to other services or applications running on the same resource, the appropriate audit trails can be established and maintained. An enterprise must work closely with the Cloud providers to have a good understanding of usage scenarios to ensure that changes in service contracts or behaviors do not result in unexpected changes in business behavior. Figure 4.12 shows three strategic enterprise process infrastructure that can help planners build a dynamic Cloud services environment [7, 31].

Furthermore, during the service planning phase, enterprises must have the ability to simulate different environments for their applications, including development, user acceptance, and performance test environments. The purpose of such an ability allows enterprises to assure the quality and completeness of their production environments.

4.7.1.3 Failure Management

Enterprises should also architect their solutions so that failures of a service can be compartmentalized. Therefore, only the parts of solutions that are dependent on that service should be affected. The IT department that will develop or adopt Cloud-based applications must design their services to be more resilient when a remote service fails (remote services are usually outside the control of the consuming organizations). The related technical and operational strategy can help maximize business continuity. Techniques such as caching reference data and store-and-forward mechanisms can allow client applications to survive service-provider failures. Additionally, traditional atomic transactions might not be appropriate when interacting with remote services. This will require architects to consider alternative mechanisms, such as compensating transactions.

As services move outside of organizational boundaries, the time to access a remote service might also increase. Solution architects need to consider alternative messaging strategies, including asynchronous-messaging techniques, to increase the scalability of their systems.

Often times, the planners may also design their applications to interact with alternate SPs to improve availability and response time. This business driver may require the application to resolve issues such as service dynamics, or even modify the protocols that are used for interaction. For large enterprises that have a large number of client applications or services that interact with external services, the service configuration must be centralized for consistent management.

Adding to the above challenges are more complex changes that apply to the existing IT flows and processes:

- How does a Cloud environment support self-healing during major application, network, processor, or data storage failures?
- What is the disaster recovery plan of the new integrated service environment, including the response to a pandemic?
- How to comply with Export and Privacy laws in a value-chain network where foreign companies are also present in the Cloud eco-system?
- Will any enterprise data disappear or lose integrity when the enterprise online storage site shuts down?

4.7.1.4 Vendors Issues

Although the industry is committed to standardizing management interfaces and processes for Cloud services and many vendors have created dynamic infrastructure ready for adopting these standards, the maturity level of management standards is still not sufficient enough to satisfy the high expectations. In the current state, many providers have coordinated with some enterprises to implement these premature specifications. Others intentionally chose to implement their special features as market differentiators. Both camps will likely have challenges when they need to integrate with each other down the road. More specifically, enterprises may be challenged in the following two areas:

- *Vendor Scalability Unknown*: Vendor scalability is currently unproven. For instance, there is no standard or guidance to determine the size of a data storage volume. Although cross-OSS security and SLA management for traditional IT is well known, what are the new elements for the Cloud environment with respect to service and resource scalability?
- *Vendor Lock-In*: Enterprises may be forced to adopt Cloud technology prematurely. If an enterprise chooses to develop their new generation business system based on a vendor solution that is later proven to be not compliant with industry

trends, meaning it does not follow standards, they may have a serious problem of porting that solution to other providers later. Irrespective to the above standard challenge, once an enterprise invests significant resources in a solution, the commitment will prevent them from moving regardless of what vendor they use due to the cost.

An enterprise must be careful when choosing their solution partners and keep their eyes on management standards development. A phased approach may be the best way to achieve the goal of total transformation.

4.7.2 Service Fulfillment

Service fulfillment includes deploying applications with desired configuration requirements such as scale-out and high-availability. These functions instantiate and activate Cloud Computing environments whether operated by SPs or internal IT providers:

- *High-bandwidth, low-latency switching*: Standards-based and widely available, 10 GB Ethernet eliminates the need for unlimited bandwidth in Cloud Computing clusters.
- *Convergence to Ethernet*: Provides the technical foundation for Web technology.
- *Massive virtualization for agile (network) workloads*: A VM-aware network is required by today's massive Cloud Computing environments.
- *Scalable management*: Enables the lowest total cost of ownership for datacenter networks. Multiple switches located in multiple blade server chasses, even across racks, operate as one large virtual switch.
- *Advanced energy efficiency*: In massive Cloud Computing environments, this translates to saving hundreds of thousands of kilowatt hours.
- *Service-oriented*: The Cloud allows enterprise clients to access multiple applications online to create their own software or service.
- *Virtualized runtime environment*: Applications are not hardware specific. Various programs may run on one machine using virtualization or many machines may run one program.
- *Linearly scalable*: A Cloud should handle an increase in data processing linearly; if "n" times more users need a resource, the time to complete the request with "n" more resources should be roughly the same.
- *Data management*: The distribution, partitioning, security, and synchronization of data.

4.7.2.1 Cross-Cloud Processes and Policy Coordination

Cloud service extends the enterprise IT environment beyond its enterprise firewall. Such a change impacts deployed technology and traditional IT roles, and adds more complexity to the management and operational domains influencing accountability, operational procedures, and policies that govern the use and operation of deployed software and services.

Adding a Cloud-enabled environment to the IT service domain raises the expectation for the department to rapidly set up collaborated services that enable users to securely interact online. Such interactions could imply interoperability with back office systems as well as human oriented exchanges. Because Cloud vendors may have different product options, policies defining infrastructure and business constraints will be varied. It all depends on whether the policy can internally or externally interact with the deployed functionality. This scenario also implies the interoperability between Public and Private Clouds.

Application platforms today are unaware of their usage context. Because business interactions have the potential to become more complex, their business functionality in Cloud platforms will have to be managed with that context in mind to assure they behave in accordance with the enterprise policy. When enterprise applications are formed as composite services and provisioned in multiple Clouds, their IT department must have the ability to uniformly provision these composite Cloud services in order to satisfy specified business policy constraints. To accomplish this, the enterprise IT department must work closely with their SPs to harmonize the policy across Cloud boundaries. As necessary, when deployed in the multi-tenant mode, service SLAs may be used to reinforce the collaboration efforts.

The typical applications that manage users and access control are no longer enough to express roles and responsibilities in a Cloud environment. While functionally the business roles may stay the same, they potentially will be operated by people outside of or across enterprise boundaries. Therefore, access control and the management of roles and responsibilities must be more feasible to composite functional behavior into a distributed environment that can be governed by enterprises' policies. For example, they can be externalized from the business functionality.

As for policy management, traditional, inter-organizational or intra-organizational policy is embedded in enterprise IT platforms and applications. Scaling businesses globally will require new ways to combine and harmonize policies within and across external process networks and value chains. It will become increasingly critical for enterprises to establish clear and explicit definitions of governance, policy (regulatory, security, privacy, etc.), and SLAs for effective operations and management.

4.7.2.2 SLA Definition and Negotiation

To conduct business within a Cloud, it is important for Cloud consumers and providers to align on graduated SLAs and corresponding pricing models. One of the most common concerns regarding Cloud service performance is service availability and security. These are the two most critical issues for line-of-business applications, since downtime implies loss in key business applications such as order taking, customer interaction, and work processes management. As mentioned in the previous subsection, maturing Cloud capabilities into more advanced offerings, such as virtual supply chains, requires support for fully abstracted, policy-driven interactions across Clouds. This can only be realized by the Cloud providers if they adequately model and warrant such policy deployment with a set of SLA that can support integrated services across distributed and heterogeneous processes and infrastructure.

Unfortunately, enterprises today cannot reasonably rely on Cloud infrastructures or platforms to support their business due to a lack of satisfied SLAs. This concern is further exacerbated by the fact that some Cloud providers do not even offer SLAs. In most cases, the presence of an SLA does not necessarily change actual operations. It merely provides a vehicle to ease their responsibility. For this reason, some Cloud providers purposely minimize their financial exposure by limiting their SLA penalty to the cost of the lost service, instead of the financial effect of the lost service. So from the providers' perspective, the purpose of an SLA is more of an after-the-fact conflict resolution guideline [12, 32, 33].

As for SLA negotiation, the presence of an SLA may entice providers to behave in a manner that meets the agreement, but may not actually address the enterprises' needs. Moreover, the more difficult the negotiating goes, the more likely the provider is to fulfill only the bare requirements of the agreement rather than solve the actual problem.

4.7.3 Service Assurance

Service Assurance involves the day-to-day activities of starting/stopping/suspending Cloud applications; monitoring software and services; taking corrective actions when problems arise; managing customer helpdesks to resolve user issues; performing routine tasks, such as backing up data; and controlling and maintaining consistent service run states to meet the required QoS.

Both enterprises and Cloud SPs can increase their operational effectiveness by designing their systems and services with operational best practices based on various standards. The service architecture and execution discipline can benefit from the guidance in the best practices over the course of planning, delivering, and operating software and services. Standardized processes and information models can help the architects improve the awareness of the transition points in their applications when stability, consistency, reliability, security, and other quality factors are affected. Unified procedures and instrumentation features within the applications can facilitate the generation of sensible information to notify monitoring tools of abnormal events.

At the enterprise solution level, operational procedures are governed by IT policies and the outcome is measured by precise system and application health metrics, such as availability and response times. The IT department can then use service thresholds with benchmark references against the data in the application health state, performance counters, management events, logs, and synthetic transactions to assess impacts and propose ways to enhance their services.

4.7.3.1 Monitoring

Monitoring service performance is one of the critical ways for enterprises to manage and improve their service assurance. Monitoring means collecting data from the CPU, memory, disk IO, transactions, and others. While the measures of hardware resources are restricted and limited by the availability of vendors' implementations, the way to retrieve data can also impact the performance of management systems. For instance, frequent data polling from the managed devices will consume more computing and storage resources from the management systems and thus reduce the management systems' efforts in other areas. On the same note, an insufficient collection can make the collected data useless [7].

Measuring applications is a different challenge. Typical measurements of how long transactions take and how much latency occurs may not be sufficient enough to represent the actual application performance. Additional context related to enterprise policy and user behavior must be considered to improve the service SA. The financial implication of an appropriate service assurance function can be illustrated by the following numbers: Amazon found that every 100 ms of latency cost them 1% in sales, Google found that an extra.5 s in search page generation time dropped traffic by 20%, and a broker could lose $4 million in revenue per millisecond if their electronic trading platform is 5 ms behind the competition.

Given the above objectives, an enterprise's IT department should deploy a monitoring capability that can collect and exchange measurements in a Cloud environment. The operation strategy of the monitoring ability should outline the performance indicators and management rules that are required to gain visibility into the performance and availability of the external services. When predefined situations occur, the monitoring systems should raise appropriate notifications and alerts so a service anomaly can be detected early.

To successfully deploy this ability, developers of the monitoring service, whether they reside at managed units in the form of collection agents or are situated at the management system to consolidate and filter the collected metrics, must be familiar with the operational behaviors of such a service and seek the most effective way to automate the process and avoid human error. However, enterprises must deal with challenges from the following three areas:

- The feasibility of providing monitoring for VMs that can offer similar metrics as the in-house assets.
- The availability of standard interfaces that can gather metrics at every level of assets and resources.
- A set of agreeable data presentations (key performance indicators and key quality indicators) from the SPs that comply with industry standards and can be implemented by the existing IT framework.

4.7.3.2 Governance and Compliance

Although the enterprise no longer controls the implementation details of the outsourced services, they should be familiar with the mechanisms and procedures of

Table 4.3 Cloud service governing

Governing in the Cloud	Operating in the Cloud
Governance & Risk Mgt	Traditional, Business Continuity Management (BCM), Disaster Recovery (DR)
Governance & Risk Mgt	
Legal	Datacenter Operations
Electronic Discovery	Incident Response
Compliance & Audit	Application Security
Information Lifecycle Mgt	Encryption & Key Mgt
Portability & Interoperability	Identity & Access Mgt
	Storage
	Virtualization

the SP that might affect the accountability and liability between the enterprise organization and its customers. The function of governance and compliance covers this requirement.

Governance determines who is responsible for what and defines the policies and procedures that the enterprise and provider personnel need to follow. Cloud governance extends from the traditional customer and provider relationship and requires enterprises to govern their own IT platform/infrastructure as well as platform/infrastructure that they do not totally control. In the case where enterprises use multiple Cloud providers in their IT solutions, the monitoring and governing capability should be performed across these solutions. It is the providers' responsibility to supply consistent formats to monitor Cloud applications and service performance and make them compatible with enterprises' monitoring systems. These inputs must be detailed enough for the enterprises to appreciate the values of compliance and risks in violating the performance goals. Table 4.3 lists the guidance domains from the Cloud Security Alliance (CSA) [33].

Third party auditors may be used by enterprises to measure the service performance of their purchased Cloud implementations. Depending upon the level of commitment to functionalities and usefulness that Cloud providers pursue to achieve a higher competitive edge, they may choose to submit themselves to regular and formal assessments in an attempt to obtain accreditation. Some accreditation process needs to be undertaken every so often. Thus, enterprises should implement continuous monitoring of the Cloud system to lower the constraints on the Cloud vendor.

Although many SPs currently provide documentation intended to comply with auditing standards in the hope of assisting enterprises in determining if their IT practices can meet the clients' business and industry requirements, there are still many technical barriers:

- The providers may not have the ability to satisfy auditors for all their clients at different levels (e.g., security, financial-service guidance).
- The Cloud vendors may not be up to speed from a guidance and auditing perspective.
- Even if the above two bullets can be accomplished, the SPs may not have the ability to perform forensic investigation.

SLAs are paired with the governance process for accountability and liability. The form of SLAs does not typically protect an enterprise from what they are designed to do—minimize loss from unexpected service behaviors. SLAs are usually limited to the cost of the hosting service itself, not the opportunity cost of an outage or degradation (i.e., the amount of money the enterprise lost or did not make).

4.8 Conclusion

Although Cloud technology presents tremendous opportunities and values for enterprises, the usual IT requirements in the areas of security, integration, and so forth are still applicable. However, many new challenges arise because of the multi-tenancy nature (information from multiple companies may reside on the same physical hardware) of Cloud services, the merger of applications and data, and the fact that an enterprise's workload might reside outside of their physical, on-premise datacenter. This chapter examined these challenges in both non-technical and technical arenas. During the discussion, we also realized that the risk assessment of Cloud hosting services should be treated as a dynamic target, not a static situation. This is because the entire technology is developing rather rapidly. Today's vendor-specific statements may not be accurate several months from now. Therefore, the authors have tried to avoid as much vendor-focused discussion as possible.

Many issues and challenges were discussed based on the nature of their management and operational functions. One of the key areas for overall management is the lack of standards for supporting Cloud-enabled enterprises. Generally speaking, Cloud standards ensure interoperability so that enterprises' tools, applications, virtual images, and other resources or assets can be shared with other Cloud environments. Portability allows enterprises to take their applications or instances from one vendor to another and still be able to perform full functions. This chapter also pointed out that both outsourced SPs and enterprises that are developing Cloud services should implement operation-related service interfaces to automate management tasks such as provisioning user accounts, setting user permissions, changing service run states, initiating data backups, managing resource priorities, and securing application partitioning. A comprehensive policy system should be in place to divide the users' view of one application from the backend infrastructure or platform supported by many Cloud providers, as well as to facilitate architecture principles and IT management guidance for automating IT operations. Additionally, standardized mechanisms for dealing with lifecycle management, licensing, and chargeback for a shared Cloud infrastructure are just a few of the management and governance issues both enterprises and Cloud providers must resolve. So far, many standard bodies have been investigating the best practices to improve the existing industry management frameworks. Organizations such as the TM Forum, the ITIL, and the *Microsoft Operations Framework* (MOF) are among the few leaders available [34].

The decision to move to Cloud-enabled services impacts the non-technical and technical aspects of an enterprise. Business owners must be convinced that the ROI is achievable. The technical staff, including enterprise architects, developers, product owners/stakeholders, IT leadership, and outsourcing teams, must understand the efforts and risks of deploying their new services. Taking into account that human capital in the enterprise may be lacking, or the planning and transition teams do not have enough incentives (job security versus marketable knowledge) to stretch and learn the Cloud technology, the transformation results can be very frustrating [35].

In the following chapters, we will present practical options and solutions to address the challenges identified in this chapter. Throughout the book, the authors will also include other new methodologies and ideas that can benefit the adaptation of Cloud technology and services. We hope the content of this book is comprehensive enough to ease questions and concerns so that enterprises can execute their transformation plans smoothly and effectively.

References

1. Chappell, C.: Preparing for Cloud Computing: the managed services revolution. CA. Nov 2008. http://ca.com/files/whitepapers/ca_Cloud_computing_en_us_1108.pdf
2. The seven elements of Cloud Computing values. MWD Advisors. 2009–2010. http://itredux.com/wp-content/uploads/2009/10/Cloud-Computing-Values.png
3. Knorr, E., Gruman, G.: What Cloud Computing really means. InfoWorld. http://www.infoworld.com/d/Cloud-computing/what-Cloud-computing-really-means-031?page=0,2
4. Golden, B.: The case against Cloud Computing, Part: II. CIO. http://www.cio.com/article/478419/The_Case_Against_Cloud_Computing_Part_Two?source=home_ts (2009). 29 Jan 2009
5. Puhlmann, Nils: NCOIC Federal Cloud storefront workshop. CSA. 21 Sept 2009
6. Cloud Computing's other Achilles' heel: software licensing. f5 Dev Central. Jan 2009. http://devcentral.f5.com/weblogs/macvittie/archive/2009/01/27/Cloud-computings-other-achilles-heel-software-licensing.aspx
7. Kaplan, J.M.: 6 key challenges facing Cloud Computing in 2010 and beyond. Seeking Alpha. Jan 2010. http://seekingalpha.com/article/180606-6-key-challenges-facing-Cloud-computing-in-2010-and-beyond
8. Unharnessing collective intelligence: a business model for privacy on mobile devices based on k-anonymity. Dec 2008. http://opengardensblog.futuretext.com/archives/2008/12/unharnessing_co.html
9. Linthicum, D: The data interoperability challenge for Cloud Computing. Info World. Jan 2010. http://www.infoworld.com/d/Cloud-computing/data-interoperability-challenge-Cloud-computing-259
10. Golden, B.: The case against Cloud Computing, Part: I. CIO. Jan 2009. http://www.cio.com/article/477473/The_Case_Against_Cloud_Computing_Part_One
11. Chong, F., Miguel, A., et al.: Design considerations for S+S and Cloud Computing. Microsoft Arch. J. (Sept 2009). http://msdn.microsoft.com/en-us/architecture/aa699439.aspx
12. Ness, G.: Bringing Cloud Computing down to earth. Sys.Com. April 2009. http://gregness.sys-con.com/node/896594

13. Fraser, J.: Cloud Camp Milan, stories from the trenches. RightScale Inc. Sept 2009. http://www.slideshare.net/gabriele_bozzi/Cloud-camp-milan-rightscale-stories-from-the-trenches
14. Fellows, R.: How real is Cloud computing/storage? Info Store. Sept 2009. http://www.infostor.com/index/articles/display/8822717913/articles/infostor/backup-and_recovery/Cloud-storage/how-real_is_Cloud.html
15. Safety in numbers: a Cloud-based immune system for computers. Science Daily. Jan 2010. http://www.sciencedaily.com/releases/2010/01/100127085540.htm
16. Bachega, L.R.: Statistical approaches for finding bugs in large-scale parallel systems. www.cs.purdue.edu/homes/xyzhang/readinggroup/leonardo.ppt
17. Gao, Q., Qin, F., Panda, D.K.: DMTracker: finding bugs in large-scale parallel programs by detecting anomaly in data movements. The Ohio State University (2007). http://sc07.supercomputing.org/schedule/pdf/pap351.pdf
18. Buyya, R., Yeo, C.S., Venugopal, S.: Market-oriented Cloud Computing: vision, hype, and reality for delivering IT services as computing utilities. The University of Melbourne, Manjrasoft Pty Ltd. http://www.buyya.com/papers/hpcc2008_keynote_Cloudcomputing.pdf
19. Khosla, P.: Information security for the next century—why we need an information-centric approach to data protection. Carnegie Mellon CyLab
20. Gartner, J.B.: Seven Cloud-Computing security risks. Info World. July 2008. http://www.infoworld.com/d/security-central/gartner-seven-Cloud-computing-security-risks-853?page=0,0
21. Scofield, M.: Issues of data ownership. Information Management. Nov 1998. http://www.information-management.com/issues/19981101/296-1.html
22. DATA CENTER FABRIC: Data-at-Rest encryption scenarios. Brocade. Sept 2008. http://www.brocade.com/downloads/documents/technical_briefs/Encryption_Scenarios_GA-TB-100-01.pdf
23. Data at rest, data in transit, data in use. Pat's Daily Grind. http://padraic2112.wordpress.com/2007/07/26/data-at-rest-data-in-transit-data-in-use/ (2007). 26 July 2007
24. Sweeney, L.: k-anonymity: a model for protecting privacy. School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. http://privacy.cs.cmu.edu/people/sweeney/kanonymity.pdf
25. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: ℓ-Diversity: privacy beyond k-anonymity. Department of Computer Science, Cornell University. http://www.cs.cornell.edu/~vmuthu/research/ldiversity.pdf
26. Johnson, K.: Courion to lead webinar on managing secure access for Cloud Computing environments. Courion. Jan 2010. http://www.courion.com/company/press_release.html?id=560
27. Melcon, C., Avoyan, H., et al.: Secure access to SaaS applications: signify extends 2FA hosted service to Apps like Salesforce.com and Google Apps, Cloud Expo: Blog Feed Post. http://Cloudcomputing.sys-con.com/node/1201472 (2009). 25 Nov 2009
28. Data-leak protection. Network World. Jan 2008. http://www.networkworld.com/community/node/23754
29. Data leakage prevention—overview. Pune Tech. June 2008. http://punetech.com/data-leakage-prevention-overview/
30. Bias, R.: Managing storage in the Cloud: challenges in embracing Cloud Storage. GoGrid/ServePath. Jan 2009. http://www.slideshare.net/randybias/challenges-embracing-Cloud-storage-presentation
31. Dynamic infrastructure: building the Cloud. Blade Network Technologies. http://www.bladenetwork.net/userfiles/image/Dynamic%20Infrastructure%20-%20Building%20the%20Cloud.png
32. Andrei, T.: Cloud Computing challenges and related security issues. Washington University, April 2009. http://www.cs.wustl.edu/~jain/cse571-09/ftp/Cloud/index.html#partnership
33. Golden, B.: The case against Cloud Computing, Part: III. CIO. Feb 2009. http://www.cio.com/article/479103/The_Case_Against_Cloud_Computing_Part_Three

34. Open Cloud Manifesto. Spring. 2009. http://www.scribd.com/doc/19708804/Open-Cloud-Manifesto
35. Hils, A.: Key issues for network, messaging, mobile security and security services infrastructure protection. Gartner. 2009. http://www.gartner.com/it/content/787500/787512/key_issues_for_network_messa.pdf

Chapter 5
Networked Service Management$_2$

Cloud Computing can best be modeled as a service offering. Enterprises use the Cloud services to augment, replace, or enhance the enterprise service offerings. This is contributed by Clouds' mechanisms for service definitions and for service deliveries. For instance, SOA defines a powerful paradigm for service definitions, while Service Delivery Platforms define powerful paradigms for service delivery mechanisms.

As a technology enabler, SOA facilitates enterprises to couple loose units of functionality that otherwise have no embedded calls to each other. SOA is discussed in Chap. 1. An SOA enables the definition of diverse services such as SaaS, HaaS/IaaS, and PaaS. Each of these services requires a transformation of current EAs to incorporate the services. To incorporate SaaS, for instance, the enterprise ITs must be transformed into Service-Centric IT Architectures that incorporate composite services. Similarly, HaaS/IaaS require enterprises to equip the quality management monitoring functionality.

As of today, there is no single agreed standard definition of Service Delivery Platforms in the industry. However, the TM Forum is working on maturing some of their existing specifications in this area, especially for the *Service Delivery Framework* (SDF) and SDF management. Depending upon the context of applications, SDP is also referred as *Service Delivery Platform* (SDP). The SDF definition provides the terminology and concepts needed to reference the various components involved, such as applications and enablers, network and service exposure, and orchestration [1].

5.1 Overview

Networked service management refers to the decoupling of server-side software from hardware and the flexible dynamics and automation with which that software is run. With networked service management, a client of Cloud services does not know where or how Cloud assets, such as applications or storage, are resourced [2]. The Cloud services obscure the location of the assets. Users want to deploy assets

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_5, © Springer Science+Business Media B.V. 2010

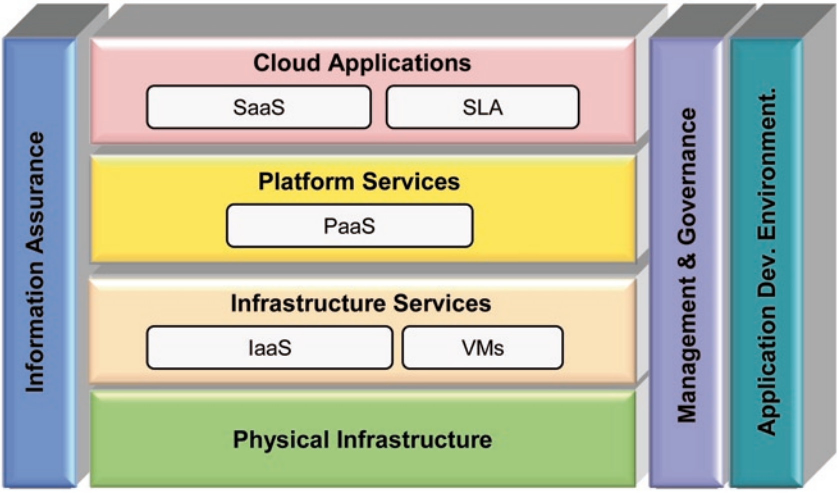


Fig. 5.1 Topics covered in Chap. 5

near-instantly over a network without binding to specific physical resources. Users also want the capacity of applications, e.g., supported number of concurrent users, transactions per unit time, or amount of storage to be adjusted automatically as demand fluctuates. This can potentially eliminate manual sizing and provisioning [2].

Figure 5.1 shows the three Cloud services categories: SaaS, PaaS, and IaaS. SaaS refers to complete applications provided by Cloud SPs, whereas IaaS refers to basic compute capability, i.e., machines with OS and storage. PaaS is between IaaS and SaaS and refers to an environment where one builds and runs an application platform in the Cloud using whatever pre-built components and interfaces are provided by that particular PaaS platform [2]. This chapter discusses the different types of Cloud services. The Cloud Applications, Infrastructure Services, and Platform Services boxes in Fig. 5.1 shows the topics covered in this chapter.

5.2 Software as a Service

When enterprises use SaaS, they need to consider the licensing models, how to transform enterprises to integrate SaaS provider offerings, and how to access the offerings. These topics are discussed in this section.

5.2.1 Software as a Service Licensing Models

In contrast to the one-time licensing model commonly used for on-premise software, SaaS application access is frequently sold using a subscription model, with customers paying an ongoing fee to use the application [3]. Fee structures vary from application to application; some providers charge a flat rate for unlimited access to some or all of the application's features, while others charge varying rates that are based on usage.

In general, SaaS licensing options include *subscription-based*, *usage-based*, *transaction-based*, *value-based*, *fixed-fee*, and *ad-based* revenue models [4]. In a *subscription-based* model, a monthly payment is calculated based on the software actually used, and it includes a commitment as to the actual number of users. Subscriptions are usually written on a per-seat or named user basis. On the other hand, in a *usage-based* model, payment is determined by application usage and is typically related to peak or near-peak levels of usage. Payment may also be tied to the number of CPUs, so that customers are charged for every computer that runs the hosted application, or payment may be tied to the number of concurrent users. In a *transaction-based* model, customers are charged for each business transaction. *Value-based* models are premised on the provision of whatever software is needed to achieve business goals, and payment is linked to the achievement of those goals. In a *fixed-fee* model, users generally pay a predetermined monthly fee based on the number of users supported, the particular application modules used, and the service and support levels specified by the customer. In an *ad-based revenue* model, users are shown advertisements in exchange for reduced fees.

5.2.2 Transforming Enterprise Architectures to Service-Centric Architectures

Figure 5.2 shows a maturity model that depicts the mannerism in which businesses procure and benefit from technology capabilities [3]. In the early stages, shown in panel 1 in Fig. 5.2, an enterprise user's needs are addressed by a collection of silo applications.

When a business initially considers incorporating technology, it is common for the business to associate the solution to its needs with a specific application that provides a narrow function. For example, if a user needs to interact with a partner on the design of a hardware component, he/she might be satisfied with a simple e-mail application as the primary collaboration and communication tool.

As an enterprise realizes that specific business needs are best met through a class of related applications, and not just one application, it evolves to adopt a service-centric view for its application portfolio, as shown in panel 2 of Fig. 5.2. Enterprise users' needs are then addressed through a service portfolio, each consisting of related applications offering a more complete set of functionalities. Going back to the partner-interaction example, the enterprise may realize that the collaboration effort can be enhanced through a Web portal that incorporates document sharing with versioning support, threaded discussions, real-time white-boarding, and slide-presentation support. As a result, the enterprise may decide to purchase and deploy

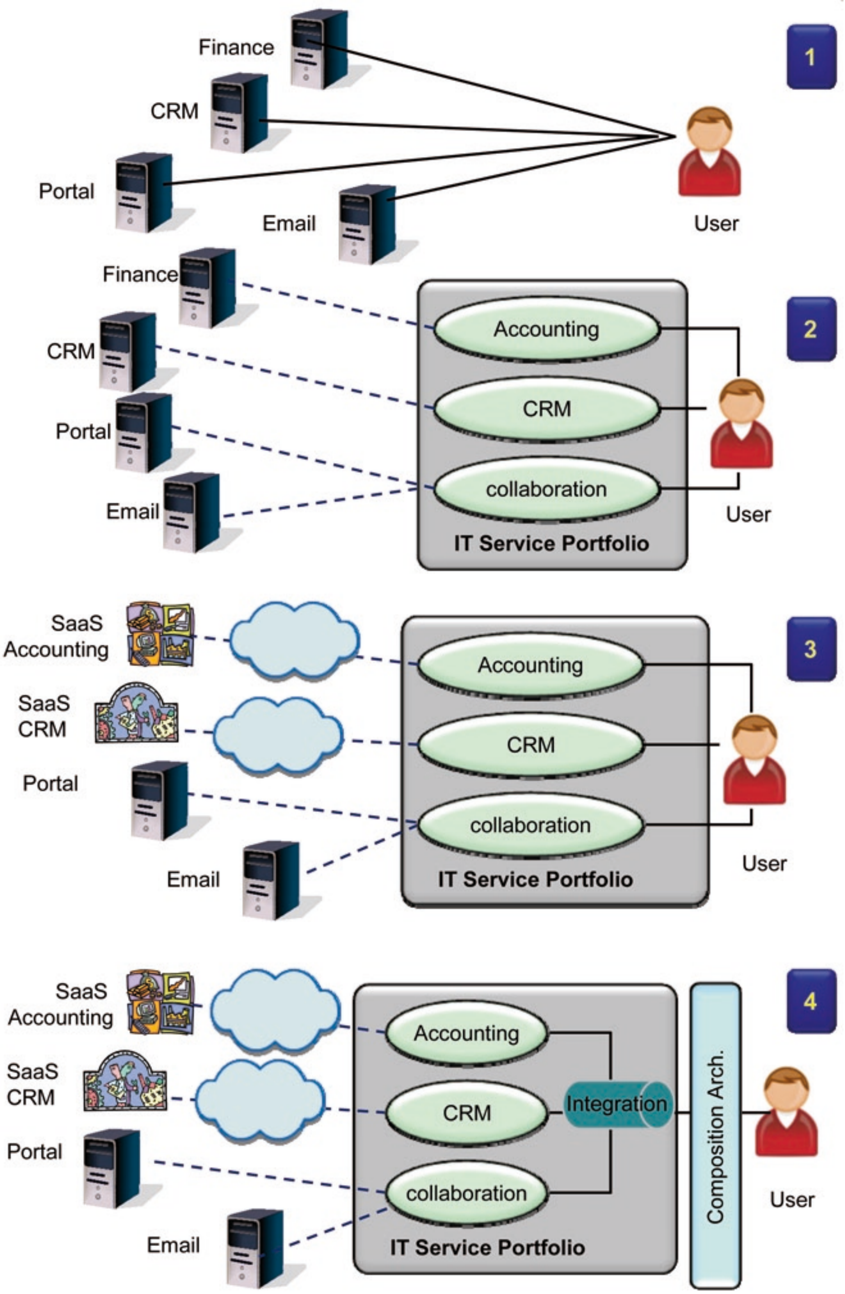


Fig. 5.2 Service-centric IT architecture maturity

a portal solution to expand the collaboration IT service capability that currently only has e-mail features.

In panel 3 of Fig. 5.2, the service portfolio is enhanced with additional options coming from SaaS providers, enabling enterprises to optimize their IT strategy and cost-allocation decisions. As platforms and line-of-business applications become delivered through a SaaS delivery model, enterprises are presented not only with an increased number of vendor options, but also increased choices for where and how the applications are delivered. SaaS influences an enterprise's allocation of resources through a variety of licensing, operation, and management models. Enterprises can trade direct control over service-implementation details for additional flexibility to optimize the strategy and execution of their core missions. Therefore, expanding the boundary of an IT's service portfolio beyond its firewall signifies another level of business and technical sophistication from the service-centric IT.

Beyond risk mitigation, an enterprise that has embraced SaaS as part of its service-centric IT can maximize the business gains by using features and data exposed through the portfolio of on-premise and in-the-Cloud services, as shown in panel 4 of Fig. 5.2. *Composite applications* provide the computing fabric for which business functions and information can be effectively composed (or mashed-up) for end users. When interacting with a composite application, end users focus on synthesizing and analyzing business information with minimal technology-related context switches.

Sections 5.2.3 and 5.2.4 below provide some detail on the roles that integration and composition architectures play in assimilating SaaS into the enterprise-computing strategy [3].

5.2.3 Enterprise Integration Architecture to Access Software as a Service Applications

Subscribing to a SaaS application means housing business data outside the controlled local network and within the Cloud infrastructure. An *integration architecture* specifies how to transform enterprises to bring this outside data into the logical enterprise infrastructure, so that internal and external infrastructure components can interoperate with one another to access needed data. Section 5.2.6 discusses some possible SaaS data architectures [3].

In most cases, implementing a SaaS application involves transferring data from one or more existing applications or data repositories local to an enterprise into a transformed system that combines internal and external infrastructure components. The following are examples of situations when such a transfer is likely to be needed [3]:

- The enterprise may need to bootstrap a SaaS application with preexisting data from an on-premise source.
- The enterprise may need to configure a SaaS application to depend on data produced by an on-premise source for part of its functionality. For example, a SaaS

CRM application may need to reference inventory data managed by an on-premise inventory application.

- The enterprise may need to configure an on-premise application to depend on data produced by a SaaS application for part of its functionality. For example, an on-premise payroll application may reference human resources data managed by a SaaS HR application.

Thus, in many cases, integrating a SaaS application with an enterprise environment means creating data dependencies that require data to be synchronized and moved between the SaaS application and in-house applications. An integration broker is used to manage data movement and system integration.

5.2.3.1 Integration Brokers

Many enterprises already use some kind of integration broker for exposing application functions, orchestrating business processes, and integrating with internal back-end systems. In many cases, the same integration broker can be customized and configured to perform integration and routing functions for a variety of internal and external data sources, including SaaS applications.

As shown in Fig. 5.3, data can originate from different sources by using different protocols and a variety of potentially incompatible mutual formats. An *integration broker* takes data from a variety of sources, determines how and where the data needs to be processed and routed, and sends each piece of data to its destination in a form that the target system can use. The broker usually has a pipeline architecture to which enterprises can add and remove modules that perform specific integration operations. Multiple logical pipelines can be used to process data traveling in different directions. In a typical case, for example, one pipeline integrates SaaS data from sources in the Cloud infrastructure with local data sources, and another pipeline takes local data and integrates it with SaaS data.

Data enters and exits the integration broker pipelines through data channels that define the protocols used to communicate with data sources. For example, one channel may be established to transmit data from a particular Web service to the broker by using REST [5] or SOAP [6]; another channel may transmit the data from the broker to a SaaS application by using FTP.

The modules in the pipelines determine how data is processed, routed, and integrated with data at the destination. A *metadata service* provides configurable rules that each module uses to perform its operations. The following are examples of typical modules:

- *Security module*: Incoming data typically is processed by a security module, which performs operations such as authenticating the data source or digital signature, decrypting the data, and examining it for security risks, such as viruses. Security operations can be coordinated with existing security policies to control access. Security is further discussed further in Chap. 9.

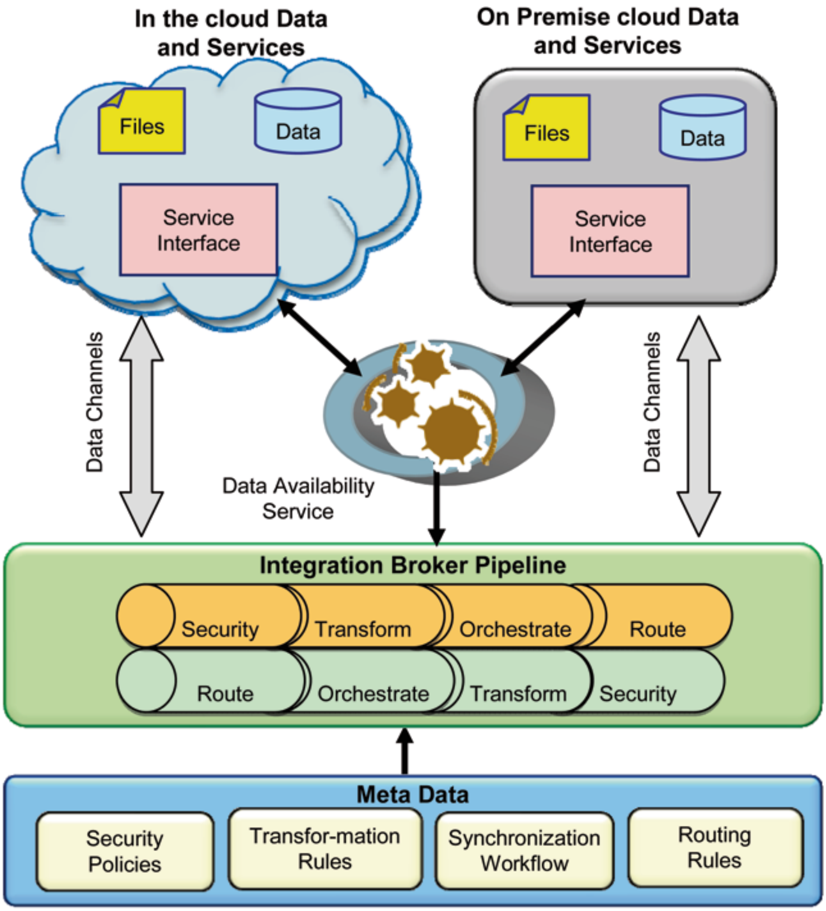


Fig. 5.3 Use of integration broker

- *Validation module*: A validation module compares the data to relevant schemas, and either rejects noncompliant data or hand it off to a transformation component to be converted to the correct format. Exchanging data with a SaaS application usually involves some degree of data transformation. For example, one of the enterprise existing on-premise systems may exchange data using the *Electronic Data Interchange For Administration, Commerce and Transport* (EDIFACT) standard [7], while a SaaS application may use an incompatible XML-based format to send and receive data. In this case, data emanating from the on-premise system must be transformed before it is sent to the SaaS application, and vice versa. Transforming data is a multi-step process. Firstly, the incoming data should be validated against the appropriate data formats and schemas, to ensure that it will be usable after transformation. Optionally, the data can be enhanced by combining it with data from another source. Finally, the data itself is converted to the target format.

- *Synchronization workflow module*: A synchronization module uses workflows and rules to determine how data changes are propagated to destinations, and in what order. In cases where one of these workflow sequences cannot be completed successfully, the synchronization component can use transactional or compensation logic to unwind the data transfer gracefully so as to guarantee data consistency across different systems.
- *Routing module*: Routing modules implement routing rules that define the destination for each piece of data. Routing can simply involve transmitting all data from a specific source to a designated target. It can also involve more complex logic, such as determining a destination from content information, such as a customer ID number.

The data-availability service in Fig. 5.3 provides the means by which the integration broker can detect when new data is available. Synchronizing data involves transferring new and changed data at regular intervals or when precipitated by an event. Three patterns are used to trigger data synchronization between a local source and a SaaS application:

- *Poll*: With polling, one source queries the other for changes, typically at regular intervals.
- *Push*: In a push relationship, the source with the changed data communicates changes to the data sink. A data source can initiate a push every time data in a data source changes, or at regular intervals.
- *Publish and subscribe*: Event-based publication and subscription is a hybrid approach that combines aspects of both polling and pushing. When a change is made to a data source, it publishes a change notification event, to which the data sink can subscribe.

Different patterns are appropriate for different data, and enterprises may decide to use a combination of patterns for a single SaaS application. The appropriate pattern to use for detecting data changes can depend on a number of different factors, including whether data changes must be reflected at or near real time, and how many data sinks must be integrated with the data update. In some cases, enterprises may need to seek a compromise that balances opposing interests. For example, a push pattern is usually best for data that must always be kept up to date. However, pushing data out to a large number of interested sources can be computationally and network intensive and may degrade application performance.

5.2.3.2 Identity Integration

From a user's perspective, whether an application is physically hosted inside or outside the enterprise firewall should not be an issue; applications in multiple locations should be made accessible in a convenient and consistent way. A component of this consistent user experience is *SSO*, i.e., users enter their credentials when signing on to an enterprise, and thereafter can access applications and network resources without having to present their credentials separately to each one. In addition to convenience, SSO means that users have fewer sets of credentials to keep track of and, therefore, may reduce the risk of misplacing a credential.

From an enterprise's perspective, SSO means that IT support staff do not have to manage independent sets of credentials. It also facilitates identity integration in other ways, such as enabling the reuse of existing application-access policies to control access to SaaS applications. For example, a policy may indicate that a certain manager has the power to approve any purchase under a certain price, and an enterprise may want a SaaS application also to recognize that permission. Integrating the enterprise's directory service with a SaaS application means that there is no need to replicate the policy information.

SaaS applications can provide SSO authentication through the use of a federation server within the customer's network that interfaces with the customer's own enterprise user-directory service. This federation server needs to have a trust relationship with a corresponding federation server located within the SaaS provider's network.

Figure 5.4 shows the interconnection between the federation servers. Chapter 9 discusses federated identity architectures in detail. When an end user attempts to access the application, the enterprise federation server authenticates the user locally and negotiates with the SaaS federation server to provide the user with a signed security token, which the SaaS provider's authentication system accepts and uses to grant the user access. Implementing a federation server that uses well-known stan-

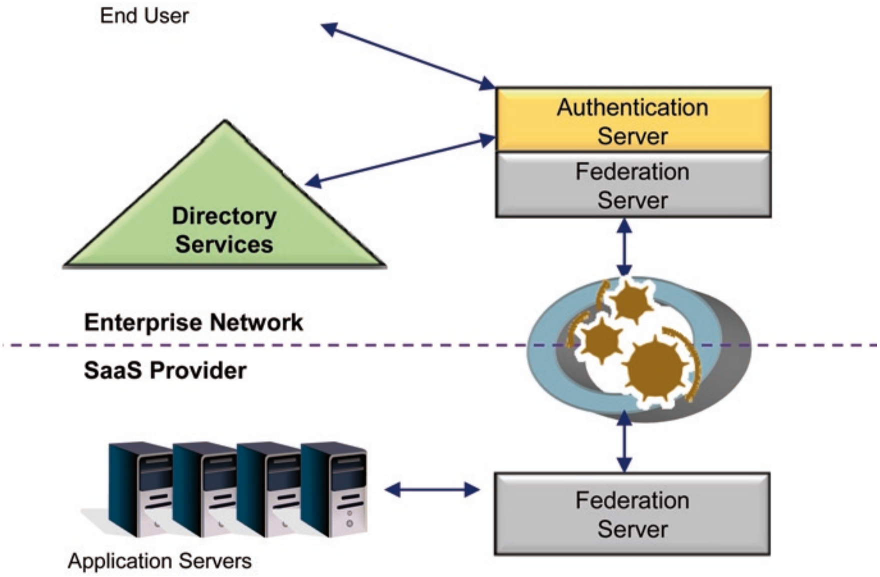


Fig. 5.4 Federated identity use

dards for remote authentication, such as the Liberty Alliance [8] or WS-Federation [9], helps in implementing SSO with a wide range of SaaS providers.

5.2.4 Enterprise Composition Architecture to Access Software as a Service Applications

A *composition architecture* makes composite applications possible. A *composite application* is where business functions and information can be integrated effectively for end users. Many vendors provide API that expose the applications data and functionality to developers for use in creating composite applications. Presenting information as a unified whole, instead of as isolated streams of data, carries benefits for users. It enables them to see relationships between data from different sources and apply their own "domain intelligence", i.e., their own preexisting knowledge of how the business and its processes work, to make informed decisions. The business benefits of a well-designed composite application include reduced redundant data entry, improved human collaboration, heightened awareness of outstanding tasks and their statuses, and improved visibility of interrelated business information. In a service-centric IT department, applications and other resources become ingredients that can be combined together to create task-focused composite applications. Creating a composite application involves integrating different applications, protocols, and technologies that were not necessarily designed to communicate with one another.

Figure 5.5 shows a proposed enterprise composition architecture to access SaaS applications. At the lowest architectural level of the composition architecture are the sources that provide stored or processed data. Sources can include internal applications, internal databases, SaaS applications, Web services, flat files, and numerous other sources.

The composition layer is where the raw data is aggregated and provided to the user in a new, unified form. Its function is to transform data into business information and process intelligence, and vice versa. The composition layer is itself composed of a number of components that manage access, data, workflow, and rules. Applications, databases, Web services, and other resources plug-in to this layer through service agents, which take care of negotiating connections and exchanging messages with each service. The identity-management component ensures that users are properly authenticated and authorized and can also manage credentials for communicating with Web services, which often require credentials that are different from the one the user supplies to access the local network.

The data-aggregation component of the composition layer takes the information from data sources and transforms it in ways defined by the application entity model. For instance, a catalog entity may need different pieces of product and inventory information from different systems. This information is then presented as a unified, correlated set of data to the end user. The workflow component organizes the information with conditions and flows to guide human interaction and collaboration; the

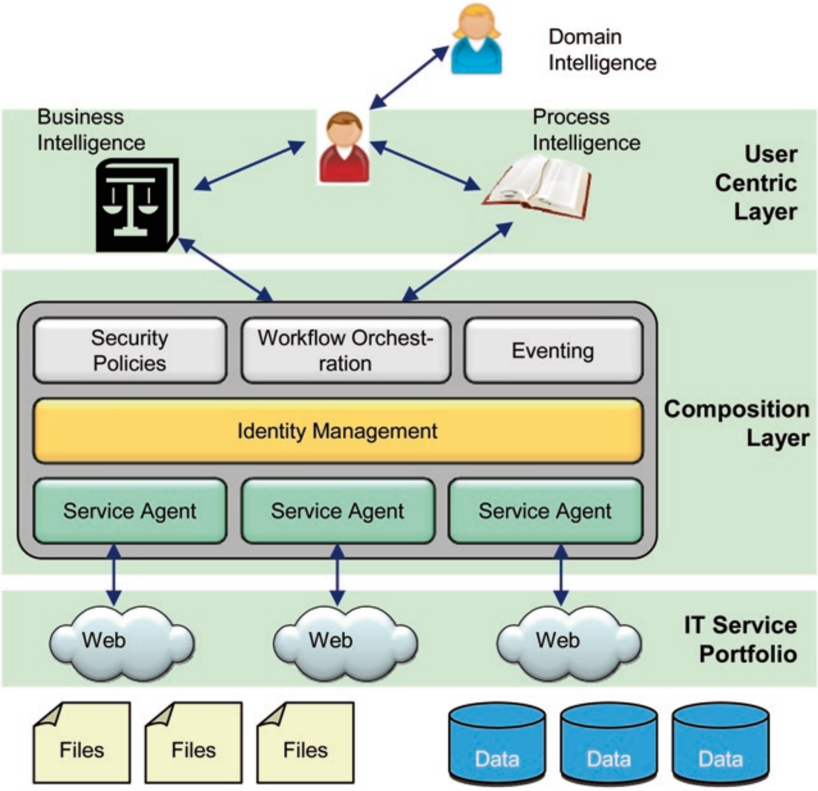


Fig. 5.5 Enterprise composite architecture

"Eventing" mechanism enables notifications to be sent and received when specified conditions are met, so that the end user can react appropriately.

The user-centric layer presents the composite data to the user in a central, integrated, task-focused UI that provides both information for decision-making and functionality for taking action.

5.2.5 *Transformation Reference Architecture for Enterprises*

Figure 5.6 depicts a reference architecture for a typical SaaS offering. The purpose of the reference architecture is to provide a proven template solution that project teams can immediately apply to specific application domains. Accordingly, it includes only a subset of the capabilities described in the conceptual architecture and is more near-term in nature. It includes summary views of data interchange, manageability, and security capabilities. Key aspects of the figure are summarized in the following sections [22].

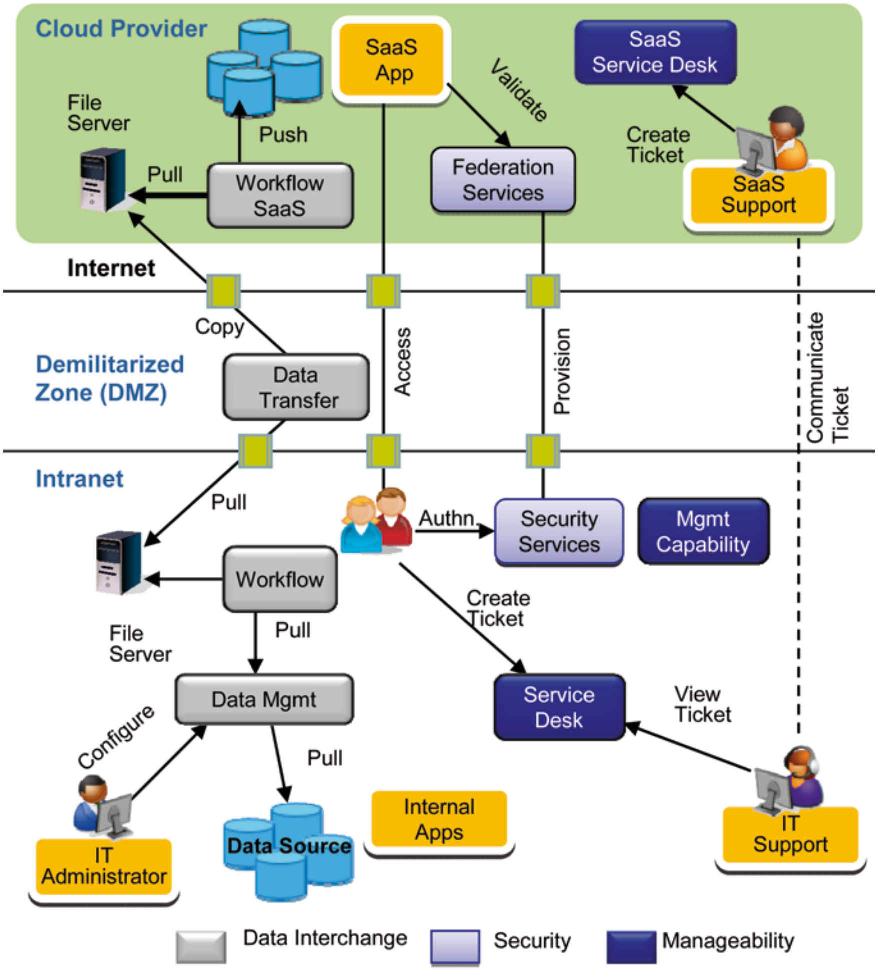


Fig. 5.6 Transformation reference architecture for enterprises

The transformation reference architecture implements the conceptual architecture shown in Fig. 5.7. The conceptual architecture depicts the key capabilities required in a SaaS offering, the logical separation of capabilities into tiers, and the logical grouping of capabilities. Figure 5.7 groups the capabilities that make up the SaaS conceptual architecture into the presentation, security, application, operations, and infrastructure categories.

The presentation category includes capabilities exposed to the user, such as the following:

- *Menu and navigation*: These capabilities provide access to the features and functionality within an application, organized in an intuitive way so that the user can select the desired function.

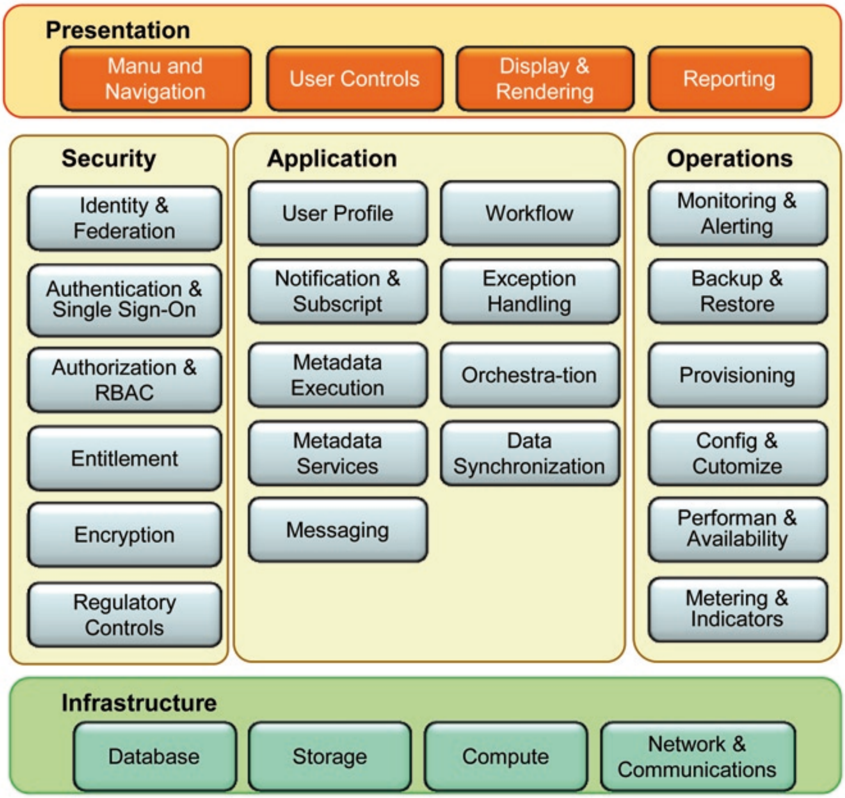


Fig. 5.7 Conceptual architecture for SaaS

- *Reporting*: This capability provides access to application-specific predefined or ad-hoc reports.

The security category includes the following capabilities. These capabilities are discussed in detail in Chap. 9:

- *Identity and federation*: Identity uniquely identifies a user or another entity such as an application or system. Federation describes the function of enabling users in one domain to securely and seamlessly access data within another domain.
- *Authentication and SSO*: This capability includes the process of identifying an individual, usually based on a user name and password. In the context of SaaS, this includes the ability to achieve SSO across multiple Cloud applications and services.
- *Authorization and Role-Based Access Control (RBAC)*: After an identity has been confirmed, authorization is the process of giving individuals access to system objects based on their identities. Identities are usually assigned to roles for ease of managing access.

- *Entitlement*: This capability includes the process of granting access to a specific resource. Tenants are usually responsible for maintaining their own user accounts using delegated administration.
- *Encryption*: Data may need to be encrypted in transit, i.e., between applications or between the layers within an application, and at rest i.e., while stored.
- *Regulatory controls*: This capability includes tracking and reporting who accessed what, when, and why. It includes tracking access to application features and data, the security rating of the data, and the implementation of a data retention policy. It also includes identifying whether individuals are located in controlled countries.

The application category represents a typical business layer or middle tier of a SaaS application and includes the following capabilities:

- *User profile*: This capability includes attributes and information that describe a user, such as name, e-mail address, and role.
- *Metadata execution engine*: This capability includes statements that define or constrain some aspect of the business. They are intended to assert business structure or to control or influence the behavior of the business.
- *Metadata services*: This capability includes information about which data is contained and exposed within an application and about how content is organized.
- *Workflow*: This capability includes a defined series of user-based tasks within a process to produce a final outcome. An example is creating a purchase order.
- *Exception handling*: This capability includes the process of raising and managing exceptions within an application. This includes how application errors are exposed to the user and how error messages are logged.
- *Orchestration*: This capability includes a series of technical tasks performed within a process to produce a final outcome. An example is an extract, transform, and load sequence to move data between business applications.
- *Data synchronization*: This capability includes synchronizing data held within the application with external data.

The operations category represents the capabilities needed to efficiently keep the SaaS application running:

- *Monitoring and alerting*: This capability includes polling application components, services, and infrastructure to detect failures. On detection, an alert is sent to the appropriate support group.
- *Performance and availability*: Performance describes how an application performs under load, both in terms of the number of users and the transaction volume. In the context of SaaS, this should allow applications to dynamically scale based on runtime usage and demand. Availability is a measure of how much of the time the application is available to users and is represented as a percentage.
- *Metering and indicators*: This capability includes tracking and reporting items specifically related to the SLA, such as usage, availability, number of failures, and mean time to respond to and fix problems.

The infrastructure category includes the underlying technical capabilities required for storing data and moving it around the network:

- *Database*: In a multi-tenant data architecture, there may be one database per tenant or one database shared by multiple tenants with the data indexed by a specific tenant identification, as discussed in Sect. 5.2.6 below.
- *Compute*: This capability includes physical clients, servers, or VMs that execute code.

5.2.6 SaaS Data Architecture

Providers of SaaS applications organize data in architectures that enable either multi-tenancy or isolation of software. *Multi-tenancy* is a software architecture in which a single instance of the software runs on a SaaS vendor's servers, thus serving multiple client organizations (tenants). By contrast, complete isolation refers to architectures where separate software instances or hardware systems are set up for different client organizations. There are three data architectures that SaaS application providers can use to vary the degree of isolation between complete isolation and multi-tenancy. They are: (1) *Separate Databases*, (2) *Shared Database, Separate Schemas*, and (3) *Shared Database, Shared Schema* [11].

5.2.6.1 Separate Databases

Storing tenant data in separate database servers provides complete isolation, as depicted in Fig. 5.8 [11].

In this architecture, each tenant gets an individual database computing resource and has a choice of either an individual application container or a shared one. The benefit of this deployment approach is that the data remains physically isolated for each tenant. Giving the tenants their own database server allows each tenant to extend the application's data model to meet their individual needs. Nevertheless, this architecture imposes a relatively high maintenance cost for maintaining data and hardware availability. Thus, this architecture may be suitable for customers who are

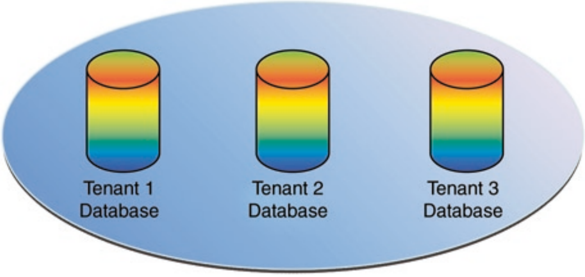


Fig. 5.8 Separate databases SaaS architecture

willing to pay extra for added security and flexibility. For example, customers in an industry such as financial services or content management often have strong data isolation requirements and may use only SaaS applications that provide tenants with their own individual database servers.

5.2.6.2 Shared Database, Separate Schemas

This deployment approach involves creating multi-tenant schemas within one database server, with tenants having access to their own sets of tables that are grouped into individual schemas created specifically for the tenants, as shown in Fig. 5.9 [11].

When a tenant first subscribes to the service, the provisioning subsystem creates a discrete set of table spaces for this new tenant schema and populates it with an appropriate set of default application tables and objects for the tenant. This ensures data separation from other tenants' data. Like the separate databases approach discussed in Sect. 5.2.6.1 above, tenants can easily extend the data model because, once tables are created from a default script, there is no need to conform to the default set, and tenants may add or modify tables as desired. This approach offers a high degree of data isolation, though not to the same degree as a completely isolated system.

This approach enables SaaS providers to back up individual tenants' tablespaces based on their volatility or SLAs. SLAs are discussed further in Sect. 5.5 below. Also, this approach can typically accommodate more tenants per server than the separate database approach can.

5.2.6.3 Shared Database, Shared Schema

This approach involves using one database and one schema to host multi-tenants' data. A given table can include records from multi-tenants stored in any order. A tenant identifier column associates every record with the appropriate tenant. The table is then list partitioned or range partitioned by tenant identifier, thereby creating an isolated set of table spaces per tenant. This approach, then, requires a robust set of partitioning methods that allows for physical data separation of each tenants' data across physical devices while providing simplification of maintenance due to shared table definitions [11].

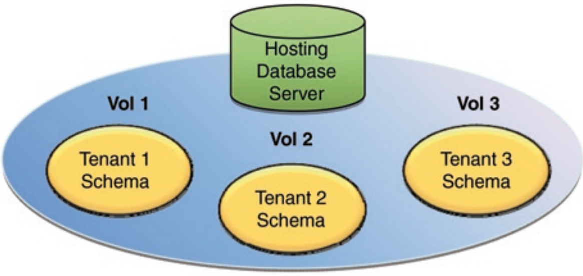


Fig. 5.9 Shared database separate schemas SaaS architecture

This approach requires re-designing the data layer to include a tenant identifier column in various tables. Nevertheless, it provides low hardware and backup costs as it allows SaaS providers to serve a large number of tenants per database server. This approach is appropriate when it is important that the application serves a large number of tenants with a small number of servers.

5.3 Hardware as a Service/Infrastructure as a Service

When enterprises use IaaS, they need to have an understanding of the services that IaaS provides, how to transform enterprises to integrate IaaS provider offerings, and how to use APIs to access the offerings. These topics are discussed in the following sections [12].

5.3.1 IaaS Hierarchy

The fundamental building block of an infrastructure is a *workload*. Workloads can be thought of as the amount of work that a single server or application container can provide given the amount of resources allocated to it. Those resources encompass the amount of processing in CPUs and RAM use, data disk latency and throughput, and networking latency and throughput. Cloud workloads are delivered frequently in virtual servers. Figure 5.10 shows how a single workload (circled in the figure)

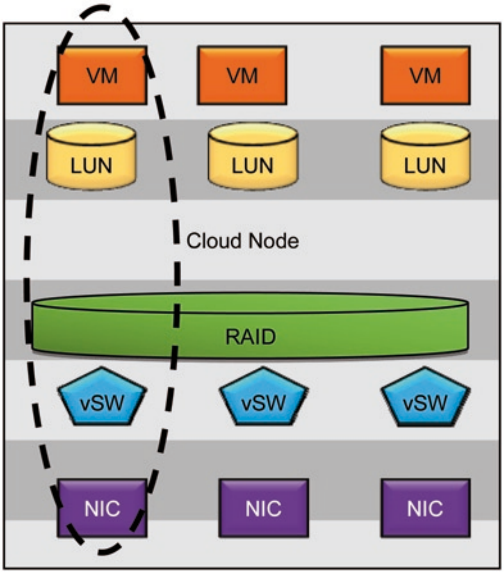


Fig. 5.10 Virtualized workload

might be delivered using a single virtual server spanning a variety of physical resources including compute, storage, and networking. In the figure, a *Logical Unit Number* (LUN) is a logical disk defined within a *Storage Area Network* (SAN). Cloud VMs use LUNs as if the LUNs were physical disks. Multiple VMs run on a Cloud node, which can have access to a *Redundant Array of Independent Disks* (RAID). A Cloud node enables communication among the VMs hosted on the node by using *Virtual Switches* (VSWs), which allow the VMs to use the same protocols that would be used over physical switches, without the need for additional networking hardware. Cloud nodes enable communication among VMs hosted on different Cloud nodes by using *Network Interface Cards* (NICs) [12].

A workload is an application or part of an application. Examples of workloads include transactional databases, fileservers, application Servers, Web servers, and batch data processing. This means that a Web application may have three distinct workload types: *database, application business logic*, and *Web serving*. These three workloads have differing requirements in terms of computation, storage, and networking. A database may require large amounts of CPU and RAM, fast storage, and low latency networking, while an application server may require large amounts of CPU and RAM only.

Since Cloud workloads in general map one-to-one to a physical or virtual server, creating a large-scale Cloud becomes an exercise of putting these workloads together as efficiently as possible. Architectural decisions directly impact this efficiency. For example, some IaaS providers do not provide a separate Ethernet network for each customer. Instead, every server has access to its own Ethernet network. This allows the providers to avoid scaling constraints. In this case, all server-to-server traffic is routed. This approach means that many kinds of network traffic, such as broadcast packets, multi-cast, and shared IP addresses that require Layer 2 networking may not be possible. These providers choose to tradeoff the impact on network usage to gain scalability. In addition, many of the protocols for Layer 2 networking, such as some VLAN tagging protocols, were not designed with the Cloud in mind, so there may be limitations on the number of VMs that can be supported. Although there are ways to modify these protocol limitations, the workarounds tend to be proprietary.

Instead of modifying protocols, a work around protocol limitations is to use *podding*. Podding partitions physical Cloud nodes into PODs. Usually, a POD can run an entire application, and applications do not cross POD boundaries. The limitations of protocols and applications to be supported determine the size of the PODs. PODs do not necessarily have uniform capabilities. In fact, different PODs can be optimized for different workloads, so that overall performance can be optimized. For example, some PODs may be designed for high performance Web applications, while others are designed for low cost, mid-tier performance applications, and yet others are designed for high performance *General-Purpose Computing on Graphics Processing Units* (GPGPU) computing on bare metal. Section 5.3.2 below discusses POD architecture. A *Cloud control system* enables IaaS providers to manage large numbers of PODs, assign customers and applications to the PODs, and group PODs into *availability zones*. Availability zones are distinct locations that are engi-

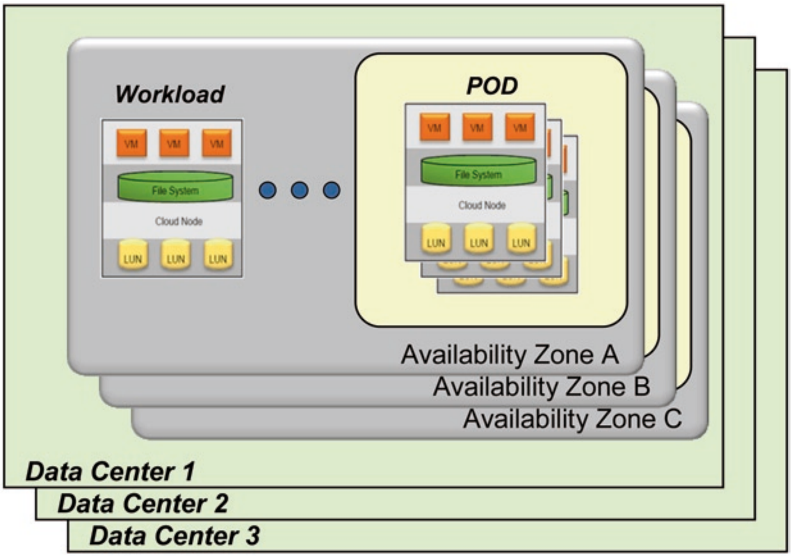


Fig. 5.11 IaaS hierarchy

neered to be insulated from failures in other availability zones and yet have network connectivity to other availability zones. Customers can protect their applications from failure of a single location by running multiple instances of the applications on PODs in different availability zones. In turn, IaaS providers often group one or more availability zones into geographically dispersed *regions* that can span several countries or continents.

Usually, each availability zone resides in a single datacenter facility isolated from other datacenters. Datacenters are then aggregated into a region, and regions form the global IaaS. Figure 5.11 shows the relationship among workloads, PODs, availability zones, and regions.

5.3.2 POD Architecture

PODs can use either *Direct-Attached Storage* (DAS) or SANs for storage needs. PODs that use DAS require every Cloud node to have its own local storage system, as shown in Fig. 5.12. This means that, from a storage perspective, a POD can be quite large as each node added to the POD also adds storage capacity. This also means that, since there is no common storage system across all nodes, some features like live migration become difficult to implement. Likewise, Cloud operations would need to manage a large amount of decentralized and distributed storage [12].

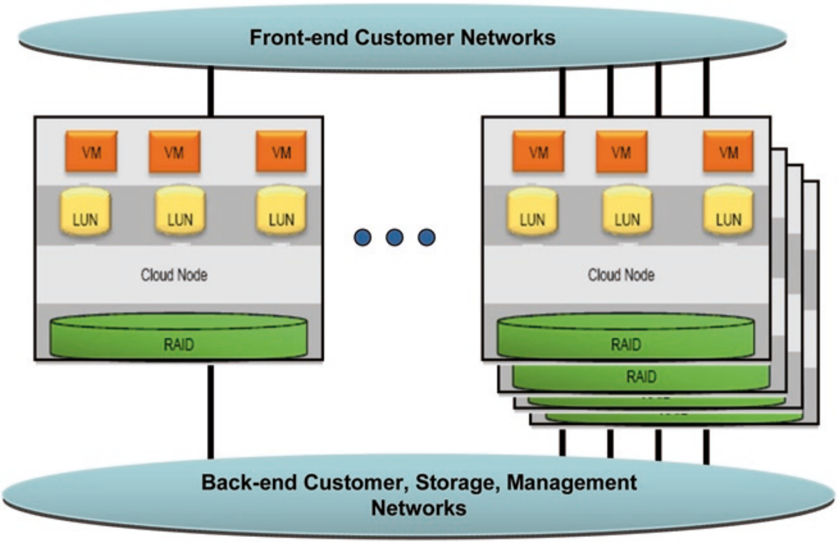

Front-end Customer Networks
VM
VM
VM
LUN
LUN
LUN
Cloud Node
RAID
VM
VM
VM
LUN
LUN
LUN
Cloud Node
RAID
RAID
Back-end Customer, Storage, Management Networks

Fig. 5.12 POD architecture using DAS

On the other hand, PODs that use SAN embrace centralization, as shown in Fig. 5.13. Features like live migration become possible, thus potentially lowering the operational overhead associated with running a large scale Cloud. Nevertheless, this means that PODs with SAN must be smaller than PODs with DAS because SANs typically have some kind of scaling limitations, e.g., large SANs are typically expensive.

5.3.3 Transforming Enterprises to Use IaaS

Some IaaS providers publish APIs that allow enterprise administrators to build their own solutions on top of the IaaS services. Usually, the APIs support a programming style based on the principles of REST or SOAP. Enterprises can use the APIs to perform operations such as browsing, where the enterprises discover the contents of a container that has an application or a virtual media image, and provisioning, where the enterprises can populate a container with entities such as virtual media ISO images [10].

5.3.3.1 Packaging and Distribution of Software

The *Open Virtualization Format* (OVF) is an open, portable, efficient and extensible format for the packaging and distribution of software to be run in VMs [13]. OVF was developed by the DMTF, a not-for-profit association of industry members

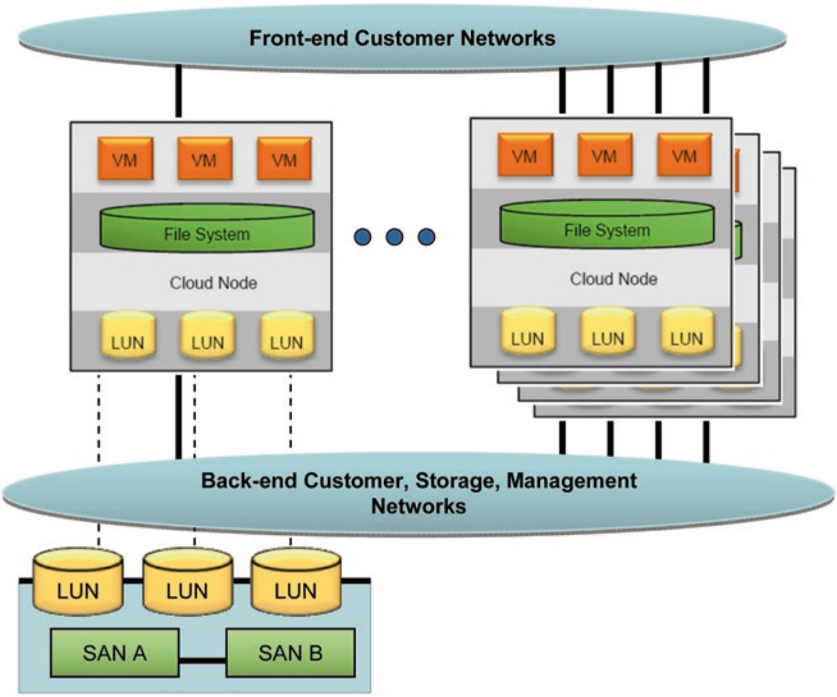


Fig. 5.13 POD architecture using SAN

dedicated to promoting enterprise and systems management and interoperability. A virtual application or VM is typically made up of one or more virtual disk files that contain the OS and applications that run on the VM, and a configuration file containing metadata that describe how the VM is configured and deployed. An OVF package includes these components, as well as optional certificate and manifest files. An OVF package includes four kinds of files [13]:

- *An OVF descriptor*: It is an XML file that contains metadata that describes a VM or collection of related VMs and the deployment environment they require
- *Virtual disk files*: Where the OVF descriptor lists these files and includes information about their format
- *Optional certification file*: It can be used to certify the authenticity of the package
- *Optional manifest file*: It contains cryptographic SHA-1 digest of each of the files in the package (SHA-1 is discussed in Chap. 9)

The package can be distributed and stored as a collection of individual files, or as an archive file in tar format. The IaaS APIs use the OVF package as a unit of distribution and storage for applications and application templates. An application template is a recipe for creating an application. This recipe, contained in an OVF envelope element described below, specifies a set of files, such as virtual disks, that the application requires. It also specifies a set of abstract resources, such as processor cycles, memory, and network connections, that must be allocated to the application in a deployment environment. Because these artifacts are uploaded, downloaded, and stored in OVF package form, the APIs support access to and deployment of a large possible variety of vApps. The APIs implement an instantiation mechanism that transforms an OVF package into deployable applications by binding the package's resource requirements to available resources in a deployment environment. In an application template, the OVF envelope element defines the capabilities and infrastructure requirements of the application, and specifies a list of files, such as virtual disks, that the application requires. In an instantiated application, the sections of the envelope that list the virtual disk files and define other aspects of the application are incorporated into the body of an application element, so the envelope is no longer needed.

Because of its generality, the OVF includes a great deal of information, nearly all of which is reused in application entities. An OVF envelope collects all of the metadata that describes a single VM into a virtual system element. An envelope that contains more than one virtual system collects them into a virtual system collection element. This arrangement supports packaging a group of related VMs as a single entity, and includes provisions for specifying global parameters such as VM startup order, network connections, and a range of resource configurations, such as processing power and memory, to which the VMs can be deployed. The APIs also support this kind of nesting of VMs in application templates and applications. Virtual system and virtual system collection element information is propagated to application and application children elements in the instantiated application. Sections that are children of the OVF envelope become children of an application element. Sections that are children of an OVF virtual system in an envelope that contain a single virtual system become children of an application element, and virtual system sections that are children of a virtual system collection become application elements contained by an application children element.

Virtual disk file information is extracted from the references section of an OVF envelope and used to populate the files element of the application. An OVF package can include exactly one references section. It lists all the files required by the package, including virtual disks and locale-specific resource files. Disk elements can also specify empty virtual disk, in which case they are not associated with a virtual disk file.

The network section element of an OVF envelope lists all the logical networks required by the package. Each network is defined by a name and an optional description. Logical network names are used when specifying connection details for a virtual NIC.

5.3.3.2 Browsing APIs

An enterprise can use HTTP GET requests to browse the contents of container entities. Response bodies returned by these requests include metadata for the container itself and the entities in it, and references to contained entities. References are typically provided as links, which the client can use to get additional information about the entities themselves.

5.3.3.3 Provisioning APIs

Provisioning APIs support a variety of operations that enable two-way transfer of images and templates between an enterprise and IaaS providers. Transfer operations are characterized as *uploads* when the operation transfers content from local hosts to remote ones, and as *downloads* when local host requests transfer content from remote hosts. In either case, the enterprise can be either a client or a server. Uploads are typically initiated by an HTTP POST request. Downloads are typically initiated by an HTTP GET request.

Provisioning APIs provide support for uploading vApps and vApp templates. A vApp is a software solution, packaged in OVF containing one or more VMs. A vApp can be authored by developers at ISVs and VARs or by IT administrators in enterprises. When uploading a *vApp template* or vApp, the workflow includes the following steps:

- The client POSTs an initial request that supplies the body document (vApp or vApp template) of the entity to be uploaded.
- The server uses the POSTed body to create a new entity of the requested type, and responds with an upload map, which is a modified version of the body document that was POSTed in the previous step. The map contains an upload URL for each of the files required by the entity.
- The client makes a series of upload requests, one for each upload URL in the map, supplying the specified file in serialized form.

Before a client can upload a vApp template, a server must allocate storage for it. The server can extract all the information it needs to allocate this storage from the references section of the vApp template's OVF envelope. Information in the server's response enables the client to construct a series of HTTP PUT requests, one for each file in the list, that upload the files referenced by the template. Each request specifies an upload URL, a content-length in bytes, and a SHA-1 hash value that uniquely identifies the file.

To monitor the progress of an upload, a client can use an HTTP GET request specifying the vApp template URL that was returned in the upload map. The response is the same upload map, with updated values for the checksum and bytes.

A client can use an HTTP GET request to get the body of a vApp template. It can then examine the body to discover the URLs of the files that the template requires. These URLs provide the basis for a series of GET requests that download the files themselves. The downloaded template includes a references element that lists each file required by the OVF package.

5.3.3.4 Datacenter Operations APIs

Datacenter operations APIs include the following:

- Instantiating vApps, which is a process that binds a vApp to a specific set of platform resources
- Deployment of instantiated vApps to a virtual datacenter
- Operating vApps by changing their power state (powering on, suspending, or powering off), resetting the virtual hardware, or shutting down a guest OS
- Providing access to the console of a VM
- Reconfiguring of instantiated vApps to modify their instantiation parameters
- Undeploying vApps, which reverses the deployment process and frees the resources being used by the vApp

Instantiation extracts the sections of an OVF envelope that specify the resource requirements of a *vApp template*, places them into a VApp element and creates virtual datacenter-specific bindings that satisfy the resource requirements. These bindings are advisory; they do not guarantee that the resource will be available when the vApp is deployed.

To instantiate a vApp, enterprises need to specify instantiation parameters that map abstract requirements specified in the core metadata sections of the VApp's envelope to concrete resources defined in a target virtual datacenter. There are several ways for a client enterprise to obtain these mappings:

- If the client does not know what resources are available in a specified virtual datacenter or how they could be mapped to the requirements specified by an envelope, the client can request the server to annotate the envelope with information about how its resource requirements can be met with the resources available in a target virtual datacenter. The client can use this information to create an instantiation parameters element, which can be appended to a *vApp body* that it uses in an upload request. When the upload completes, the vApp will be instantiated.
- If the client has enough information about the resources available in a virtual datacenter to map them to a template's requirements, the client can append an instantiation parameters element to a vApp body that it uses in an upload request. When the upload completes, the vApp is instantiated.

Instantiation parameters provide explicit mappings of an abstract requirement specified in an OVF core metadata section to a concrete resource available in a virtual datacenter. An enterprise client can reconfigure an instantiated vApp by making HTTP PUT requests for special configuration URLs that the server inserts in an instantiated vApp body. These URLs are references to specific core metadata elements, such as the network section and virtual hardware section. Clients can use these URLs to add, delete, or edit an element of the vApp body document. Each of these URLs has a type attribute that specifies the kind of element on which it operates, and other attributes that specify the kind of operation, such as add, edit, and remove, it performs.

5.4 Platform as a Service

PaaS is positioned between SaaS and IaaS. PaaS generally refers to internet-based software delivery platforms for which third-party ISVs or custom application developers can create multi-tenant, Web-based applications that are hosted on the PaaS provider's infrastructure and offered as a service to customers [14].

The main premise of PaaS is providing software developers and vendors with an integrated environment for development, hosting, delivery, collaboration, and support for their on-demand software applications [14]. Like other software platforms, PaaS aims to be a foundation for a broad, interdependent ecosystem of users and businesses. It can support tasks from code editing to deployment, runtime, and management. The current PaaS ecosystem shows a wide range of different levels of service and is described briefly in Sect. 5.4.2. Some platforms offer little more than a set of APIs on top of an elastic infrastructure, while others offer fully functional Web-based IDEs or fourth-generation programming language environments [15] allowing an easy creation of metadata-level mash-ups. Additionally, a PaaS could support built-in backend functionalities of applications like billing, metering, advertising, etc.

5.4.1 Implications of PaaS on Transforming Enterprises

In this section, we will discuss the potential implications of PaaS on the evolution of software development and delivery. Considering the global nature of the network-based PaaS paradigms, the section specifically looks into the concepts of distributed work and collaboration within a global software development and delivery framework [14].

5.4.1.1 Software Development

Software development involves designing, developing, testing, supporting, and implementing applications. For enterprises that spread software development among globally distributed teams, these tasks maybe executed simultaneously from multiple locations. Therefore, in order to provide a sustained quality of final code, enterprises have to rely heavily on industry-specific standards, or else projects that have only partial environments replicated offshore may have significant integration problems once the code is brought back onsite. In using PaaS, enterprises would need to continue to rely on standard development tools and languages in order to reduce training time and provide integration of development environments among distributed teams, thus reducing the costs of development. Section 5.4.2.1 discusses the topic of standard tools and languages.

5.4.1.2 Service Delivery

Service delivery in an enterprise involves core IT service management processes that have a tactical or strategic focus, namely, SLM, capacity management, *IT Service Continuity Management* (ITCM), availability management, financial management for IT services, and delivery of IT services to customers. Some of these functions can be transformed into a Cloud environment by using PaaS.

Capacity management is responsible for ensuring that adequate capacity is available at all times to meet the requirements of the business. PaaS in a Cloud environment is particularly suited for distributed development because programmers can use a shared, high-capacity platform that is easy to provision to additional developers to code and test software. The platform also enables easy expansion of work groups when necessary.

ITCM refers to the process that ensures that required IT technical facilities, such as computer systems, networks, applications, and telecommunications infrastructure, can be recovered within required and agreed business timelines. Enterprises can provide a high level of continuity by using PaaS in a Cloud infrastructure that employs architectures similar to the architectures described in Sect. 5.3.1 above. Continuity management by using PaaS, in turn, translates into *availability management*, which is a term that represents similar aspects to continuity management but from the perspective of a client of the enterprise. This relationship between continuity management and availability management when using PaaS can be seen as one of the major benefits of PaaS.

Financial management refers to IT accounting, charging, and budgeting. Enterprises can include some support of this delivery discipline through the billing mechanisms based on the use of the underlying PaaS.

5.4.1.3 Collaboration

Inadequate collaboration can pose serious challenges to a distributed project in terms of unexpected rework, mismatched processes, and poor project synchronization and team dynamics. By using PaaS, enterprises can provide tools that make seamless real-time interaction between teams possible. These tools include shared source code development and IDE integration, Web-based dashboards, project management tools, discussion threads, and automatic tracking systems and can even be integrated with popular social networks.

5.4.2 Example PaaS Techniques

While SaaS gained some market traction in current business scenarios, PaaS is still in the early stages of development. A number of companies have developed their own PaaS offerings, each with a slightly different approach. In this section, we will discuss some of the differences in these approaches [14].

5.4.2.1 Software Development

Some PaaS providers introduce their own version of a fourth-generation language at the metadata level that would simplify the creation of new applications even by inexperienced programmers. The same providers also support, with different levels of complexity, Web-based IDEs for these languages. Other PaaS providers rely on standard programming languages, such as Java, Python, .NET, PHP, and Ruby. For these providers, development is not Web-based but mostly done with the help of downloadable *Software Development Kits* (SDKs) for standard development platforms. Normally, designers and testers then run the applications on a custom runtime environment on a local host that simulates the platform.

5.4.2.2 Collaboration

Not all PaaS providers offer collaboration tools as a core component of their architecture. Some PaaS providers offer solid support for collaborative development based on their Web-based IDE. Other providers support collaboration by acting as live repositories for development projects that can be universally accessible by authorized development team members. Yet other providers do not offer collaboration as a functionality of the platforms themselves but through auxiliary tools.

5.4.3 Public Cloud vs. Private Cloud

Besides the split into SaaS, PaaS, and IaaS, Cloud Computing has divided along another dimension. Though initial uses of Cloud environments were to access software over the public Internet or Web, enterprises can setup environments internally to have the essential Cloud environment characteristics of network-based self-service deployment and elastic capacity [2]. Such “on-premise” or “internal” Clouds have come to be referred to as “Private Cloud.”

Because integration flexibility and control over QoS and security are a high priority for many enterprises, and because such enterprises likely have the financial resources to optimize for costs over time rather than up-front costs, the enterprises may gravitate towards the private variant in their adoption of Cloud Computing. An added attractive feature of Private Cloud’s is that adopting Private Cloud practices is likely to be a relatively small change for many enterprises. IT departments in many cases have already gone down the path of consolidating infrastructure and setting up shared services, and enabling a Cloud’s self service and automated dynamic capacity is often a relatively small incremental step. This is in contrast to the adoption of a Public Cloud offering, which can dramatically change how departmental users obtain application support.

In setting up a Private Cloud, a natural organizational structure comprises a central IT function that sets up and manages the Cloud itself and various functional or product departments across the enterprise that are "customers" of the Cloud. For a Private Cloud, the most appropriate type of computing is PaaS. If the central IT functions were to set up an offering at the IaaS level, the departmental users would need too much IT expertise themselves to make use of the Cloud, thus defeating the economical purpose of centralizing the IT function in the first place. At the other end of the spectrum, an internal SaaS offering would not likely make sense in many cases because departments would not have the flexibility to create the specific functionality they require—there are very few applications that would fulfill a majority of functional needs across multiple departments. The platform level of PaaS is the right balance between flexibility and ease of use for the departmental Cloud customers.

Depending on industry or domain characteristics as well as company-specific business strategies, different enterprises have different balances between what is in a shared Cloud platform and what individual departments need to create. For example, in a consumer products company organized such that different departments represent different products, the Cloud-based platform may have functionality such as a consumer-facing portal, a catalog, order processing, and customer service, and each product department would have only minimal customization beyond the basics provided by the platform. In another example, a telecommunications company's Private Cloud platform may provide basic customer record functionality, with each department creating deeply specialized applications. In the former example, most of the IT expertise resides in the central IT function with very little expertise required of the departments; in the latter example, each department likely has programmers. Each enterprise has a unique set of capabilities provided in the central Cloud platform; the goal is to centralize as much as possible while allowing the departments the flexibility they need for their roles in making the overall business competitive.

5.4.3.1 Reference Architecture for PaaS Private Cloud

Figure 5.14 shows the basic architecture of a PaaS Cloud offering that a central IT function can set up within an enterprise. The physical infrastructure includes servers, legacy systems such as mainframes, integrations, and database resources. The lowest layer of software above this is at the OS level and may include virtualization technologies such as hypervisors. Above this resides middleware, including application servers and technologies such as SOA, BPM, UI technologies, and identity management. Systems management spans the entire stack [2].

Upon this foundation, the central IT function builds custom elements, including shared components such as SOA services and BPM processes as well as the self-service interface that the enterprise's internal Cloud customers interact with [2].

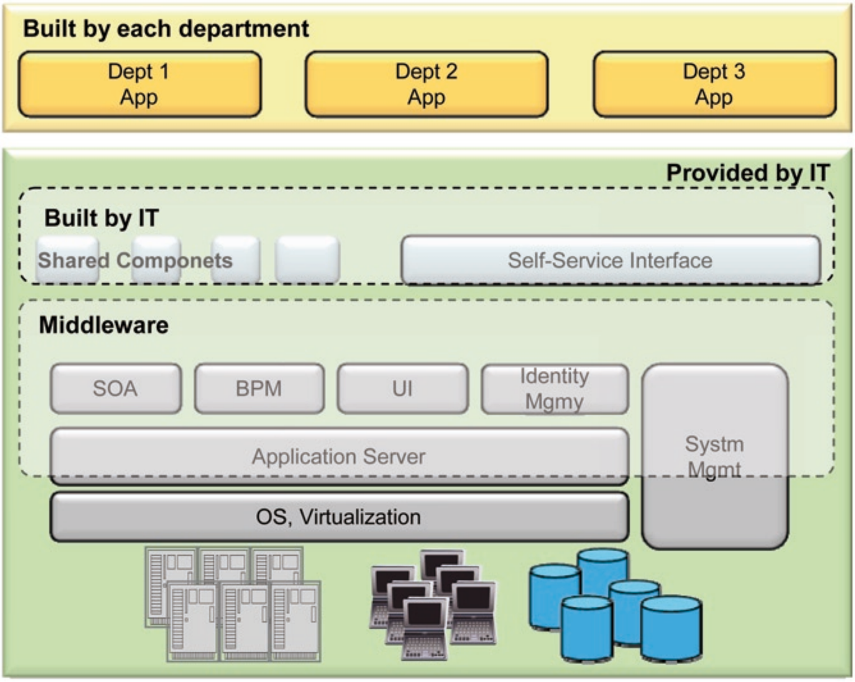


Fig. 5.14 Basic Platform-as-a-service Cloud architecture

5.4.3.2 PaaS Private Cloud Life-Cycle

Getting underway with a PaaS Private Cloud involves four macro-level steps. These are shown in Fig. 5.15. First, the central IT function builds the platform, starting with out-of-the-box middleware to create the enterprise-specific shared components and self-service interface. Once the basic platform is up and running, the application owners within the enterprise's departments can set up their respective applications. Depending on the nature of the domain and enterprise, this may involve fairly simple application composition using platform components, or it may involve a substantial amount of custom application development [2].

Once an application has been deployed on the platform, the third step is simply the use of the application. From the users' perspective, the application is no different from any other network/Web-based application they would use within the enterprise—there is nothing special about the fact that it is running on a Cloud platform as far as they are concerned.

The central IT function carries out ongoing administration of the platform as well as the applications. Depending on the nature of the applications, the application owners in the fourth step may carry out some amount of administration, such as adding and removing users or other high-level functions specific to the application.

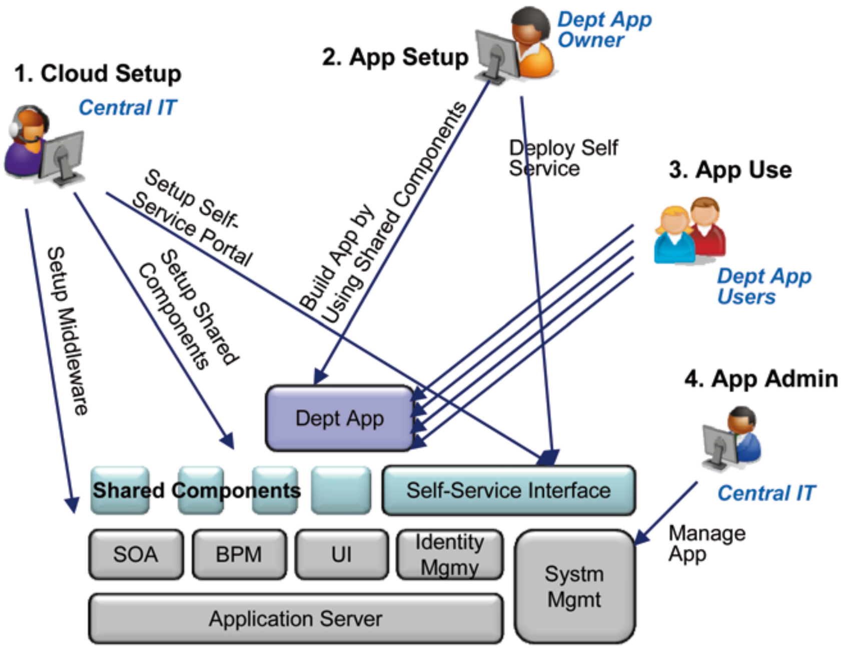


Fig. 5.15 PaaS private Cloud life-cycle

Central IT is concerned with lower-level issues such as whether the application is resourced appropriately, if it is meeting its SLAs, etc. One of the goals in setting up shared infrastructure in general and Private Clouds in particular is to exploit as many economies of scale and opportunities for efficiency as possible. Among these is the opportunity to automate dynamic resource allocation and optimization, enabling the elastic capacity that characterizes Cloud. This also enables the continuously high responsiveness demanded by users irrespective of load and minimizes manual intervention.

5.4.3.3 SOA, BPM, and UI

A Private Cloud consists of shared components. Hence, a starting point to build Private Clouds is to create a SOA that uses modular application components that are accessible through standardized interfaces such as XML or SOAP. In addition to SOA components, enterprises may want to include business process components managed within a unified BPM framework as part of their PaaS [2].

Like SOA and BPM components, UI components can be included in an enterprise's PaaS. A centrally-managed library of UI components can give department application owners a head start in composing their solutions and also gives the central IT function a consistent level of control over the enterprise's UIs. At the same time, a UI framework can give the departments the flexibility to accommodate their specific functionality, customization, and personalization needs for applications and portal solutions.

UI technologies play an additional role in a PaaS environment as the basis of the self-service interface for the Cloud. In many cases, this can be an extensive portal that works closely with an identity management system to authenticate users, filter their access based on roles, and present the platform's shared components for application development and composition.

5.4.3.4 Identity Management and Systems Management

Security is a high-priority concern for many enterprises in creating a Private Cloud, particularly for firms in domains with a high level of regulation or sensitive customer data. Enterprises, therefore, require balancing rich mechanisms for identity and access management with convenience features such as SSO [2].

Implementing PaaS with a high degree of self service in a security-critical environment requires an approach where security pervades the entire architecture in a well-integrated manner rather than being bolted on as an afterthought. Chapter 9 discusses security in detail.

Like security, systems management is a characteristic that depends partly on the functionality manifested in a particular software utility and partly on capabilities infused throughout the other technologies in the platform. Systems management needs to provide an insightful set of visualizations that enable system administrators to monitor performance, diagnose problems, and make adjustments. In addition, the systems manager needs to sense when inputs cross certain user-specified thresholds and automatically take appropriate actions, such as adding capacity to applications that see responsiveness compromised by load spikes. Such automation is required both for the elastic capacity and self-service provisioning aspects of the Private Cloud. Another direction many enterprises take with the Private Cloud is departmental "chargeback," which is an economic regime where departments are charged by the central IT function based on their usage. In this case, the system manager collects and logs the kinds of information, e.g., items such as times and numbers of users logged in to particular applications and amounts of data transferred, that an IT department would use as the basis for chargeback. By processing log files and generating notifications, an effective internal billing system can then be created.

5.5 Service Definition and Instance Management

Most SOA governance environments only skim the surface of enterprise IT environments: managing only the subset of services operating in the application layer, and only those Web services built on XML, SOAP, WSDL and other core SOA specifications [16]. By contrast, many Public Cloud services provide a deep stack of on-demand services, spanning the application, software platform, integration middleware, and hardware layers. By proliferating services deep into the stack, beyond the capabilities of today's SOA governance tools, Cloud environments make unified planning, design, provisioning, monitoring and control of all services difficult. One key area where Cloud governance differs from traditional SOA is in its focus on life-cycle governance of VMs. To facilitate automated provisioning of deep application and integration stacks on VMs, Cloud management environments can offer prepackaged *server templates*. These templates embed prepackaged policy definitions that govern important life-cycle service VM governance functions, including deployment, setup, booting, monitoring, control, optimization and scaling of VMs on one or more public or Private Clouds.

Cloud governance encompasses the periodic need to decommission and throw away old VM instances, and launch new ones in their place [16]. The problem of unchecked proliferation of VM instances across public and private virtualization infrastructures is sometimes known as VM sprawl. A growing range of commercial management tools provide the ability to control VM sprawl across disparate hypervisors. Preventing VM sprawl is referred to as *instance management*, and it is a feature that is lacking from traditional SOA governance tools.

Traditional SOA-style development is top-down. It requires upfront architectural design that factors functional primitives into platform-independent, loosely coupled service contracts that are exposed to developers through open Web services standards. It often also includes a core service catalog, such as *Universal Description Discovery and Integration* (UDDI) [17] to broker abstract service contracts, as well as tools and platforms that support key interface standards such as *Web Service Definition Language* (WSDL) and *Simple Object Access Protocol* (SOAP). By contrast, Cloud services encourage a style that uses Web-oriented architecture or REST for service provisioning, development and management. Thus, anyone with a browser can mash up available Cloud service components into applications that may deviate significantly from corporate-standard design patterns and may also lack the stringent security expected from enterprise-grade services. Therefore, transforming enterprises to use Cloud services requires enterprises to export enterprise SOA governance practices into similar practices in Cloud environments. For example, the transformation requires a service catalog that maintains metadata about services and enables enterprises to control development and construction of services and publish visibility and availability of services to consumers. Also, the transformation requires federation agreements that set up auto-provision service definitions between Public Clouds and enterprises' SOA, REST and other application environments, as discussed in Chap. 9 [18].

5.5.1 Virtualization and Cloud Infrastructure

The mass scale adoption of server virtualization in datacenters and public/Private Cloud environments creates the need for high-speed, low latency and resilient Cloud networking. Building a combination of virtual and physical Cloud network that is commensurate with virtual and physical servers demands an architectural approach to infrastructure build-out. The performance, latency and elasticity must be considered as well as the management of the networking infrastructure. Once architected and deployed, the solution can offer services over a common shared infrastructure [18].

Virtualization technologies encapsulate existing applications and isolate them from the physical hardware. Unlike physical machines, VMs are represented by a portable software image, which can be instantiated on physical hardware at a moment's notice. With virtualization comes elasticity where compute capacity can be scaled up or down on demand. Additionally, applications that run on VMs can be migrated while in service from one physical server to another. Extending this further, virtualization can abstract where an application workload runs. As Clouds begin to interact programmatically with other Clouds, a single Cloud can consist of geographically distributed datacenters that are virtualized. Due to virtualization and multi-tenancy, Cloud environments allocate, replicate and migrate resources while keeping the workloads and data logically isolated. This enables economies of scale to reuse resources during idle hours [18].

Portability, elasticity, mobility and density of VMs and application workloads demand a high performance network that offers low latency and resiliency. Additionally, consistent network-driven policy and controls are necessary for visibility to VMs' state or location as they are instantiated, torn down, or roam across virtualized Cloud environments.

As was mentioned above, a direct consequence of the virtual server deployments is VM sprawl. Simultaneously, there is a proportional sprawl of VSWs to which a set of VMs connect within a physical server. Since more than one VSW can be instantiated for every physical server, there is a one- or two-fold increase in the number of switches to be managed compared to physical top-of-rack switches, as shown in Fig. 5.10 in Sect. 5.3.1 above. Essentially, the network access layer formed by VSWs moves inside the server.

The massive VM and VSW infrastructure, as well as Cloud applications that run on the virtualized infrastructure, place new demands on the underlying Cloud network fabric for seamless user-to-VM, VM-to-VM and VM-to-data store communications. Specifically, since many VMs can be instantiated on one physical server, the utilization of the physical NIC bandwidth increases proportionally. This NIC link is no longer heavily undersubscribed, which implies that traditional oversubscribed network topologies need to be re-architected for Clouds. Portable VMIs are of several gigabytes in size; hence large amounts of data are moved over the network to spin VMs up or down. Also, workload elasticity implies that applications are scaled up or down programmatically, based on various conditions such as load, time of day and power/cooling availability. Thus a Cloud network still needs to be designed with peak bandwidth in mind.

Cloud applications can liberally integrate rich media technologies, often through mash-ups, which can be accessed by millions of users dispersed over widely separated geographical areas. This leads to a large number of flows and transactions that traverse the network with high amounts of VM to user traffic. Cloud application workloads are architected to distribute computing tasks across multiple layers of worker and data nodes, thus requiring large amounts of VM-to-VM interactions.

Workloads discussed in Sect. 5.3.1 above can be moved while in service off of low utilization servers so they can be shut off to save power, or perhaps opportunistically to use low-cost compute enclaves. This migration requires Cloud networks to have large Layer-2 domains.

Server administrators, and not network administrators, typically manage virtual networks, because network administrators do not have direct access to built-in VSWs. For Cloud providers, this creates a challenge as consistent network-wide policies, monitoring and diagnostics need to be applied to a large number of VSWs across a multi-vendor hypervisor environment.

5.5.2 Virtualization-Optimized Cloud Infrastructure

In building a combination of virtual and physical Cloud networks to cope with virtual and physical servers, performance, latency and resiliency must be considered as well as policy control and management. Characteristics of virtualized Clouds listed in Sect. 5.5.1 require the use of the following Cloud networking architectures [18]:

- *High-bandwidth network switches*: As VMs pump large amounts of bits into the network through NIC links, and as gigabytes of VM images move across the network fabric due to elastic workloads, high-bandwidth network switches become needed to build high performance and highly responsive networks capable of handling peak bandwidth demands of Cloud workloads.
- *Symmetric cross-sectional bandwidth*: Highly subscribed NIC links in Cloud infrastructures and symmetry in user-to-VM and VM-to-VM traffic require that ingress and egress switching bandwidths be highly balanced, having an ingress-to-egress bandwidth ratio of 1:1 or 2:1.
- *Leaf-Spine Architecture*: Constant inter-VM communication and VM mobility demand large Layer-2 domains, thus necessitating two-tier leaf-spine architecture over traditional three-tier designs.
- *Low-latency Switching*: Improving application response time requires reducing latency and provisioning proper bandwidth in network switches. This may require switches that use a cut-through packet processing mode rather than a store-and-forward mode.
- *Resilient Networking*: To ensure that workloads are not impacted when physical nodes or VMs fail, switch software needs to be architected with a fault-tolerant extensible OS and network stack.
- *Virtualization*: For seamless consistency between embedded VSWs and external physical switches, the embedded VSWs use transparent redirection, for example via standards based mechanisms, including VLAN tags, MAC addresses and inter-VSW tunnels, which are transparent to the physical switches. Proprietary tags need to be avoided.

- *Network Management*: Consistent network management across both physical and virtual networks requires that heterogeneous VSWs be managed by administrators that use well-understood mechanisms, such as the *Command Line Interface* (CLI), in order to maintain configuration and management consistency across virtual and physical networks as well as during VM migration.

5.6 Service Level and Quality Management

A SLA serves as the foundation for the expected level of service between a consumer or an enterprise and a Cloud services provider [19]. QoS attributes, such as response time and throughput, usually form a part of an SLA. Since the QoS attributes change frequently over time and are based on traffic conditions, enterprises need to monitor these attributes [20]. To monitor the QoS attributes, enterprises can demand that monitoring data, such as raw transaction count, be exposed by a SP without further refinement. Alternatively, enterprises can request that collected monitoring data be put into a meaningful context, such as statistical measures of average or standard deviation. This request requires that the Cloud SP create processes to collect data from several different sources and apply suitable algorithms for calculating meaningful results. A second alternative is for enterprises to request certain customized data to be collected. Yet another alternative is for enterprises to dictate the way monitoring data is collected.

Web Service Level Agreement (WSLA) is a language and framework that is designed to capture SLAs in a formal way [21]. The WSLA language is designed to capture SLAs in a formal way to enable automatic configuration of both the service implementing system of providers as well as the system that is used to supervise the agreed QoS. In particular, WSLA specifies the following:

- A description of the parties, their roles (provider, consumer, third parties) and the action interfaces they expose to the other parties of an SLA contract. The tasks of third parties vary from measuring service parameters to taking actions on violations as delegated by either the SP or SC. In order to protect the confidentiality of consumers, an SLA must be decomposable into the configuration information that is needed for third parties to perform their role in the SLA supervising without having access to the complete SLA.
- A detailed specification of the SLA parameters, which are specified by metrics. Metric descriptions also include which party is in charge of measuring and aggregating and how the metrics can be retrieved.
- A representation of the parties' obligations. This representation includes *Service Level Objectives* (SLOs) that contain a formal expression of the guaranteed condition of a service in a given period, and includes action guarantees that represent promises of parties to do something, for example, to send a notification in case the guarantees are not met.

5.6.1 Specification of Service and Quality Levels

A WSLA agreement complements service descriptions. While a service description, for example, defines the service interface relationship between a service and its using application, the WSLA defines the agreed performance characteristics and the way to evaluate and measure them. Thus, whereas service descriptions are input to the design and implementation of the service system and the client application using its service, WSLA provides input to the measurement and management system of an organization to check and manage compliance with a WSLA [21].

Figure 5.16 shows the role that WSLA plays between SPs and SCs. Both the SP and SC may run their own instrumentation, measurement and management systems. Each organization may access measured metrics from various sources, such as server-side metrics from the provider and client-side metrics from the consumer. This allows parties to determine both a service's performance within a SP's domain and its performance as experienced by a user [21].

Figure 5.17 shows a model of the runtime management of a WSLA. The model assumes that the measurement and management functionality is divided in three groups of functionality [21]:

- The *measurement* functionality receives the measured metrics from the system's instrumentation. Instructions on how to measure a particular system parameter are defined in the measurement directives of a WSLA. The role of the measurement functionality is also to compute high-level metrics, e.g., the average re-

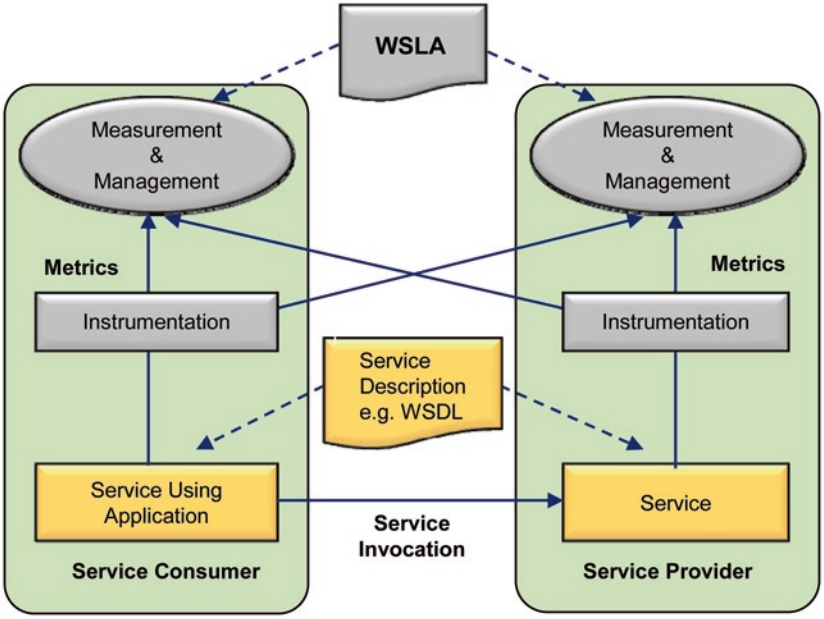


Fig. 5.16 Role of WSLA

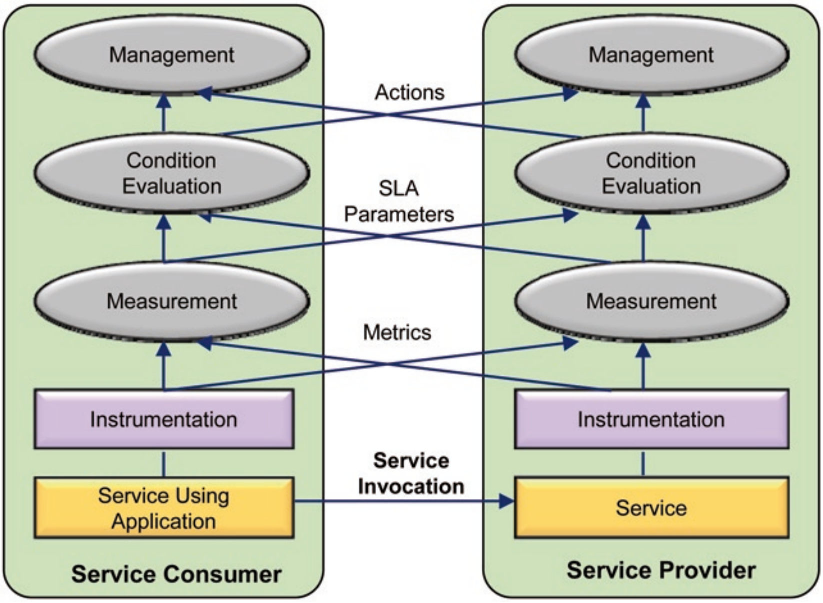


Fig. 5.17 WSLA runtime management

sponse time of a complete cluster of servers in a particular period, as defined in the metrics definitions of a WSLA. The measurement functionality must implement the functions that are required to compute the high-level metrics. The set of metrics that are used in the guarantees of the WSLA are made available by the measurement function as SLA Parameters.

- The *condition evaluation* function evaluates the guarantees of the WSLA as defined in the WSLA. Guarantees are defined as predicates over SLA Parameters. The value of these parameters can be obtained from the measurement function. The condition evaluation function must implement the relevant predicates to perform the guarantee evaluation. In the case of a guarantee violation, an action is invoked on the management function.
- The *management* function implements actions that are invoked upon guarantee violations.

Figure 5.17 illustrates that interactions between parties may occur at various function levels, if agreed upon in the WSLA. Measured and high-level metrics may be exchanged by the measurement function, the condition evaluation function may retrieve SLA Parameters from various sources, and management actions may be triggered from both the provider and the consumer.

As mentioned before, a provider or a consumer may choose to commission a part of the WSLA management activity to third parties, which are sometimes referred to as supporting parties. Third parties can implement the measurement, condition evaluation, or management functions.

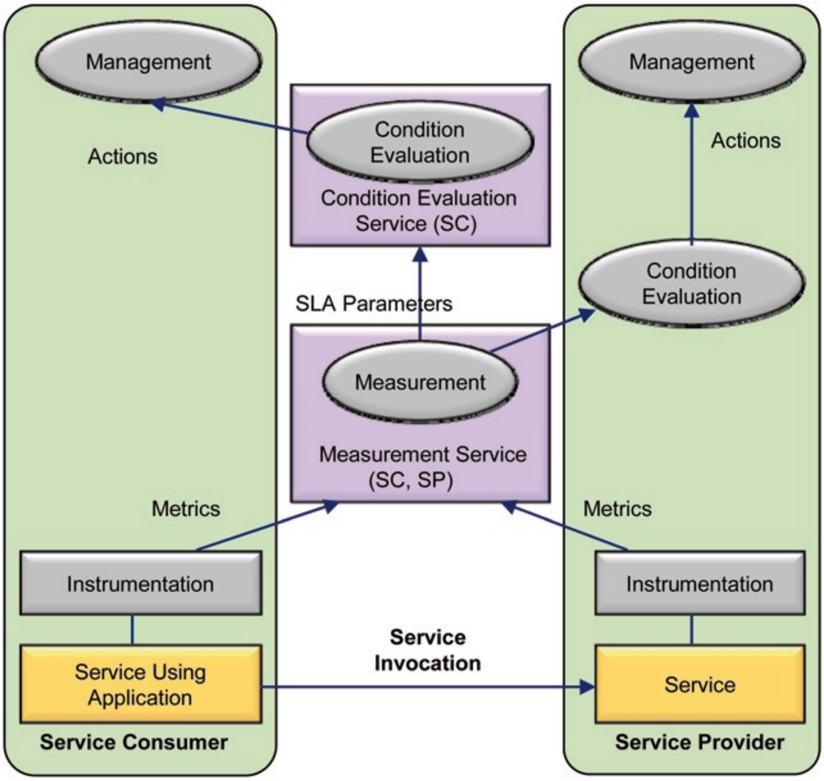


Fig. 5.18 Third parties in a WSLA

Figure 5.18 depicts two supporting parties: a measurement service that implements the complete measurement function for the SP and SC, and a condition evaluation service for the SC. In the figure, the SP implements the condition evaluation functionality itself. The SP and SC implement their own management functions [21].

A complete WSLA document is composed from all the information negotiated and agreed upon by the two parties, SP and SC. Information on supporting parties, e.g., roles, and details of their actions are specified by their sponsors. Since the sponsored parties do not participate in creating a WSLA, full details of a composed WSLA, other than information related to their roles, are not visible to the sponsored parties. During deployment of a WSLA, however, the appropriate information is passed to various supporting parties by their sponsors.

In many scenarios, one of the parties, e.g., a SP, defines most of the content of a WSLA, and a SC may simply agree to such information and provide additional client-specific information, e.g., party information. In another scenario, while the SP defines many of the aspects of a service, including definition of specific SLA metrics and measurement directives, the SP may offer a choice to consumers on the details of the guarantees, e.g., violation thresholds and actions to be invoked upon violation.

The authoring process can be off-line, where the information is exchanged between the parties via e-mail or other human communication mechanisms. Alternatively, the WSLA creation can be negotiated in an online process. A template can be published in a registry such as UDDI [17]. After a sequence of information exchange via negotiation steps, a WSLA document is created.

The interpretation of the WSLA and the corresponding setup of the components required to supervise the WSLA is called the *deployment process* and is depicted in Fig. 5.19. Each signatory party is responsible for the deployment of its function and the setup of the supporting parties that it sponsors. The information passed on to various functions may not be the WSLA but derived setup information in a proprietary format. Nevertheless, in some cases, such as when using a condition evaluation service, passing on information in standard WSLA format may be appropriate [21].

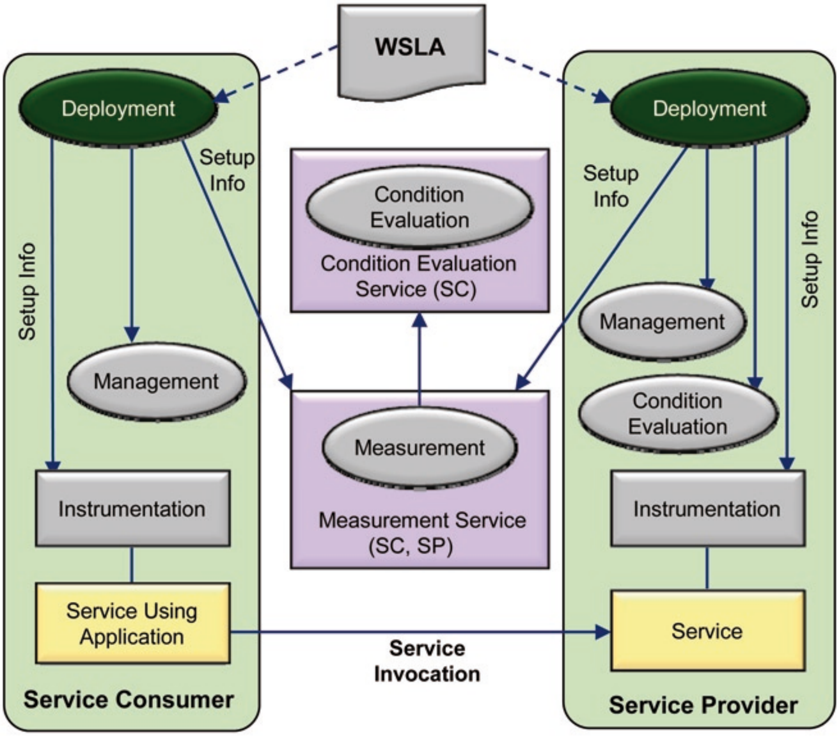


Fig. 5.19 WSLA deployment process

5.6.2 Cloud Service Level and Quality Management Architecture

In transforming enterprises to use Cloud services, the following considerations affect the direct use of WSLA:

- Any system that enforces SLAs needs to take into account that Cloud resource usage changes dynamically. Hence, all measuring tasks in a Cloud context need to be performed via the WSLA-defined functions.
- To alleviate consumer concerns regarding privacy and data security, privacy-sensitive and security-sensitive tasks can be delegated to trusted third parties.
- Cloud services are subject to load fluctuations, and provider SLA violations are likely to happen during these transitions. The nature of these fluctuations is unpredictable, hence, a static schedule for evaluating conditions may not suffice. THerefore, SLAs in the Cloud context may need to use dynamic schedules for condition evaluations.

5.7 Conclusion

Cloud services can be divided into three categories: SaaS, PaaS, and IaaS. Subscribing to a SaaS application means housing business data outside the controlled local network and within the Cloud infrastructure. An integration architecture specifies how to transform enterprises to bring this outside data into the logical enterprise infrastructure, so that internal and external infrastructure components can interoperate with one another to access needed data. In most cases, implementing a SaaS application involves transferring data from one or more existing applications or data repositories local to an enterprise into a transformed system that combines internal and external infrastructure components. Many vendors provide API that expose the applications data and functionality to developers for use in creating composite applications. Presenting information as a unified whole, instead of as isolated streams of data, carries benefits for users. The fundamental building block of an infrastructure is a workload. Workloads can be thought of as the amount of work that a single server or application container can provide given the amount of resources allocated to it. IaaS providers publish APIs that allow enterprise administrators to build their own solutions on top of the IaaS services. Usually, the APIs support a programming style based on the principles of REST or SOAP. Enterprises can use the APIs to perform such operations as browsing, where the enterprises discover the contents of a container that has an application or a virtual media image, and provisioning, where the enterprises can populate a container with entities such as virtual media ISO images. OVF is an open, portable, efficient and extensible format for the packaging and distribution of software to be run in VMs. OVF was developed by the DMTF, a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability [10, 12].

PaaS is positioned between SaaS and IaaS. PaaS generally refers to internet-based software delivery platforms for which third-party ISVs or custom application developers can create multi-tenant, Web-based applications that are hosted on the PaaS provider's infrastructure and offered as a service to customers. The main premise of PaaS is providing software developers and vendors with an integrated environment for development, hosting, delivery, collaboration, and support for their on-demand software applications. Some platforms offer little more than a set of APIs on top of an elastic infrastructure, while others offer fully functional Web-based IDEs or fourth-generation programming language environments allowing an easy creation of metadata-level mash-ups.

Besides the split into SaaS, PaaS, and IaaS, Cloud Computing has divided along another dimension. Though initial uses of Cloud environments were to access software over the public Internet or Web, enterprises can setup environments internally to have the essential Cloud environment characteristics of network-based self-service deployment and elastic capacity. Such "on-premise" or "internal" Clouds have come to be referred to as "Private Cloud" Because integration flexibility and control over QoS and security are high priorities for many enterprises, and because such enterprises likely have the financial resources to optimize costs over time rather than up-front costs, enterprises may gravitate towards the private variant in their adoption of Cloud Computing. An added attractive feature of Private Clouds is that adopting Private Cloud practices is likely to be a relatively small change for many enterprises. IT departments in many cases have already gone down the path of consolidating infrastructure and setting up shared services, and enabling a Cloud's self service and automated dynamic capacity is often a relatively small incremental step. This is in contrast to adopting of a Public Cloud offering, which can dramatically change how departmental users obtain application support.

A SLA serves as the foundation for the expected level of service between a consumer or an enterprise and a Cloud services provider. QoS attributes, such as response time and throughput, usually form a part of an SLA. Since the QoS attributes change frequently over time and are based on traffic conditions, enterprises need to monitor these attributes. To monitor the QoS attributes, enterprises can demand that monitoring data, such as raw transaction count, be exposed by a SP without further refinement. Alternatively, enterprises can request that collected monitoring data be put into a meaningful context, such as statistical measures of average or standard deviation. This request requires that the Cloud SP create processes to collect data from several different sources and apply suitable algorithms for calculating meaningful results. A second alternative is for enterprises to request certain customized data to be collected. Yet another alternative is for enterprises to dictate the way monitoring data is collected [20].

References

1. Service Delivery Framework reference architecture, TMF061, Release 1.0. TM Forum. Nov 2009
2. Piech, M.: Platform-as-a-Service private Cloud with Oracle fusion middleware. Oracle™ white paper. Oct 2009. http://www.oracle.com/ocom/groups/public/documents/webcontent/036500.pdf

3. Carraro, G., Chong, F.: Software as a Service (SaaS): an enterprise perspective. Oct 2006. http://msdn.microsoft.com/en-us/library/aa905332.aspx
4. Software & Information Industry Association: Software as a Service: strategic backgrounder. Feb 2001. http://www.siia.net/estore/pubs/SSB-01.pdf
5. Rodriguez, A.: RESTful web services: the basics. IBM. Nov 2009. http://www.ibm.com/developerworks/webservices/library/ws-restful
6. Simple Object Access Protocol (SOAP). W3C. http://www.w3.org/TR/soap/ (27 April 2007)
7. United Nations Economic Commission for Europe, *Electronic Data Interchange For Administration, Commerce and Transport,* http://www.unece.org/trade/untdid/directories.htm, Version D.09B, 2009
8. Liberty Alliance: http://www.projectliberty.org/liberty/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates/
9. Goodner, M., Hondo, M., Nadalin, A., McIntosh, M., Schmidt, D.: Understanding WS-Federation. May 2007. http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fed/WS-FederationSpec05282007.pdf?S_TACT=105AGX04&S_CMP=LP
10. vCloud API programming guide, Version 0.8. VMware Inc. 2009. http://www.vmware.com
11. Pandya, R., Upadhyay, S.: Oracle SaaS deployment architecture. Oct 2009. http://www.oracle.com/technology/tech/saas/pdf/saas-data-architecture-whitepaper.pdf
12. Cloudscaling.com: Infrastructure-as-a-Service builder's guide, v1.0, 4Q. 2009. http://cloudscaling.com/files/iaas-building-guide-v1.pdf
13. Distributed Management Task Force, Inc.: Open virtualization format white paper, Version 1.0.0. Feb 2009. http://www.dmtf.org/standards/published_documents/DSP2017_1.0.0.pdf
14. Stankov, I., Datsenka, R.: Platform-as-a-Service as an enabler for global software development and delivery. Proceedings of Multikonferenz Wirtschaftsinformatik 2010, MKWI, Göttingen. 23–25 Feb 2010. http://webdoc.sub.gwdg.de/univerlag/2010/mkwi.pdf
15. Fourth-generation programming language. Wikipedia. http://en.wikipedia.org/wiki/Fourth-generation_programming_language
16. Kobielus, J.: Storm Clouds Ahead: SOA governance clashes with cloud computing model. Network World. http://www.networkworld.com/news/2009/030209-soa-cloud.html?page=5 (2009). 2 March 2009
17. Organization for the Advancement of Structured Information Standards, UDDI Version 3 specification. 2004. http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3
18. Arista Whitepaper: The impact of virtualization on Cloud networking. http://www.aristanetworks.com
19. Patel, P., Ranabahu, A., Sheth, A.: Service level agreement in cloud computing. White paper, Knoesis Center, Wright State University. Sept 2009. http://knoesis.wright.edu/library/resource.php?id=742
20. Keller, A., Ludwig, H.: The WSLA framework: specifying and monitoring service level agreements for web services. J. Netw. Sys. Manag. **11**(1), 57–81 (March 2003)
21. Ludwig, H., Keller, A., Dan, D., King, R.P., Franck, R.: Web Service Level Agreement (WSLA) language specification, Version 1.0. 2003. http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf
22. Spence, C., Devoys, J., Chahal, S.: Architecting software as a service for the enterprise. Intel white paper. Oct 2009. http://download.intel.com/it/pdf/ArchitectingSoftwareasaService.pdf

Chapter 6
Cross-Domain Policy-Based Management$_{1,3}$

As enterprise IT departments increasingly delegate their functionalities to Cloud providers, the business relationship between consumers and providers, as well as service quality management, emerges as a new challenge for enterprises. Due to the dynamic nature of Cloud services, maintaining a satisfactory level of QoS becomes a necessary mission for these enterprises to enforce their business operations. From a provider's perspective, the Cloud forces SPs to speed up their business integrations with other providers and demands dynamic collaborations with CoI members. Such requirements add complexity to the already complicated business operations and introduce uncertainty to the supplier-consumer relationship. Additionally, numerous other factors such as information assurance and cross-domain operations lay another dimension upon the Cloud landscape and warrant a thorough and protective means of service management. Stacks of traditional management systems are no longer sufficient to meet the needs.

To eliminate expensive human interactions for service planning, provisioning, and management, an automatic system that performs overarching collaboration for Cloud services becomes necessary. In this chapter, cross-domain, PBM is presented as a solution to facilitate effective, distributed management for various types of services across Clouds. The functionalities of this solution include distribution and execution of inter-Cloud policy, intra-Cloud policy, inter-security domain policy, and intra-security domain policy. These policies must be planned, created, executed, and managed in an external environment to achieve the most flexibility that is possible under the new requirements. This chapter will begin by revealing the existing IT policy and PBM standards. Based on these standards, the following sections illustrate how a policy template builds up a management framework. After reviewing the existing methodology, this chapter will offer some new ways of thinking about policy management by using a multiple-level policy hierarchy suitable for Cloud services. The chapter will also highlight how the layered hierarchy can be integrated with other enterprise support systems. Critical features, such as policy negotiation and adaptations, are key for effective PBM operations in a Cloud environment. In the final sections of this chapter, we will learn how these features can be utilized to assist enterprises in performing a successful transformation to a completely Cloud-based environment.

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_6, © Springer Science+Business Media B.V. 2010

6.1 Overview

While customers may perceive the benefits of Cloud Computing, they often still have a number of concerns. These concerns may be relevant to cost and flexibility, compatibility with existing applications, lack of a migration path from existing applications to Clouds, federation of internal and external resources, lack of SLA coordination, service interoperability, or a multifold of security issues. The solutions to the above concerns require effective coordination among the managed and managing resources regardless of whether they are virtual or not. For instance, we discussed federation of resources in Chap. 5. In Chap. 8, we will address SLA and SLA management; and in Chap. 9 we will communicate the need to manage Cloud security. These coordination efforts must be governed and provisioned by certain rules and guidance that make sense to enterprises' business [1, 2].

For instance, when enterprises need to implement a management infrastructure to provision a pool of virtual Cloud resources that are independent from the underlying physical infrastructure, the ability to integrate other service partners can help the enterprises avoid having to deal with a variety of resource differences (e.g., vendors, versions, management/control interface, functionalities, etc.). To facilitate a smooth integration from a business perspective, there exists a need for workflows regarding requesting, approving, provisioning, and billing management. Once the workflows are installed, managed equipment can be conscious of the business objectives and mission context, and therefore can carry out accurate infrastructure-level management functions such as forwarding, security, usage analysis, traffic, and utilization (for billing) in a federated manner. Some cross-(security)-domain or cross-organizational interfaces also possess management functions to inspect, transform, mediate, or protect data passed among these resources in Cloud ecosystems. With the complete OSS and BSS integration, enterprises can realize the full benefits of their Cloud strategies by establishing a flexible Private Cloud or extending the Private Cloud into the public domain as needed. To ensure these workflows are performing at the most effective state, an automatic process that can govern the execution of these business and operational objectives with common rules and guidance is essential.

Although these rules and guidance exist in the form of enterprise policies, the word *policy* can have a number of meanings when used in conjunction with different IT architectures and systems. For example, a policy can mean "governance" relating to system architecture development and implementation, or it can mean "operational rules and standards" for administering a deployed production system. For instance, in telecom services, bandwidth can be made available to users during a particular time of day as a function of the total number of users present. Just as policy is used in the telecom services to shape the use of critical resources, Cloud policy can be used to shape the execution of the business functionality. PBM is generally perceived as a system that can facilitate integrated processes and controls of enterprises' management systems. PBM is an integral part of BSS or OSS, as shown in Fig. 6.1, and helps enterprises automate their operations, thus minimizing the

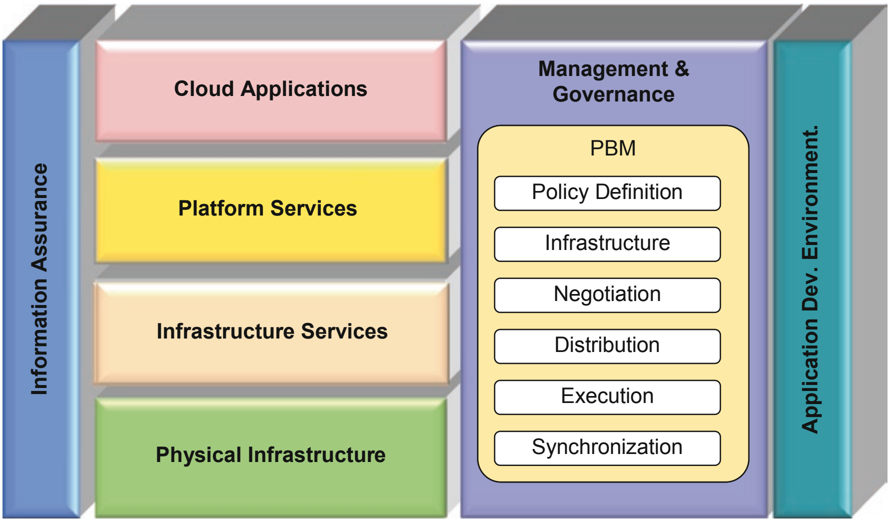


Fig. 6.1 General Cloud infrastructure

operational complexity in dealing with end-to-end management and security. PBM assists Cloud providers in provisioning new services quickly and cost effectively with a high degree of scalability, flexibility, and transparency. It brings resources to the requesting user quickly and effectively, regardless of who owns the resources, who makes it, or where it is. For backward compatibility sake, a PBM is mandatory to support and address any legacy systems and devices that enterprises are supporting [3, 4].

Figure 6.1 shows the general Cloud infrastructure with specific decompositions of PBM. In this chapter, we will discuss the needs and potential solution options to formalize the way that policy is used to manage Cloud resources within and across service/security boundaries.

6.2 PBM Benefits and Potential Applications

In a traditional IT management environment, PBM is considered an administrative approach that is capable of simplifying service management with definable policies to deal with predictable situations and conditions. Policies are a set of operating rules referred to as a means of maintaining order, security, consistency, or other ways of successfully furthering a goal or mission. For instance, delineated policies are used to control access to and priorities for the use of networking resources. From an operational perspective, rules are used in response to certain situations or the creation and operation of a computing environment. A high-level policy is called a business or mission policy. It is entered into the management system as a guidance and communicates with lower-level resources to execute the service designer's intentions. In the following sections, we will look at the benefits and applications of PBM in the current IT industry [5].

6.2.1 The Benefits and Business Drivers of PBM

PBM has been portrayed as an effective mechanism to orchestrate the behavior of distributed systems. Different generations and vendors of policy management concentrate on different aspects of this problem, ranging from defining business goals to specifying low-level configuration changes in a device. However, they can be categorized by the following three goals in an IT environment:

- Compliance entails providing operators with assurance and complying with operational, security, and business guidelines.
- Consistency entails ensuring the committed levels of performance, security, availability, and reliability.
- Cost drives the business strategy to control overall cost and enables more efficient and effective management.

These goals are applicable to many IT applications in commercial, enterprise, or research environments and affect all types of SPs and operators. Table 6.1 outlines the generic drivers of PBM in these IT environments. They are categorized by four application areas with different corresponding drivers.

6.2.2 PBM Support OSS and BSS

BSS and OSS are two key groups of systems that support enterprise IT management infrastructure. At the enterprise level, the methodology of designing and implementing PBM may include the following steps:

1. *Identify and Define Use Cases*: Describe the behavior of a policy-based system.
2. *Describe Policy Categories*: Identify and describe the general groupings of policies across the use cases.
3. *Design Logical Policies*: Design policies for each use case by applying the policy template.
4. *Transform and Implement Policies*: Transform policy designs into implementable, machine-readable policies.
5. *Monitor Policies*: Gather metrics and evaluate the impacts of policy implementation.

From a functional perspective, PBM plays an essential role in supporting the automation of collected enterprise management systems. It provides tremendous value to operators, allowing them to facilitate internal and external integration of their

Table 6.1 Generic drivers of PBM in IT environments

Area	Driver	Details
Technology	Complexity	Meet business (SLA) demands Elevate resource values Minimize errors Utilize vendor and technology management Improve user experience: e.g., self-management
	Adaptation	Adapt new services and technology
Business	Agility	Reflect market changes quickly
	Scalability	Apply the policy uniformly to large sets of devices/objects across different management domains Avoid the strenuous task of re-coding Embed decision making ability
	Flexibility	Separate policy from the implementation of managed systems Separate policy from management entities Make dynamic modifications
Time/Resource constraints	Efficiency	Maximize operation efficiency (e.g., load sharing, dynamic re-prioritization) Minimize personnel shortfalls Maximize ROI
Systematic solutions	Effectiveness	Increase predictability (know what actions result from what events under what conditions) Increase consistency (take the right actions at the right time in the right way) Automate complex operational rules to manage/provision resources and services Streamline workflow in a distributed and heterogeneous environment Implement regulatory compliance

offerings, and helps enterprises ensure all defined policies, processes, and technology seamlessly work together to achieve optimized results. The following list is a collection of management systems/functions that potentially interact with a fully functional PBM:

- *Customer Management Platform*:
  - Customer order management
  - Customer relationship management
  - Customer problem management
  - Customer SLA management
- *Supply Chain Management Platform*:
  - Supply chain operations
  - Supply chain development
- *Service Management Platform*:
  - Service configuration
  - Service performance

  - Service quality assurance
  - Trouble tickets
  - Resource configuration

- *Workforce Management Platform*:
  - Workforce management
- *Billing Platform*:
  - Customer billing
  - Supply chain billing
  - Service usage
- *Enterprise Management Platform*:
  - Security management
  - Identity management
  - Asset management

6.3 PBM Standards and Commercial Implementations

Using policy to automate equipment provisioning has a long history in the telecom industry. Some standard bodies previously tried to establish a common rule set so vendors and operators could follow a common practice. Although many good ideas were introduced, only a few specifications survived. Part of the reason for this was because the previous specifications were developed bottom-up. Many proven concepts for circuit and signal-level policies were adopted and implemented at the chipset level. As the concept continues to stream up to service and business operations, modern policy specifications focus more on system integration and service federation. In this section, we will review the two most popular standards, the details of their policy models, and some commercial implementations.

6.3.1 TM Forum SID's Policy Aggregate Business Entities

A policy framework describes the architecture for policy management. It includes the policies, how policies are used, and where they are stored. It also defines objects, which are managed by certain rules. To effectively manage resources residing in different operational scopes, the architecture also covers a domain space, rule space, policy driver, and action space.

In Chap. 3, TM Forum's NGOSS and its *Information Framework* called SID were first discussed. Within the SID *Aggregate Business Entities* (ABE) Common Business domain, policy is listed as a sub-domain. This relationship is shown in Fig. 6.2. In this section, we will describe in detail the policy management section of SID and its origin: the *Directory Enabled Networks* (DEN)-*Next Generation* (ng) policy model.

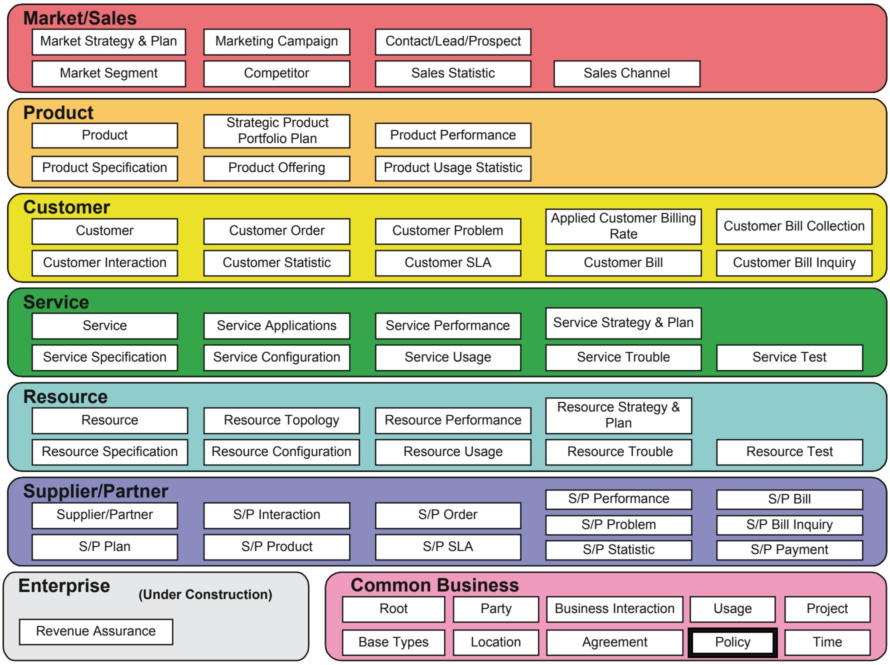


Fig. 6.2 Policy as one of the common business entity domains of the TM Forum SID Framework

DEN is a specification of an object-oriented information model describing the elements and entities in a managed environment and how they are related to each other. It also specifies a model mapping to a format that can be stored in a directory with LDAP as the access protocol. An earlier version of DEN-ng was adopted by TM Forum as the basis for its PBM function in the SID specification. DEN-ng is an enhanced version of the DEN specification and was incorporated in the TM Forum standard because of its strong tie to the NGOSS architecture. However, after TM Forum's adaptation, the DEN-ng has continued to evolve and is now becoming much more complex and generic. However, the new development is no longer part of the SID specifications.

Like DEN, DEN-ng is an object-oriented information model that describes the business and system views of managed entities and their relationships. This definition is created using UML. The DEN-ng model differentiates between the management of policy and controlling a managed entity using policy. Specifically, the former refers to managing policy rules, groups, and components, while the latter signifies using policy entities to control the state of a managed entity (or set of managed entities). DEN-ng uses a Finite State Machine for representing states. According to the NGOSS document, its ability to "chang[e] and [maintain] the state" is one of the most distinguished features of the DEN-ng model that is lacking in other similar policy models [6].

The purpose of the policy continuum, shown in Fig. 6.3, is to provide a semantic linkage between different types of policies that exist at different levels of abstrac-

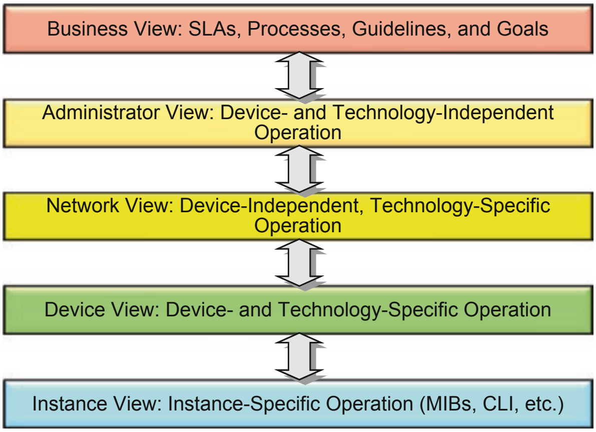

Business View: SLAs, Processes, Guidelines, and Goals
Administrator View: Device- and Technology-Independent Operation
Network View: Device-Independent, Technology-Specific Operation
Device View: Device- and Technology-Specific Operation
Instance View: Instance-Specific Operation (MIBs, CLI, etc.)

Fig. 6.3 The policy continuum

tion. Five levels of abstraction are: the *business*, *administrator*, *network*, *device*, and *instance* views. The policy continuum defines a semantic relationship between each of these levels, and each level represents one set of related policies. In other words, the notion of a single policy is eliminated because there is always a set of policies that exist at different levels of the policy continuum. Furthermore, one of the most important features required to translate policies at one level of the policy continuum to another level is to align the needs of the user at each of the different levels. This special feature enables the flexibility to align the inputs to, outputs from, and behavior of each set of policies at each level of the policy continuum.

The DEN-ng policy information model includes a set of managed entities that can be used to relate different forms of policy to each other. It provides an infrastructure to enable a set of mappings to be defined that transform the data between each type of policy in the policy continuum. The specification provides a layered set of policies with different levels of abstractions, and model mappings to translate between them. It is the job of the policy system to translate these entities and concepts between layers of the policy continuum. The DEN-ng policy model includes the information to represent policies as well as to enable policies to be treated as managed entities. From the SID perspective, this means that policies can interoperate with and manage other managed entities defined in SID. Note that the DEN-ng policy model is the only information model that uses the concept of a policy continuum. The representation and application of policy are abstracted into a set of concepts that are appropriate for businesses. This set of abstractions, which are defined by the business view of the DEN-ng Policy model, can then be used for the following tasks: [7–9]

- Define a canonical model for policy
- Standardize the representation of policy independent from the content of the policy
- Provide an infrastructure that supports extending policy to support application-specific domain information

Each of the five views depicted in Fig. 6.3 is optimized for a different type of user and thus requires slightly different information at these layers. For example, a business user may need only the SLA information and is not concerned about how the purchased network service is programmed to deliver the specified QoS. This user is only interested in the fact that the network is delivering the right type of QoS based on the SLA. Conversely, network administrators need to "translate" the QoS that is implied by different SLAs into sets of CLI languages in order to program the appropriate devices. This is a completely different representation of the same policy that executes the SLA specifications. Furthermore, network administrators are not responsible for the financial or other contractual information in the SLAs. They are only concerned that the sets of CLI commands governed by network policies are correct. It is obvious that these two views are both correct but different. This is a simple case of why the policy continuum is used. In most cases, such diverse views require different policies that may demand different syntaxes to be related to each other. Therefore, the concept in DEN-ng treats policy as a continuum, where different policies take different forms and address the needs of different users. In Chap. 8, we will discuss how SLAs are related to customer experience via a service policy.

When we further zero in on the policy domain highlighted in Fig. 6.4, it shows the details of two other levels of Policy ABEs. The first level of policy ABEs is

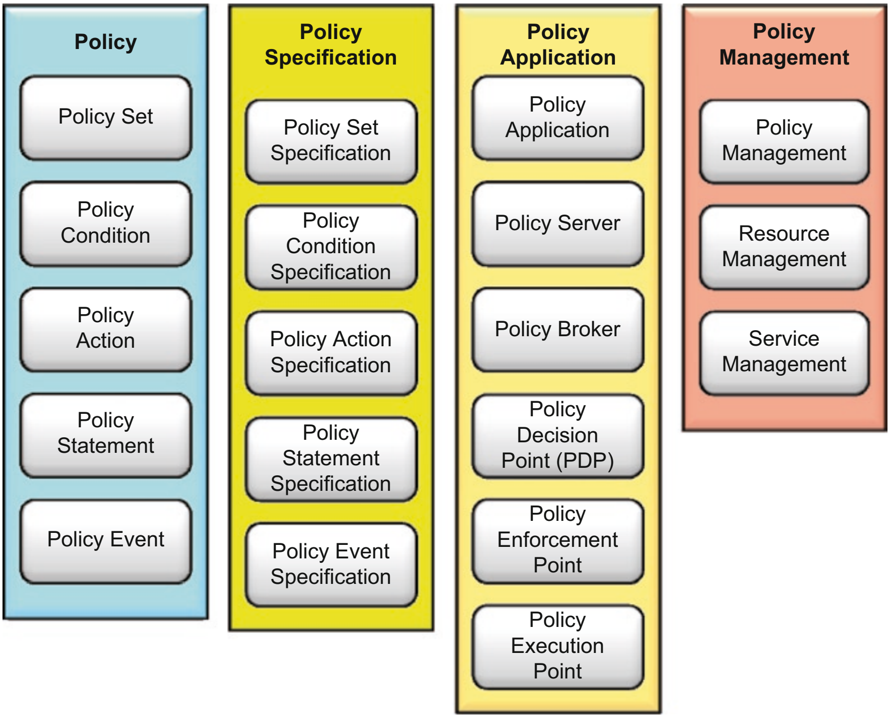


Fig. 6.4 Level 2 of the policy domain in the SID framework

defined by the four rounded rectangles labeled “Policy,” “Policy specification,” “Policy application,” and “Policy management.” Each of these four levels has three or more Level 2 ABEs. In other words, the Policy Level 1 ABE is made up of five different ABEs: *Policy set*, *Policy condition*, *Policy action*, *Policy statement*, and *Policy event*. These five Level 2 Policy ABEs define the Level 1 Policy ABE in greater detail. The ABEs not only group similarly managed entities together, but also help define important relationships between different ABEs within the Policy domain as well as between the Policy domain and other domains.

6.3.2 IETF Policy Workgroup

In addition to the TM Forum’s policy specification, another influential research activity on policy standards is carried out by the IETF Policy working group. Instead of using programming language to specify policies, the IETF team extended the CIM from DMTF and created a new object-oriented information model. In the IETF model, a policy rule is modeled as an aggregation of policy conditions and policy actions. Policy rules, conditions, and actions are represented as object classes and their associations are modeled with association object classes. Network QoS policies within the IETF Policy framework are represented according to the information model that is extended from the *Policy Core Information Model* (PCIM, RFC 3460). It includes QoS-related policy actions, values, and variables in order to incorporate QoS-specific semantics in the framework [10–12].

Storing policies in a central directory is a key component of the PBM framework, which is accepted as a powerful technology for the management of large networks. Apart from providing an information model for representing policies, the IETF framework also defined a schema for storing policies in a directory. This framework also uses the LDAP as the access protocol, similar to the DEN specification. The IETF architecture proposes the enforcement of policies as presented in Fig. 6.5.

This figure (Fig. 6.5) depicts how the Policy Administration Console can interact with the Management Tool to define policy, control system and service behaviors, and verify that intentions are executed at the management resources. The abstraction of service requirements, service rule representation, and device-specific instructions are exchanged through a well-defined open interface. The role information for these three layers is stored in the central repository [13].

In the IETF architecture, directories are used for storing policies but not for grouping subjects and targets. Because the concepts of subject and target does not exist, the determination of mapping components to a policy must rely upon other means such as interface roles. Furthermore, the IETF model does not offer an architecture for policy rules to be triggered by events dynamically in order to reconfigure the managed system based on different circumstances. In accordance with the published specifications, IETF’s policy work seems to focus only on the network layer and lacks the consideration of the interaction between application and network poli-

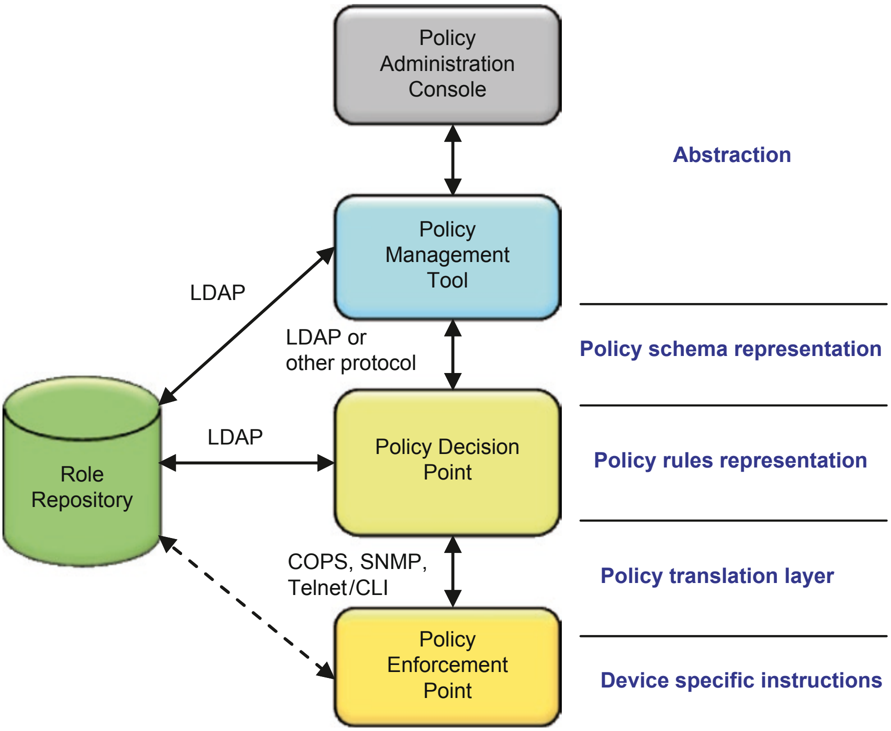


Fig. 6.5 IETF policy enforcement architecture

cies, let alone business and service policies—two critical pieces of the IT applications and services. Figure 6.6 depicts an application of the IETF policy management framework for satellite communication management, developed by DISA.

6.3.3 Market Players

A number of vendors are marketing policy management toolkits in the form of *Commercial-off-the-Shelf* (COTS) systems. The majority of these commercial tools are specific to QoS management, but many also include access control configurations. Listed below are a few examples of major commercial products that are specific to QoS management:

- *Cisco QoS Policy Manager* (v3.0): This policy manager supports a broad range of Cisco devices, including routers and switches. Following the IETF policy representation, a QoS policy rule consists of a set of conditions and a set of actions. Policy actions (actions for classification, limiting, shaping, and queuing traffic) are applied on a traffic flow if the flow matches the filters (conditions) defined in the policy. Filters define traffic characteristics. In addition, the Cisco QoS Policy

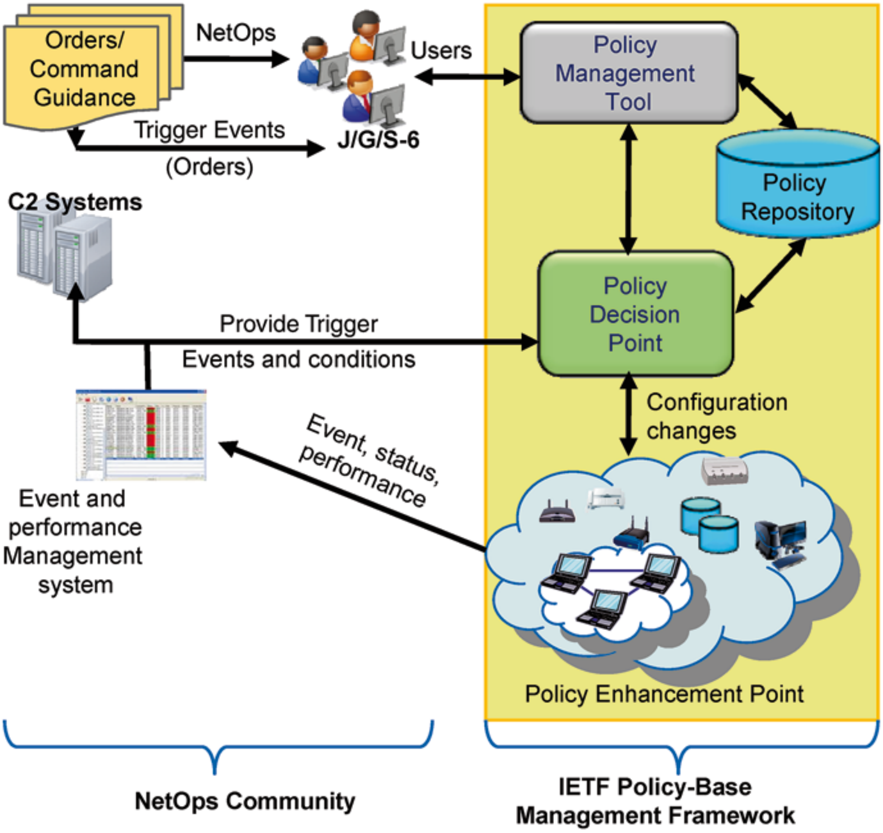


Fig. 6.6 Policy management framework and policy engine

Manager provides a Web-based interface to define QoS policies and translates the policies into device-specific CLI commands. Since policies do not specify their target elements, prior to deployment, the administrator manually assigns a set of devices to each implemented policy rule through the management console. Policies are stored in the manager's QoS database, which is vendor-specific; policies are not stored according to a directory schema that follows a standard information model [14].

- *HP PolicyXpert*: PolicyXpert defines policy as a combination of one or more sets of rules. Policy rules consist of a single action and one or more condition lists. These are constructed from one or more conditions, which match against time/date or packet/traffic characteristics. Policy actions are used to manage DiffServ and RSVP mechanisms. The product offers support for the management of devices from a number of vendors. It also offers an Agent *Software Development Kit* (SDK). This SDK enables vendors to develop support for specific QoS mechanisms on their devices [15].
- *Allot Communications NetPolicy*: NetPolicy also follows the IETF standards. A policy rule consists of conditions and actions. Conditions are used for matching

IP addresses, protocols, application data, *Type of Service* (ToS) settings, and the time of day. The administrator can group devices together in domains and manually enforce a set of policy rules to an existing domain. The COPS protocol is used only when NetPolicy uses the NetEnforcer device as the enforcement point. Communication with other devices is realized through the CLI or SNMP. Directories are used not for storing policies, but for retrieving users and applications information [16].

- *CA*: CA's business-driven automation, service management, application performance management, and database management solutions now support the Amazon EC2. Support of the CA *Spectrum Automation Manager* (CA CMDB), the CA service desk manager (CA Wily Introscope), and the CA *Insight Database Performance Manager* (DPM) for Amazon EC2 can enable customers to achieve Lean IT by provisioning capacity to Amazon EC2 [17].
- *IBM's WebSphere XD*: WebSphere XD is designed to use policy-based request routing in order to control routing requests for system administrators. Its 6.0 and later versions enable administrators to define rules-based routing and service policies. The routing policies control if and where a request is routed, while the service policies control how fast a request is serviced. Other similar products include *IBM Tivoli* and *Smart Business Storage Cloud* [18–20].

In addition to the products mentioned above, there are also compatible or competitive products from other companies. For example, Lucent's RealNet Rules, Nortel's Optivity, Extreme Networks's ExtremeWare, Gold Wire Technology's Formulator, and Dorado Software's Redcell Suite are among the few available systems. Although each product has its own niche features, the majority of these products are alike. For instance, most of the tools specify the policies in the form of "if <condition> then <action>" rules. Target elements are assigned to policies either manually through the administrator console or by using a role-based model. Different products allow for specifying various degrees of conditions in policy rules, including a number of time attributes, source or destination IP addresses, IP ToS, TCP, and UDP port numbers, as well as higher-level user-defined data. They also allow users to permit or deny traffic based on those conditions. However, none of the aforementioned products support a policy specification language and none of the products appear to have considered the automation of the policy lifecycle and how to adapt the configuration of target network elements when conditions change within the managed network. In most cases, new configurations need to be imposed manually by the administrator through a management console [21].

6.4 Policy and Management Framework

With PBM approaches, the distribution of business and mission intentions, the automation of the management process, and the dynamic adaptation of the behavior of the managed systems are achievable by using different flavors of policies. In a nutshell, policies are derived from the goals of management and defined based on the desired behavior of distributed heterogeneous systems and networks in an IT environment.

Although policies can potentially be presented various syntaxes for different applications, they are generally broken down into three categories for the ease of definition coordination:

- *High Level policy*: Large scope, ambiguously worded directives possible, guidance, or instructions
- *Command or Operation policy*: High-level policy interpreted for enforcement across an *Area of Responsibility* (AOR) or across a CoI
- *Executable policy*: A translated version of the command or operation policy that can be processed and implemented through automation; usually defined by the following attributes: [22]
    - *Events*: Underlying events (e.g., performance, security breach, failure, etc.) that trigger policy execution pending the satisfaction of any policy conditions
    - *Conditions*: Circumstances that must be true before policy execution can be triggered
    - *Actions*: Tasks to be executed as a result of the policy being triggered (occurrence of event AND satisfaction of the conditions). Types of actions include: configuration/requesting, monitoring, reporting, filtering, aggregation/fusion, processing, etc.
    - *Scope*: The domain (e.g., set of services, capabilities, or resources) to which the policy applies
    - *Metrics*: Describe the effectiveness of the policy in terms of the performance of the operational processes of the system as well as the performance of the services that the system provides

6.4.1 Policy Template

Consolidating data from different sources, this section is intended to list a set of policy templates that is useful and available in the industry. The following attributes have been used to express most of the needed resources' management in the IT industry: [23, 24]

- *PolicyRule*: A PolicyRule is an intelligent data container. It contains data that define how the PolicyRule is used in a managed environment as well as a specification of behavior that dictates how the managed entities that it applies to will interact. The contained data consists of four types: (1) data and metadata that define the semantics and behavior of the policy rule and the behavior that it imposes on the rest of the system, (2) a group of events that can be used to trigger the evaluation of the condition clause of a policy rule, (3) a group of conditions aggregated by the PolicyRule, and (4) a group of actions aggregated by the PolicyRule. For instance, the DEN-ng policy model is deceptively simple, a triplet defined as an event clause, a condition clause, and an action clause. In this context, a "clause" means one or more expressions can be used to define events, conditions, and actions. An event is a condition used to trigger the evaluation of one or more other conditions. If the set of conditions evaluates to TRUE, then one or more of the set of actions associated with this PolicyRule will be executed.

- *PolicyEvent*: A PolicyEvent is an occurrence of an important event, and can be used to trigger the evaluation of a PolicyCondition or PolicyCondition clause in a PolicyRule. An event can be an alarm, a user inserting a card, and so forth.
- *PolicySet*: A PolicySet class represents an aggregation of PolicyEvents, constrained according to the eventConstraint attribute of the EventDetails aggregation class. This set of PolicyEvents is then presented to one or more PolicyRules to trigger the evaluation of their condition clauses. This enables an external application, such as a Policy Server, to dynamically adjust the set of events that are being used to trigger the evaluation of a PolicyRule.
- *PolicyCondition*: A PolicyCondition class is an aggregation of individual PolicyConditions. It is treated as an atomic object aggregated by a PolicyRule. PolicyCondition is normally represented as a Boolean expression and includes the definitions of necessary state and prerequisites to determine if the actions aggregated by that same PolicyRule should be performed. This is signified when the PolicyCondition clause associated with a PolicyRule evaluates to "True."
- *PolicyAction*: A PolicyAction clause is an aggregation of individual PolicyActions, and is treated as an atomic object that is aggregated by a PolicyRule. It represents the necessary actions that should be performed if the PolicyCondition clause evaluates to "True." These actions are applied to a set of managed objects, and have the effect of either maintaining an existing state, or transitioning to a new state of those managed objects.

6.4.2 Policy Implementation and Usage

The IETF Policy Framework (POLICY) and TM Forum SID framework working group has developed a policy management architecture that is considered the best approach for policy management for the E2E solution. The purpose of this section is to illustrate terms that are used in the DEN-ng Policy model to represent how policies are used in a PBM system. Figure 6.7 depicts a typical PBM implementation breakdown. A typical lifecycle diagram is showed in Fig. 6.8 [6, 24].

6.4.2.1 Policy Domain, Conditions, and Entities

To effectively conduct policy management and perform needed service automation and refinement, a PBM system must include a set of vocabularies that can express the management scope, objects, and conditions to support the policy template described in Sect. 6.4.1:

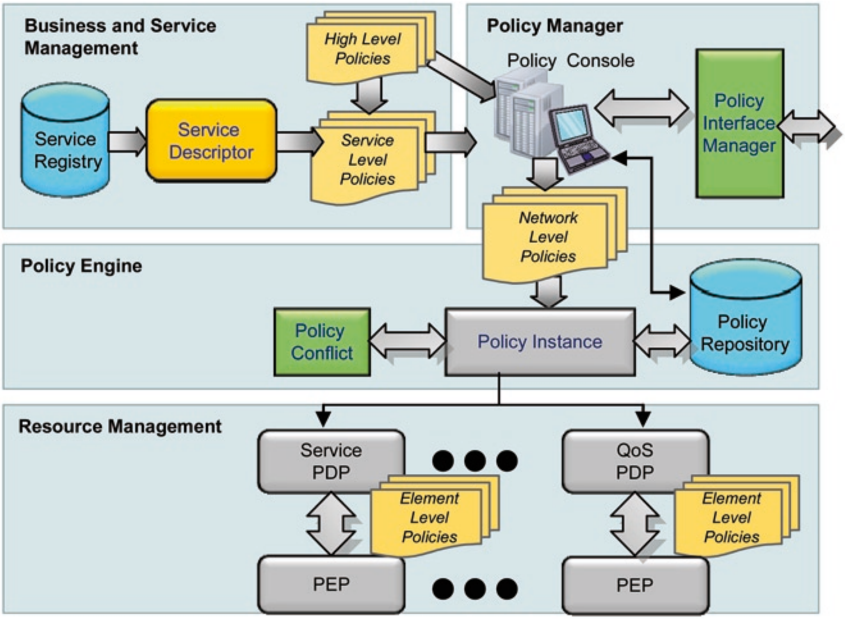


Fig. 6.7 Typical PBM implementation

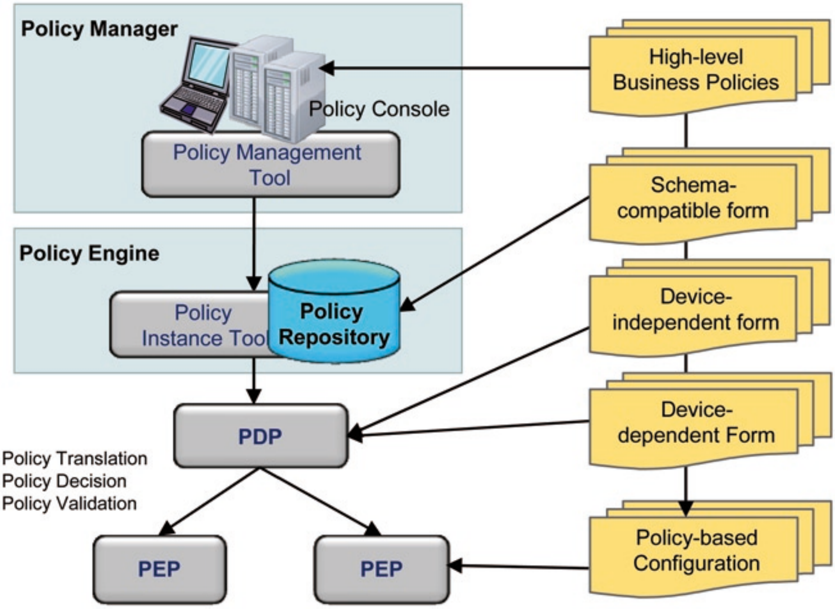


Fig. 6.8 Typical policy lifecycle flowchart

- *Policy Domain*: A Policy Domain is a collection of managed entities that are operated on using a set of policies. The policies are used to administer and control the set of characteristics and behavior of these managed entities. The purpose of defining a Domain is to define a set of managed entities that are all operated on in the same way. While administration is important, it is only one of a set of operations that are targeted on entities in a domain.
- *Policy Subject*: A Policy Subject is a set of entities that is the focus of the policy. The subject can make policy decisions and information requests, and can direct policies to be enforced at a set of policy targets. Note that a Policy Subject is an architectural concept, as defined in the literature. However, DEN-ng defines a *role* to implement the concept of a Policy Subject.
- *Policy Target*: A Policy Target is a set of entities that a set of policies will be applied to. The objective of applying policy is to manage the state transitions of the Policy Target. A Policy Target could be a *device* (e.g., power it on), a *device interface* (e.g., check if it is up or down), or a *device configuration* (e.g., define traffic conditioning, protocols, and other operations). Note that this definition uses the notion of using a finite state machine to control the behavior of the Policy Target.
- *Policy-Aware (or Policy-Enabled) Entity*: A Policy-Aware Entity is one that can understand and use policies to make present and future decisions. These decisions are used to manage and control change and/or maintain the state of one or more managed objects that are the targets of the policy
- *Policy-Unaware Entity*: A Policy-Unaware Entity is one that can neither understand nor use policies to make present and future decisions. A Policy-Unaware Entity cannot use policies to manage and control change and/or maintain the state of one or more managed objects.
- *Policy Conflict*: A policy conflict occurs when the conditions of two or more PolicyRules that apply to the same set of managed objects are simultaneously satisfied, but the actions of two or more of these PolicyRules conflict with each other.
- *Policy Evaluation*: A Policy Evaluation is the set of computations necessary to determine if the PolicyCondition clause is satisfied.
- *Policy Decision*: A Policy Decision is the determination that one or more PolicyActions that are aggregated by a PolicyRule should be applied to a set of managed objects. These PolicyActions correspond to either maintaining the current state or transitioning to a new state of each of the managed objects that it is affecting.

6.4.2.2 Policy Management Processes

In addition, the model must contain application entities assisting the PBM system or solution designers to appreciate the architecture of the management system. These “processes” can exist in the form of a software component, a software function as part of a software component, or an independent software solution. These processes are:

- *Policy management service (Policy console)*: The Policy console is responsible for creating and managing policies, in addition to providing a *Graphical User Interface* (GUI) for specifying, editing, and administering policy.
- *Policy Decision Point (PDP)*: A PDP is an entity that makes Policy Decisions for itself or for other entities that request such decisions. It is also a resource manager or policy server that is responsible for handling events and making decisions based on those events, and for updating the PEP configuration appropriately.
- *Local Policy Decision Point (LPDP)*: This is a scaled-down PDP that exists within a network node and is used in cases when a policy server is not available. Basic policy decisions can be programmed into this component.
- *Policy Enforcement Point (PEP)*: An entity that is used to verify that a prescribed set of PolicyActions have been successfully executed on a collection of PolicyTargets. Note that a PEP is an architectural concept. PEP exists in network nodes such as routers, firewalls, and hosts. It enforces the policies based on the "if <condition> then <action>" rule sets it has received from the PDP.
- *Policy Execution Point (PXP)*: An entity that is used to execute a prescribed set of PolicyActions on a set of PolicyTargets. Note that a PXP is a defined architectural concept.
- *Policy Repository*: A policy repository is an administratively-defined virtual container that is used to hold policy information. Virtual container can be a standalone data store or a collection of many data storages. Information stored in the virtual container includes policy rules, policy groups, and relevant data used to support PBM.
- *Policy Server*: A Policy Server is a collective set of entities that can be used to replicate the core policy management functionality in a distributed implementation. It consists of at least one PDP, one PEP, control logic to detect and resolve policy conflicts, and optionally one or more proxies to communicate to the external world.

6.4.3 Policy Management and Policy Engine

A policy engine is a collection of functionalities that accepts predefined policies, verifies policies, executes policies, and reports results of policy execution. In a more comprehensive engine implementation, the verification function includes certain intelligence to mediate policies from different sources, check for their consistency, and even perform policy negotiation in case there is any conflict or ambiguity. This is because in an enterprise IT environment, enterprise management systems often need to perform policy collaboration and enforcement processes. It is critical that PBM eliminate conflicting policies in a multiple domains environment to prevent the enterprise from provisioning incorrect resources or configuring inappropriate service behaviors for the customers.

To achieve this objective successfully, certain guidance for a policy engine's deployment and operations must be followed. Firstly, PBM processes (e.g., PDP and PEP) must be deployed to all the managed resources in the service environment that need to be managed. Secondly, the multi-level policies must be understandable and executable in all deployed PBM processes. Thirdly, these processes must be extensible and have the ability to detect policy violations that are defined by the users or operators. For practical purposes, all policy languages, processes, components, and systems must be able to manage version differences. Version control for policies and events is necessary so the PBM decision making process can be uninterrupted. To improve the overall service efficiency, policy exception management should be fully automated if possible, with a configurable option to include human in-the-loop for final justification [25].

In situations where events may have service implications from an abnormality in other resources instead of the reporting resource, the policy engine may require more sophisticated intelligent to perform root cause analysis. For instance, users' inability to connect to an application should be automatically handled in three categories: (1) *software system exception*, (2) *network unavailability*, and (3) *security violation*. If the deployed management system is incapable of making a fair judgment from the available service data, the procedure in the SP should invoke human-in-the-loop to determine the next course of action according to the SLA. Further discussion on this subject can be found in the following sections.

6.5 Transforming PBM to a Cloud Environment

The existing policy solutions described in the previous sections have several key limitations and restrictions to support the Cloud and Cloud services. This is mainly due to the requirements for supporting the dynamic nature of service participants and complex community relationships of Cloud ecosystems. From an implementation perspective, the main obstacle in moving to Cloud environments is that both business processes and policies of the traditional PBM systems are mainly embedded in monolithic applications. This is because an automated service federation is a rather new business practice in the IT industry as policies and processes were not treated as formal architecture components. As a result, the scalability and flexibility of provisioning PBM capabilities has been limited to the domain boundary such as regulatory, industry, market sector, provider, network, or vendor. As enterprises are now entering a new technology era, their offerings must be able to deal with globalization via loosely coupled interactions with their business partners. The need for policy to be made explicit, scandalized, and automated becomes critical for enterprises to conduct successful business in a quickly changing business environment. In the following sections, we will provide the theory, method, and implementation concepts for how the existing PBM can be transformed to an effective management capability in Cloud environments [26].

6.5.1 Cloud-Focused Policy Stack

In this section, we propose a hierarchical view of service operations for Cloud environments, as shown in Fig. 6.9. This operational view for a Cloud can be mapped to the Cloud policy stack portrayed in Fig. 6.10.

At the higher levels of the policy stack, such as the federation and business levels, are abstract operational guidelines. While the lower levels of policy, such as policy, configuration, and parameters for network and device resources are more physically-oriented specifications.

In a side-by-side comparison of Figs. 6.9 and 6.10, the service perspectives defined in Fig. 6.9 can be directly related to a policy layer in Fig. 6.10. For instance, the most abstract view, the E2E Business View, can be translated to the Level 1 policy stack, Federated Cloud Policy. The applications of this type of policy include Hybrid Clouds or enterprises that integrate their internal Cloud with Public Clouds or other Private Clouds. In such a business scenario, all participating Clouds may maintain their own operational policies, but any cross-domain or cross-organization interactions must be operated and compliant with a common set of policies and guidance. Additionally, the end-to-end QoS (for services and operations) must be regulated by a universal agreement, in the form of a SLA.

The second layer in Fig. 6.9, the Cloud Designer/Architect View, corresponds to the second level of policy in Fig. 6.10, the Cloud Policy. This is a Cloud or enterprise-focused policy where it governs the operations and business directions of the Cloud or enterprises with derived and extended policies from the federation

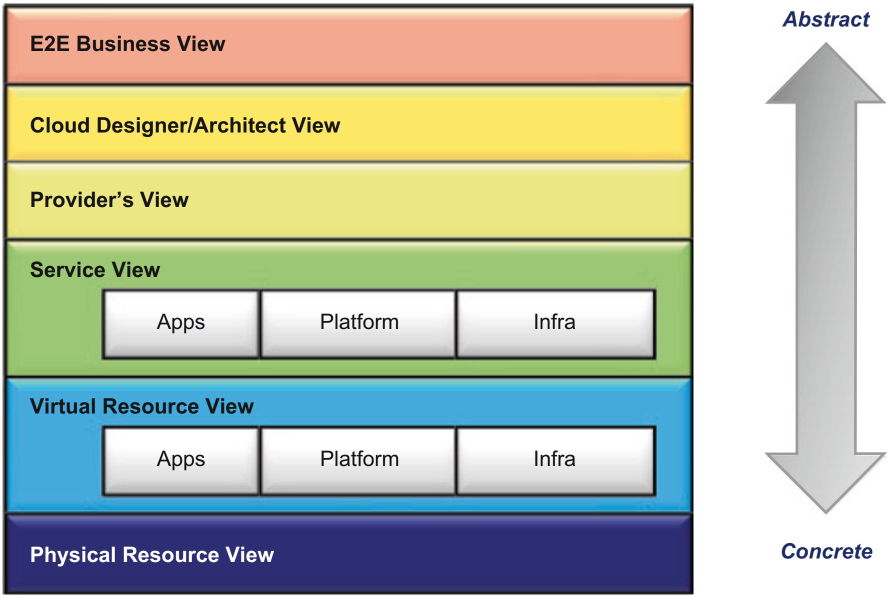


Fig. 6.9 Hierarchical view of policies for Cloud services

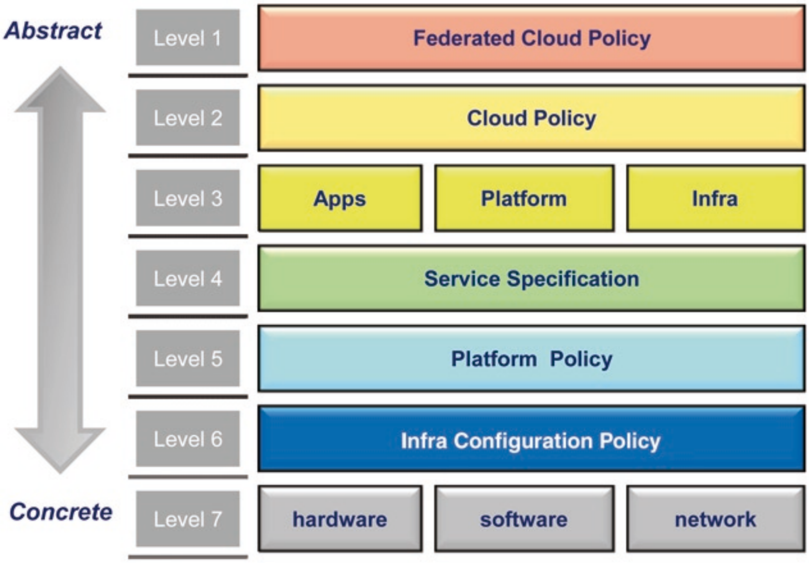


Fig. 6.10 Policy stack for Cloud services

level. Keep in mind that this policy stack stems from our observation that successful enterprises that adopt a PBM solution must make their IT infrastructure sensitive to the provision of a service that can exchange economic values with their providers. It is essential that Cloud providers make the PBM capability clearly aware of business-level considerations. Therefore, a clear distinction of business and service policies is necessary in our model. This correlation offers a workable stack that includes a business aware layer and the underpinning policy-based service and resource management layer. In other words, the business policy layer provides business context to the supporting service and resource layers.

The business aspect of the Cloud Policy (i.e., business policy) is a set of enterprise-wide rules and regulations assigned to business objectives and strategies. Business policy dictates user profiles, allowing services and operations to take effect at the task (execution) level. In our model, business policies cannot place constraints upon the affiliated business functionality, they can however harmonize constraints at the infrastructure (hardware and software) level that provision the functionality through policy negotiation (will be expounded on in Sect. 6.4.2). The constraints can include accounting rules that enterprise businesses follow, RBAC on business functionality, corporate policies, rules on deploying new virtualized application images, infrastructural policies that might prefer one customer over the other when critical resource contention happens, and so forth. Because the main objective of the business policy layer is to drive the management of Cloud resources and services from a business point of view, Cloud providers and enterprises must always look for all available options that can grant the possible minimum cost or the least amount of disruption to the service offerings. Critical to the trade-off analysis, unless the impact to the chosen course of action onto the business layer is well understood, there is always a risk of solving the wrong problems or applying the solution to the wrong places. Because of this concern, the business policy layer must maintain service knowledge in accordance with the rules that pertain to visible business implications from underneath Cloud resources.

The Cloud provider view in Fig. 6.9 can be mapped to the service-oriented Service Catalog Policy shown in Fig. 6.10. The Catalog may include information such as types, characteristics, availability, and billing of each service type specified. To improve management efficiency, service types must be distinguished, in other words, differentiating whether it is a SaaS, PaaS, or IaaS. In a modern service catalog, active service information and references, including a set of rules, regulations, liabilities, and responsibilities, is associated with enterprise service organizations. This allows Cloud providers to proactively manage the offering in accordance with the given business goals and objectives. An active service catalog also often includes policies that can provide abstract service models. These assist providers' task executioners without having to deal with physical assets.

Going down the stack hierarchy, the policy entities become more concrete and specific. The Service View in Fig. 6.9 corresponds with the Service Specifications in the policy stack, shown in Fig. 6.10. The policy at this layer focuses on service fulfillment, assurance, and billing. Examples of the management information include governing metrics and rule sets for SLAs. The Service Policy is activated, compliant, and provisioned by the information from the upper Federated Cloud Policy and the intra-Cloud Policy. In complex cross-domain Cloud environments, Service Specification must comply to both the regional SLA as well as the end-to-end SLA, both will be further discussed in Chap. 8. For Cloud providers, sets of OLAs that govern the network-layer rules and operations for network operators often appear in this layer. Both SLA and OLA contain several SA report templates for monitoring the performance and fault data of the offering. Functionally, Service Specification is linked to and based on SLAs and OLAs in order to:

- Enable and disable user profiles
- Support the policy file (the profile is based on the agreed-on Federated Policy and enterprise-wide Cloud Policy)
- Instantiate and deploy the Service Catalog Policy file (policy files are injected into the SA subsystem)
- Determine how SA reports will be disseminated

The Virtual Resource View in Fig. 6.9 maps to Levels 5 and 6 in the policy stack. Platform Policy is mainly used to address policies in a PaaS and Infrastructure Configuration Policy is mainly used to address a generic infrastructure-specific policy for managing the network, VPN, CPU, rate, etc. Because the platform sometimes acts as a portal to the Cloud infrastructure resource, users can have a choice of managing these resources at either layer. Finally, the Physical Resource View in Fig. 6.9 corresponds to the lowest level of policy that governs the physical resources such as hardware, software, and the network. This type of policy determines the fulfillment preference, limitation of managed assets, access rules, recovery behaviors, and regulations for *Fault, Configuration, Accounting, Performance, and Security* (FCAPS) management. Each deployable asset will be governed by the network rule for that specific policy based on its upper level policies. The network policy also supports the QoS parameters defined in the SLA and OLA templates.

6.5.2 Design Considerations

The latest business trends in the IT industry focus on providing services to clients and providing the ability to access information and people that is not restricted by location or time. Such a feature relies upon scalable architectures for use in increasingly larger communities, as well as a new management framework that is far more configurable and compositional than before. The control and management functions in this management framework must possess the ability to expose policy to external systems or services in order to facilitate service collaboration and integration across the entire Cloud stack, not only vertically (with other vendors) but also horizontally (with other service partners or community users). The result of such pervasive policy integration at the infrastructure and platform levels can improve controllability over the provisioned Cloud environment as well as establish a foundation for supporting the upper-level business or mission objectives. Such improvement makes the integration of business polices feasible because the infrastructure and platform management policies are the base of the process for multiple level SLAs and enterprise-level management, as shown in Fig. 6.10.

As discussed in the previous chapters, Cloud technology relies on SOA to provide an open architecture for the efficiency of provisioning interfaces that allow easy alignment with their enterprise business processes and minimize architecture complexity. Applying the same principle to the PBM architecture can also greatly improve the solution efficiency, particularly in an environment where a *Federation of Systems* (FoS) is a must-have configuration. One of the most effective ways to facilitate an open policy architecture is to make the policy externalized. Externalization of policy can provide the opportunity for external policies (horizontal entities) or other internal policies (vertical entities) to participate in the decision making process and share control of an enterprise. The principles of adopting an external policy can be summarized in the following: [26]

- Because most IT businesses interoperate at the business process level rather than at the technology level, externalizing policies can increase business agility and simplify how businesses interoperate. As polices can be easily modified with minimal impact on system implementations, this can achieve higher service effectiveness.
- Using policy-driven business processes, all service participants in a CoI can now construct and manage their own solutions as well as influence other community members' solutions entirely based on externalized policies. It is a common practice for Cloud vendors to often use containers to deploy functionality.

These containers become a flexible vehicle to provide configuration, management, and business information to drive their values in the end-to-end value chain community through this permeable means. Once agreed with other parties, these policies can then be distributed to the designated parts of the Cloud environment.

- The separation of policy management from enterprises' internal process implementations enables the policy to be managed and harmonized explicitly and independently from other business functionalities. Technology and business negotiations (e.g., SLA negotiations) between different parties are no longer at risk for leaking business insights to the open market. Because this external policy has to be based on pre-agreed syntaxes and formats, there will be less ambiguity in the language of agreement.

6.5.3 Implementation Considerations

After seeing the value of the principles of policy management for Cloud environments, let us look at the generic steps that enterprises can take into consideration for designing, developing, and implementing their next generation PBM solution:

- *Functional specifications and implementation plan*: The first step of the project is to determine the mission and business objectives, with an association of the main service functionalities for the new PBM solution. As the new PBM solution will be supplemented by business-specific policies and processes, internal stockholders and external contributors must be identified and engaged. If COTS tools are considered, vendors and product evaluations must be part of the implementation plan. Functional specifications should address the basic principle of the adopted policy, policy controls, directory service authentication, security management, automated workflow, external interfaces, configuration change management, integration with other OSS and software tools, and the overall solution architecture.
- *Standard-based business, system, and network implementations*: Technical evaluations should include the adaptation of standards or best practices of the existing policy-related specifications. Most standards are implemented by vendors who recognize the values of these specifications. Adopting standards can make the solution more flexible in taking in new industrial developments through product upgrades and make the final solution more compatible with other vendors' products. For instance, enterprises can incorporate their enterprise IT process and information models with the TM Forum NGOSS specifications, where eTOM and SID models can provide a comprehensive framework with potential flexibility to integrate many products to satisfy the enterprises' business, system, and network requirements. Many service vendors who are familiar with the standards can quickly provide in-depth, value-added services to the enterprise to speed up the implementation.

- *Service integration and E2E collaboration*: A pilot implementation followed by a scoped deployment can help enterprises test the functionalities as well as the reactions from (provider and customer) community members. One critical consideration in this phase for the enterprise is its ability to ensure its information that resides in data repositories is secured and accessible. Secure accessibility can be accomplished by technologies such as the XML standards, SOAP, Java, J2EE, JNDI, JINI, and LDAP. For cross-domain policy management, the integration effort can deploy an enterprise service bus to facilitate effective event correlation, service registration, and attribute sharing. In a service environment requiring multiple providers, such as a value chain relationship, inter- and intra-organizational SLAs can be exercised to ensure end-to-end service metrics and regulations are commonly acceptable and can accurately reflect the aggregated service objectives at different layers of the service hierarchy.
- *External policy agents' deployment*: *Cloud Policy Extension Point (CPEP)*: The CPEP acts as an external policy agent to facilitate harmonization of policy requirements among policy processes. The deployment of such agents in a Cloud environment can assist policy to be exposed to the businesses' policies and corresponding infrastructural functionalities. The purpose of this capability is to allow appropriate behavior justifications to take place at the execution phase. Unlike PEP behaviors that are usually known in advance, CPEPs are bound to real-time conditions of businesses and infrastructural functionalities and offer a dynamic means to refine the execution of services.
- *Policy filter and forward capability*: Although roles and responsibilities are standard elements in a PBM architecture, they are mainly used to address system users and administrators. We learned the variety of participating entities in a Cloud Ecosystem in Chap. 1. Additionally, we also see Cloud resources must carry different identities in order to satisfy complex business operations where virtual-to-physical, virtual-to-virtual, and physical-to-physical relationships are interchangeable. For instance, for an enterprise to set forth a security policy to automate a group of networked devices to better respond to emergency situations, a predefined security filter and forward capability can assure important roles and information are delivered securely and accurately to the appropriate destinations. The concept of this filter and forward capability is similar to the defense-in-depth strategy in the cyber security world, where coordinated firewall configuration, network quarantine, and threat mitigation are built upon.
- *Automated network device configuration and change management capability*: Once the management framework is evaluated in a scoped environment, the next step is to ensure all managed resources can in fact accept the multi-tier policies across the entire Cloud environment. For new generation IT resources, most facilities and equipment are capable of supporting automated network configuration, security management, auditing and reporting, and activation and provisioning. For legacy resources, enterprises must evaluate their values for the end-to-end service and assess the effort to develop adopters or interface programs to perform compatible features.

- *Architecture conformance*: The values of Cloud services come from both the business and technical aspects. Although it relies on the form of technology to display tangible measures, the Cloud business model is indeed driven by the management architecture. For instance, the compliant PBM capability can serve as a means to increase service (e.g., network and applications) availability through consistency in configuration and reduction of human error. Because service automation is driven by PBM, enterprises can increase the level of service quality by a set of effective maintenance and management policies. Therefore, a good practice of architecture conformance can assist enterprises in maintaining their high degree of efficiency and effectiveness across disparate technologies.

As mentioned in the discussion in Sect. 6.3 about PBM standards, DEN-ng continues to evolve and the most current version is now independent from the TM Forum specifications. Because of its flexibility and goal of solving generic system problems, there are very few implementations available due to the lack of clear guidance from the specification to orient/drive this model. The newer model is now becoming more complex and requires more skills and experience to select the needed components for implementation. With respect to the version adopted by TM Forum, it can provide a decent starting point and rather solid foundation for developing practical policy for Cloud applications. However, this version lacks the focus on how to automate the process with a policy-based model, implicated in Fig. 6.2. This subject will be further discussed in the following sections. For true Cloud service management, the DEN-ng model can benefit from the additions that associate the existing policy continuum and process to the concept of virtualization more explicitly. Such an enhancement will help Cloud system and service developers appreciate the values of the model more clearly, thus helping facilitate the adaptation.

With respect to service federation from the PBM perspective, externalizing policy highlights a significant distinction between the traditional and service-oriented management paradigms, where most cross-domain issues must be worked on from outside in. In most legacy systems, policy is customized for specific application needs and is often times embedded in the application implementation. This not only prevents the flexibility of changes but also limits the ability for cross-system integration. Many middleware vendors intend to implement policy adjudication logic that can perform peer-level collaboration. Without a dynamic model like the one described in Sect. 6.5.1, the capability will be limited. We will have a more detailed discussion on this subject in Sect. 6.6.

6.5.4 Service Policy and SLA

For a Cloud service user, the service order and support management processes should be completely transparent. In an ideal situation, a user only needs to issue a service order to the provider and can expect a result of the business inquiry with a service confirmation and associated SLA. One of the revolutionary advantages of Cloud services is its ability to link the service with its customer's business aspects,

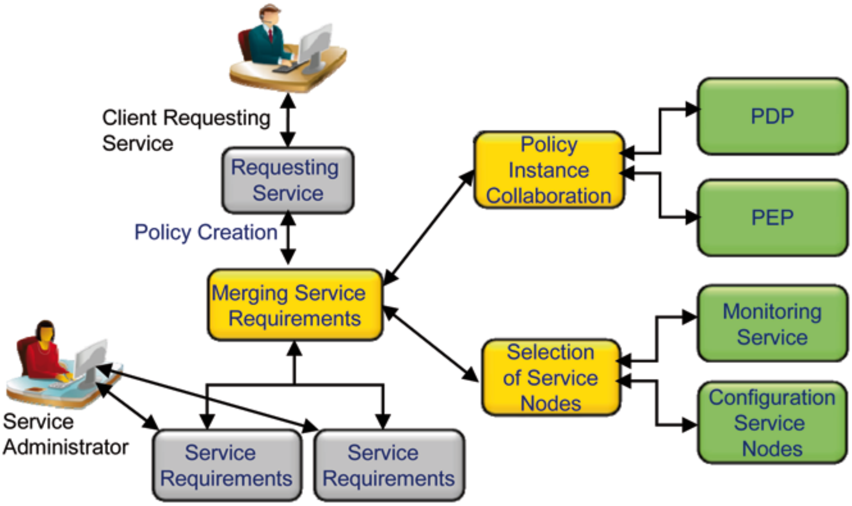

Client Requesting Service
Requesting Service
Policy Creation
Merging Service Requirements
Service Administrator
Service Requirements
Service Requirements
Policy Instance Collaboration
PDP
PEP
Selection of Service Nodes
Monitoring Service
Configuration Service Nodes

Fig. 6.11 Using policy to enforce the deployment and management of Cloud services

where expectations of service agility and flexibility in business operations can be justified in real-time and on-demand. Cloud services enables a new business model that can improve SLA dynamically with a customizable cost structure for small or large service customers to do their business faster, better, and cheaper. Using this new technology, enterprises are now capable of orchestrating small business partners into powerful federations so they can compete on a more global scale. Because of its automation nature, this new model enables suppliers and customers to facilitate unprecedented knowledge sharing among the CoI and turns the traditional transaction processing model into new value-added services [27].

Figure 6.11 illustrates a use case of Cloud service deployment and management. The Cloud service management process starts when an authorized user requests a service from the management system through the policy interface. At this time, the client will specify the name of the requested service and the QoS requirements for its deployment, if they are required. Once the management system receives the client request, it exchanges information related to the service in order to process the resources' service requirements.

In addition, Fig. 6.12 demonstrates a typical business-driven management framework in accordance with the PBM framework. Each contracted service relationship is modeled as a set of parties in which each party plays one or more roles to achieve the SLA objectives. Each SLA is associated with a set of SLOs to be achieved; as well as a set of intrinsic policies related to role behavior. Furthermore, a special engine (i.e., the role-to-policy mapping engine) translates roles, SLOs, and rules into a set of enabling policies. These policies are further refined to lower level specifics that enclose all the low-level logic required to correctly utilize resources. Business objectives affect the way SLAs are defined and managed at the service policy layer. Whenever a business objective is changed, added, or removed, an important impact takes place on the long term time scale of the SLA database [28].

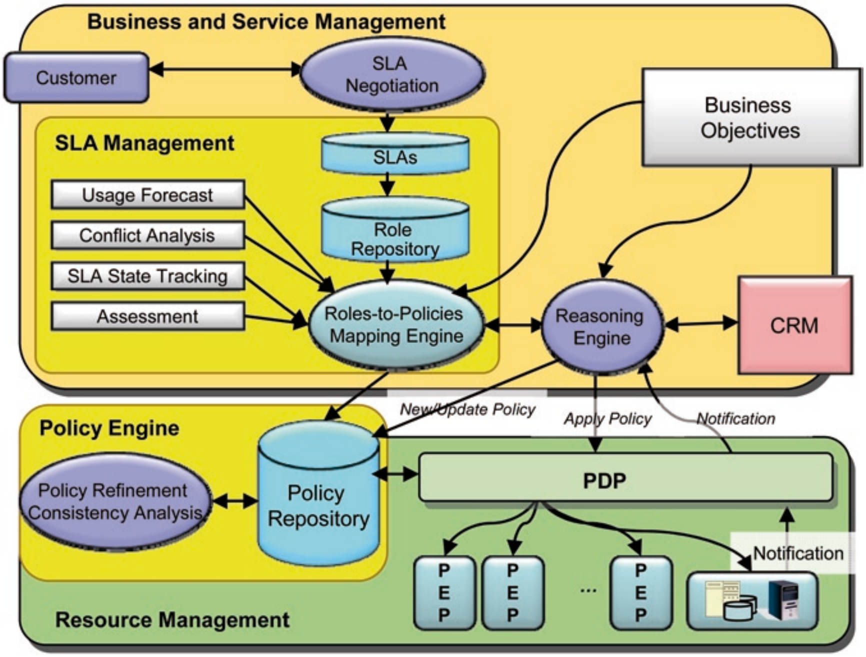


Fig. 6.12 PBM and SLA interactions in the business-driven management framework

6.5.5 *Service Policy and Resource Allocations*

In a pre-Cloud service flow, SCs normally have to express their requirements of resource usage using low-level primitives that are very difficult to change. With Cloud technology, providers can now handle these resource inquiries in a more dynamic way. No longer needing to hard-code service specifications, Cloud applications can coordinate and orchestrate Cloud resources at run time. For this to occur, two very essential Cloud features, management autonomy and fault-tolerance, are required. A successful implementation of these two features requires redundant resources (e.g., computation, databases, networks), an autonomous process, and a self-regulatory model to support the proper functionalities. All these rely upon a comprehensive PBM to seamlessly integrate these components together.

6.5.6 *Security in Cloud Policy Management*

Traditional IT security functionalities support the definitions of authentication and verification processes, encryption procedures (for instance, key/certificate management, encryption/decryption procedures), user profiles, user roles, and firewall configuration. The policy engine to manage the above features can be deployed as a single device that handles every policy action, or a set of servers that work together to process policy actions. In this SoS approach, service users must communicate with a policy server in order to obtain permissions for accessing a resource. When security rules and policies are applicable across different providers or organizations, the policy engines of these providers must deal with other policy engines separated by enterprise firewalls. Today, most enterprises choose to deal with these policy collaborations manually because of general security considerations and a lack of mature technology to protect the organization's internal information.

In a primitive security management environment, security policy collaboration through human interaction is feasible and sufficient. However, security requirements in Cloud environments increase the operational and informational sophistication to a whole new level. These requirements influence the management of IT resource operations, IT SPs' interactions with external actors in CoIs, and service customers' behaviors in relation to the service offerings. They impact the baseline architecture of the service framework, service usage patterns, application regulations, service monitoring capability, and accessibility of users or user groups. To appreciate the potential complexity of the relationship description among Cloud users, customers, providers, processes, and procedures, let us look at the scenario below. To satisfy a community-based Hybrid Cloud, automated security management (e.g., provision, monitoring, compliance) demands appropriate correlation between the above managing and managed identities with the policy stack illustrated in Fig. 6.10 of Sect. 6.5.1. This mapping also needs to be extended to incorporate the policy domains, policy subjects, and policy targets that are consistent with the business objectives of the Cloud applications and sensitive to technical efficiency. For instance, the organizational directives and corresponding policy rules that associate with the management of both internal and external SLAs over various SaaS, IaaS, and PaaS resources, add another security variable to the enterprise business and technology operations [29].

Using SOA technology, the security management functions can at certain degrees detached from the main management architecture. Along with an externalized policy management configuration, enterprises can simplify their management efforts to deal with other internal management processes as well as with external third-party policy components. One key ingredient to a successful integration depends upon a unified information and process model consistent across all security management elements in the entire CoI. With a common model, all policy servers can perform the best judgment and actions with respect to their domain and end-to-end security policies and rules. There is a fundamental difference between the SOA-based distributed management architecture and some central security architectures, where a trustworthy server node performs global security policy coordination from a single location. The central control method creates potential security threats to all participating systems because of its vulnerability of attacks for any targeted systems in the community. Any functional or performance degradation of the security controller or engine can become a single-point of failure for the entire community. On the other hand, distributed security policy engines can achieve functional duplications with a technique that will be discussed in the following sections and decrease impacts from internal or external security attacks. Using the new solution, policies, certificates, and keys are delivered in advance and can be dynamically reconfigured based on new business needs. Global security verifications (e.g., remote testing), constant compliance audits, and service refinements can all be automated. Policy negotiation will be discussed in the following sections, further security defense and management will be illustrated in Chap. 9.

6.6 Externalizing Policy and Management

The new Web applications open a new frontier for service clients to access unlimited objects in the form of documents, videos, digital images, and music through Cloud services. With Cloud technology, enterprises can link data from hundreds of locations to serve millions of users concurrently around the world. This phenomenon leads to new market requirements for information management on a global scale [30].

In a fully virtualized Cloud environment, services are offered by a group of federated providers (e.g., enterprises) as a pool of virtual IT resources that are independent from their underlying physical IT infrastructure. These services are operated by different vendors but are collaborated with common process and information models in order to appear to their clients as a harmonized collection of service offerings.

Figure 6.13 depicts a system architecture scenario where scaling policies can be tedious and difficult to manage. In this scenario, business policies exist in multiple

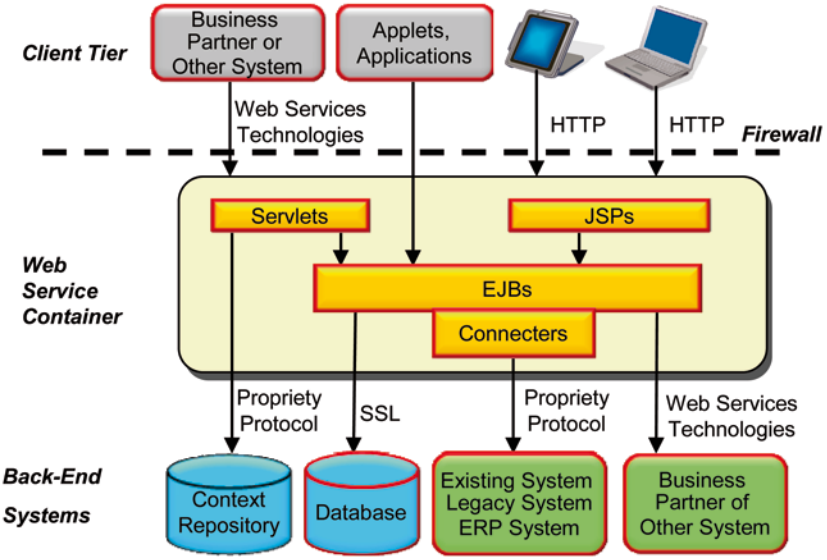


Fig. 6.13 Scattered business policies over multiple places

locations of the serving environment. The functional areas relevant to PBM are outlined in red. Due to this scattered system distribution, policies have to be distributed over a large and diversified environment with the potential for duplications. Deployed policies can be out of synch overtime and result in content inconsistency after several rounds of system upgrades or configuration changes. An appropriate PBM infrastructure is a key to make such seamless integration possible in terms of provisioning and management.

From a customer relationship perspective, regardless of which client issues a service request, the Cloud environment must be able to quickly identify the client's profile and invoke the requested service efficiently. Keep in mind that invoking a Cloud service even through other providers does not require the action of Cloud composition. Instead, the affiliated SPs must work together to provision, manage, and govern the demanded resource with unified policies, even across different Clouds. Such unified policies can assist the collected SPs in the same value chain to support the same management and control interfaces automatically. This feature allows providers to quickly construct a composite service offering that is operated by different Cloud vendors or to disassemble collected services and delist offerings from the service catalog in real-time.

The externalization of policy provides a flexible management framework that makes the above scenarios feasible. Furthermore, external policy architecture can work with any existing enterprises' SOA solution more effectively. It can support composition of Cloud services and manage policy compliance more effectively. As shown in Fig. 6.14, the business services and utility computing (or infrastructure resources) are gathered in a virtualized service pool. The only interface to the ESB is the public service portal, where a user can order services directly from the pool or through the composite service portal, where additional service federations are needed. Based on the SOA paradigm, the external management entities can harmonize a service bundle from different providers or vendors by incorporating other service features (e.g., security or management features), whether they are inside or outside the existing pool. Figure 6.14 shows a logical view of a Cloud service architecture where service (and network) policy is completely externalized. All management processes and tasks are potentially federated and controlled under the policy management services. It is important to mention that both business and service infrastructure policies must be completely detached from the central management functions in order to gain the best performance in service harmonization. Once the policy of business functionality is also externalized, the enterprise can achieve a fully compartmentalized, end-to-end, SOA-based architecture [31].

6.6.1 Policy Negotiation

Policy within or across organizational boundaries has traditionally been embedded in IT platforms and applications. Cloud-based services require providers to scale their businesses globally, thus demanding new ways to collaborate and harmonize

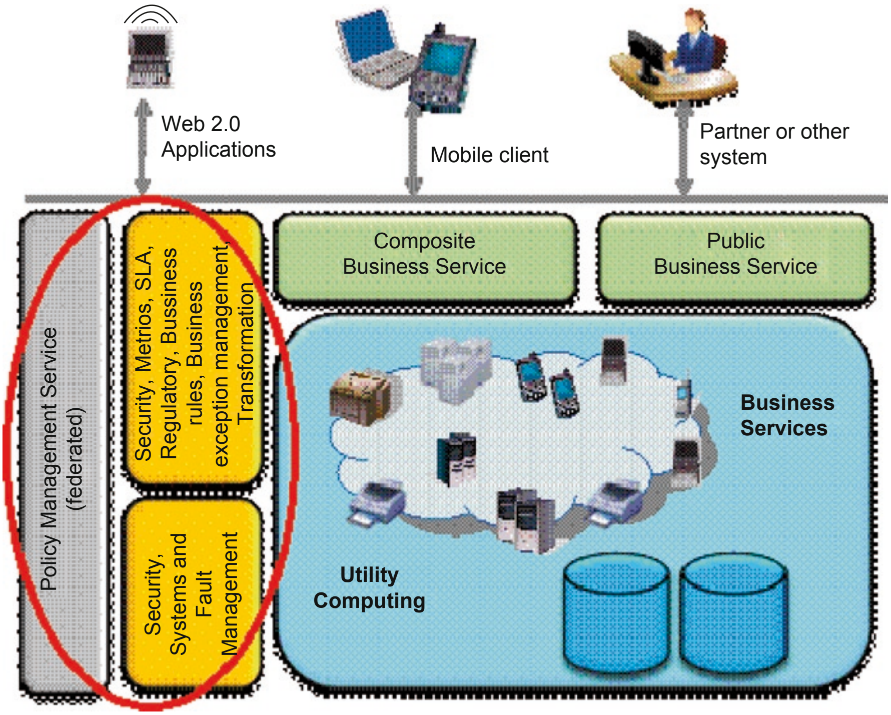


Fig. 6.14 Aggregated PBM functionalities in one unit

policies within and across external process networks and value chains. This type of framework management relies on distributed capabilities to enable technology-neutral, vendor-independent, policy enforcement and execution functions across multiple operational or security domains. The PBM described in this chapter aims to provide a new way of thinking of coordination and automation requirements. It includes necessary implementation details of clear and explicit definitions on governance, policy (regulatory, security, privacy, etc.), and SLAs when dealing with diverse entities in the Cloud.

The process of negotiation between SC and SP produces a contract that captures and formalizes the agreement between the parties. The contract embeds the rules for the interaction between SC and SP. As it happens in normal business contexts, service delivery follows contract acceptance by both parties. Policy negotiation is the process of determining the most appropriate communication policy that all of the parties involved can agree on. The core problem here is how to reconcile the various (and possibly conflicting) Cloud management protocols used by different enterprises. Current policy negotiation approaches focus on limited Web services, client-server capability negotiation.

Developing a practical methodology to compose policies that support cross-organizational cooperation of different policy sources is a difficult and long-standing challenge. It includes, for instance, the difficulty of dealing with inconsistencies of different policy focuses and features. A set of simple principles that can support desirable policy negotiation methodology should have the following characteristics:

- Simple and intuitive to policy designers.
- Encompass a formal foundation that allows careful reasoning about the correctness of algorithms. It also needs to include consequences of composition in boundary scenarios where systems may be vulnerable to security attackers.
- Enforcement should be efficiently implementable based on open interface or industrial standards.

Figure 6.15 depicts a typical policy negotiation scenario where human intervention of a third-party is often required. This figure shows how policy negotiation, combination, de-confliction, distribution, and execution take place, where the human is in the center of the process. Although human interaction can assist in understanding and resolving discrepancies between two Cloud environments, the step two process still requires some form of protocol to appreciate the policy specifications from the parties involved. These protocols must be commonly available, the "best" results depend on whether the information is complete and compatible between these two Clouds.

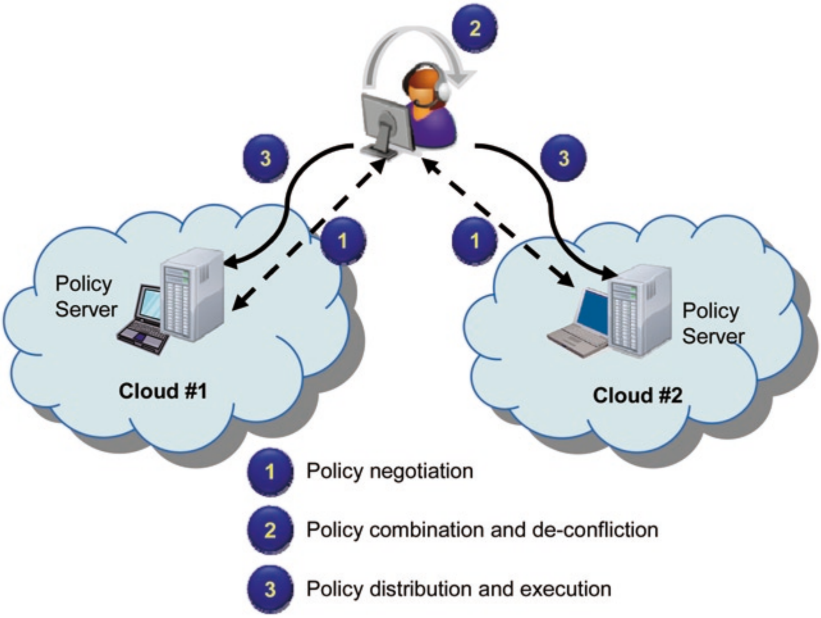

2
3
3
1
1
Policy Server
Policy Server
Cloud #1
Cloud #2
1 Policy negotiation
2 Policy combination and de-confliction
3 Policy distribution and execution

Fig. 6.15 Traditional policy negotiation

6.6.2 Automated Policy Negotiation

A key in automatically collaborating a joint service from different SPs in Cloud environments is the ability to negotiate policies among providers with a set of processes and technology. The negotiation effort across a global Cloud infrastructure must be performed without any business or security compromise. This is because different organizations have different goals. Dynamic federations need to operate within the constraints of potentially changing groups of goals, participants, and service features. All these changes must take place without interfering with the continuation of the services. Although automatic policy negotiation is an intractable problem, efficient policy negotiation methods have been suggested for some classes of policies. For example, policies are represented in de-feasible logic and composition is based on rules for non-monotonic inference. Here, a policy writer constructs *meta-policies* describing a set of policy as well as associated annotations for their composition preferences. Meta-policies are specified in de-feasible logic, a computationally efficient and non-monotonic logic that is developed to model human reasoning. These annotations indicate whether the specified policy assertions are required or if they allow other assertions to take precedence when certain circumstances occur. This implementation presents a sound method that can perform effective coordination to automate negotiation of Cloud management policies [32–34].

Figure 6.16 provides an example process of a general policy negotiation algorithm which allows dynamic authorization and QoS agreements between negotiating partners. To achieve the most effective dynamic (automated) Cloud management capability, a number of important capabilities must be present:

1. *Computer readable infrastructure description*: The PBM system must be able to understand the infrastructure in order to dynamically manage resources in other organizations. Service infrastructure information varies widely between organizations whether they are virtual or physical. The information can range from a configuration database in a service or resource management system to a Visio drawing for service topology. So far, the IT industry has not agreed on the representation of application and network infrastructure, thus automatic cross-organizational service integration is not yet feasible. Moreover, the representation must accommodate individual providers' privacy requirements, so sensitive information in these infrastructures can be hidden from others.
2. *Real-time policy negotiation*: Today's client-server service negotiation and peer-to-peer trust negotiation are insufficient to accomplish the needed dynamic policy negotiation for Cloud services. The challenges of managing the complex Cloud service relationships in dynamic environments have been discussed in the previous sections. To satisfy the constraints of the dynamic group of organizations, a mechanism to achieve true peer-to-peer policy negotiation in real-time must be developed [35, 36].
3. *Capture and reuse domain knowledge*: SPs in Cloud ecosystems often have their own specific business processes. These processes are purposely preserved as their business differentiators, and thus should not be considered as candidates

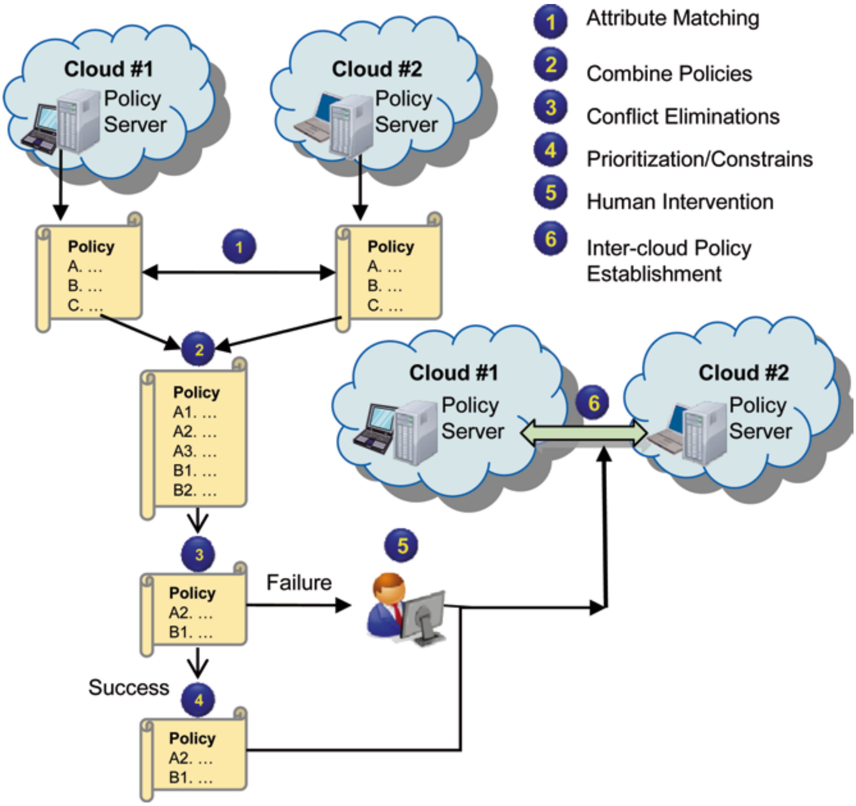


Fig. 6.16 Dynamic policy negotiation process

for integration. To achieve effective cross-organizational federation, the PBM solution needs to ensure these business processes are acceptable to all federated service partners. This PBM should possess an ability to capture and dynamically reuse such domain-specific knowledge or processes as necessary.

4. *Interpret and forward the intent*: Each local policy engine must be able to fully understand the policy intent assigned to them. If the terms or conditions from other policy engines derive a conclusion that other policy management elements are needed, either locally or remotely, the policy engine should span distributed ranges to reach these elements that belong to different Clouds. This includes an appropriate interpretation of the original policy intent and additional security features to handle cross-firewall data exchanges.

One implementation of the dynamic policy negotiation is the NASA Jet Propulsion Laboratory's PBM software solution called *Policy-based Adaptive Network and Security Management* (PAM). Implemented for space and defense applications, this system established a revolutionary concept for automated infrastructure management between federated partners (e.g., organizations) using open interfaces. In a typical service environment involving different SPs, the provisioning process normally requires a cycle of configuration, testing, and evaluation processes. Therefore, when any providers in the value-chain are not capable of supporting the committed levels of QoS, the primary provider can repeat this cycle until a satisfactory point is achieved. For NASA, PAM provides a practical approach that allows different operators to integrate and evaluate their network resources while maintaining their own controls of the resources. Federated operators use an open source language named *Integration Markup Language* (IML) for information exchanges and policy negotiations.

Facilitated by a dynamic decision-making engine, PAM enables automated E2E measurement, testing, and corrective actions. Supporting this functionality is a non-implicated service definition between operators so the primary operator can quickly identify the location and nature of the problem. In cases when the problem description is not obvious and further analysis is needed, IML service information can provide service topology and key service attributes to assist the fault and performance management systems with their root-cause analysis in order to precisely determine the impact areas. In the center of PAM's policy engine is a functional component called *De-feasible Policy Composition* (DPC). DPC integrates and correlates policies from two systems (operated by two operators) needing to be combined, and produces a consolidated specification that describes the means of communication agreeable to both systems. The file format of these consolidated policies is based on an XML-based language called RuleML. These new policy files are then parsed into three objects: *facts*, *rules*, and *priority* relations. Certain restriction rules are treated as requirements to be fulfilled, all other rules, such as de-feasible and defeater rules, are treated as Reasoning rules [37].

6.6.3 Policy Adaptation

In a pre-Cloud IT service environment, providers specify SLAs that include service categories, service coverage, and QoS metrics such as delay, throughput, error rates, and availability. These are part of the service order management. The majority of these specifications are essentially static and providers often require provisioning a single type of service at a time. Like we discussed about the new Cloud environments, service clients are given an expectation that services are virtual and ad-hoc, thus they can be added and removed dynamically. This demands the Cloud-based policy management function to be flexible in order to adopt any order changes. In addition, a "fallback" QoS specification must be part of the new active service catalog or features in SLA management to handle failure conditions when meeting the primary QoS is not feasible. This is particularly important in a virtual environment where the upper level SPs normally have very limited knowledge of the underlying physical or logical infrastructure. Therefore, service adaptation can take place either as a result of service failures or changes from clients' application requirements. For example, an SLA may ask its SP to downgrade the primary video service to audio services when the image quality cannot keep up at compatible HDTV resolution [38].

From a provider's perspective, BSS or OSS must react to modifications of existing configurations from service orders by mapping the changing policy to corresponding service configurations. Similar requirements for policy adaptation may exist in different applications of PBM systems. For instance, policy may have a need to adapt changes in firewall or router packet filtering rules to respond to certain network events such as denial-of-service attacks. Adaptation of a ubiquitous computing environment is another case of policy adaptation where a user enters a new location or enters an attribute to the service that can trigger new behaviors of the service offerings. For device and system manufacturers, the management system must be able to interact with the supporting devices or systems to verify if the units support the required functionality and have the necessary resources to carry out the committed services. If there is any known restriction or limitation, there must be a way to mitigate impacts to the end-to-end service relationship by using an alternative service or environment.

Figure 6.17 depicts an application of policy adaptation, individual service elements at the bottom of the figure build up to large organizations to reflect the needs of unstructured service assets. Such integrated service applications range from

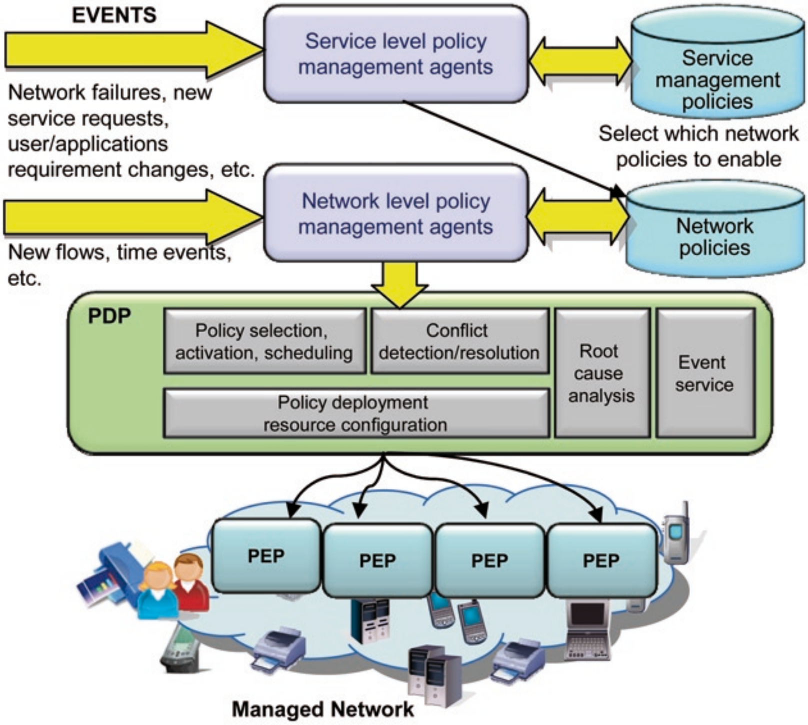

EVENTS
Network failures, new service requests, user/applications requirement changes, etc.
New flows, time events, etc.
Service level policy management agents
Network level policy management agents
Service management policies
Select which network policies to enable
Network policies
PDP
Policy selection, activation, scheduling
Conflict detection/resolution
Root cause analysis
Event service
Policy deployment resource configuration
PEP
PEP
PEP
PEP
Managed Network

Fig. 6.17 Service management policies for policy adaptation

digital photos to videos to mobile content of all kinds. The Cloud service model provides participating organizations with an efficient infrastructure that leverages many highly distributed resources and acts as a single, local entity trough a distributed PDP and PEP management infrastructure. Its information framework helps expose the service and network values' availability from information mobility through an any-to-any architecture.

Both the service and network levels of management are leveraged by this architecture to gain a highly flexible and granular policy management function for significant competitive advantage. Functional modules in the PDP process portray some key functionalities that help facilitate policy adaptations. Dynamic policy adaptation for modifying service behaviors can take place in the following three levels of action: [38]

- Learning and identifying the most suitable policy configuration from the existing system and service behavior. The newly adapted configuration can be incorporated into the OSS or BSS to update management strategies. For instance, adding new policies that react to different events or updating new versions of policies to perform new actions on the managed objects. This is particularly important for Cloud-based PBM in selecting appropriate, real-time policies or generating new ones when needed.
- Selecting and controlling policies from a set of pre-defined policy databases. Policies at the service level are triggered by events that indicate a need of change in the managed environment; the policy engine determines which lower-level policy must be invoked (either enabled or disabled) to adapt the configuration. The advantage of this feature is the flexibility to manipulate management strategies from a top-down approach, allowing Cloud service designers to install service management policy with a more effective, streamline method, as shown in Fig. 6.17.
- Changing the parameters of management policies dynamically. Appropriate attribute values or benchmarks are specified at the run-time for configuring the desired behaviors of the managed objects. Using programmable policy parameters, new actions may be updated via a management interface without having to change the policy rules. This way, policy behaviors can be updated dynamically while the policy organization and logical flows remain intact in the policy repository. The complexity of programming policy behaviors can be dramatically reduced when new actions or new calculation methods are loaded at run-time either through the management console or by new policy rules.

For instance, a Cloud-based storage service enables enterprises to leverage different tiers of storage (disk and tape) applications. These products provide great flexibility and allow archived data to be accessed through either a Private or Hybrid Cloud, even if it is stored on tape media. Using the dynamic policy adaptation feature of PBM, the service can automatically move less active information to more cost-effective storage systems to improve system efficiency. The service designer can define policies to determine how information is distributed and handled on a global basis. For example, information that is current and valuable may be defined as premium, and therefore require more copies in more locations than information that is older and accessed less frequently. Older information may be compressed and retained with fewer copies in fewer locations. This feature can be treated as PaaS, so Cloud service developers can build applications that offer secure online services, pay-for-use, and other models. When used as IaaS, enterprises can use this feature for global distribution, management, and retention of digital media assets [39].

6.7 Conclusion

PBM is not a new concept in the telecommunications industry. Network operators have been using PBM to improve certain degrees of efficiency in network operations such as configuration and trouble ticket reporting. A more sophisticated implementation concerning policy-driven management for better QoS is a rather new trend. Before being driven by the Cloud service model's openness, most SPs were essentially domain-centric. In other words, service environments for providers are self-sufficient because all their resources are controlled and managed by a central OSS, inter-provider iterations are typically done manually due to very little business dependency with other SPs.

The new generation of Web services opens up a whole new business relationship that drives industry players to act more closely together. Whether they are business allies or competitors, service clients can access their service portals equally. Therefore, the concept of market domination no longer has the same meaning. We have seen that standard bodies are proactively addressing Cloud issues, as discussed in Chaps. 2 and 3, but the improvement to PBM for Cloud-specific business is still slowly catching up. Section 6.3 discussed a good progress in two distinguished organizations' efforts. Fundamentally, these existing PBM specifications are useful because their well-defined rules have been deployed in the IT industry for some time. These PBM architectures are also integrated with many enterprises' management systems. However, issues such as the design of the policy-based directory, Cloud QoS standards, and cross-provider negotiation protocols are still under development. Although policy information models such as the ones from the IETF and the DMTF are available and capable of touching upon some issues, they are often found difficult to implement. This is because of their overly complicated data relationships for low-level resources, and lack of corresponding information to make appropriate linkages to specific business processes and relationships. This barrier prevents vendors in the industry from designing or developing products that can sponsor the concept of high degrees of automation and interoperability for Cloud services. From Sects. 6.5 to 6.6, the authors intended to introduce some new perspectives for Cloud management based on the PBM framework mentioned in the earlier sections. Through the discussion of the Cloud policy stack, external policy architecture, and cross-domain policy negotiation, we are hoping these subjects can become useful references for assisting in the future development of standard specifications.

References

1. Private Cloud Computing for enterprises: meet the demands of high utilization and rapid change. Cisco Systems. http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns836/ns976/white_paper_c11–543729.html
2. Datar, K.: Cloud the future of computing: adoption of Cloud Computing in Indian market. Cisco WebEx. 2008. http://www.telecomindiaonline.com/telecom-station-cloud-the-future-of-computing-kiran-datar-webex.html
3. Cole, A.: Cloud storage: blending the old with the new. IT Business Edge. Dec 2009
4. Winans, T.B., Brown, J.S.: Cloud Computing, a collection of working papers. Deloitte. May 2009
5. Information, Computer and Network Security Terms Glossary and Dictionary. http://www.networkdictionary.com/security/p.php
6. Common Business Entity Definitions—Policy, NGOSS Release 4.0, GB922 Addendum 1-POL. TM Forum. Aug 2004
7. Strassner, J.: Policy-based network management: solutions for the next generation, Elsevier, 2004
8. Strassner, J.: Policy Based Network Management. Morgan Kaufman Publishing, San Francisco (2003). ISBN: 1–55860-859–1
9. NGOSS Architecture Technology Neutral Specification—Policy Management, TMF053P. TM Forum, Q4 (2003)
10. IETF Policy Working Group. http://www.ietf.org/html.charters/policy-charter.html
11. Snir, Y., Ramberg, Y., Strassner, J., Cohen, R.: Policy framework QoS information model. Internet Draft. draft-ietf-policy-qos-info-model-03.txt. April 2001
12. Moore, B., Ellesson, E., Strassner, J., Westerinen, A.: Policy core information model—version 1 specification, RFC 3060. Feb 2001
13. Strassner, J.: Directory Enabled Networks, Chapter 10. Macmillan Technical Publishing, Indianapolis (1999)
14. COPS QoS Policy Manager. http://www.cisco.com
15. HP Policy Expert. Cisco Systems Inc. http://www.openview.hp.com/products/pxpert/index.html
16. Allot Communications NetPolicy Policy Based Management System product documentation. http://www.allot.com
17. Cloud solution: master your dynamic service supply chain. CA. http://www.ca.com/us/cloud-computing.aspx?WT.svl=header_link
18. WebSphere Extended Deployment. IBM. http://www-01.ibm.com/software/webservers/appserv/extend/
19. Bergin, S.: IBM Tivoli—security solutions for the Cloud. IBM. http://www.slideshare.net/IBMNZ/ibm-tivoli-security-solutions-for-the-cloud
20. IBM Smart Business Storage Cloud. IBM. http://www.sbgadget.com/182/ibm-smart-business-storage-cloud.html
21. Lymberopoulos, L.A.: An adaptive policy based framework for network management. Dissertation, Dept. of Computing, Imperial College London, University of London (Oct 2004)
22. Kumar, N., Kumar, P., Jiang, X., Kowtha, S., Chow, E., Chang, H.P., James, M., Freides, D., MacArthur, G., Mayer, A.: Applying policy based enterprise management towards realization of disn real-time services. MILCOM, San Diego (2008)
23. Strassner, J.C.: Policy-Based Network Management, Solutions For The Next Generation. Morgan Kaufmann, San Francisco (2003)
24. Information framework concepts and principles, TMF GB922, Release 9.0. TM Forum. 1 April 2010
25. Winans, T.B., Brown, J.S.: Demystifying Clouds, exploring Cloud and service grid architectures. Deloitte. April 2009

26. Winans, T.B, Brown, J.S.: Motivation to leverage Cloud and service grid technologies. Deloitte. May 2009
27. Cloud services: technology evolution or business revolution? Deloitte. 2009. http://www.deloitte.com/assets/Dcom-UnitedStates/Local%20Assets/Documents/us_consulting_debates_CloudServices_170909.pdf
28. Aib, I., Sallé, M., Bartolini, C., Boulmakoul, A., Boutaba, R., Pujolle, G.: Business aware policy based management. Business-Driven IT Management (BDIM), IEEE Communications Society, Vancouver (2006)
29. Mai, C.: Policy based management system. Aeronautical Telecommunication Network (ATN) Seminar, International Civil Aviation Organization and Aeronautical Radio of Thailand Limited, Thailand, 11–14 Dec 2001
30. McGaughey, K.: EMC delivers policy-based information management solution for building Cloud storage infrastructures. EMC, Nov, 2008. http://www.emc.com/about/news/press/2008/20081110–01.htm
31. Winans, T.B., Brown, J.S.: Moving information technology platforms to the Clouds. Deloitte. May 2009
32. Gong, L., Qian, X.: The complexity and computability of secure interoperation. Proceedings of the IEEE Symposium on Security and Privacy, IEEE Computer Society, Washington (1994), pp. 190–200
33. McDaniel, P., Prakash, A.: Methods and limitations of security policy reconciliation. ACM Trans. Info. Sys. Secur. **9**(3), 259–291 (2006)
34. Lee, A., Boyer, J., Olson, L., Gunter, C.: Defeasible security policy composition for web services. Proceedings of the 4th ACM workshop on Formal Methods in Security, ACM, New York (2006), pp. 45–54
35. Cheng, V., Hung, P., Chiu, D.: Enabling web services policy negotiation with privacy preserved using XACML. Proceedings of the 40th Hawaii International Conference on System Sciences, IEEE Computer Society, Washington (2007)
36. Li, J., Zhang, D., Huai, J., Xu, J.: Context-aware trust negotiation in peer-to-peer service collaborations. Peer-to-Peer Netw. Appl. **2**(2), 164–177 (March 2009)
37. Chow, E., James, M., Chang, H.-P., Vatan, F., Sudhir, G.: An Intelligent network for federated testing of NetCentric systems. NASA Jet Propulsion Laboratory California Institute of Technology, California
38. Lymberopoulos, L., Lupu, E., Sloman, M.: An adaptive policy-based framework for network services management. J. Netw. Sys. Manag. **11**(3), 277–303 (Sept 2003)
39. Atmos – Multi-tenant, distributed Cloud storage for unstructured content, 2010. http://www.emc.com/products/detail/software/atmos.htm

Chapter 7
Building and Configuring Enterprise Cloud Services$_{3,2}$

Advanced enterprises must stay competitive by continuing to invest in compelling and attractive new products for the market. These new business cases must be justifiable and set apart from their traditional cost system to adopt the utility pricing model common in Cloud services. Although cost and pricing are two significant financial challenges that SPs must deal with in terms of customers' experience and expectations, they are not alone. The pressing issues on the technology front also cause unavoidable impacts to SPs, they include the effect of SOA strategies, impact of disaster recovery plans, data management policies, and risk profiles for mitigation strategies that are common to typical enterprise processes. Depending upon the market size of the enterprises, the complexity level of their management systems varies. However, regardless of the amount of money enterprises invest in these management systems, they must be able to answer challenges from service levels, privacy matters, compliances, data ownership, and data mobility in order to fully participate in a Cloud ecosystem. Enterprises that are heavily technology-dependent must also be sensitive to the timing for introducing new methodologies or technologies to their target markets. This can avoid unproductive results caused by either a premature or late deployment of a new or updated service.

Building and configuring Cloud services is a challenge to all participating providers and vendors in a Cloud value-chain environment, due to a Cloud's dynamic nature and variety of service categories. Service configurations for a Private Cloud, Public Cloud, Community Cloud, or Hybrid Cloud can take different design philosophies and implementation directions. Areas that are impacted by configuration management include: which services can reside in the Cloud and which should be present internally within a business; if enterprise data can be approached by someone else from within or outside of the Cloud without one's knowledge or approval; keeping consistent levels of compliance for data stored in the Cloud; what the state of community sharable data is after a Cloud relationship is terminated; and protection of data ownership and mobility. Hidden costs, such as management, governance, and transition costs, are also factors in how enterprises configure their Cloud services. These are the questions we plan to tackle and answer in this chapter.

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_7, © Springer Science+Business Media B.V. 2010

7.1 Overview

Aside from the variety of service and technology categories a SP can design and implement for its customers, there are many supporting functionalities that the provider needs in order to execute, deploy, and manage these services. Among many management processes and procedures, planning and fulfillment are two important initial stages of a service deployment. These involve many management functionalities such as product lifecycle management, product strategy/proposition management, resource strategy management, resource domain planning, SLA planning, inventory management, order management, and provisioning management. Following these two stages are the assurance and billing processes. The assurance process can be decomposed into performance, fault, test, SLA, and inventory management at the product, service, and resource levels. For the billing process, the management areas include asset management, bill calculation, bill format/render, billing account management, billing data mediation, and finally the billing inquiry dispute and adjustment management. These functionalities can be implemented as an integrated system in the form of a SoS, or a Cloud-based management solution in the form of a FoS. Regardless of these two options, most large enterprises are comfortable in dealing with their own service deployments in most recent implementations. This is due to the full controllability of the environment. In other words, even with Cloud technologies, the option of using a Private Cloud architecture in their datacenters has no fundamental difference from a SoS implementation. However, when enterprises step outside their operational domains and interact with service or business management components from external providers via public or Hybrid Clouds, the rules of business process change.

To take advantage of new business and technology offerings from the open market to reduce cost and increase competitive edge, enterprises are beginning to construct their process framework differently than their current SoS model. With this new mindset, the enterprise process framework proliferating with external providers exhibits a new trend of a giant packaging process within/outside the enterprise business chains. The challenge now is how to build an effective solution to link resources to the value-chain process so the offerings can be accessible to the target clients, while the managing framework remains consolidated, virtualized, and automated. Furthermore, managing the existing or legacy services must be incorporated. For instance, configuration changes to existing computer resources such as storage, devices, switches, or routers may need to be processed manually. When moving to virtual platforms or infrastructure, manipulating the configurations of these assets can become an execution nightmare for operators. With the new technology and process that will be discussed in this chapter, automation and orchestration can happen in the Cloud in real time. There will no longer be a need for engineers to roll trucks out for making changes across all the routing, switching, and firewall platforms in response to change orders [1]. For instance, configuration management records and updates service information that describes an enterprise's computer systems and networks, including all hardware and software components. Such information typically includes the versions and updates that have been applied to

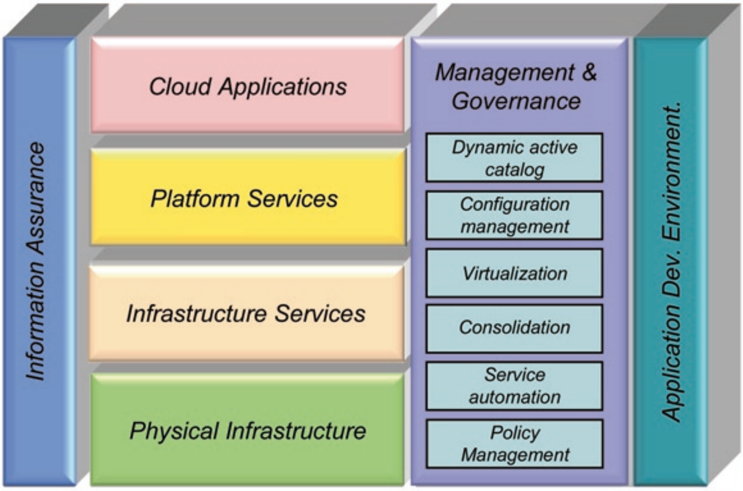
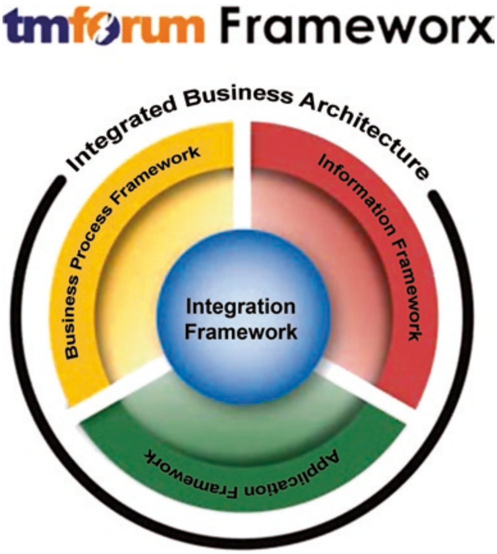


Fig. 7.1 Service planning and configuration in Cloud architecture model

installed software packages and the locations and network addresses of hardware devices. Advanced configuration management assists operators in the Cloud management environment in accessing and manipulating specific resources (virtual or not) in the entire collection of services, systems, and business. The same management system also possesses the knowledge and ability to ensure changes to any resources that do not adversely affect any of the other services, systems, or business. Figure 7.1 portrays the functional areas that impact the Cloud architecture model we introduced in Chap. 1. It includes dynamic active catalog, configuration management, virtualization, consolidation, service automation, and policy management. With these technologies, and in an evolutionary management manner, service clients can achieve a high degree of flexibility and dynamics to manage the service products they purchase from the enterprise [2, 3].

The ongoing innovation and development of new additions to the Cloud ecosystem expedite the implementation of the componentized infrastructure based on standardized mechanisms. An ideal case of adopting standard specifications is to unify the process of managing services such as provisioning, monitoring, security, integration, and deployment. In this chapter, we will use TM Forum as well as ITIL models to illustrate how a set of best practice guidance can formulate a sound foundation to answer these new challenges. As first seen in Chap. 1, TM Forum's Frameworx is an integrated business architecture that provides a service-oriented approach for rationalizing operational IT, processes, and systems. It enables SPs to significantly reduce their operational costs and improve business agility. Frameworx uses standard, reusable, generic blocks called Platforms and Business Services. These services leverage industry concepts (e.g., SOA and ITIL), allowing SPs to assemble new services using standardized methods while providing the flexibility of customization. Frameworx is an enabler for SPs to realize ITIL-compliant

Fig. 7.2 TM Forum Frameworkx



implementations through the Business Process Framework, and is an Integrated Architecture Built on NGOSS. There are four major components in this framework, their relationship is shown in Fig. 7.2:

- *Business Process Framework (eTOM)*: The industry's common process architecture for both business and functional processes
- *Information Framework (SID)*: A common reference model for Enterprise information that SPs, software providers, and integrators use to describe management information
- *Application Framework (Telecoms Application Map* or *TAM)*: A common language between SPs and their suppliers to describe systems and their functions, as well as a common way of grouping them
- *Integration Framework*: A service-oriented integration approach with standardized interfaces and support tools

In the following sections, we will lay down business considerations and justifications for different implementation options and their associated supporting infrastructures. With the introduction of some key standards for enterprise processes and information frameworks, we wish to bring forth management guidance and configuration recommendations that are practical and useful for an enterprise's reference.

7.2 Design Principles and Deployment Options

The purpose of this section is to share our thoughts about the design patterns for a new generation of applications that are referred to as Cloud services. In this section, we will provide architectural considerations and patterns as they affect common architectural domains such as enterprise, software, and infrastructure architecture.

Cloud delivery models, such as SaaS, PaaS, and Iaas, are discussed in Chaps. 3 and 5. Cloud deployment models, such as Private Clouds and Public Clouds, are discussed in Chap. 3.

7.2.1 Service Automation

In order to satisfy the requirements of the next generation of computing, Cloud environments will need to be more than just externalized datacenters and hosting models. Rather, they will need to have fully autonomic virtual organizational computing. In addition to the already-established characteristics of Cloud environments today, e.g., uniquely identifiable, dynamically configurable, alignment of business constraints with infrastructure constraints, etc., the fully autonomic virtual organizational computing architecture would be able to address global-scaled collaboration and partner network problems in all aspects. To be more specific, it will be necessary to set up collaboration networks rapidly to enable secure interactions online, where interactions could imply interoperability with back office systems as well as human-oriented exchanges. In addition, it will be necessary to provision and uniformly manage composite Cloud services since business interactions will be much more complex than personal transactions. Finally, it will be necessary to implement relevant adequate access control management since roles and responsibilities will potentially be played by people outside of or across enterprise boundaries in an online context just as frequently as they are inside [4, 5].

All of these coarse-grained characteristics of autonomic computing can be represented in the form of finer-grained architecture drivers that are useful in characterizing steps toward an autonomic computing architecture [6]. Cloud technology offerings that are available today share many of the same drivers as what we have organized into systems and application management drivers.

7.2.1.1 Systems Management Drivers

Figure 7.3 shows step-wise characteristics towards autonomic computing. The path on the left demonstrates the steps of systems management. It begins with no systems management, and ends with a systems management capability that is policy driven and enables automated systems management. It is able to harmonize business and infrastructure policies within and across Cloud boundaries, in both single- and multi-tenant modes. This path is further divided into two groups of specific characteristics: *system management* and *utility computing* [7, 8].

7.2.1.2 Applications Management Driver

The path on the right in Fig. 7.3 demonstrates the steps of architecture style. It begins with common monolithic enterprise applications, and ends with applications

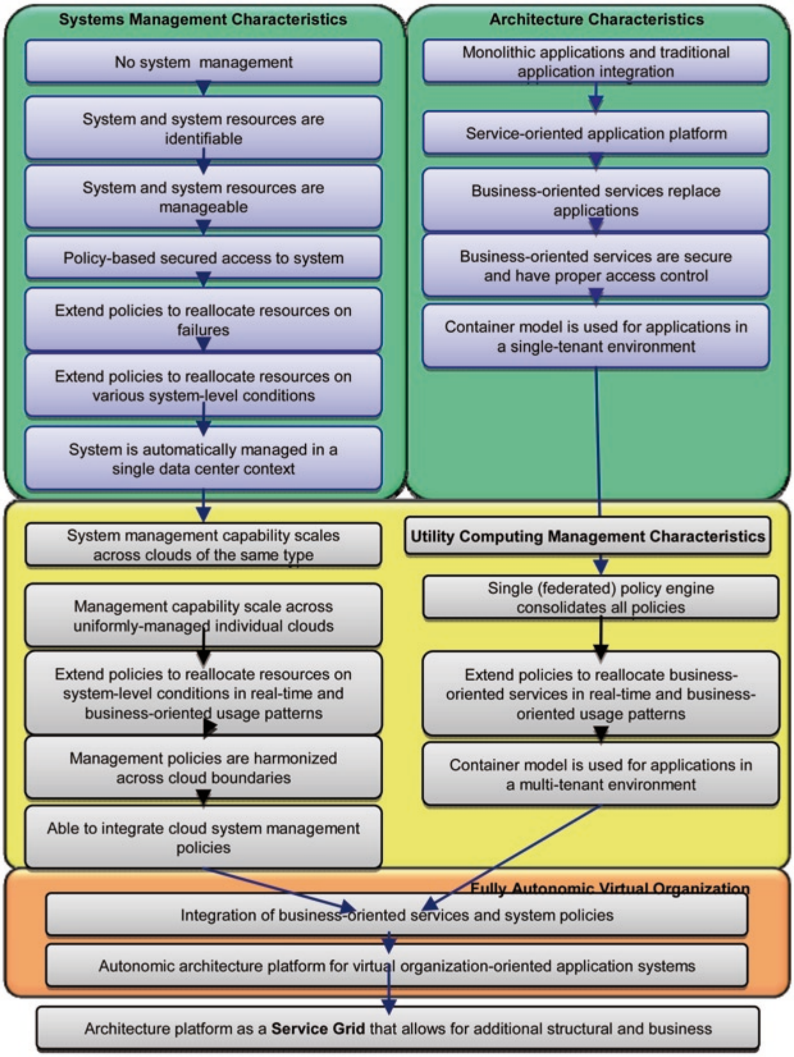

Systems Management Characteristics
No system management
System and system resources are identifiable
System and system resources are manageable
Policy-based secured access to system
Extend policies to reallocate resources on failures
Extend policies to reallocate resources on various system-level conditions
System is automatically managed in a single data center context
Architecture Characteristics
Monolithic applications and traditional application integration
Service-oriented application platform
Business-oriented services replace applications
Business-oriented services are secure and have proper access control
Container model is used for applications in a single-tenant environment
System management capability scales across clouds of the same type
Management capability scale across uniformly-managed individual clouds
Extend policies to reallocate resources on system-level conditions in real-time and business-oriented usage patterns
Management policies are harmonized across cloud boundaries
Able to integrate cloud system management policies
Utility Computing Management Characteristics
Single (federated) policy engine consolidates all policies
Extend policies to reallocate business-oriented services in real-time and business-oriented usage patterns
Container model is used for applications in a multi-tenant environment
Fully Autonomic Virtual Organization
Integration of business-oriented services and system policies
Autonomic architecture platform for virtual organization-oriented application systems
Architecture platform as a Service Grid that allows for additional structural and business

Fig. 7.3 Steps towards autonomic computing

having been replaced with service-oriented ones. In addition, policies are externalized so that business policies can be harmonized with utility management policies, so that it is possible to implement E2E SLAs and enforce conformance to business and regulatory constraints. The usage of business functional and infrastructural components is also to be metered and elastically load balanced. At this endpoint, business services and infrastructure can be organized into a Cloud and be used in both single- and multi-tenant modes [4, 9].

As the figure indicates, the systems and applications management drivers' paths converge at the point where it is necessary to manage both the business and the infrastructure using common management capabilities, and where related policies must be harmonized. The end point will be able to support business ecosystems and emergent and fluid virtual organizations.

7.2.2 Adapting to High Utilization and Rapid Growth

The fact that today's IT environment is only a small portion of each dollar spent on IT creates a direct business benefit. It is estimated that customers spend 70% of their budget on operations, and only 30% on differentiating the business. Since datacenter IT assets become obsolete approximately every 5 years, the vast majority of IT investments are actually spent on upgrading various pieces of infrastructure and providing redundancy and recoverability. This expenditure makes up approximately 60–80% of IT expenditures without necessarily providing optimal business value or innovation. Thus, it is obvious that the current, tightly-coupled model and siloed infrastructure of IT hinders enterprises from adjusting dynamically to new business requests [10–12].

Enterprise datacenters are currently facing a critical challenge: the rapid growth of both the number of applications and the amount of data in the datacenters. Cloud technologies address this challenge in a way that allows transactional data and high-performance file share applications to be best handled within the enterprise datacenter. In addition, Cloud technologies also demonstrate the ability to handle increasing Internet data from rich web applications, services from online SPs, large data processing jobs, and digital media creation with follow-on global distribution.

7.2.2.1 Consolidation and Virtualization

From a datacenter's perspective, the movement toward Cloud services transformation began with datacenter virtualization and consolidation of server, storage, and network resources to reduce redundancy and wasted space and equipment with measured planning of both architecture (including facilities allocation and design) and process. Virtualization technologies enable the abstraction and aggregation of all datacenter resources in order to turn them into a unified logical resource that can be shared by all application loads. Virtualization decouples the physical IT infrastructure from the applications and services being hosted, allowing for greater efficiency and flexibility, with any effect on system administration productivity handled by tools and processes [4, 13].

Consolidation is not only viable for a datacenter, it is also a critical process to the enterprise's application development. The virtualization ability enables the enterprise to regain control of distributed computing and development resources by creating shared pools of standardized resources that can be rationalized and centrally managed.

Many customers use server virtualization not only for server consolidation, but also to improve flexibility, speed up service provisioning, and reduce planned and unplanned downtime. Virtualization of application software, the software development environment, servers, storage, and networks will enable the mobility of applications and data not only across servers and storage arrays in the same datacenter, which customers are already implementing in production, but also across datacenters and networks [14, 15].

7.2.2.2 Automation and Optimized Virtualization

After the resources are consolidated in a Cloud environment, enterprises need to move from managing underlying infrastructure to managing service levels based on what makes sense for the user of applications. Enterprises must implement automation for central IT and self-service for end users, thus extricating IT from the business of repetitive management procedures and enabling end users to get what they need quickly. Virtualization optimizes IT resources and increases IT agility, thus speeding time-to-market for services.

The IT infrastructure undergoes a transformation in which it becomes automated. Critical IT processes are dynamic and controlled by trusted policies. Through automation, datacenters systematically remove manual labor requirements for the runtime operation of the datacenter. With the hardware and provisioning freedom that comes with a Private Cloud, a major pharmaceutical company can perform multiple drug trials that cost far less in computing power than a single drug trial had cost previously. As a result, the company can now rethink the way that it conducts its research and product development, dramatically improving time-to-market. While self-service and metering are breakthrough Private Cloud capabilities for end users and business units, maintaining service delivery in a fully virtualized multi-tenancy environment and providing security, especially for information and services leaving the datacenter environment, are essential enterprise requirements for IT administrators [13].

7.2.2.3 Service Federation

To go beyond organizational boundaries through Cloud internetworking to reach a third-party, businesses will need SPs and virtual Private Cloud services. Ultimately, SPs will offer both public and virtual Private Cloud services on a secure infrastructure, and that will allow enterprises to include and consume those services as part of enterprise Private Clouds, without exposing content to the general public.

The use of federations to link disparate Cloud infrastructures with one another, for instance by connecting their individual management infrastructures, allows disparate Cloud IT resources and capabilities, such as capacity, monitoring, and management, to be shared, much like power from a power grid. It also enables unified metering and billing, one-stop self-service provisioning, and the movement of application loads between Clouds, since federation can occur across datacenter and organization boundaries due to Cloud internetworking [16].

Creation of an open, competitive marketplace, in which IT capabilities in a utility model can be procured, allocated, and provisioned over the Internet on demand by the consumer, with self-service and metering, requires federation. Cloud internetworking is the network technology enabling the linkage of disparate Cloud systems in a way that accommodates the unique nature of Cloud environments and the running of IT workloads [13]. The benefits gained from this stage can be concluded as the following:

- Enable choice through open inter-Cloud standards and services
- Support federation across internal and external Clouds
- Deliver Cloud services with security, QoS, and manageability
- Use standards for consolidated application and service management and billing

After going through this process, enterprises will be able to select services freely among SPs, and SPs will be able to use other providers' infrastructures that allow federation to handle exceptional loads on their own offerings.

7.2.2.4 Consolidation of Management Information

Federations typically distribute IT information across multiple repositories. Therefore, a mechanism is needed to create a more complete and accurate view of IT information across multiple data sources. The *Configuration Management Database* (CMDB) Federation specification was created by the DMTF to provide such a mechanism. CMDBs give IT organizations visibility into the attributes, relationships, and dependencies of the components in their enterprise computing environments. It helps service managers maintain mappings between the virtual configuration and the physical configuration. For providers who use ITIL as a framework to manage IT, a CDMB federation can help track service assets and configuration management data. A CMDB can help predict Cloud usage by first knowing what is available and how it is being used [17, 18].

Information in a CMDB can help the service management function to validate pre-installation or pre-version upgrade storage and server configurations. The result can be used to generate reports and alerts to notify either the service customers or the providers of the status of key configuration variables. When a patch is available, the CMDB can facilitate the management infrastructure in providing a proactive patch-notification service [19].

7.3 Standards-Based Business Process Framework

Today's IT leaders are operating in a business climate in which intense commoditization and change force deployment of new IT-enabled business processes. It is clear that business processes and architectures that are fixed/rigid will not scale to large networks of practice. IT budgets may have reached the point where conventional internal cost cutting can wring out only nominal additional value unless business and IT processes make the corresponding adjustments. In the next two sections, we will discuss in detail the usage of the two most relevant existing standards for operation and information management: *business process* and *information/data frameworks*.

7.3.1 The ITIL and eTOM Frameworks

As stated in Chap. 3, eTOM is a business process framework, a reference framework or model for categorizing all the business activities that a SP will use. eTOM provides the definition of common terms concerning enterprise processes, sub-processes, and the activities performed within each. Common terminology makes it easier for enterprises to negotiate with customers, third party suppliers, and other enterprises as well.

The latest eTOM (release 8.0) [20] also includes many ITIL elements. For instance, the Enterprise Effectiveness Management incorporates event management, incident management, request fulfillment, service asset and configuration management, and continual service improvement. The Strategic & Enterprise Planning has new additions of release and development and change management. The Enterprise Risk Management is expanded with problem, information security, and service continuity management.

7.3.2 Level Zero Key Concept

The highest conceptual view of the eTOM framework is the Business Process Element Enterprise Framework. It represents the whole of an organization's enterprise environment.

Figure 7.4 shows the highest conceptual view of the eTOM framework. This view provides an overall context of the key concepts within the framework. eTOM encompasses three major process areas. First, eTOM differentiates strategy and lifecycle processes from operations processes in two large process areas: *Strategy, Infrastructure & Product* (SIP) and *Operations*. SIP covers planning and lifecycle management, Operations covers the core of operational management, and Enterprise Management covers corporate or business support management.

The horizontal layers across these two process areas are the key functional areas. Market, Product, and Customer processes are constituted by sales and channel management, marketing management, product and offer management, CRM, ordering, problem handling, SLA Management, and billing. The Service processes include service development and configuration, service problem management, quality analysis, and rating. The Resource processes support the development and management

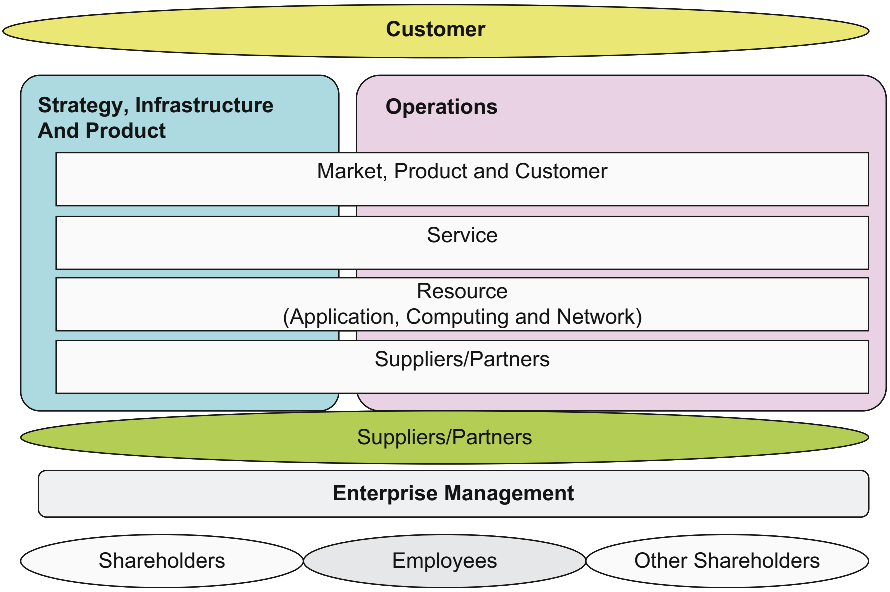


Fig. 7.4 eTOM business process framework—level 0 key concepts

of the enterprise's service and operational infrastructure. The Supplier/Partner processes support the enterprise's interaction with its suppliers and partners.

The third major process area, Enterprise Management, is shown as a separate box in the lower part of the diagram. It includes basic business processes that are required to run any large business. These processes are sometimes considered to be the enterprise functions and/or processes.

Finally, customers, shareholders, employees, and other stakeholders are the four types of entities that interact with the enterprise, shown as ovals in the diagram. Suppliers and Partners are involved in all three areas of the processes that manage the product and infrastructure. The employees, shareholders, and stakeholders are the internal and external entities that interact with Enterprise Management. The customers interface with the SIP and the Operations of the SP. All of these functional groups reflect the major expertise and focus required to pursue the business.

7.3.3 Level One Processes

Figure 7.5 shows how the three major process areas of Level 0 are decomposed into their constituent Level 1 process groupings. Note that the Level 1 process in the eTOM has seven vertical process groups conducting the end-to-end processes required to support customers and manage business. This view typically is considered the overall view of the eTOM framework.

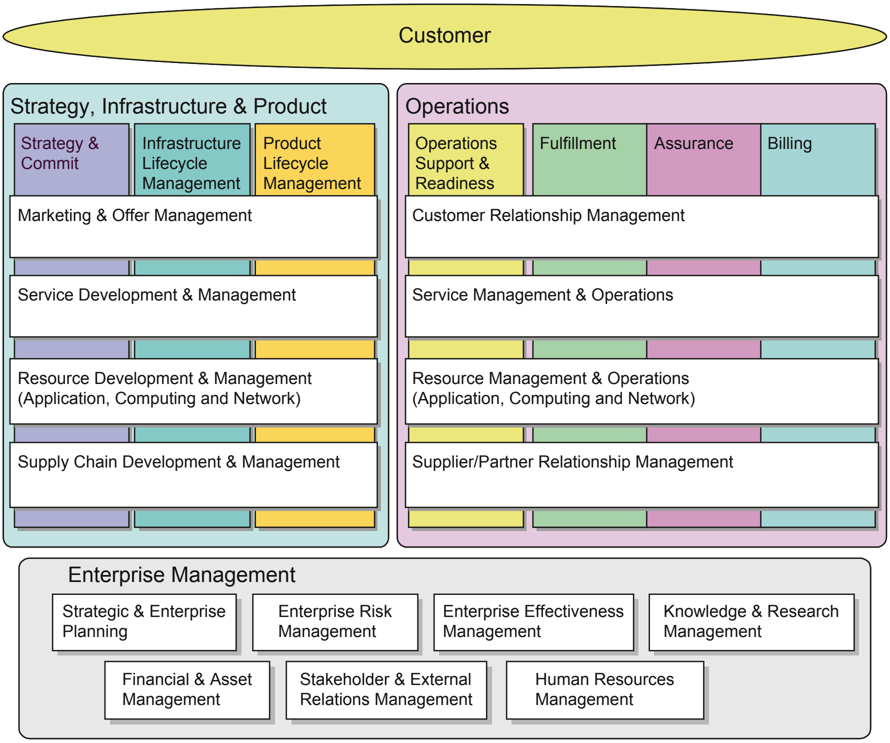

Customer
Strategy, Infrastructure & Product
Strategy & Commit
Infrastructure Lifecycle Management
Product Lifecycle Management
Marketing & Offer Management
Service Development & Management
Resource Development & Management (Application, Computing and Network)
Supply Chain Development & Management
Operations
Operations Support & Readiness
Fulfillment
Assurance
Billing
Customer Relationship Management
Service Management & Operations
Resource Management & Operations (Application, Computing and Network)
Supplier/Partner Relationship Management
Enterprise Management
Strategic & Enterprise Planning
Enterprise Risk Management
Enterprise Effectiveness Management
Knowledge & Research Management
Financial & Asset Management
Stakeholder & External Relations Management
Human Resources Management

Fig. 7.5 eTOM business process framework—level 1 key concepts

The two major Level 0 areas (i.e., SIP and Operations) are further categorized into seven vertical bins. It is important to note that Operations Support & Readiness processes are concerned with activities that are less "real-time," or customer-facing, than Fulfillment, Assurance, and Billing & Revenue Management processes are. Thus, they are typically detached from individual customers and services.

In addition, Strategy & Commit, Infrastructure Lifecycle Management, and Product Lifecycle Management process groups work on different business time cycles from the operation time cycle perspective. The Strategy & Commit process group is responsible for generating specific business strategies in support of the Infrastructure and Product Lifecycle processes, and for gaining buy-in within the business to implement this strategy. The Infrastructure Lifecycle Management process group is responsible for addressing the needs of the Product Lifecycle Management processes. It is also responsible for identifying, defining, planning, and implementing all necessary infrastructures, supporting infrastructures, or business capabilities to support the provision of products to customers. The Product Lifecycle Management process group is responsible for defining, planning, designing, and implementing all products in the enterprise's portfolio.

Finally, the horizontal functional process groupings shown in the diagram distinguish functional operations, processes, and other types of business functional processes.

7.3.4 *Level Two and Three Processes*

The Level 2 processes decompose the previous specifications into functional components to show detailed capabilities for supporting vertical end-to-end processes. In the TM Forum specifications, some Level 3 processes are delineated as samples in Level 2 discussions, due to the fact that the majority of level 3 capabilities are application specific. Note that eTOM has evolved since the inception of the TM Forum's NGOSS program to include a number of other artifacts, in addition to levels of decomposition down to Level 3.

These newer artifacts represent the interaction among eTOM processes and defined SID business entities using a number of techniques that develop process flow diagrams, user case diagrams, state chart diagrams, activity diagrams, etc. Fig. 7.6 shows the corresponding eTOM Level 2 processes in the three process areas: *SIP, Operations*, and *Enterprise Management* respectively.

The SIP area, as shown in Fig. 7.7, supports the management and operations of marketing and offers services, service resources, and supply chain interactions.

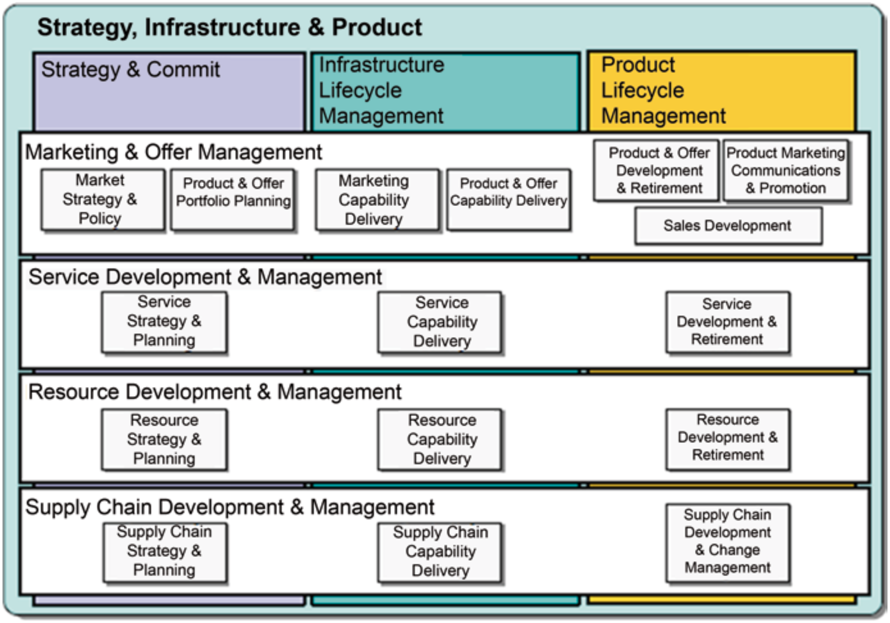

Strategy, Infrastructure & Product
Strategy & Commit
Infrastructure Lifecycle Management
Product Lifecycle Management
Marketing & Offer Management
Market Strategy & Policy
Product & Offer Portfolio Planning
Marketing Capability Delivery
Product & Offer Capability Delivery
Product & Offer Development & Retirement
Product Marketing Communications & Promotion
Sales Development
Service Development & Management
Service Strategy & Planning
Service Capability Delivery
Service Development & Retirement
Resource Development & Management
Resource Strategy & Planning
Resource Capability Delivery
Resource Development & Retirement
Supply Chain Development & Management
Supply Chain Strategy & Planning
Supply Chain Capability Delivery
Supply Chain Development & Change Management

Fig. 7.6 eTOM level 2 business process—SIP

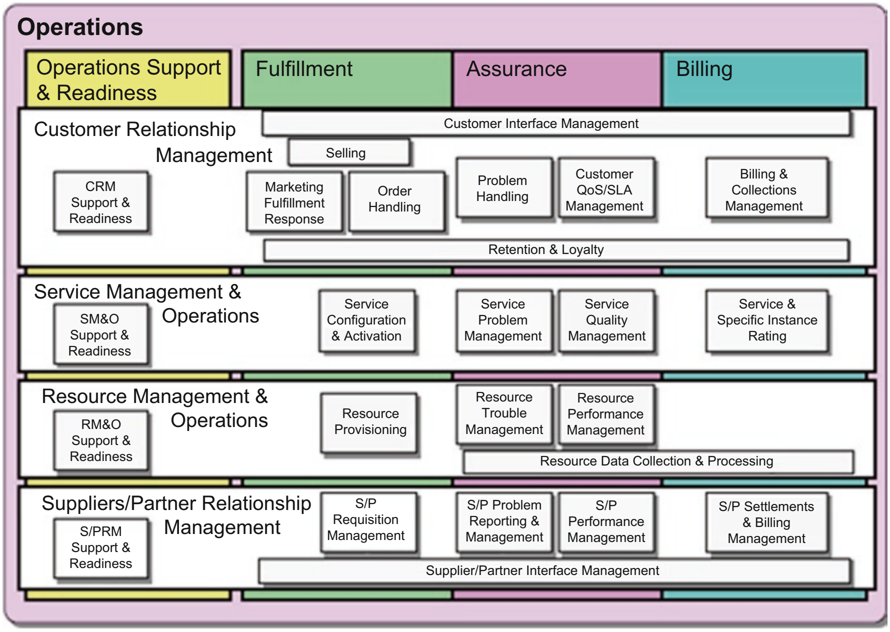

Operations
Operations Support & Readiness
Fulfillment
Assurance
Billing
Customer Relationship Management
Customer Interface Management
Selling
CRM Support & Readiness
Marketing Fulfillment Response
Order Handling
Problem Handling
Customer QoS/SLA Management
Billing & Collections Management
Retention & Loyalty
Service Management & Operations
SM&O Support & Readiness
Service Configuration & Activation
Service Problem Management
Service Quality Management
Service & Specific Instance Rating
Resource Management & Operations
RM&O Support & Readiness
Resource Provisioning
Resource Trouble Management
Resource Performance Management
Resource Data Collection & Processing
Suppliers/Partner Relationship Management
S/PRM Support & Readiness
S/P Requisition Management
S/P Problem Reporting & Management
S/P Performance Management
S/P Settlements & Billing Management
Supplier/Partner Interface Management

Fig. 7.7 eTOM level 2 business process—operations

Starting at the top horizontal group, the *Customer & Offer Management* process group includes defining strategies, developing new products, managing existing products, and implementing marketing and offering strategies. The *Service Development & Management* process group is responsible for planning, developing, delivering, and retiring services to the Operations domain. The *Resource Development & Management* process group is responsible for planning, developing, delivering, and retiring resources needed to support services and products to the Operations area. The *Supply Chain Development & Management* process group is responsible for interactions with the supply chain suppliers and partners, as required by the enterprise. These processes assist the enterprise in having information flows and financial flows in place. They also assure that suppliers/partners are available to deliver the required support in a timely manner.

Based on the previous organization, the Operations area can be divided into four horizontal groups. The CRM process group includes the functionalities necessary for a SP to collect customer information, identify potential buyers, and acquire, enhance, and retain a relationship with a customer. The *Service Management & Operations* (SM&O) process group includes functionalities necessary for the delivery (on-demand or not), management, and operations of services required by, or proposed to, customers. The *Resource Management & Operations* (RM&O) process group is responsible for managing all the resources required to deliver and support services specified by, or proposed to, customers. The *Supplier/Partner Relationship Management* (S/PRM) process group enables the direct interface with the appropriate lifecycle, E2E customer operations, or functional processes with suppliers and/or partners, which supports core operational processes.

The Enterprise Management area manages enterprise-level actions, and needs to provide a clearer focus on relevant process responsibilities. Enterprise Management processes are, in part, responsible for setting enterprise strategies and directions in order to provide guidelines and targets for the rest of the business. TM Forum has not developed process models for this area because they do not require significant specialization for SPs. Shown in Fig. 7.8, the Strategic & Enterprise Planning process group drives the mission and vision of the enterprise by developing strategies (market, financial, and acquisition) and plans. The Financial & Asset Management process group is accountable for the overall management of the enterprise income statement, corporate balance sheet, asset resource, and corporate procurement. The HRM process group manages people, resources, and organizational development that the enterprise uses to fulfill its objectives. The Knowledge & Research Management process group performs research and development of technology within the enterprise as well as evaluates potential technology acquisitions. Its knowledge management function directs and supports the marketing processes in the SIP and Operations areas of the enterprise. The Enterprise Effectiveness Management process group is in charge of developing and improving key architectures of the enterprise. This group includes management of the program, project, process, and facilities, as well as enterprise quality and performance assurance. The Enterprise Risk Man-

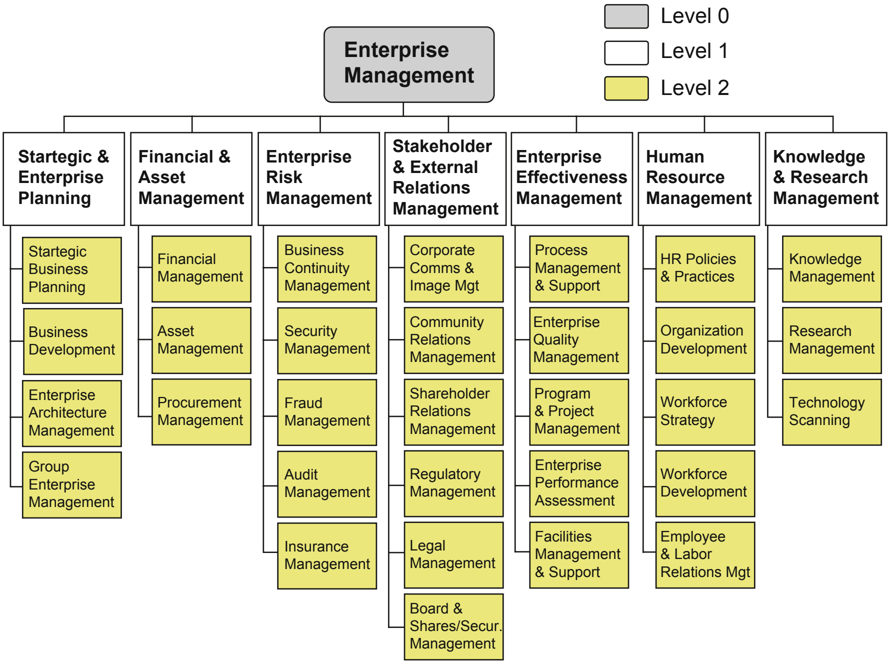

Level 0
Level 1
Level 2
Enterprise Management
Startegic & Enterprise Planning
Startegic Business Planning
Business Development
Enterprise Architecture Management
Group Enterprise Management
Financial & Asset Management
Financial Management
Asset Management
Procurement Management
Enterprise Risk Management
Business Continuity Management
Security Management
Fraud Management
Audit Management
Insurance Management
Stakeholder & External Relations Management
Corporate Comms & Image Mgt
Community Relations Management
Shareholder Relations Management
Regulatory Management
Legal Management
Board & Shares/Secur. Management
Enterprise Effectiveness Management
Process Management & Support
Enterprise Quality Management
Program & Project Management
Enterprise Performance Assessment
Facilities Management & Support
Human Resource Management
HR Policies & Practices
Organization Development
Workforce Strategy
Workforce Development
Employee & Labor Relations Mgt
Knowledge & Research Management
Knowledge Management
Research Management
Technology Scanning

Fig. 7.8 eTOM level 2 business process—enterprise management

agement process group assures that the enterprise can support its mission critical operations, processes, applications, and communications in dealing with disasters, security threats, and fraud attempts. Finally, the Stakeholder & External Relations Management process group manages the enterprise's relationship with stakeholders and outside entities (e.g., regulators, local community, and unions) [21].

7.3.5 Improvements to Current eTOM for Cloud Services

As one can see, eTOM is rather sophisticated in terms of specifying processes and functional aspects of the telecommunications industry. Even though a large portion of the eTOM can be adopted for Cloud services, there are still a number of modifications that need to be made:

- *Enhancements to the value-chain interface*: The community relationship is no longer one-way (client-supplier); the interface specifications should include elements for more collaborated service relationships.
- *Policy-based management*: Policy features require the current model to "negotiate" with other providers peer-to-peer. The current supplier management may need to be extended in the enterprise management area. Policy should be relevant to the service topology, service specification, SLA, SLO, and security at both the physical and virtual levels.
- *Security management*: The biggest challenge in security management is that it has to manage security configurations across physical, virtual, and Cloud environments. In addition, from a cross-Cloud perspective, incompatible log format outputs by physical devices will continue to be a problem for virtual and Cloud environments, at a much larger scale.
- *Enterprise management*: One of the trends for enterprise Cloud computing is that enterprise policies for dealing with external suppliers and partners start to emerge. New enterprise management platforms must be developed to apply policy and automation across thousands of transient servers that belong to different suppliers and providers, fluid underlying storage and network resources, and variable workloads which often need to be dynamically migrated.
- *Software development*: Product lifecycle management in the current eTOM model provides a good foundation, but needs to expand to develop collaboration and extended relationships with other providers and development communities (or standards bodies).
- *Enterprise outsourcing model*: The IT outsourcing business still has plenty of mileage shifting software development and support work offshore, but eventually this will dry-up. Many of the high-end enterprises have already moved much of their commodity work offshore, and they now have to look at more of the complex infrastructure areas for the next wave of productivity gains. Cloud delivery will play a pivotal role in the heart of the future global sourcing delivery business, but the critical question is how quickly it will become adopted. In ad-

dition, it needs refined functional areas to address how enterprise management will use services from other providers (PaaS and IaaS).

- *Multiple tendency*: The multiple tendency environment in a Cloud is much more flexible than the existing multiple-instance environment. The flexibility of Cloud services at the technical and process domains must reflect to appropriate levels of procedures and methodologies in the runtime fulfillment, service assurance, and billing of the eTOM sub-models.
- *New kind of customer relationships*: In Cloud environments, it is apparent that service customers have more freedom to provision purchased services and make contributions (e.g., add/update content) to the services, in addition to Web 2.0/3.0's impact on the process. The traditional idea of "customers" has to change drastically; their roles are now between the existing client and the value chain partner relationship.

7.4 Standards-Based Information Framework

NGOSS's information framework (SID) [22] provides the communications and information industry enterprises an effective way to organize their business processes and communicate with each other. The SID business view model can be viewed as a companion model to eTOM. SID provides an information and data reference model and a common information and data vocabulary from a business entity perspective.

Teamed with eTOM, the SID model provides enterprises with not only a process view of their business, but also an entity view. In simpler terms, SID provides the definition of the 'things' that are to be affected by the business processes defined in eTOM. eTOM and SID together offer a way to explain how things are intended to fit together to meet a given business need.

7.4.1 The SID Business View

The business view model uses the concepts of domains and aggregate business entities to categorize business entities and reduce duplication and overlap. The SID business view focuses on business entity definitions and associated attribute definitions. A business entity is a thing of interest to the business, while its attributes are facts that further describe the entity. One direct benefit of this partitioning is that it allows distributed working groups to build out the model definitions while minimizing the flow-on impact across the model.

A domain is defined as a collection of Aggregate Business Entities (ABEs, see next section) associated with a specific management area. An ABE is a well-defined set of information and operations that characterize a highly cohesive, loosely coupled set of business entities. Domains that make up the SID framework are consistent with eTOM Level 0 concepts shown in Fig. 7.9. At the top layer, a set of domains is identified which are broadly aligned with the eTOM business process framework.

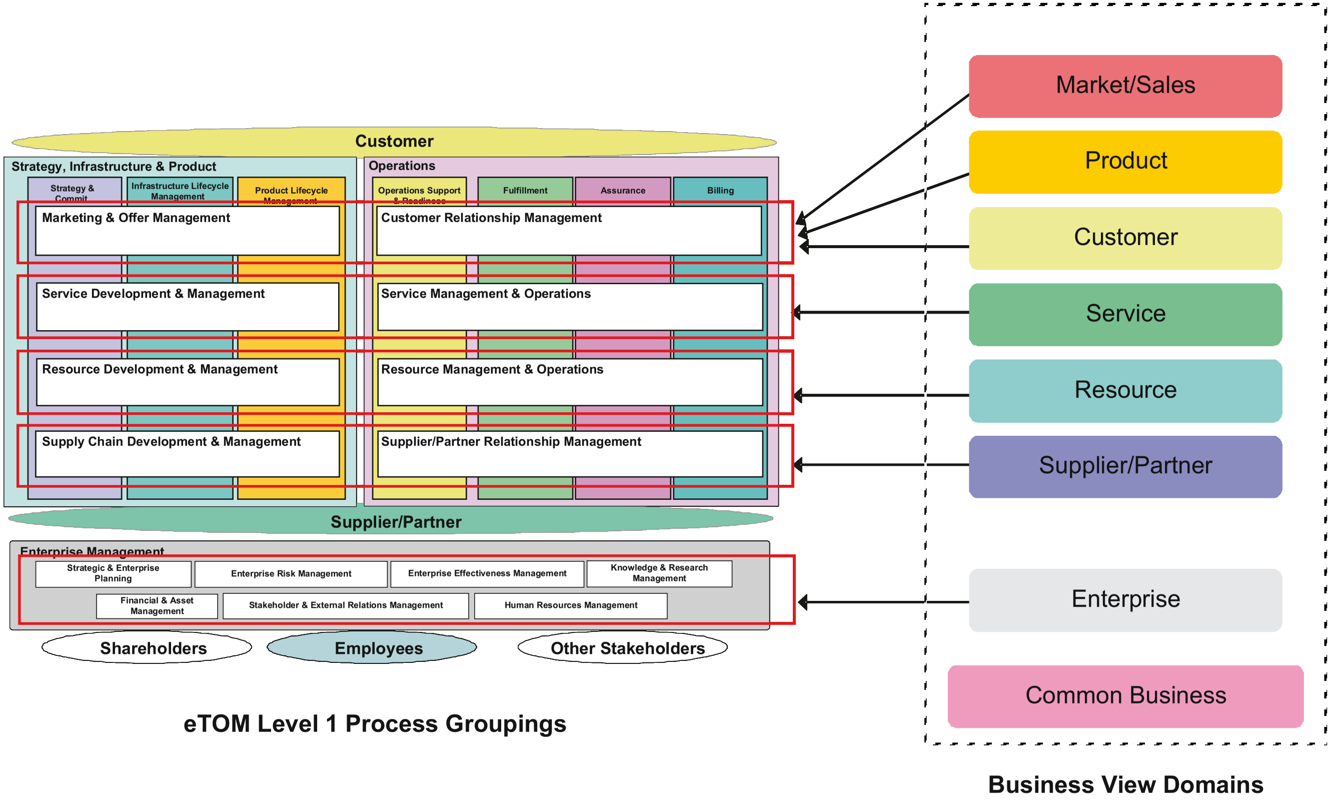

Customer
Strategy, Infrastructure & Product
Strategy & Commit
Infrastructure Lifecycle Management
Product Lifecycle Management
Operations
Operations Support & Readiness
Fulfillment
Assurance
Billing
Marketing & Offer Management
Customer Relationship Management
Service Development & Management
Service Management & Operations
Resource Development & Management
Resource Management & Operations
Supply Chain Development & Management
Supplier/Partner Relationship Management
Supplier/Partner
Enterprise Management
Strategic & Enterprise Planning
Enterprise Risk Management
Enterprise Effectiveness Management
Knowledge & Research Management
Financial & Asset Management
Stakeholder & External Relations Management
Human Resources Management
Shareholders
Employees
Other Stakeholders
eTOM Level 1 Process Groupings
Market/Sales
Product
Customer
Service
Resource
Supplier/Partner
Enterprise
Common Business
Business View Domains

Fig. 7.9 SID framework domains aligned with eTOM domains

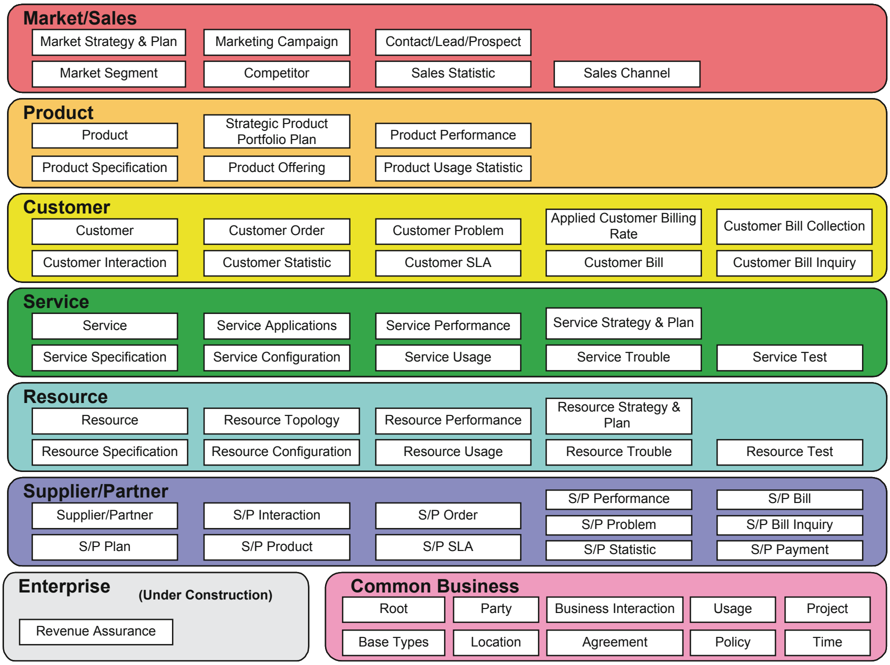


Fig. 7.10 SID domains and level 1 ABEs

7.4.2 SID Domains and Level One ABEs

Within each domain, further partitioning of the information is achieved through the identification of ABEs. Figure 7.10 shows the Level 1 ABE's. As the SID business view is further expanded and defined, further partitioning of the ABE's occurs as more explicit business entities are identified.

The business entities, along with the attributes and relationships that characterize the entities, provide a view of the model that is easily understood from a business perspective. The business entities, attributes, and relationships are developed using textual descriptions in a consolidated UML-based model. The UML model provides an architecturally oriented business view of business entities, attributes, and relationships to other business entities.

7.4.3 Service Domains and Level Two ABEs

Figure 7.11 shows Level 2 ABEs identified within the Service domain. Note that at some point in the decomposition of ABEs, business entities appear within the ABEs. Business entities represent the lowest level of entity decomposition within the SID framework. The SID, along with business entities, their attributes, and as-

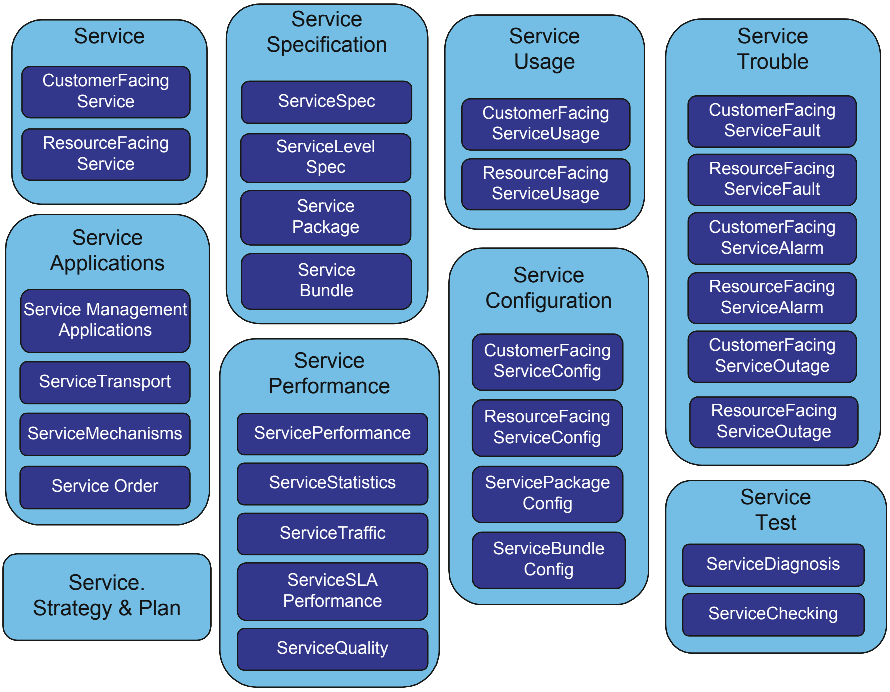


Fig. 7.11 Level 2 ABEs identified in service domain

sociations, are organized into a UML model. An example of such a model is shown in Fig. 7.12.

7.4.4 Improvements to the Current SID for Cloud Services

After examining SID closely, it is rather obvious that there are many aspects of SID can be borrowed for Cloud services, just like the eTOM. However, there are several adjustments that need to be made:

- *More specific to cover PaaS, IaaS, and SaaS*: These three are the most referenced Cloud services thus far. SID currently does not reflect the characteristics and distinctions of them. In order to make SID more functional and relevant to the Cloud environment, object and relationship definitions and aspects of PaaS, IaaS, and SaaS need to be created.
- *The current billing and pricing model needs to be upgraded to reflect the new Cloud paradigm*: More specifically, replacing the "pay-up-front" model to reflect the "pay-as-you-go" characteristic of Cloud services.
- *New value-chain model*: Understanding the structure of the Cloud and its potential value creation schemes is challenging due to the diversity in requirements, inherited technical complexity, and unstructured service schemes. Clarifying the

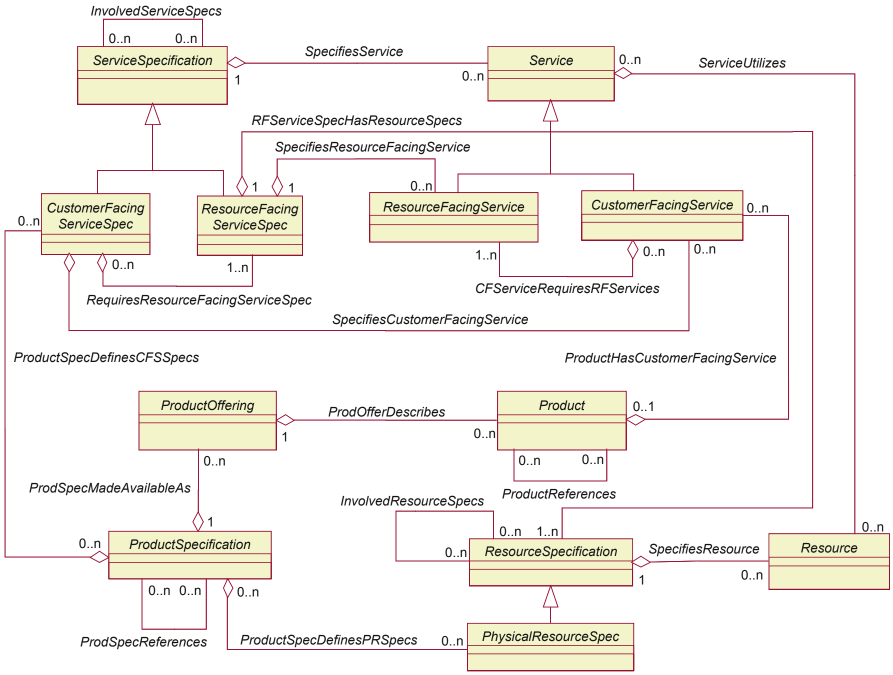


Fig. 7.12 SID framework and entities within UML model

value structure and corresponding primary and support activities in the Cloud value chains would be beneficial to both the business and Cloud communities.

- *More focus on SLO than SLA*: Currently, automation is not often considered as a feature in a SLA. For competitive reasons, providers offer many features as part of a standard package that are invisible to their clients. This relies on (enterprise) internal SLO to guide through the service life cycle.
- *The new integration data model/value chain model must be more relevant to service policy and tied back to the eTOM model at a higher level*: The current SLA specification relationship is not sufficient, because many policies are implicit in service offerings as part of built-in automation features. Once the appropriate and necessary features are added in SID, references need to be made to the Cloud-enhanced eTOM accordingly in order to reflect both process and information aspects of Cloud services.

7.5 Technology-Neutral, Service-Centric Architecture

The term composite service, as explained in Chap. 6, means combining business services or methods together to form coarse and larger business functions that are peered with an application functionality. For example, services that manage order fulfillments, invoice submission and payment processing, orchestrations for invoicing, logistics planning, etc.

Orchestration is often equated to workflows used to coordinate ordering of service method invocations. Workflows and other BPM technologies are well-known within today's enterprises. Workflow engines for Web services have been commoditized through open source initiatives and by commercial software vendors. These engines make it possible to implement composite Web services as either state machine or sequential workflows. Use of state machine flows makes it possible to avoid prescriptively dictating how systems interoperate. They also provide the opportunity to incorporate human intelligence tasks to help resolve exception conditions that often emerge from composite services or straight through processing flows [4].

Because today's context is qualitatively different than just a few years ago, many relatively recent innovations challenge traditional wisdom. This old wisdom states that it is better to evolve and extend an existing platform then it is to create a new one to circumvent problems. A few of these innovations include significant broadband capacity, economic storage (both self- and Cloud-hosted), cheap memory and modern caching services, commodity 64-bit OS, XML accelerators and sophisticated application protocol management capabilities, commoditized integration/interoperability technologies, virtualization and utility computing, Cloud and service Grid Computing, and other many others.

7.5.1 Next-generation Datacenter Management

Compute Clouds are going to be housed in datacenters, big in both size and amount. This is good news for enterprise network management vendors because those datacenters will need to be managed. People running small- or medium-sized datacenters are likely to be the people most attracted to Cloud technologies. Therefore, it is likely that these datacenters will be consolidated into a few large datacenters instead of having numerous small and medium datacenters. The only people who will be able to justify the cost of running a small or medium datacenter are those with special requirements that cannot be easily accommodated using a Cloud-based solution. Generally, datacenter costs are comprised of three main components: *hardware costs*, *physical costs* (such as power and cooling), and *administrative management costs*. Particularly, the administrative and management costs account for a significant portion of the overall cost. As such, removing manual processes, errors, and repetition is a great way to reduce and control IT costs [23].

It is hard to see the transition as anything other than party time for enterprise vendors. Open source enterprise vendors will be in a very good position to win new customers. The transition to Cloud technologies is a once in a lifetime disruption causing a lot of Cloud vendors to look for new, more flexible tools to help them manage their new, ultra-flexible infrastructure. In addition, new Cloud vendors are utilizing open source software extensively while building their offerings, so they will be more amenable to open source-based network management solutions [24].

The business case behind datacenter strategies is changing. Datacenters represent a very logical starting point for a new consumer of Cloud services. They provide relatively low risk and potentially significant cost savings and efficiency gains. Transitioning existing systems to the Cloud offers opportunities to outsource non-core functions for most businesses. At the same time, it provides experience with a Cloud-oriented way of organizing and accessing digital technology that is necessary for building out a roadmap for sensible Cloud adoption.

To satisfy the requirements of the next generation of computing, Cloud technologies will need to be more than just externalized datacenters and hosting models. Although architectures that enterprises deploy in datacenters today could be run in a Cloud, simply moving them into a Cloud is certainly not what one might hope Cloud technologies will come to be. In fact, tackling globally-scaled collaboration and trading partner network problems in different sectors such as government, military, scientific, and business contexts, will require more than what the current architectures can readily support. For example:

- It will be necessary to rapidly set up a temporary collaboration network, enabling network members to securely interact online, where interaction could imply interoperability with back office systems as well as human oriented exchanges, all in a matter of hours. Examples that come to mind include emergency medical scenarios, global supply chains, and other business process networks. Policies defining infrastructure and business constraints will be varied, so policy must be external to and must interact with deployed functionalities. These examples also imply the need for interoperability between Public and Private Clouds.
- Business interactions have the potential to become more complex than personal transactions. Because they are likely to be formed as composite services, and because services on which they depend may be provisioned in multiple Clouds, the ability to provision and uniformly manage composite Cloud services will be required, as will be the ability to ensure that these services satisfy specified business policy constraints.
- The way that users and access control are managed in typical applications today is no longer flexible enough to express roles and responsibilities that people will play in next generation business interactions. Roles will be played by people outside of or across enterprise boundaries in an online context just as frequently as they are inside the enterprise. Access control and the management of roles and responsibilities must be externalized from business functionalities so that it becomes more feasible to composite functional behavior into distributed service-oriented applications that can be governed by externalized policy.

7.5.2 Architectural Planning, Simplification, and Transformation

Moving IT platforms to Clouds represents the next logical step in a service-oriented world, and the new decision framework in service selection will migrate to build, buy or lease. Understanding the level of the Cloud and the internal enterprise maturity will guide decisions, such as how and when to leverage Cloud services to support the core business objective as well as non-core business capabilities, and how software assets should interoperate to provision business functionalities. Note that it is critical to give explicit focus to policy-based architectures that support agility and innovation.

One solution available is that the Cloud vendor provides detailed usage statistics of the Cloud through the Cloud vendor's management portal. In order to use this information in the enterprise's own network management system, the information needs to be available in a format that can be read by the enterprise's network management software. Many network management systems have fine extensibility mechanisms so that the enterprise can wire up the network management system to use the vendor's instrumentation. A better solution would be for a standard to emerge that all Cloud vendors implement. This is not very likely given the diverse offerings in the Cloud computing market. Amazon EC2 has little in common with Google Apps for instance. The more likely scenario is that a winner will emerge eventually and that will become the de facto standard. The winner looks like Amazon at the moment, but do not underestimate either Google or Microsoft. Microsoft in particular has a good deal to lose if they do not allow Microsoft-centric web developers to take seamless advantage of Cloud technologies.

7.5.2.1 Using eTOM and SID

For a commercial Cloud SP, the eTOM framework outlines a clear roadmap with many neutral reference data points that benefit not only the provider's internal process reengineering needs, but also the establishment of partnerships, alliances, and general working agreements with other enterprises. For enterprises operating in a generic business domain, the eTOM framework also provides potential boundaries of system components and the required functions, inputs, and outputs that must be supported by their products [25, 26].

eTOM is one of the most recognized and adopted process frameworks and there are a wide range of uses of the eTOM in the lean and adaptive Cloud enterprises. It can be deployed as:

- A tool for cataloging enterprise processes
- A tool for developing operational process flows
- A requirements capture framework
- A tool for mapping organization responsibilities

Extending from the scope of providing business process and integration standards for enterprises, eTOM also focuses on information and communications enterprises' development and integration of BSS and OSS. eTOM analyzes all of the business activities of an enterprise and categorizes them into different levels of detail, according to their significance for the business. One interesting perspective of eTOM is that it is a guild for developing and managing key processes within an enterprise by offering a catalogue of industry-standard names, descriptions, and scopes at multiple hierarchical levels. eTOM positions itself at commercial applications and is designed as a customer-centric and service-centric framework, viewing business processes for customer services, which is aligned with what Cloud computing emphasizes. From this perspective, differentiations of some internal processes that deal with the enterprise infrastructure or business supporting tasks may be less obvious and significant. By moving up from the original telecommunications operational supports into a business process framework, the TM Forum roadmap that used to satisfy telecommunications SPs is now redefining itself for broader, generic audiences. In the next section, we will try to lay down the roadmap as to how an enterprise may take advantage of eTOM in enterprise Cloud services. As a result, the current eTOM model is more valuable for system architecture design rather than system development. To benefit enterprise system developers, future iterations of eTOM are expected to elaborate lower level processes, as well as the linkages between them. Adding the application concepts of eTOM to the development of Cloud services can improve bottom-line sensitivity and the customer-centric objective. With the availability of many eTOM-compliant off-the-shelf products, the government, or large scale enterprise IT operations, can benefit from the marriage of these two models.

A set of steps have emerged for implementing eTOM in an enterprise. It generally involves the following three steps:

1. Accessing the level of process maturity.
2. Mapping existing enterprise processes into the standard eTOM framework. Until this step is carried out, it is very difficult to see how the wealth of material existing in the eTOM can benefit an enterprise.
3. Mapping enterprise business objective process flows using the eTOM. This step refers to the task of using their process map as the building blocks for the development of process flows.

Besides providing a vocabulary for common information concepts, the SID framework, the SID model, and its contents can be put to a number of uses within a Cloud enterprise such as:

- Part of an application integration framework
- Defining new or enhancing application development
- Organizing enterprise application user cases
- Organizing existing information models

As we briefly mentioned in Chap. 3, SID can potentially be used for multiple purposes. Using SID as part of an application integration framework involves the following steps:

1. Adopt the SID XML schema and use them to develop application-specific XSDs
2. Use SID to develop application-specific extensions
3. Use the application specific SID XSD extensions to form the basis for API message payloads

7.5.2.2 Framework-based SOA Methodology

In order to meet the efficiency and agility challenges of the new Cloud service paradigm, next generation enterprises and SPs have to reach the business modularity stage in their EA. This is accomplished when the enterprise capabilities have been captured and structured in modular, reusable business services, which support the enterprise business operations. Realistically, enterprises must deal with a barely manageable mesh, i.e., a legacy of applications, processes, data models, organizations, etc. Thus, enterprises need to first move away from these meshed operations to structure their capabilities and assets. The first step in meeting this challenge involves reducing enterprise complexity by introducing an EA that structures the enterprise assets and capabilities in architectural layers. To be really agile and modular, enterprises should take the next step towards business modularity, where these capabilities and assets are captured and structured in flexible, reusable modules, i.e., SOA services.

As mentioned in Chap. 1, SOA provides a synergistic approach to transform an enterprise's business architecture to the Cloud paradigm. In this section, we will provide a methodology for leveraging eTOM, SID, and the general Solution Framework in a systematic SOA analysis and design approach, which consists of the following four phases, as shown in Fig. 7.13:

- *Capturing business requirements*: This is the initial phase of the lifecycle. This is an analysis phase, which focuses on tasks including collecting business requirements, scoping, identifying expected business goals, and proposing required changes to the business operations. In the frame of an enterprise's business transformation, these requirements typically address the challenges of the new service paradigm. More specifically, this step can be further broken down to the following sub-steps:
  - Collect business requirements in business use cases (structured along the eTOM and SID)
  - Model the collected business use cases based on eTOM and SID

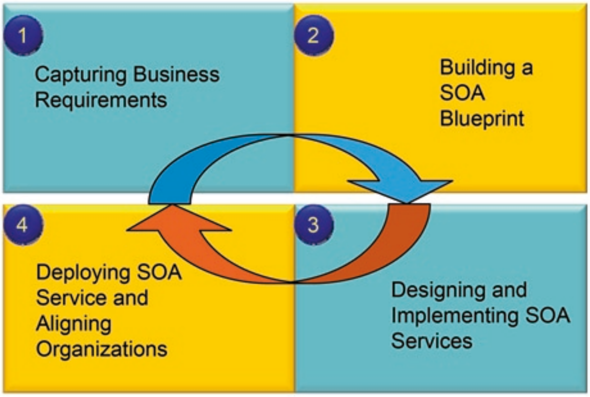

1
Capturing Business Requirements
2
Building a SOA Blueprint
4
Deploying SOA Service and Aligning Organizations
3
Designing and Implementing SOA Services

Fig. 7.13 Four phases of framework-based SOA methodology

  - Structure the analysis model with the business role, activity, and entity models
  - Complete the analysis model with a process model.

- *Building a SOA blueprint*: This phase involves transforming the process-oriented results collected in the business view into representations appropriate for the application and service-oriented system view. This provides a technology- and implementation-neutral blueprint for further SOA service design and implementation. This phase can be decomposed into three sub-tasks:
  - Specifying services and roles
  - Allocating the service candidates to logical applications into a platform-independent model
  - Consolidating the services and roles with real-world capabilities
- *Designing and implementing SOA services*: This phase consolidates the SOA blueprint with real world constraints and orchestrates the SOA services in business processes. This phase results in two key characteristics: (1) the libraries cover and encapsulate all business requirements relevant to the scoped implementation project, and (2) the service and role specifications are validated regarding their feasibility in the physical layer.
- *Deploying SOA services and aligning organizations*: The deployment phase focuses on technical and organizational infrastructure. This includes SOA infrastructure and new and modified applications, as well as the business processes and required organizational changes. SOA deployment changes the way that businesses operate, which in turn, requires organizational alignment.

Note that this framework-based SOA methodology built on the four subsequent phases is not bound to a specific industry or a given process, information, or application framework. We explained the methodology in the context of TM Forum's Solution Framework (i.e., the business framework eTOM, the information framework SID and the application framework TAM), since it is well-established, well-accepted, and widely-adopted in the telecommunications, information, communication, and entertainment sectors. This framework-based SOA methodology helps enterprises structure, organize, and align their service entities in gearing towards transforming their enterprise services to the Cloud paradigm.

7.5.2.3 Dynamic Cloud Active Catalog

In today's business arena, enterprises want and need to launch new services in a short period of time in order to stay competitive in the market. Currently, the knowledge of service and product bundling is spread across many OSS and BSS systems used by enterprises. In addition, service provisioning from the Cloud platform not only entails decomposing the order received from self services, but also provisioning (i.e., activation/deactivation) E2E processes (semi) automatically and dynamically reconfiguring infrastructure resources.

An active catalog is a great tool for assisting enterprises in bundling, delivering, and provisioning services and products. An active catalog is the place where all of the service and product building blocks are modeled. For example, network equipment, applications, or even more abstract building blocks, such as work instructions and instructions for rating. These building blocks are modeled as "items" within the active catalog and can be assembled into service or product offerings that make sense to the customer or product manager. This is especially applicable in Cloud environments in the sense that the active catalog within an enterprise can serve as a systematic and organized shelf for Cloud services and products, which are abstractions/virtualizations of physical resources. In addition, an active catalog can also encompass services and products from a collection of Clouds, providing the customers with a comprehensive view and easy browsing/purchasing. More specifically, TM Forum's Active Catalog describes the interface requirements of multiple providers in detail [27]. The "active" in active catalog refers to the fact that while resource, service, and product items are created in a *Computer Aided Design* (CAD)-like fashion; each item definition includes full instructions for automated handling of the item. The item specifications understand or reference the processes that live within the traditional OSS/BSS stack needed for delivery and management of that item. For example, the activation process, the billing process, or the element managers that manage resources directly. Each existing system remains in control of its own specialist area, with its own service creation tools.

There has been an increase in demand for product/service convergence. Therefore, the time and cost to introduce and manage product/services need to be significantly reduced. In addition, customers continue to raise their expectations, which drives the need to operate more rapidly and effectively in increasingly complex environments. The current solutions cannot afford long product development cycles and there is an amicable need for open standards and technologies to facilitate interactions among Cloud SPs. There is an increasing need for a holistic model-driven approach for service development that (1) is able to align business, operations, and network all together; (2) is able to manage processes, rules, and data supporting the services; and (3) is standards-based in order to facilitate rapid partnering. The third capability refers to the fact that more Cloud platforms need to communicate relevant data to other Cloud platforms and/or mediation systems in order to enable business model flexibility. In conclusion, creating and implementing a holistic Cloud service model (such as shown in Fig. 7.14) enables flexible and systematic creation and deletion of services, creates customized service bundles, creates real time enforcement of eligibility and compatibility rules, and finally allows seamless integration with ordering processes.

The service model shown in Fig. 7.15 that consists of a product catalog, service catalog, and model components enables association of resources to products, regardless of the resource providing the services. This is done in addition to facilitating converged product offerings across technical and enterprise boundaries.

Figure 7.15 depicts Cloud product/service bundling, especially the decomposition for service offerings using active catalog. Note that the active catalog product consists of assembled service definitions from multiple sources and acts as a cen-

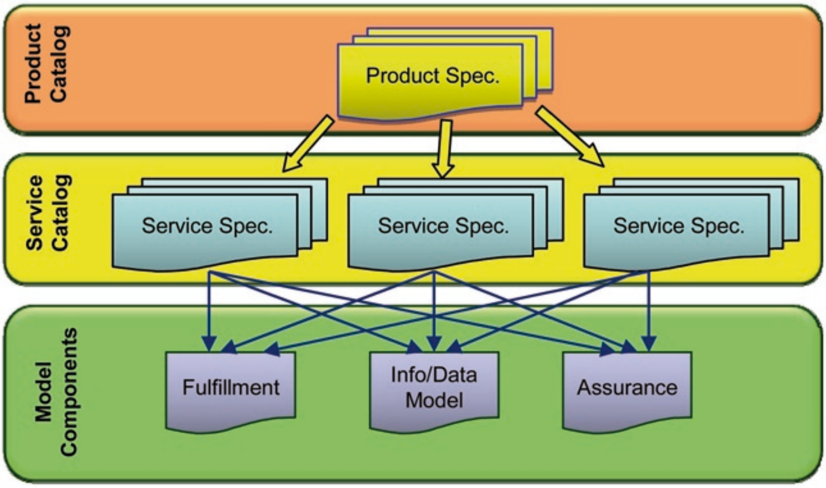


Fig. 7.14 Example holistic service model with product/service catalogs

tral catalog for the order management system. However, let us focus on the active service catalog for the sake of explanation. The active service catalog aggregates catalogs from different Cloud providers. In the service active catalog, service administrators within each Cloud provider create, modify, and delete services, while

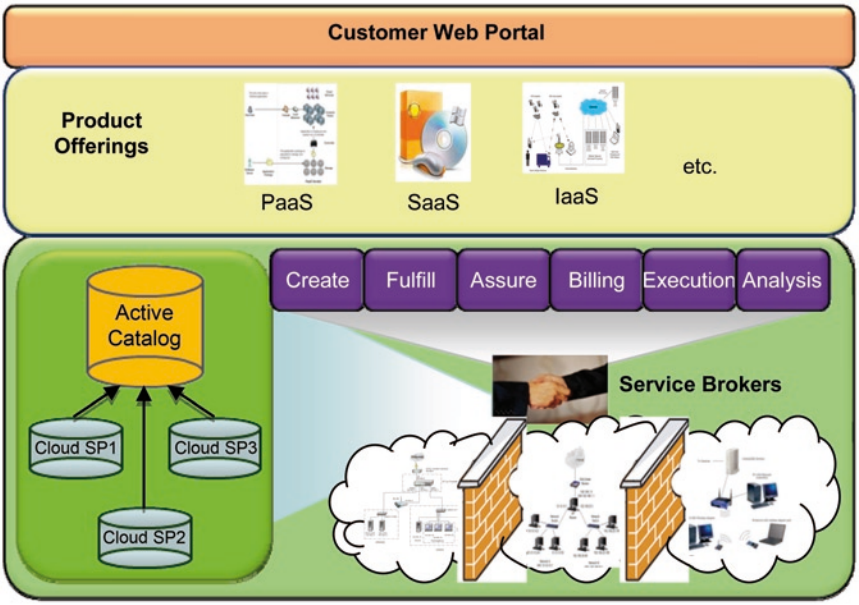


Fig. 7.15 Cloud active catalog with multiple Cloud providers

the active service catalog deals with the unique characteristics and interactions of different services. The active catalog in this case is effectively a database with well-defined structure and interfaces, exposing service brokers to manageability and inter-catalog messaging.

Service brokers need to understand the details in the service management functional definitions (e.g., fulfillment, billing etc.). Their responsibilities include, but are not limited to, orchestrating the execution of automated or manual workflows, interfacing to provisioning systems, performing service fulfillment and activation, interfacing with service assurance, determining the impact or interaction of services, and retrieving *Key Performance Indicators/Key Quality Indicators* (KPI/KQI) data, which are discussed in Chap. 8. In addition, as mentioned in Chap. 6, service brokers can also act like BPM, defining, enforcing, and monitoring policies, dynamically applying rules to data, and maintaining fault/alarm correlation rules or workflows.

7.5.2.4 Policy-Oriented Business and Risk Management

Policy within and across organizational boundaries has traditionally been embedded within enterprise IT platforms and applications. However, scaling businesses globally will require implementing new ways to combine and harmonize policies within and across external process networks and value chains. It will become increasingly critical for companies to establish clear and explicit definitions of governance, policy (regulatory, security, privacy, etc.), and SLAs if they are to operate effectively with diverse entities in the Cloud. Aspects of PBM are described in detail in Chap. 6.

7.5.2.5 Cloud Service Monitoring and Management

Cloud technologies make network and server infrastructure invisible. One of the big selling points of Clouds is not only outsourcing the provision of a scalable, enterprise-grade network, but also the necessity to manage it as well. A large part of existing network management is involved with making sure that the network and server infrastructure is working properly. The focus of network management in a Cloud environment will shift away from managing infrastructure to managing service availability and performance. In addition, root cause analysis will effectively come down to ring Cloud vendors' technical support team. Instead of the network management system tracing the root cause of outages, the enterprises have to rely upon the Cloud vendors' network management system instead.

To conduct business within a Cloud and recognize what is available today, it is important for Cloud consumers and providers to align on graduated SLAs and corresponding pricing models. Maturing Cloud capabilities into more advanced offerings, such as virtual supply chains, requires support for fully abstracted, policy-driven interactions across Clouds. This is a big jump and will become a major challenge for Cloud providers to adequately model, expose, and extend policies in order to provide integrated services across distributed and heterogeneous business processes and infrastructure. The data associated with these business processes and infrastructure will need to be managed appropriately to address and mitigate various risks from a security, privacy, and regulatory compliance perspective. This is particularly important as intellectual property, customer, employee, and business partner data flows across Clouds and along virtual supply chains.

The obvious place is to put the service-oriented monitoring in the Cloud right alongside the enterprise's applications. Whilst such an approach will work most of the time, repercussions must be taken into account when the Cloud vendor's network fails. All of the major Cloud vendors have had network outages so it is not a theoretical risk. An alternative to deploying an enterprise's own monitoring solution could be to use one of the many vendors promoting SaaS-based online monitoring solutions. Whilst that is probably going to be a more robust solution, it may be difficult to know precisely how the vendor has deployed their solution. One must consider if the vendor is using the same Cloud vendor that the enterprise has chosen. If so, it will not be any better than deploying the enterprise's own monitoring solution in the Cloud.

One of the side effects of managing a Cloud world is that vendors will need to be more open about their infrastructure arrangements. If an enterprise's management vendor is using a Cloud then they need to be open about it. Otherwise, there may be a danger that both management and managed services use exactly the same infrastructure.

7.5.2.6 Configuration Management

The first question we ask is whether or not it is possible to use a standard configuration management approach for Cloud technologies and virtualization. There is an increasing awareness that the existing notion of a CMDB is an unrealistic and problematic model, due to the fact that a CMDB is usually the central repository for all information about the datacenter and the decisions made for its management. With the sheer amount of information that enterprises expect it to hold and the requirement that it is always up-to-date, accurate, and complete, it is clearly impossible to expect seamless integrate of all the discovery information. To make matters even more complicated, with the myriad of domain management tools that span applications, servers, network devices and storage, we also have to sort out and answer the question of how the physical environment is related to the virtual environment.

One of the major issues with compute Clouds is the process of configuration management of the software image to be deployed. If the software running in the Cloud has a bug, then the enterprise needs to be able to revert to a previous image or upload a new one quickly. In addition, controlling when new software is deployed is likely to be very important. Enterprises cannot afford to wait around for an off peak time period to upload new software, it would be useful to have one's own network management system to do it on behalf of the enterprise.

Secure configuration in the Cloud requires controls for provisioning, administration, monitoring, validation, and management. PCI DSS recommends the use of industry-accepted hardening standards. Implemented as baselines, the ideal Cloud will offer on-demand instances that are pre-configured according to a specified baseline with tools for managing their configuration and detecting configuration drift.

Cloud providers "must protect each entity's hosted environment and cardholder data" [28]. These requirements extend PCI DSS configuration requirements for a shared environment and proscribe specific control requirements for segmenting each entity's data, identity, application, audit, and incident response capabilities. When looking at these together, one can quickly see components that must be managed and controlled by the entity, components that are up to the Cloud provider, and perhaps a few that require both. Configuration management in the Cloud will require a multi-tenant solution that addresses these PCI requirements with a methodology that supports self-service by the entity and a common control infrastructure managed by the provider [29].

REST in Configuration Management

As mentioned in Chap. 3, REST [30] greatly simplifies the world by eliminating the need for SOAP or *Remote Procedure Call* (RPC) protocols. It is not debated that this is a great architecture for large-scale systems. Most of the Internet runs on REST and is perhaps the best example of interoperability ever built. In this section, we examine REST in the configuration management domain. Even though configuration management may seem less trendy, it is just as useful if not more in understanding the practical value of REST for IT management. Especially in terms of IaaS, managing the configuration of everything that runs on top of the VMs remains as a main challenge.

At the first glimpse, REST seems to be ideally suited for Cloud configuration management. Applying REST to the task of retrieving configuration data from a CMDB or other configuration storage should be relatively simple, especially in the IT management world, where there are already explicit resource models and a rich set of relationships defined. The benefits of REST in the configuration management domain include:

- A URI-based scheme makes the protocol independent from the resource topology, unlike today's data stores that usually struggle to represent relationships between stores.
- It makes it trivial to browse the configuration data from a Web browser. The resources provide an HTML representation based on content-type negotiation, or a simple transformation could generate it for the Web browser.
- REST-induced caching and scalability.

Although RESTful Cloud APIs have no problem retrieving resource descriptions, they seem somewhat hesitant in the way of dealing with resource-specific actions.

In complex situations where actions may require some time to be executed or the required actions are not quite clear, applying REST is a lot less straightforward than performing document or data retrieval. As a matter of fact, there are quite a few things that RESTful configuration management does not solve:

- The ability to process queries involves multiple resources, thus no CMDB
- The ability to retrieve the configuration change history and compare configurations across resources or to a reference configuration

This is not to say that these two features cannot be built on top of a RESTful IT resource model, just that they are the real meat of configuration management rather than a simple resource-by-resource configuration browser. It is not necessarily true that a REST-style foundation makes these two features harder to implement. However, there are a few aspects one needs to consider before using REST. First, in hypermedia systems, the links are usually part of the resource representation, not resources of their own. In IT management, relationships/associations can have their own lifecycle and configuration properties and it is important to make sure it is possible to maintain the address of a resource. It is one thing to make sure that a UUID is maintained as a resource configuration change, it is another to ensure that a non-referenceable URI remains unchanged. For example, the administrative server of a cluster may move over time from one node to another. More fundamentally, the ability to deal with multiple resources at the same time and/or to use the model at different levels of granularity will remain a challenge. One solution is to make the protocol more complex or pollute the resource model.

Queries require information from multiple resources, as mentioned recently became a DMTF standard, called CMDBf, which is discussed in detail in the next section. CMDBf is SOAP-based and does not have too much association with REST. CMDBf is mostly a query interface and is more about CMDB interoperation than federation. There are a number of things in the query operation that can be made RESTful. REST can make the discovery/reconciliation tasks of the CMDB more efficient. The CMDBf query result format can be improved so that from the returned elements, one can navigate among resources by following hyperlinks. The query operation itself looks fundamentally RPC-like. This is similar to an interaction with the Google search page, which is really a RPC call that happens to return a Web page full of hyperlinks. In a way, this query (whether Google or CMDBf) can at best be the transition point from RPC to REST. It can return results that open a world of RESTful requests while the query invocation itself is not RESTful.

CMDB and Configuration Management

In July 2009, several industry forefront companies announced the release of a Cloud standard [31]. CMDBs give IT organizations complete visibility into the attributes, relationships, and dependencies of the components in their enterprise computing environments. The federation standard provides a way for accessing IT information in CMDBs distributed across multiple repositories to create a more complete and accurate view of IT information spread out across multiple data sources.

The service model stored in the CMDB holds all the tangible and intangible IT infrastructure items that support the service and the relationships that exist amongst them. Without extensive federation, configuration management can be complicated and enterprises are forced to restrict the scope of their CMDB projects and limit the realization of their potential value. For those using ITIL as a framework to manage IT, the CMDBf can help track service assets and configuration management data.

Some of the challenges that must be addressed include: how to build a service model; whether some or all of the information about the underlying components are stored in radically different data sources; and how to maintain the current model after it is built. Until an organization can address these challenges, ITIL *Service Asset and Configuration Management* (SACM) may be daunting. On the other hand, CMDBf focuses on a standards-based approach to data access in support of the SACM.

A CMDB can help predict Cloud usage. Knowing what the enterprise has and how it is being used is the very first step. Using a CMDB to manage assets and store information, even federated, allows for an initial baseline analysis of where the enterprises are today. Enterprises continue to work on better CMDBs. For example, in 2009, BMC Software highlighted the importance of CMDBs with the release of its Business Service Management platform. Rackspace also announced an open, standards-based API for The Rackspace Cloud. This API can deliver data about a VM instance, relate files to it to create a server, ensure that a customer's VMs do not congregate on one physical host, and create shared IP groups to ensure high availability. In addition, Rackspace Cloud servers have access to local and disk storage, much like one would expect in a physical server. The Rackspace Cloud is also the only services suite where one can get a Cloud, dedicated hosting options, or unique hybrid hosting offerings.

One of the biggest challenges of Cloud configuration management is how to implement a CMDB using virtualization and Cloud computing, and how to leverage federated CMDBs to support specialized devices such as routers and network devices. Maintaining the mappings between the virtual configuration and the physical configuration is one of the key features to implementing the CMDB in the Cloud. Currently, open industry standards are making the effort to bring in information from distributed sources and help achieve success in implementing CMDBs much easier. The essence of the Cloud is the fact that it is dynamic. There are virtual configurations in the Cloud, and users/consumers of Cloud services have no idea how that virtual configuration maps to the physical configuration. This is precisely why enterprises want to have a Cloud in the first place, so users only need to be concerned with virtual configuration. In order to handle that in the CMDB, enterprises must distinguish both and maintain a map between the physical and virtual resources.

Usually, this is the way enterprises want their services to be supported, which means planning ahead of time using the human time scale. On the other hand, mappings must be made on the electronic time scale. Federated CMDB, different tools for managing IT infrastructure, vendors' network management systems, asset management tools, special management tools, etc., all have information on configuration. Pulling information from diverse sources such as different systems, devices, and programs, and putting them into one single federated CMDB, is the standard.

7.6 Conclusion

Although the concept and current practices of Cloud technology show great potential, it is still very much in its infancy. Clouds in their current form are more likely to appeal to enterprises that are considering shared or VM hosting, for example Google Apps, or one or more dedicated servers, for example Amazon EC2. In either case, neither would be in the market for network management systems.

After the early adopters have ironed out various problems, the next wave of Cloud technology adopters will be from enterprises replacing small or medium-sized datacenters with a computing Cloud. These enterprises will still require a management system no doubt, but it will be more tuned to managing a Cloud environment, not a datacenter environment. The management systems will need to monitor the services the enterprises provide and manage their interactions with the Cloud.

On the other hand, Cloud vendors themselves will require full-blown, enterprise–grade, network management systems. A great opportunity is being presented to vendors who are able to quickly fine-tune their products to the particular requirements of Cloud vendors. The open source, enterprise–oriented, network management vendors will find that their offerings mesh well with Cloud vendors, as Cloud vendors are already heavy open source software users.

References

1. Gubbins, E.: Linesider launches to automate Cloud configuration. http://connectedplanetonline.com/service_delivery/news/linesider-launches-cloud-101909
2. Glossary. MicroConvergent. 2009. http://www.microconvergent.com/glossary.html
3. Configuration Management. Shavlik Technologies. http://www.shavlik.com/info/configuration-management.aspx
4. Winans, T.B., Brown, J.S.: Cloud Computing, a collection of working papers. Deloitte. May 2009
5. Van Alstyne, M.: The state of network organization: a survey in three frameworks. J. Organ. Comput. Elect. Commer. **7**(2, 3), 83–151 (1997)
6. Mell, P., Grance, T.: Effectively and securely using the cloud computing paradigm. NIST, Information Technology Laboratory. 3 June 2009
7. Systems Management: Centralization, automation and administration. http://www.fr-tech.net/systems-mgmt.php
8. Cloud Computing, Information security briefing. Centre for the Protection of National Infrastructure. May 2010. http://www.cpni.gov.uk/Docs/cloud-computing-briefing.pdf
9. Papazoglou, M.P., van den Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. VLDB J. **16**(3), 389–415 (2007) (Springer Berkin/Heidelberg)

10. Ryan, P.: Today's challenges, a roadmap for America's future. www.roadmap.republicans.budget.house.gov/plan/challenges.htm
11. SOA Market and Products 2006: Current state, future directions. Enterprise Management Associates. April 2006
12. IT Next. March 2010 Issue. http://www.slideshare.net/shashwatdc/it-next-march-2010-issue
13. Private Cloud Computing for enterprises: meet the demands of high utilization and rapid change. Cisco whitepaper. http://www.cisco.biz/en/US/solutions/collateral/ns340/ns517/ns224/ns836/ns976/white_paper_c11–543729_ns983_Networking_Solutions_White_Paper.html
14. Chong, F., Miguel, A., Hogg, J., Homann, U., Zwiefel, B., Garber, D., Joseph, J., Zimmerman, S., Kaufman, S.: Design considerations for software plus service and Cloud Computing. MSDN Architecture Center. http://msdn.microsoft.com/en-us/architecture/aa699439.aspx
15. From Cloud Computing to new enterprise data center. IBM Software Group. 2008. www.dataline.com/FromCloudComputingtoDataCenter.pdf
16. Toward private Cloud Computing. NetLINK. http://www.netstarnetworks.com.au/news/netlink_2009_Sep.asp?Cloud (Sept 2009)
17. The CMDB distributed management taskforce (DMTF)—a standard for connecting CMDBs and MDRs. The BSM Review Blog. http://www.bsmreview.com/blog/2009/11/the-cmdb-distributed-management-taskforce-dmtf---a-standard-for-connecting-cmdbs-and-mdrs.htm (2009)
18. Glodman, A.: Cloud Computing firms: yes to open standards. http://itmanagement.earthweb.com/netsys/article.php/3830701/Cloud-Computing-Firms-Yes-to-Open-Standards.htm
19. Wan, S.H.C., Chan, Y.H.: Improving service management in outsourced IT operations. J. Facil. Manag. **5**(3), 188–204 (2007)
20. Business Process Framework Concepts and Principles, TMF GB921, Release 9.0. TM Forum. 18 Aug 2010
21. Chang, W.Y.: Integration of defense applications into the TM Forum framework—system of systems roadmap. TM Forum Whitepaper. http://www.tmforum.org/WhatsNew/IntegrationofDefense/37008/article.html
22. Information Framework Concepts and Principles, TMF GB922, Release 9.0. TM Forum. 1 April 2010
23. Amrhein, D., Willenborg, R.: Cloud Computing for the enterprise: using WebSphere to create private clouds. IBM WebSphere Dev. Tech. J. (June 2009)
24. Hughes, J.: How will cloud computing change network management. The Tech Teapot. Sept 2008. http://www.openxtra.co.uk/blog/how-will-cloud-computing-change-network-management/
25. The business process framework, for the information and communications services industry, TMF921. TM Forum. 11 March 2010
26. Chou, T.H., Lee, Y.M.: Integrating E-services with a telecommunication E-commerce using service-oriented architecture. J. Software **3**(9), 60–67 (Dec 2008)
27. Active Catalog NGOSS Contract v1.0.1, TMF 211. TM Forum (2009)
28. PCI-DSS 2.4. http://pcidssfaq.org/forum/showthread.php?t=34
29. Berman, M.: PCI Configuration management in the Cloud. Cloud Slam Event. Dec 2009. http://www.cloudslamevent.com/pci-configuration-management-cloud-part-c
30. Representational state transfer. Wikipedia. http://en.wikipedia.org/wiki/Representational_State_Transfer
31. Configuration Management Database (CMDB) federation specification. DMTF. June 2009. http://www.dmtf.org/standards/published_documents/DSP0252_1.0.0.pdf

Chapter 8
Service Monitoring and Quality Assurance$_2$

Enterprise administrators and users can trust Cloud Computing services only if the quality of the Cloud services is superior to what administrators and users expect from their existing private enterprise networks. For this reason, service monitoring and quality management are two key functional areas that can provide the metrics that assure the success of transforming enterprise networks.

Chapter 2 explains that, in order to ensure a successful transformation, enterprises need to incorporate quality management and monitoring agents in the service architectures. Chapter 5 further discusses SLAs and explains that they are an important part of a CRM program. SLAs are only meaningful in the context of customer experience. In addition, Chap. 5 discusses services and SDFs. Thus, this chapter ties these concepts together and discusses how service monitoring and quality assurance can be implemented in enterprise networks to ensure that they can take advantage of the services and capabilities offered by Cloud providers while enhancing end user and customer experiences.

8.1 Overview

Integrating service monitoring of Cloud services with the internal processes of enterprise monitoring requires careful design and planning. Often times, enterprise monitoring services are proprietary and offer limited interfaces to outside agents. Therefore, transforming enterprises to use Cloud services requires the use of monitoring agents that interface between the enterprise networks and Cloud services. The agents must be designed to distinguish between service levels that are offered by Cloud services and service levels that are offered by the enterprise networks themselves. The agents then need to react to quality levels in appropriate ways to ensure that Cloud services deliver the agreed upon quality metrics to the enterprise and then manage these levels to comply with the enterprise's own commitments.

In addition to monitoring service quality, agents need to manage enterprise assets. These asset agents keep an inventory of dynamic and static assets that are currently in use or can be provided to requesting users and services. The asset agents

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_8, © Springer Science+Business Media B.V. 2010

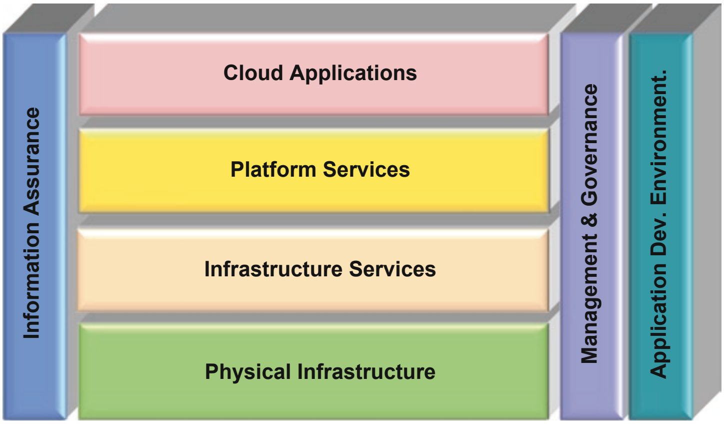


Fig. 8.1 Topics covered in Chap. 8

also work with monitoring agents to request and assign resources in a dynamic or static fashion to maintain the quality levels of service that are needed.

This chapter discusses the key aspects of specification of service and quality levels; the interface between service levels and monitoring and asset agents; multi-level, multi-vendor, SLM interfaces; enforcement of SLM; and reporting. The Management & Governance box in Fig. 8.1 contains the topics covered in this chapter.

8.2 Enterprise Quality and Performance

SLAs are meaningful to enterprises as the agreements relate to the provider and consumer relationship. When consumer is a service customer, the customer experience is a key driver of SLAs. This topic is covered in detail in this section.

8.2.1 Service Level Agreements, Enterprises, and Customer Experiences

SLAs have been a common product in support of services offered by telecommunications SPs for many years [1]. As discussed in Chap. 5, SLAs define agreed performance and QoS or product metrics and are an important part of a CRM program. Achieving quality and performance targets for the products or services may require an enterprise to establish and manage a number of SLAs. The complexity of global services brings together a myriad of services, suppliers, and technologies, all with potentially different performance requirements. Thus, the goal of enterprise SLAs is to improve the *customer experience* (CE) of the service or product to the enterprise clients, whether they are internal or external to the organization. CE is a collective term to form a measure of the quality of a service or product and includes all aspects of service: its performance, level of customer satisfaction in the total experience, pre and post sales, and the delivery of its products and services. Determining the CE provides a discriminator between various types of service or products that an enterprise provides, and leads to opportunities to balance the level of quality offered against price and customer expectation [1].

The relationship between a CE and SLA is that the CE relates to the perception of the quality of a product or service, whereas an SLA refers to the definition, measurement, and reporting of objective measures of the service or product. As such, the CE and the SLA are related in that if the perception of the service or product is poor, yet the service parameters fall within the limits defined by the SLA, the SLA must be redressed. The key concept is to map the perceptive measures from the CE into objective measures for the SLA. This mapping may be multidimensional, empirical, functional, or complex in nature.

In support of enterprise or business applications, business services facilitate the applications. For example, in a call center (the application), an obvious business service is voice communications. Business services in themselves usually do not raise revenue, but they support business objectives and the effectiveness of a business application. One or more business services may be necessary to support a business application. Business services in turn use a number of service resources, such as network services. In particular, business services that support Layer 4 and below in the OSI model are categorized as network services. Network services may be either supported internally or outsourced to external providers, such as Cloud providers.

Figure 8.2 shows an example end-to-end SLA. Important issues to note in the figure are that business applications, e.g., a call center or online stockbrokers, are supported by a number of business services, e.g., voice and databases, that in turn

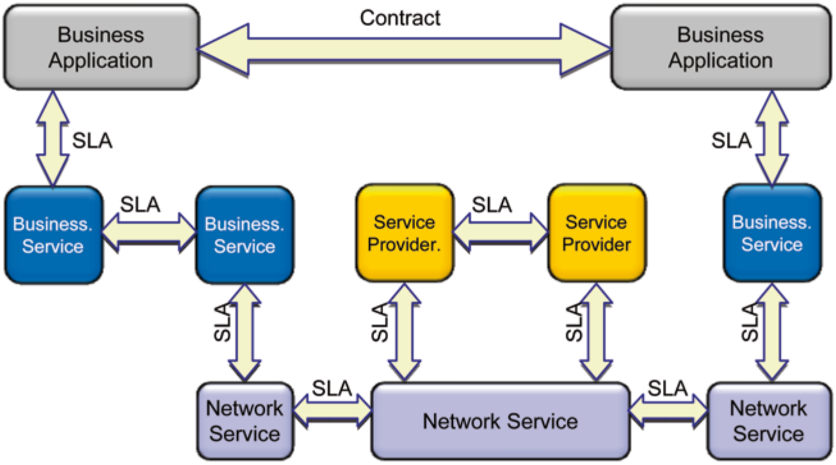


Fig. 8.2 Example end-to-end SLA

are supported by network services, e.g., IP. There may be internal network services or external network services from Cloud providers. Some business services, such as shipping, do not require network services to fulfill their business applications, such as parcel delivery, but increasingly rely upon network services to provide value added services, such as online parcel tracking.

For an SLA to add value in providing business applications for an enterprise, the enterprise infrastructure needs to be instrumented adequately so that metrics can be determined to ensure conformance, prevent or warn of non-conformance, and measure non-conformance. An audit log may also be necessary for capacity planning, cost control, and dispute resolution.

Typically, implementing or monitoring services or products requires the application of service functions and resources in a relationship similar to that shown in Fig. 8.3. *Service functions* allow services to be physically implemented and can be decomposed into three main functional areas:

- *Primary functions*: It implement the primary service. Email is an example of a primary function.
- *Enabling functions*: It allow the primary function to be implemented. Examples of enabling functions include OS, and *Heating, Ventilation, and Air Conditioning* (HVAC).
- *Support functions*: It support the primary and enabling functions. Examples of support functions include accounts, help desk, operations, administrations, and maintenance.

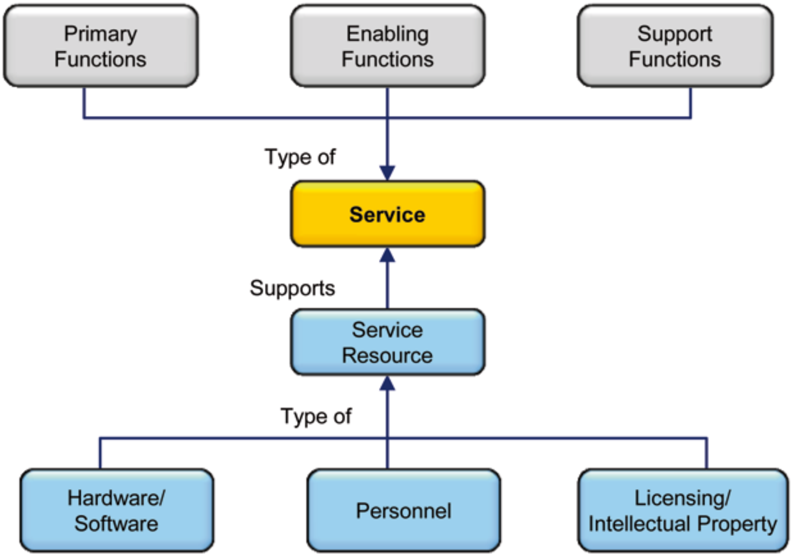


Fig. 8.3 Service functions and resources supporting a service

On the other hand, *resources* include such assets as hardware, software, personnel and training, licenses and intellectual property, facilities, and budgets. Section 8.3 below discusses service functions and resources in further detail.

8.2.2 Key Quality Indicators and Key Performance Indicators

There is a difficulty in mapping service-specific parameters to technology-specific parameters that are more easily measured and reported. As a consequence, traditional SLAs have focused almost solely on the performance of the supporting service. By contrast, KQIs and KPIs focus on service quality rather than network performance. KQIs and KPIs provide measurements of specific aspects of the performance of applications or services. A KQI is derived from a number of sources, including performance metrics of a service or underlying support service KPIs. As a service or application is supported by a number of service elements, a number of different KPIs may need to be determined to calculate a particular KQI. The mapping between the KPI and KQI may be simple or complex, and the mapping may be empirical or formal [1].

Being subjective, some KQI parameters can be difficult to include as a contractual requirement in an SLA. Nevertheless, there are a number of KQIs that relate to a CE and should be included in an SLA. To meet these KQIs, a number of KPIs must also be defined, measured, and agreed on in the SLA. These relationships are depicted in Fig. 8.4.

In Fig. 8.4, KPIs can be defined in SLAs, and KQIs are derived and monitored during a SLM process. Each KPI or KQI has a lower and upper warning threshold and a lower and upper error threshold, as shown in Fig. 8.5. The KPIs are then

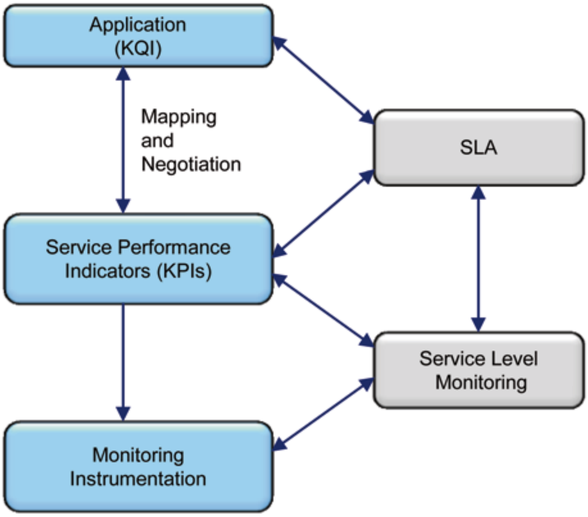


Fig. 8.4 Relationship among SLA, KQI, and KPI

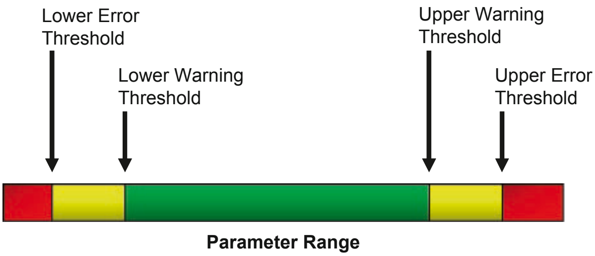


Fig. 8.5 KPI and KQI parameter thresholds

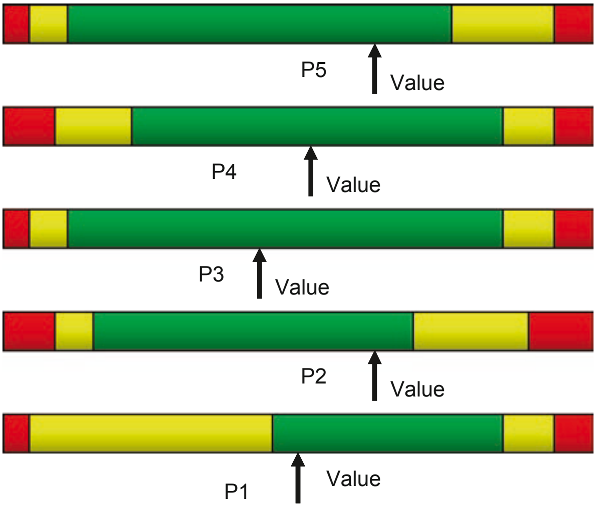


Fig. 8.6 Combining KPIs to determine a KQI

combined by some empirical or theoretical function to lead to a measure of KQIs as illustrated in Fig. 8.6.

The importance of the warning thresholds from Fig. 8.5 can be seen in Fig. 8.6, as in some instances a single indicator in the warning zone may indicate that an SLA threshold may be violated for a particular KPI; a collection of such KPIs in the same state may indicate a violation of a KQI threshold.

The exact form of the function linking KPI to KQI is an important concept for SLA negotiation. There has been considerable work in determining the functional relationship between KPIs and KQIs by standards bodies such as TM Forum and the International Telecommunications Union (ITU) and by commercial entities. If no relationship can be determined, measurements can, in real or laboratory environments, determine the relationship.

8.2.3 *Sample Key Quality Indicators and Key Performance Indicators*

In determining enterprise SLAs, it is important to determine the KQIs for the applications or services and then map them to KPIs that can be used to measure the KQIs.

SLAs reflect the most rigorous requirements for KQIs and KPIs for all services supported by the SLAs. SLAs are further discussed in Sect. 8.5 below.

Examples of generic KQIs include the following:

- *Availability*: It measures whether a service is available for use at the time required. As a KQI, this includes all aspects of the service, physical terminal availability, network, etc.
- *Speech/Video Quality*: It measures whether speech or video has sufficient quality such that, within the context of the application, the information can be conveyed and interpreted in an audio or visual form. Information may include inclination, expression, body language, and content.
- *Response Time*: It measures how quickly a service responds to an internal or external stimulus.
- *Round Trip Delay*: It measures the time lapse between making a request and seeing the response. This includes network round trip delay, client and server processing delay, and any manual intervention in the system (like servicing a work order).
- *Delay*: It measures one-way delay in the system. Delay may be different in the forward direction than the return direction.
- *Jitter*: It measures variation in delay over time. The period over which the delay variation is to be measured should be understood and defined.
- *Locking Information*: It assesses whether information is locked for read or write to ensure integrity of the data and to inform others that the information may change.
- *Transaction Rate*: It measures the rate that the system or service can service requests. Burst rates, sustained rates, and their periods should be defined along with how the system or service reacts when presented with transaction rates higher than the value required.
- *Goodput*: It measures the amount of valid information that is carried by the system and processed by the customers. For voice systems, this represents the total amount of voice traffic serviced, i.e., total calls minus blocked calls. For data applications, it is the total data, minus errored data, minus lost data, minus retransmitted data.
- *Throughput*: It measures the total amount of information that is offered to the system. Throughput includes all processed information, including retries and replications. For voice and data systems, this represents the total amount of traffic presented to the system, but not necessarily serviced. For example, throughput includes lost calls, and retransmitted and errored information.
- *Idle Time*: It measures the amount of time that a system or service is idle, i.e., not performing a service or request.
- *Authorization*: It measures metrics related to authorized resource access at allowed times.
- *Confidentiality*: It measures metrics related to ensuring that data can be seen by only those authorized to see it.
- *Integrity*: It measures metrics related to ensuring that data is available as required and has not been changed from the original. Integrity includes loss of service due

to denial of service (DOS) or system failure. Loss of integrity implies either loss of information or a change of information.

- *Non-repudiation*: It measures metrics related to ensuring that data has come from the source shown in the data and is a valid, authorized source.
- *Disk Space*: It measures whether space is available to service requests for the duration of the service request.
- *Help Desk*: It measures whether a help desk is available to handle information requests. Requests may include information about the service, support, etc.
- *Training*: It measures whether training is sufficient to perform the required task, including the use of services, responsibilities of users and providers, etc.
- *Interoperability*: It measures the degree to which a service or product inter-works with all systems and services required.
- *Pick-up Time*: It measures how long it takes a human to respond to a request, normally by voice or video phone.
- *Time to Close*: It measures how long it takes to close, to the user's satisfaction, a support or information request.
- *Hold Time*: It measures the time a support or information request is held in a queue without being processed.
- *Connect Time*: It measures how long a service takes to start.
- *Graceful Degradation*: It measures the degree to which a system or service in a controlled and gradual manner degrades when the system is overloaded.
- *Revocation or Termination*: It measures the speed of recalling authorization to use a service or product.

The particular KQIs for an enterprise depend to some extent on the enterprise's objectives and business rules. However, only KQIs pertinent to the business service should be considered. For example, in telecommuting, only those KQIs that represent a positive CE in gaining access to the enterprise systems are considered, as it cannot be generalized as to how or what applications the telecommuter wishes to use. If, for example, the end user wishes to use email in a telecommuting environment, the full service would contain the KQIs for both telecommuting and email.

8.2.4 Quality Equations and Measurement

There has been much research in the development of standard equations that provide quality measurements from performance-related data. These equations can be used to model a network before it is deployed, assign values for an SLA contract, and perform analysis of data to predict the performance enhancement or degradation due to changes in the service, such as the addition of a route controller or a move from narrowband to broadband connections. These equations can be used to determine thresholds and sensitivity analysis of PKI parameters for SLA monitoring and reporting.

The Open Group *Application Resource Measurement* (ARM) model allows applications to be instrumented to allow the performance and availability of single-system and distributed applications [1]. These may be visible to the users of the business application and those within the IT infrastructure, such as client/server requests to a data server. ARM establishes transactions that are meaningful within the application. Typical examples are transactions initiated by a user and transactions with servers. Applications on either client or server machines call ARM when transactions start or stop. The agent in turn communicates with management applications, which provide analysis and reporting of the data. The management agent collects the status and response time, and optionally other measurements associated with the transaction. The business application, in conjunction with the agent, may also provide information to correlate parent and child transactions.

The standard measurement for user perception of voice call quality is the *Mean Opinion Score* (MOS). While the MOS is useful and certainly valid, it is subjective and not easy to measure. In an effort to remedy this, ITU has developed the E-Model standard, specified in the ITU G.107 [2] and G.108 [3] standards, as a means of objectively measuring call quality. The output of an E-Model is called an R-value, with a value between zero and 100. This has been shown to reliably map to an estimated MOS. The E-Model takes account of impairments that lead to speech degradation and includes impairments that typically occur in packet-based networks. In the E-Model, impairment values are assigned to a number of independent parameters, which are then combined to give a transmission rating factor R as follows:

$$R = Ro - Is - Id - Ie + A,$$

where:

- Ro represents the signal-to-noise ratio, including noise sources such as circuit noise and room noise.
- Is is a combination of impairments which occur simultaneously with the voice signal. This includes loudness, sidetone, and quantizing distortion from analogue-to-digital conversions. This also includes impairment by packet loss.
- Id represents impairments caused by delay and includes talker and listener echo and end-to-end delay.
- Ie represents impairments caused by low bit rate CODEC. This includes effects from packet jitter and loss.
- A represents an advantage factor to compensate for impairment factors when there are other advantages such as mobility.

To ensure high speech quality, the following KPI measurements may be appropriate: *delay*, i.e., end-to-end packet delivery delay; *jitter*, i.e., variations on packet delivery times; *packet loss*, i.e., percentage of packets dropped during transmission; and *CODEC selection*. For packet-based technologies, the R-value can therefore be determined from the E-Model by measuring these KPIs. Similarly, response models have been developed for *Internet Protocols* (IP) to predict and measure performance. A transmission rating factor R for IP can be written as follows:

$$R = 2(D + L + C) + (D + C/2)((T - 2)/M) + D * ln((T - 2)/M + 1) \\ + max(8 * P * (1 + OHD)/b, D * P/W)/(1 - sqrt(L)),$$

where:

- B is the minimum line speed in the path, where the speed is in units of bits per second
- C equals CC plus CS, where CC is the client processing time measured in seconds, and CS is the server processing time measured in seconds
- D is the round trip delay measured in seconds
- L is the packet loss measured as a fraction
- M is a multiplexing factor
- OHD is the overhead fraction
- P is the payload size measured in bytes
- R is the response time measured in seconds
- T is the number of application turns
- W is the effective window size in bytes

The transmission rating factor for IP depends on a number of factors:

- Application design, which defines the number of turns between the client and the server
- Client processing time, which depends on processor load, processor speed, and client application design
- Server processing time, which depends on processor load, processor speed, and server application design
- Payload, which depends on application design and the request made
- Effective windows size, which depends on the configuration of the network, client, and server
- Packet loss, round trip delay, and line speed, which depend on network characteristics and performance

Note that the transmission rating factor for IP implies that the response time of an application requires KPIs from the network services to be combined with KPIs for the servers and clients. ARM can then be used to measure client and server response times.

8.3 Service Quality Management

Digital media services industries deliver services through a value chain or an ecosystem of cooperating partners and SPs. Delivering high quality customer experience over complex value chains supported by an ecosystem requires the cooperating partners to measure customer satisfaction, police SLAs, pinpoint problems across the value chain or ecosystem, and apportion payments whilst maintaining security. Thus a *Service Quality Management* (SQM) framework needs to define a holistic framework for measuring and effectively managing service quality; key service quality metrics at each point along the service delivery network; service quality issues and the necessary accounting and rebating information, usage information, and problem resolution information; management capabilities to support each step in the service delivery network; and appropriate interfaces and API's to enable the interchange of such information electronically between the various providers in a service value chain [4].

In a value chain, each relationship between a provider and a customer can be modeled as a customer having specific needs. These needs are generally captured in some form of an SLA. SLAs are discussed in greater detail in Chap. 5.

In a Cloud environment, enterprises expect the best possible quality from Cloud providers so as to pass a similar quality to the enterprise customers. Furthermore, new services are expected to have greater complexity in the end-to-end service delivery chain than current services. The quality of experience that a customer perceives depends on many factors, such as behavioral and image factors, marketing, components that set up the service, business processes related to the service, resources on which the processes are supported, and the performance of the underlying network and applications. Thus, to quantify the perceived quality of experience, SPs should know key customer needs metrics for measuring CE, KQI, and KPI for networks and services [4].

Chapter 5 discusses services and SDFs. Management models for networks and IT services expose resource models that are closer to a service view than to a simple exposure of the detailed components that are used to realize the service. By using the resource models, the management models can provide capacity forecasting and planning services, resource service provisioning that take into account CE and targets, assurance services including SLA violation alerts and threshold crossings, and usage and billing services.

The following subsections discuss these ideas further.

8.3.1 Value-Chain SQM

From a value chain viewpoint, an SQM supports a set of APIs and metrics that allow collaborating partners, such as an enterprise and its Cloud provider partner, to collect, process, and exchange information. This data is used to manage and report the end-to-end service quality offered to an end user at service access points and support the management of SLAs amongst partners and end customers. Figure 8.7 shows the essential elements of the value chain viewpoint of SQM. The figure depicts several applications as follows: [4]

- *CRM Applications*: These applications hold information about customers and the relationships or groupings among them. The applications also have a history of customer incidents and metrics assessing customer satisfaction.
- *Value added CE/SQM Applications*: These applications may take various forms. One form is for the applications simply to aggregate information from various sources and display them in a consistent form over a consistent time interval on a management dashboard. The objective is to flag the relative importance of

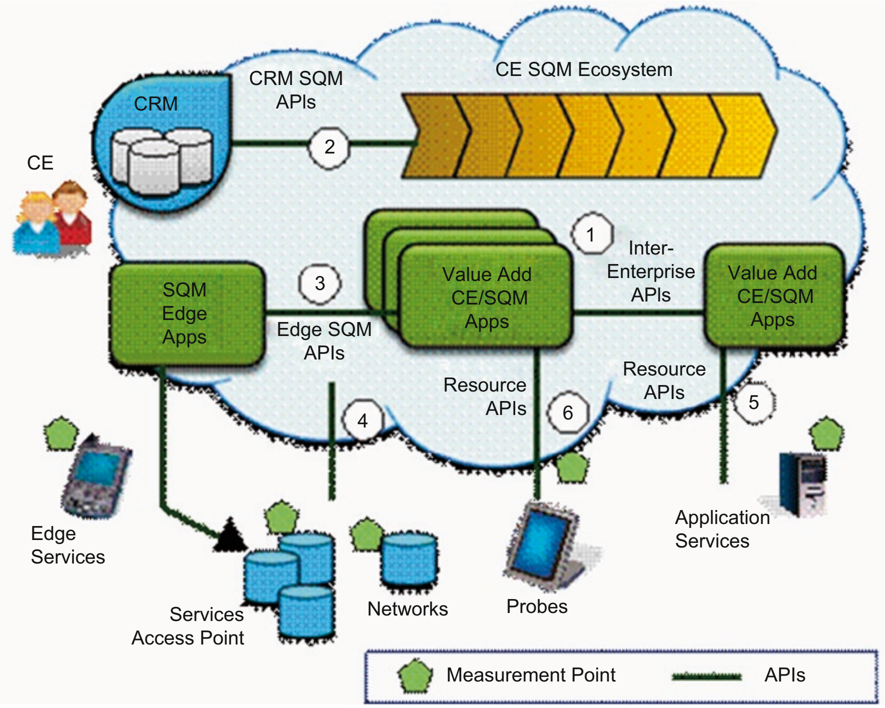


Fig. 8.7 Elements of value chain viewpoint of SQM

incidents and to observe trends. Another form is for the applications heuristically or algorithmically to process resource measurements to predict service performance, e.g., performance of product features. Yet another form is for the applications to correlate and optimize resources and derived services measurements against the customer incident history in order to optimize customer service actions. The objective is to improve customer satisfaction and gauge CE metrics.

- *Resource Management Applications*: These applications provide services that abstract networks, IT applications, and IT resources.
- *Edge Application Probes*: These applications support proactive monitoring of the service experienced by customers at service access points.
- *Network Probes*: These applications monitor the technical performance of *network* resources and provide diagnostic functions. For example, these functions monitor network quality, such as error rate or latency over several integration periods, flag when moving performance measurements exceed specified thresholds, apply diagnostic test conditions, and report back results of diagnostic tests.
- *Application Probes*: These applications monitor the technical performance of *application* resources and provide a set of standards monitoring and diagnostic functions similar to those for networking. It may include measures of application delay, processor, and storage utilization.

Figure 8.7 shows sets of APIs that correspond to various interfaces, as follows: [4]

- *Interface 1 corresponds to an inter-enterprise CE interface*. It includes the communication of SLA agreements between two cooperating partners and the exchange of CE metrics, events, and reports. Example APIs include APIs to order services and specify SLA targets; trouble-to-resolve APIs (SLA violation events, SLA jeopardy events, performance metrics reports, usage reports, trouble tickets, diagnostics, etc.); and concept-to-market APIs (plan forecast service capacity, policy business rules, capacity exhaustion events, capacity metrics report, etc.).
- *Interface 2 corresponds to a CRM CE interface*. This provides information about customers and their relationships including group memberships. It also provides CE and service quality incident records that are used in the assessment of SLA problems. Moreover, it facilitates the means to flag and communicate SLA events and jeopardy to customers and facilitates initiating customer rebates and settlements. Example APIs include business intelligence APIs (request user details and request group details); CRM customer information APIs (SLA violation events, SLA jeopardy events, performance metrics reports, usage reports, etc.); and billing rebates APIs (request billing change, rebates, etc.).
- *Interface 3 corresponds to an edge device CE interface*. This allows proactive monitoring of the service experienced by the end user at a service access point and can also be used to proactively test the service via KQIs and KPIs. Example APIs include set APIs (set URL or channel monitored, set reporting interval, set monitoring profile, etc.); get APIs (get channel name, get encoded profile, get average throughput, etc.); and service APIs (reconstruction of detailed records, generation of network traffic and usage measurements, monitoring of QoS and time parameters such as delay and jitter, real-time events, and alarms).
- *Interface 4 corresponds to a resource network CE interface*. This includes support for resource service provisioning requests against a service model; network performance measurements, possibly via probes setting policies and SLAs; CE and SLA reporting and threshold crossing events; and capacity forecasting and planning services. Example APIs include resource or network service provisioning requests against a service model; trouble-to-resolve APIs (network performance measurements, possibly via probes, to be communicated to the SQM Value Added applications, and CE or SLA reporting and threshold crossing events); and concept-to-market APIs (setting of policies and SLAs, capacity forecasting and planning services, capacity exhaustion events, capacity usage metrics, etc).
- *Interface 5 corresponds to an application's CE interface*. This allows performance measurement of IT resources and IT applications and infrastructure. Example APIs include IT applications and infrastructure resource service provisioning requests against a service model; trouble-to-resolve APIs (setting of policies and SLAs, capacity forecasting and planning services, capacity exhaustion events, capacity usage metrics, etc.); and concept-to-market APIs (network performance measurements, possibly via probes, to be communicated to the SQM Value Added applications, and CE or SLA reporting and threshold crossing events).
- *Interface 6 corresponds to probe CE Interfaces*. This provides a general set of capabilities controlling active, passive, and agent-based probes.

8.3.2 SQM Metrics

SQM metrics are used currently to create inputs to monitoring and analysis applications that drive enterprise or SP dashboards in support of service and customer management functions. Metrics aimed at dashboards need only to show trends reliably, e.g., upwards and downwards and do not need to be absolutely correct, i.e., some measurement artifacts or defects can be tolerated [4].

When exchanging metrics between enterprises and Cloud providers, however, the metrics need to be defined to the level where measurements carried out by one organization with one tool are directly comparable to measurements carried out by another organization using a different tool. Not only does this imply defining the measurement method, but it also requires calibrating both the tools and the organizations to be sure that the measurements created are comparable.

CE and quality metrics are collected in very high volumes and, for practical reasons, it is necessary to summarize them in some suitable form. Typically, summarization metrics represent average values, percentage of time or value measures below or above a threshold, and general percentages. Usually, these metrics are used to support operational functions such as network operations and service operations for customer service representatives and product managers, where these statistical summarizations are used in reports and on line dashboards to establish trends and patterns. Where metrics from individual organizations are exchanged across interfaces to estimate overall end-to-end performance in a value chain, as opposed to directly measuring with probes, the metrics need to be presented differently than traditional averages and thresholds. For example, if service or product feature availability metrics need to be computed from several resource availability measures, as seen in the SDFs discussed in Chap. 5, then the computation needs information about dependencies between resources and products or service features. Likewise, it needs resource resilience mechanisms to prevent one resource's failure from impacting exposed product features, conditional probabilities among events from these resources, such as the degree of independence of statistical events, etc. Valid statistical calculations need distributions to be characterized and conditional probabilities to be estimated. Ideally, raw data is provided but practically some data reduction may be needed.

To compute overall end-to-end averages from individual sub-domain averages, additional information is needed. This includes comparability of measurements such as benchmarking evidence measurements; estimators for probability distribution functions by using methods such as specific probability distributions (*Binomial*, *Poisson*, *Normal*) or generalized distributions (*Kurtosis*); and estimators for conditional probabilities, i.e., the degree of independence of the sub-domain distributions measured in different organizations.

To display metrics, a distinction can be made between presentations to product and customer service managers and presentations to network and service operations. For presentations to product and customer service managers, simple statistical indicators usually suffice so that information for complex subsequent statistical processing is not needed. Therefore, the display metrics in this case comprise MIN or MAX acceptable threshold for service. For example, SLA reports should report metrics such as exceed performance, within performance, and below performance percentages. On the other hand, for presentation to network and service operations, detailed information, or even raw data, and some indication of the skew of event distributions would be needed.

8.4 Probes

Probe systems are a fundamental tool for network operators and SPs to monitor and manage the QoS. Probes can be placed at any point in the network, so they provide a greater flexibility than the systems based on network elements or other data sources. *Active probes* inject traffic in the network, and send requests to services' servers as an end user does. They are usually used to provide an end-to-end view. On the other hand, *passive probes* sniff packets from different services. They can only provide a view of a part of the network at several protocol levels [4].

Probes create a single monitoring tool for all services that enables systems to evaluate the QoS and can correlate information from different measures and services. In particular, probes provide the following functions:

- *Real-time network supervision*: By continuously monitoring the status of network elements and their traffic and quality parameters, failures can be detected and their impact analyzed in real-time.
- *Network planning based on updated data*: Traffic data obtained through detailed probes can be used to make network planning estimates such as routing capacities.
- *Detailed control of network use*: Probes can monitor the type and amount of traffic, which can help prevent abusive use by customers or partners.
- *Performance management*: Probes can measure parameters such as the number of calls, roaming attempts, requests for advice to intelligent network platforms, and failure statistics. In this way, in the event that a quality parameter exceeds pre-defined thresholds, the system can inform users and provide sufficient data to precisely characterize problems.
- *Data for billing services*: Probes can act as an additional billing support system. Since probes have access to network traffic, probe-based systems can reconstruct the services carried out by a user and, thus, verify billing.

Probe-based systems place probes at specific points on the system networks, where the information generated by the probes is received and pre-processed in remote sites, usually physically close to the probes, so that the remote sites draw up specific traffic and quality measurements. In addition, frames captured during a configurable amount of time are also stored in these remote sites so that they can be later accessed for the study of any reported abnormality. Measurements from remote sites are sent to a central system for processing, grouping, and correlation. Also, data from the remote sites is consolidated in a database in the central system.

This architecture involves remote probes that support, on one hand, passive network card interfaces, and, on the other hand, active probes that generate end-to-end sessions. In addition, passive probes monitor live traffic going through the network and conduct a set of call quality measurements. Two types of probe entities can be envisioned: *customer probe entities*, which simulate the behavior of service customers, and *network probe entities*, which non-intrusively gather inside the network the real traffic that customers generate when using the services. Both of these entities provide operators with the full scheduling capability in order to design self tests. Service tests can then be used to build QoS reports.

Passive probes gather traffic generated in the network in order to monitor its signaling. The information is classified by a specific service and customer. Information is gathered by means of non-intrusive probes located in the network. It is based on the traffic extracted by probes that are deployed on the network. Furthermore, it is directly fed by real traffic, neither from network elements nor intrusive equipment. Instead, the necessary data to monitor the network is generated from the signaling and IP traffic. The information can then be used to reconstruct detailed records from any service, generate network traffic and usage measurements, monitor CE and time parameters such as delay and jitter, use troubleshooting tools, and create real-time events and alarms to be exported to external fault management tools.

On the other hand, *active probes* enable end-to-end tests, where tests automatically and periodically behave as a customer. Active probes are usually used to supervise services provided by a SP, although the probes may also be used to obtain quality indicator measures. The probes generate one or several registers per test in a general purpose database. These registers contain information on the type of the executed test, the affected service, parameterization, partial results of each step, global results of the test, and measures of intermediate times of execution. These probes interact with the offered services from customers' point of view. They provide information at certain moments or locations even with a lack of traffic.

8.5 SLA Management and Reporting

Conformance with an SLA is ensured by using instruments in the systems to provide appropriate KPI and KQI measures at required sample rates. The references discusses that it is important in the design process to ensure that the measurement process itself does not create or worsen system conditions by adding further load to the system, e.g., by using additional processing power or adding additional management traffic overhead. If a KQI for a first service is determined by correlating KPI or KQI data from a second service, the information from the second service may be required in real time. This would allow for true measurements to be made for proactive management of the first service to allow fault prevention rather than aggregate or stored information. Thus, an SLA should be monitored continually at a

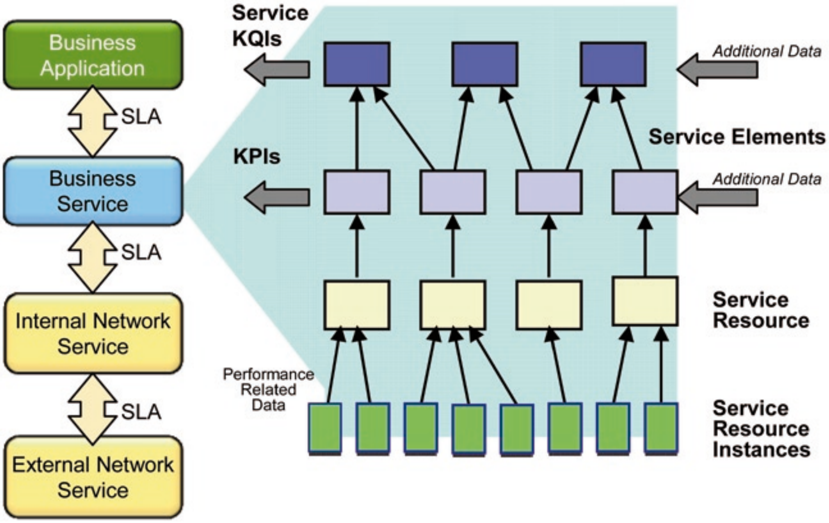


Fig. 8.8 Relationship among service resources, KQI, and KPI

rate appropriate to the requirement for a service to assure that corrective actions can be taken and collated to form management reports [4].

Once an SLA is in place, conformance against the SLA is demonstrated by the production of reports. As with the SLA itself, the display and interpretation of the report data should be clear and concise, and it should be clear when the SLA is out of conformance and not hidden within a myriad of conformance data. For each service, performance-related data is retrieved from the relevant instances of the service resource. These are collated and combined to form KQI for each resource and further combined to form the service and product KQI, as shown in Fig. 8.8. The following subsections elaborate on these ideas [4].

8.5.1 SLA Monitoring and Reporting Process

By using instrumentation within a system to measure the KPI and KQI, service performance data is collected and collated into a form that can be manipulated to allow diagnosis and report generation. Instrumentation can include user satisfaction surveys; test applications, including applications such as phantom callers; client-based monitor agents; server-based monitor agents; and network-based monitor agents. SLAs should define the periods of collection and provide the granularity necessary to ensure that the SLAs meet their requirements. Instrumentation data may need to be passed across service access points so that it can be used as KQI and KPI data to determine KQI or KQI for other services. In addition, for *proactive management*, systems need to collect real time information so that the system can perform proactive management for fault prevention. For *reactive management*, systems can collect near-real time or off-line, time-stamped data so that it can be correlated with other time-stamped data for improved performance information. Also, systems can correlate offline, aggregate information with other aggregate data to provide trend analysis for reactive management. It is important to distinguish between performance *events* and performance *parameters*, as follows [4]:

- *Events* are instant or near-instant phenomena that occur within a service or its environment that affect the KQI of the service. Examples include lost or misdirected packets, loss of signal, and power failure. Events may be signaled via mechanisms such as SNMP traps and interrupts or may be inferred by catastrophic loss of service.
- *Parameters* are derived by processing a series of measurements or events over a measurement period in a defined metric that can be reported. These may be time-related, ratios, or event rates. Examples are availability, throughput, utilization, average call response time, and ethernet collision rate per packet.

Systems can collate SLA performance data to form internal reports that can be used to diagnose the performance of the systems both for internal diagnoses and to produce customer reports. Collation may require combining KQI or KPI from different services or products, covered by different SLAs, from potentially different providers. Collation may be performed directly by collection tools or by using middleware applications that use common interface languages such as CORBA, XML, or SQL. The sampling period may be real time, semi-real time, or historical.

Internal reports may be in a different format than external reports in order to fit in with internal procedures and tools. Additionally, systems may set conformance thresholds at more aggressive values than those defined in SLAs to ensure corrective actions can be taken before non-conformance ensues. For this reason, internal reports are likely generated at more regular intervals than external reports, in order to allow remedial action to be taken and to improve or enhance a system. In an enterprise application requiring a multi-tiered SLA, systems may need to demonstrate how a service has been performing recently (typically a few hours) as a result of a support call or in the diagnosis of non-conformance at a different tier.

Systems should present external reports to customers at appropriate time intervals and in formats agreed a priori. Enterprises then can use the external reports to provide assurance of conformance and trend analysis for future growth or new opportunities.

8.5.2 SLA Reporting Mechanisms

A number of different functional groups may wish to see SLA management reports. These may include senior management groups that may be concerned with high-level achievement targets, finance groups that may be concerned with billing and cross-charging, engineering groups that may be concerned with diagnosis and planning, and end user groups that may be concerned with CRM. Therefore, the format, language, and style of each report should be appropriate to the audience. For example, for senior management, reports may consist of presentations with color and pie charts. By contrast, reports for finance would consist of spreadsheets that may be machine-readable. For engineering, reports would consist of trend graphs and raw data, and for end users, reports would consist of Web-based reports [4].

8.6 Enterprise SLA Negotiation

SLAs are largely dependent on enterprise objectives. Hence, it is difficult to create an SLA format that applies to all enterprises and objectives. Nevertheless, the following subsections describe general themes related to SLA development processes and SLA formats.

8.6.1 SLA Development Process

Enterprises work towards high-level objectives that an SLA or collection of SLAs support. Business processes are judged against these high-level objectives. Conflicts may require modification to application objectives or requirements or, in some cases, changes to the enterprise objectives themselves. Figure 8.9 depicts a possible SLA development process [4].

In Fig. 8.9, the process starts with the business decision, for instance from a product manager or board level, to pursue an opportunity by providing an application, say a product or service, to customers or partners. In normal business processes, the objectives of applications are defined, from which the requirements for the applications are derived. The application requirements are fulfilled by acquisition of products or services, development of a new product or service, outsourced arrangements, enhancement of an existing product or service, or integration of new or existing product services [4].

In general, applications require the use of business and network services. The relevant services are then determined. If a service is not already in use, then the service is initiated and an instance created for the application. Where the service already exists, a new instance is created. SLA metrics can then be applied to the service as a whole or for a particular instance.

For each service or instance required, the KQIs of an application are determined and mapped to the service requirements. This allows definition of KPIs for the service instance that can be used in monitoring and performance reporting. Other secondary indicators may be derived and monitored for diagnostics, fault prevention, and resolution. If the service does not exist or requires modifications to existing services, the timescales, costs, and impact on other applications and services should be considered. If there is a conflict with the requirements for the service from oth-

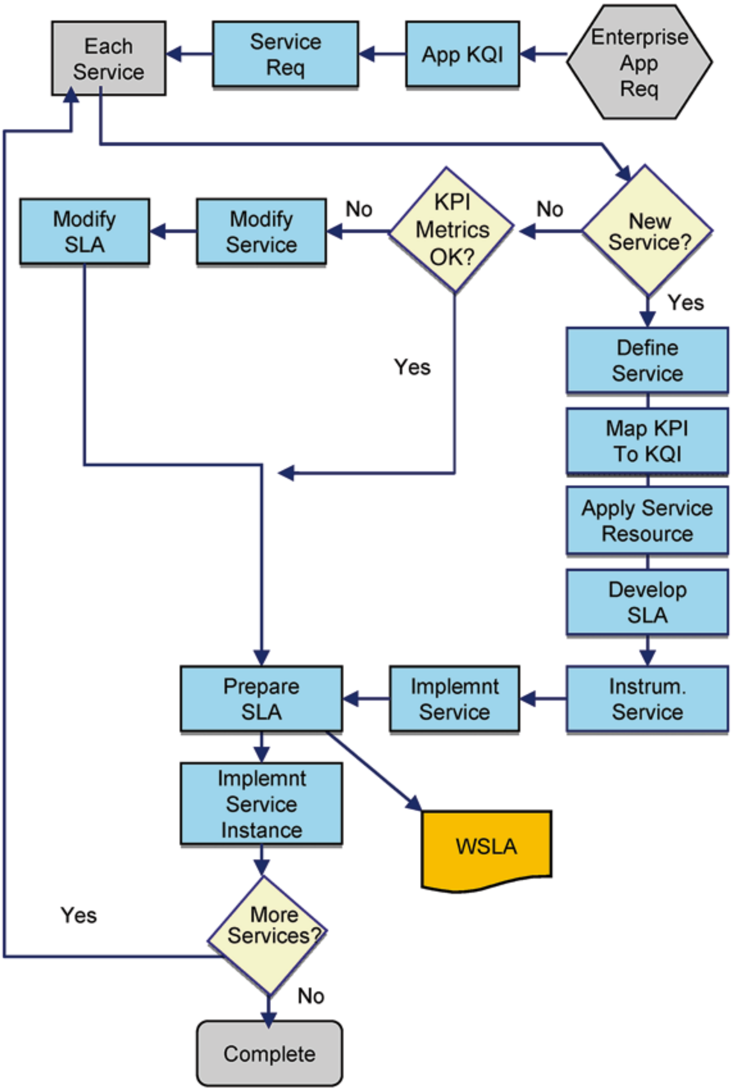


Fig. 8.9 Enterprise SLA development process

er applications, as defined by the enterprise application, these must be escalated through management to resolve the conflict. The costs of commissioning, lifetime support, and decommissioning of each service instance and the service as a whole need to be considered in the decision-making process. If no such conflicts exist, the level of training required must be assessed in terms of cost and time to ensure the appropriate personnel are trained adequately. The impact of any extra training must be assessed to ensure that the enterprise objectives are not compromised.

Monitoring should be considered to ensure that the KPIs can be measured in the time periods required and in a manner that does not impact on this or other services, say because of management bandwidth overhead. Once all these factors are considered for an individual service, an SLA can be developed. This process is repeated for all relevant services until either a conflict is found that cannot be resolved or all relevant SLAs have been developed.

It is important to note that the KQIs and KPIs for services should be the minimal set required for the services. If, however, other applications or services require underlying services with more stringent and rigorous KQIs or KPIs, it is those KQIs or KPIs that should be considered in the SLA. The SLA therefore reflects the most rigorous requirements for the KQIs and KPIs for all services supported by the SLA.

8.6.2 Form of an Enterprise SLA

The exact form of an SLA depends on the two entities that are entering into the agreement or contract. In particular, the form of the SLA will be different, especially in the area of penalties, when the SLA is between an enterprise and an external party, such as a Cloud provider, when the SLA is between internal enterprise parties, and when the SLA is between the enterprise and its customers. The SLA is a mutual agreement between two parties with expectations from both sides defined. It also defines the course of action to be taken when deviations from these expectations occur. An SLA is, in general, a legal contract between the parties, especially for SLAs between an enterprise and external parties, such as Cloud providers. It is therefore important to take legal advice as to the exact form of the contract and the language used. If the SLA is to span international boundaries, such as may occur in a Cloud environment, enterprises need legal advice that has an understanding of the differences in contract law, environmental, employment, and any relevant regulatory environment in the relevant countries. Even internal SLAs, where the SLA spans international boundaries, may have to take these issues into consideration [4].

The language and terminology used in SLAs should be appropriate to the audience. A glossary may be necessary to explain common terms, but in principal, the SLA should be written in a manner such that it can be read by someone versant in the particular service or technology in question. This also applies to any legal advice taken in the preparation and negotiation of an SLA. If the SLA is written between enterprises or between an enterprise and a Cloud provider, it is likely that legal language and terms are used. On the other hand, this type of language may give a negative view of the enterprise in terms of CE if this language is used in SLAs between an enterprise and an end residential user.

Relevant law should be stated and considered in the negotiation and preparation of an SLA. If the relationship is aimed to be long-term and strategic, then a mutually acceptable law should be considered; if tactical, then it would likely favor the party creating the initial draft. The SLA should make clear, in plain language, the aim of application that a service supports. Although unlikely to form part of the contract itself, this may help both parties to understand the requirements.

SLAs should also specify whether any part of the SLA, such as the existence of the relationship, the contract itself, or SLA reports, should be considered confidential because there may be competitive advantages in the service offered or the application supported. Those areas considered confidential should be clearly identified and the duration of the confidentiality stated.

When a provider defines a service for the first time, an SLA template is defined to form the basis of all instances of the SLA.

An outline of the main topics for inclusion in an SLA is discussed in the following bullets. The exact form of the SLA depends on a number of factors, including whether the SLA is a separate contract in its own right or forms a part or annex of a larger contract. It may ease further negotiations if annexes can be added to an SLA for new services without having to re-negotiate the main body. In this case, the SLA should be written appropriately for the first service [4].

- *Introduction*: This section documents the relevant parties that are entering into the SLA agreement. The introduction should also contain a brief overview of the need for the SLA and the application or services it serves. This information should include the KQI for the application to be included and how the KPIs of the service support the concept of the application KQI.
- *Customer Requirements*: This section documents how the customer is to use the service so that it clearly explains what the service supports. For example, if the requirement is to support a round trip time of less than 1 s for a transaction, then it will be necessary to understand the peak value and length of any bursts of transactions that are anticipated. It may be necessary to determine how the service should respond when, in this example, the transaction rate is exceeded.
- *Overview of Service*: This section describes the service including the location of the physical and logical interfaces between the two parties, who owns which part of the interface, the number of locations, and any other information that describes the service or product adequately.
- *Term*: This section details the period over which the SLA is valid, perhaps with a commencement date.
- *Responsibilities*: This section details the responsibilities of both the customer to the provider to ensure conformance and those of the provider to the customer. Expectations from both sides can be detailed in this section.
- *Details of Service*: This section describes the parameterization of the service in terms of the KPIs as they will be reported to the customer. This probably takes the form of a table. It should clearly show the levels of acceptable performance and non-conformance and out-of-specification conditions.
- *Exceptions*: It is likely that exceptions need to be included and clearly documented in the SLA. Downtime for upgrades or routine maintenance may be necessary but need to be described with such parameters as notice periods. In multi-site environments, care must be taken to ensure that the downtime is explicit. For example, if the SLA is between an enterprise and a Cloud provider that enables

connectivity between a corporate headquarters and its branch offices, and the total maximum downtime is 10 h per month, it is likely that the SLA may define the maximum downtime of the headquarters and each branch differently.

- *Sampling and Reporting*: SLAs define how often the KPIs are measured as a measure of conformance and how often they are collated in the form of a report to calculate the application KQI. The method of reporting, for instance via Web or paper, may also be necessary. Reports for non-conformance may require a different frequency from the normal collation process. The method of reporting non-conformance may also be different from normal KPIs and should be documented. Similarly, the reporting of asynchronous events, such as alarms, alerts, and traps, may also be different, and it may be necessary to establish the maximum frequency of asynchronous events from the customer or the provider.
- Sample reports should be agreed and included with the SLA document. If the SLA performance data is required to determine KQI or KPI performance metrics in real time or near-real time, then the format of this data, the interface, e.g., SQL, XML, or CORBA, and the support, availability, integrity, and confidentiality for this interface needs to be defined. In addition, it is possible that tiers of reports may be available in an online and offline form. For example, customers may be able to view the reports for the SLA for their own use. Therefore, access control would have to be agreed on in the SLA, along with how long these reports are to be stored either online or as an archive.
- *Penalties*: The penalties for non-conformance should be detailed. Example penalties include lost fees, repayment of fees, compensation for lost earnings, and termination.
- *Dispute Resolution and Escalation*: This section documents how differences of opinion on the SLA in either the contract, its reports, or performance are resolved. It may be necessary to provide contact details for these instances and also to document how the situation can be escalated to senior management if the situation cannot be resolved. For SLAs between external parties, arbitration may be necessary. For internal parties, this section is likely to be absent and resolved within the normal management process.
- *Change Requests*: This section details procedures for how change requests to the SLA can be made and handled, with any expense detailed. Maximum frequency of change requests should be detailed. Notice periods for change requests should be documented. Performance of these change requests may be subject to the penalty clauses.
- *Termination*: This section documents reasons for terminating SLAs along with notice periods for termination and any costs associated. Notice periods may differ for supplier-to-customer and customer-to-supplier SLAs. The SLAs should also specify what would happen in the event that one of the parties is acquired by another party or acquires another enterprise such that the service requirements may be different from what is specified in the SLAs. Consideration should be given to whether the SLAs should terminate, continue as-is, or be renegotiated.
- *Relevant Law*: This section details which relevant law is to be considered for the SLA and under which jurisdiction any breach of contract is to be resolved. It is

likely that this section may be missing between internal parties unless the two parties are located in different countries, and there are significant differences in relevant law pertinent to the operation and performance of the service between the different countries.
- *Confidentiality*: This section details and highlights any aspects of the SLA, such as its existence, performance, reports, and report data, that are confidential.
- *Warranties*: This section details areas that are covered by warranty conditions. Where warranties already exist for some service resources, how these effect the SLA should be detailed.
- *Indemnities and Limitations of Liability*: This section specifies who is liable in the result of failure of the SLA, either the provider or the customer.
- *Signatories*: The SLA should be dated and signed by relevant signatories from both parties to the SLA.

8.7 Policies and Monitoring

Business Service Fabric (BSF) is a model for heterogeneous virtualization and abstraction of services, applications, policies, capabilities, resources, infrastructure and people. In the BSF model, these mentioned entities can be partitioned logically and virtually, into distributed Virtual islands of *Business Service Sub-Fabrics* (VBSFs). A BSF may span company, geographical, and technological boundaries, public and Private Clouds, and enterprise datacenters. Bridges between VBSFs, provided by sub-fabric mediator services, manage and control inter-sub-fabric interactions, manage protocols, including protocol conversions, and monitor and manage the underlying sub-fabrics. In a business sense, the sub-fabric mediator services manage the interaction between partner environments [5, 6].

The BSF and the VBSF concepts are a virtual aggregation of business services, from diverse sources, in a networked services environment that permit consistent usage, manageability, and operability. In a VBSF, diverse, discrete sets of services work together to perform some tasks while communicating over business services protocol stacks [5, 6].

In the BSF model, users, i.e., end-users of services, service developers, or administrators, operate in their permitted BSFs, and each user BSF is configured to include the necessary VBSFs. A BSF hides the characteristics of the underlying resources from the way in which other service systems, applications, or end-users interact with those resources. Users have isolated, fully functional service environments based on their rights and their roles [5, 6].

One common method for creating VMs in Cloud environments splits the OS into two discrete systems, a *hypervisor* that manages the VMs and a SDF for managing the application and providing needed services, as discussed in Chap. 5. In the model presented in [5], business services bind or utilize just the needed services from BPM systems, database systems, middleware frameworks, etc. The business services also use business manageability and operability services such as fault, configuration, accounting, performance and security management.

A *process* is a coordinated set of activities that collaborate to deliver some specified output. Processes can be composed of and can interact and collaborate with other processes. KPIs can be monitored to verify that business process actions are being performed and expected targets and results are being achieved. The KPI monitors are services and can be internal or external. Action, preferably policy-driven and automated, is taken when there is a gap between expected and achieved results. In addition to processes, resources, messages, and people are also often referred to as *services* [5]. Sub-fabrics restrict the possible interactions, the type of resources, their location, and manageability and operability options. Thus, mobility can be restricted to well-specified service implementations or agents, middleware, network segments, client devices, compute servers and data servers. An *agent* realizes or implements a service. A service can be realized by multiple agents where the agents have certain capabilities, for instance because different agents conform to the laws of different countries. The choice of provider agents depends on the agents' capabilities, performance, management policies, and cost considerations [5, 6].

Policies constrain the behavior and utilization of resources and apply to agents. Services realized by external agents or incorporated within a managed service enforce policies. An agent (A0) realizing a service (S0) may consist of a number of management service agents that manage the set of capabilities for the agent (A0), as shown in Fig. 8.10, and a set of external management service agents.

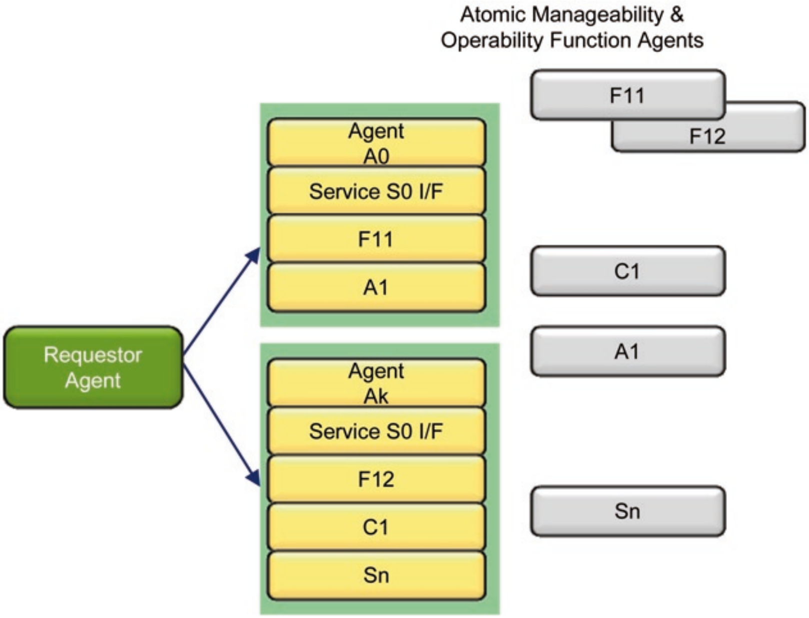


Fig. 8.10 Agents and services

Only certain policies are internally manageable, while others may require external coordination and management. For example, external management agents may enforce policies for legacy applications.

8.7.1 Monitoring Agents

Multiservice platforms present unique demands on event management systems because of the volume of traffic they process and the volume of alarms they can generate [7]. An Event Manager component within a managed service can support event correlation and filtering to reduce the potential flood of events. Event filtering and correlation policies define the filtering and correlation performed. A correlation policy can be defined to link all associated events to a given root event, provided they arrive within the specified time interval. As a result, only the root event is forwarded, thus reducing the alarm overload on the management system.

The instrumentation and management interfaces of a service are important aspects of its manageability. The service would be unmanageable if it does not have the proper instrumentation to provide information and control. An external management system can structure and initiate a query for all instrumented measures for status or trend analysis. Centralized performance monitoring and trending are difficult to perform in a highly dynamic, very large, distributed environment. In centralized performance monitoring systems, the volume of data available from a large number of services is likely to be too high for a performance monitoring component to collect, store, correlate, and process. By partitioning the monitoring capability among virtual fabrics, the volume of data can become manageable.

A monitor component of the Event Manager monitors counters against configurable thresholds. An alert is generated whenever the threshold is crossed during some configurable monitoring collection interval. The thresholds and collection interval can be configured individually for each attribute being monitored. Different levels of alerts can be raised when an attribute has multiple thresholds. For example, a pressure monitoring system may have critically low, low, high, and critically high thresholds. Rules may specify the generation of alarms after a threshold condition is met. For example, in the absence of a corrective action, the attribute value may continue to be above the threshold, and the rule can either restrict further alarms being generated, restrict their generation frequency, or raise the alarm level or the alarm receivers.

Instrumentation is required to protect services from losses caused by security problems, and, to ensure instrumentation security, access to that instrumentation must also be protected. To help ensure software image integrity, loadable software is digitally signed and authenticated by the installation manager during the installation process. If a package fails authentication, it is not executed. Access to the information and control enabled by embedded instrumentation is gained through interfaces and messages.

Fault management is the collection and analysis of alarms and faults in the service. These faults can be either transient or persistent. Transient failures are not alarmed if their occurrence does not exceed a threshold. For example, sporadic message losses or delays. These events are, however, logged. Some transient problems can be automatically corrected within the service, while others may require different levels of management services for resolution. Faults can be determined from unsolicited alarm messages or by log analysis; the latter may be the only course when, say, existing services or applications do not have internal monitoring or alarm generation capabilities.

Fault management analyzes and filters fault messages and coordinates the messages so that the number of actual events reflects the real conditions of the services. The root cause is reported, while suppressing other related fault messages. While all faults are logged, and fault management at some layer may have been able to resolve the fault, the resolving fault manager creates a trouble ticket that records the fault details and any corrective actions performed. For example, while the fault management may have decided that a particular service resource Ra has failed and elected to use an alternate service resource Rb, Ra still needs to be fixed. For every service, there is a mapping to the underlying services or resources that can trace a failure to the service or resource. Some faults can be easily corrected in real-time. For example, in the case of an input message queue build-up of an otherwise functioning service, the fault can be corrected or mitigated by provisioning one or more service instances and distributing the load between the various copies. If, however, the output message queue is growing, then the fault may be in the recipient services, the messaging service, or any one of the underlying resources that implement the messaging service. The diagnostics of these various services can identify the service that needs to be addressed. Therefore, if the fault is in network congestion, then provisioning an alternate network path would correct the problem.

The effects of a fault are that the results are wrong or that the result does not meet the performance requirements. Two methods are commonly used to detect faults: *acceptance testing* and *comparison testing*. In *acceptance testing*, the service is executed with known inputs, and the actual result is compared with the expected result for the given inputs. *Comparison tests* are used in an environment where multiple versions of the service execute concurrently; the results from all of the versions for the same inputs are compared, and the majority result is accepted.

A *fault tolerant service* manages to keep operating, perhaps at a degraded level, in the presence of faults. For a service to be fault tolerant, it must be able to detect, diagnose, contain, confine, mask, compensate and recover from faults, i.e., it must have self-management capabilities.

Fault isolation is the process of determining what caused the fault, or exactly which component is faulty. In comparison testing, fault isolation requires an odd number of versions to concurrently run, and then a majority vote is taken to isolate the faulty versions. In well–designed, fault-tolerant services, faults are contained before they propagate to the extent of affecting service delivery. This leaves a portion of the service unusable because of residual faults. If subsequent faults occur, the service may be unable to cope because of this loss of resources, unless these resources are reclaimed through a recovery process that ensures that no faults remain in service resources or in the service state. A service can mask faults by ensuring that, even in the occurrence of a failure, only valid results are propagated beyond services where a user may be impacted. For example, in the case of an account balance enquiry, the last valid data and the date and time is presented to the user. If a fault occurs and is confined to a component, it may be necessary for the service to provide a response to compensate for the output of the faulty component. This is possible in certain situations, such as when reporting weight, where the balance component has been determined to consistently return the actual weight plus some fixed known amount.

When a service completely fails, recovery may entail restarting the service. A Configuration Manager restarts the service based on the recovery process defined for the service. The Configuration Manager may provision a service recovery choreographer service that would enforce the recovery constraints on message order, state consistency, and communicate progress to interested parties.

8.7.2 Manageability and Operability

Manageability is the composite result of a number of different facets, including, availability, scalability, performance optimization, reliability, risk management, business continuity, and change management. The more frequently a system needs to be managed, the more steps involved in each management action, or the longer each management step takes, the poorer the system manageability.

Business services adapt to an environment through composition or by interacting with appropriate services. The service policies specify availability and scalability, performance optimization, monitoring and security requirements. Services, being stateless, achieve seamless incremental scalability and high availability through service replication. Services provide visibility into their performance, in particular, along KPIs through a combination of constituent and external monitoring services.

A service becomes manageable when it exposes a set of management operations that support management capabilities. These operations may only be exposed to services with the necessary permissions. The management operations provide for monitoring, controlling, and reporting functions, in addition to policy management capabilities. Agents can raise alarms based on policy infringements. Information may also be provided in response to a manageability query on request and response counts, begin and end timers, etc. The service's interface specifies the supported management capabilities to monitor, diagnose, and manage service performance. Although the provision of management capabilities enables a service to become manageable, the extent and degree of permissible management are defined in management policies that are associated with the service. Management policies therefore are used to define the obligations for, and permissions to, managing the service.

The manageability and operability of services is simplified when services operate in well-defined and managed environments.

A *management ecosystem* supports the set of processes and activities necessary to deliver services and operate them to meet some of the service objectives. In a management ecosystem, there is at least one agent acting as the manager and at least another agent acting as the *managed agent*. The manager requests information or the performance of some action. The manager agent facilitates the performance of the request by interacting with the managed agent via a link between the manager and the managed agents. In the management ecosystem, an agent can assume the manager role or the agent role.

Orchestration agents allow business services to interact with agents realizing elements of a service and make it possible to synchronize many different events or operations that may apply. Orchestration agents enable the performance of complex operations on a dynamic and diverse grouping of agents and control behavioral changes during operation.

Operability is the ability to operate the system while it is performing its intended function during its up-time. It includes reliability, maintainability, supportability, flexibility, safety, operating costs, and usability. Reliability is a composite of availability and its ability to recover quickly to a fully-operational state. Supportability is the ability to operate the system and adapt to changing demands. Maintainability is the ability to quickly make changes to the service and keep the unavailability of the service to the bare minimum. Operability determines costs that include the costs for support, maintenance, training, technical publications, spares, support equipment, and some facilities.

The following shows a subset of the steps in the service creation process that endows the service with operability capabilities.

- *Create Services*: Services can be created using service development tools by completely creating a new service, by adapting existing services, or by encapsulating an existing application in a service. During the service creation process, the non-functional capabilities would also be created for the service. The actual realization of these non-functional properties may be provided by incorporating existing management agents. The policies that govern the service are also defined, and policy management agents are incorporated.
- *Register Service*: The registration process entails the discovery of the description, interface, capabilities, etc. It also requires specifying the management capabilities and interfaces. Services register to be visible within some defined network depending on the service creators and service credentials, such as certification of manageability and operability. The manageability and operability capabilities of the service specify the interaction patterns between the service and external management agents.
- *Operate Service*: A service can interact with another service by using a mediator service. The mediator service manages authentication and interaction and also provides protocol conversion, for example encryption and policy management.

8.8 Conclusion

This chapter discusses the key aspects of specification of service and quality levels; the interface between service levels and monitoring and asset agents; multi-level, multi-vendor, SLM interfaces; enforcement of SLM; and reporting.

Achieving quality and performance targets for the products or services may require an enterprise to establish and manage a number of SLAs. The complexity of global services brings together a myriad of services, suppliers, and technologies, all with potentially different performance requirements. Thus, the goal of enterprise SLAs is to improve the CE of the service or product to the enterprise clients, whether they are internal or external to the organization. CE is a collective term to form a measure of the quality of a service or product and includes all aspects of service: its performance, level of customer satisfaction in the total experience, pre and post sales, and the delivery of its products and services. Determining the CE provides a discriminator between various types of services or products that an enterprise provides, and leads to opportunities to balance the level of quality offered against price and customer expectation.

There is a difficulty in mapping service-specific parameters to technology-specific parameters that are more easily measured and reported. As a consequence, traditional SLAs have focused almost solely on the performance of the supporting service. By contrast, KQIs and KPIs focus on service quality rather than network performance. KQIs and KPIs provide measurements of specific aspects of the performance of applications or services. A KQI is derived from a number of sources, including performance metrics of a service or underlying support service KPIs. As a service or application is supported by a number of service elements, a number of different KPIs may need to be determined to calculate a particular KQI. The mapping between the KPI and KQI may be simple or complex, and the mapping may be empirical or formal.

There has been much research in the development of standard equations that provide quality measurements from performance-related data. These equations can be used to model a network before it is deployed, assign values for an SLA contract, and perform analysis of data to predict the performance enhancement or degradation due to changes in the service, such as the addition of a route controller or a move from narrowband to broadband connections. These equations can be used to determine thresholds and sensitivity analysis of PKI parameters for SLA monitoring and reporting.

A SQM framework needs to define a holistic framework for measuring and effectively managing service quality; key service quality metrics at each point along the service delivery network; service quality issues and the necessary accounting and rebating information, usage information, and problem resolution information; management capabilities to support each step in the service delivery network; and appropriate interfaces and API's to enable the interchange of such information electronically between the various providers in a service value chain.

Probe systems are a fundamental tool for network operators and SPs to monitor and manage QoS. Probes can be placed at any point in the network, so they provide a greater flexibility than the systems based on network elements or other data sources. Active probes inject traffic in the network and send requests to services' servers as an end user does. They are usually used to provide an end-to-end view. On the other hand, passive probes sniff packets from the different services. They can only provide a view of a part of the network at several protocol levels.

Conformance with an SLA is ensured by using instruments in the systems to provide appropriate KPI and KQI measures at required sample rates. It is important in the design process to ensure that the measurement process itself does not create or worsen system conditions by adding further load to the system, e.g., by using additional processing power or adding additional management traffic overhead. If a KQI for a first service is determined by correlating KPI or KQI data from a second service, the information from the second service may be required in real time so that true measurements can be made for proactive management of the first service. This allows for fault prevention rather than aggregate or stored information. Thus, an SLA should be monitored continually at a rate appropriate to the requirement for a service to assure that corrective actions can be taken and collated to form management reports.

Enterprises work towards high-level objectives that an SLA or collection of SLAs support. Business processes are judged against these high-level objectives. Conflicts may require modification to application objectives or requirements or, in some cases, changes to the enterprise objectives themselves.

The exact form of an SLA depends on the two entities that are entering into the agreement or contract. In particular, the form of the SLA will be different, especially in the area of penalties, when the SLA is between an enterprise and an external party, such as a Cloud provider, when the SLA is between internal enterprise parties, and when the SLA is between the enterprise and its customers. The SLA is a mutual agreement between two parties with expectations from both sides defined and defines the course of action to be taken when deviations from these expectations occur. An SLA is, in general, a legal contract between the parties, especially for SLAs between an enterprise and external parties, such as Cloud providers. It is therefore important to take legal advice as to the exact form of the contract and the language used. If the SLA is to span international boundaries, such as may occur in a Cloud environment, enterprises need legal advice that has an understanding of the differences in contract law, environmental, employment, and any relevant regulatory environment in the relevant countries. Even internal SLAs, where the SLA spans international boundaries, may have to take these issues into consideration.

Multi-service platforms present unique demands on event management systems because of the volume of traffic they process and the volume of alarms they can generate. An Event Manager component within a managed service can support event correlation and filtering to reduce the potential flood of events. Event filtering and correlation policies define the filtering and correlation performed. A correlation policy can be defined to link all associated events to a given root event, provided they arrive within the specified time interval. As a result, only the root event is forwarded, thus reducing the alarm overload on the management system.

Manageability is the composite result of a number of different facets, including, availability, scalability, performance optimization, reliability, risk management, business continuity and change management. The more frequently a system needs to be managed, the more steps involved in each management action, or the longer each management step takes, the poorer the system manageability.

Operability is the ability to operate the system while it is performing its intended function during its up-time. It includes reliability, maintainability, supportability, flexibility, safety, operating costs, and usability.

References

1. The Open Group: SLA Management Handbook, vol. 4. The Open Group, Berkshire (Oct 2004)
2. ITU-T: The E-model, a Computational Model for Use in Transmission Planning. ITU-T Recommendation G.107. International Telecommunication Union (May 2000)
3. ITU-T: Application of the E-model: A Planning Guide. ITU-T Recommendation G.108. International Telecommunication Union (Sept 1999)
4. Technical Report: Part: I—Holistic e2e customer experience framework, TR149, Release 1.0. TM Forum. Nov 2009
5. Goyal, P.: The virtual business services fabric: an integrated abstraction of services and computing infrastructure. Proceedings of 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE 2009, Groningen, 29 June–1 July 2009
6. Goyal, P., Mikkilineni, R., Ganti, M.: Manageability and operability in the business services fabric. Proceedings of 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE 2009, Groningen, 29 June–1 July 2009
7. Goyal, P., Mikkilineni, R.: FCAPS in the business services fabric model. Proceedings of 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE 2009, Groningen, 29 June–1 July 2009

Chapter 9
Security for Enterprise Cloud Services$_2$

Companies and individuals have natural concerns about the security of their data. The term “security” is rather ambiguous, in that it can mean “confidentiality,” “authenticity” “timeliness,” “availability” or many other definitions. We use the term “security” to mean ensuring that the data can be accessed only by *authorized* entities and that the data is *confidential, authentic, up-to-date*, and *exists*.

Only authorized entities can access secure data. The authorized entities must access the data only when the entities are authorized to perform certain functions, i.e., when the entities have pre-defined *roles*. For example, a doctor may have access to patient records stored in a Cloud when the doctor is in the role of a doctor. If the doctor no longer is in that role, say because the doctor is on a temporary leave, then the authorization of the doctor can be suspended or revoked. During an emergency, the doctor may re-gain access to the records when the doctor is assigned an appropriate role. A new doctor can gain access to the same records when the new doctor obtains an appropriate role.

Confidentiality ensures that the plaintext data cannot be read while in storage or in transit except by entities with specific characteristics. For example, the data may be stored encrypted in the Cloud. Only those entities that have the decryption key can read the plaintext data. On the other hand, *authentication* refers to authentication of data integrity or authentication of the source of data. Even if the data confidentiality is assured, the integrity of the data can still be compromised. For example, encrypted fields within messages can be swapped to alter the content of the messages, even if the plaintext content of the fields is not known.

Attackers may substitute old versions of software that have known security holes for robust versions of software in the Cloud. Therefore, data security must guarantee the freshness of the data, i.e., that entities access the most *up-to-date* data. Furthermore, a hacker can launch an attack by deleting data, so data must not be deleted except via secure methods.

In the following, we expound on these concepts and explain the ways that security management enables them. The discussion centers on the following concepts:

- Security for Cloud services and infrastructure
- Security for enterprises that use Cloud services
- Security for data stored in the Cloud

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_9, © Springer Science+Business Media B.V. 2010

Security is part of *information assurance*, a term that the US DoD defines as "Measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. These measures include providing for restoration of information systems by incorporating protection, detection, and reaction capabilities" [1].

9.1 Overview

The security concerns for Cloud providers differ from the concerns of Cloud users. The Cloud providers want to ensure that only authorized Cloud provider personnel can modify the basic services provided by the Cloud. On the other hand, the Cloud providers want to enable authorized users of the Cloud to use the IaaS and SaaS functions to customize the services of the Cloud to suit the enterprise users' security needs. Thus the Cloud providers need to balance two seemingly conflicting objectives; prevent users of the Cloud from modifying the basic services that the Cloud provides to the users while enabling the users to customize the services to suit individual enterprise needs. In addition, Cloud providers need to allow individual enterprises to protect their security services from use by or disclosure to other enterprises. In addition, Cloud providers must ensure that any security vulnerabilities introduced by the security practices of individual enterprises do not affect the security of the Cloud itself or the security of other enterprise users in the Cloud.

The enterprise users of the Cloud want to ensure that the security services provided by the enterprise meet the security expectations of the enterprise users. In

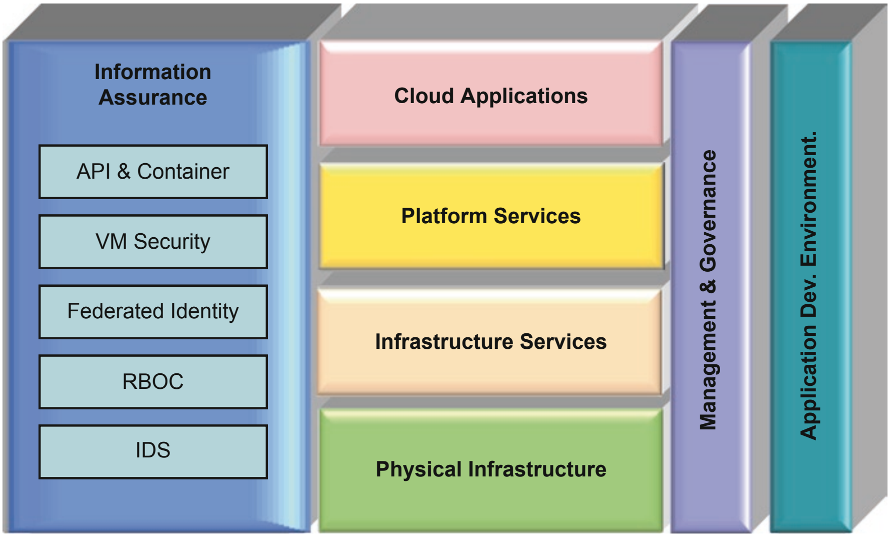


Fig. 9.1 Topics covered in Chap. 9

some cases, the security expectations may be more stringent than those offered by the underlying Cloud services. In other cases, the enterprise security requirements may be less stringent than the security provided by the underlying Cloud services. The enterprise users of the Cloud need to have the flexibility to trade-off security with speed and efficiency, regardless of the limits imposed by Cloud services.

The data from competing enterprises may reside alongside one another in the same Cloud servers. In fact, data from one enterprise may reside in the Cloud servers of a competing enterprise. Hence, Cloud providers need to have services that ensure the anonymity of the sources of the data and the randomization of the location of the data.

In this chapter, we will discuss the transformation of enterprise security into Cloud services and infrastructure security. The Information Assurance box in Fig. 9.1 shows the topics covered in this chapter.

9.2 Security for Cloud Services and Infrastructure

Management of security for Cloud Computing requires RBAC architectures in the Cloud that can integrate well with customer systems. Security management comprises security for the Cloud network itself and security for customer data and infrastructure hosted in the Cloud. Security for the Cloud network itself requires secure APIs so that users of the Cloud are assured of the security of the services the Cloud offers. Security for data and infrastructure hosted in the Cloud requires that VMs for different customers operate autonomously so that the hardware and software resources used by one VM are securely protected from other VMs.

9.2.1 Authorization and Role-Based Access Control

One of the most challenging problems in managing large networks is the complexity of security administration [2]. RBAC has become the predominant model for advanced access control and is adopted as an ANSI/INCITS standard since the year 2004 [3]. Today, most IT vendors have incorporated RBAC into their product lines, and the technology is finding applications in areas ranging from health care to defense, in addition to the mainstream commerce systems for which it was designed.

9.2.1.1 Access Management Architecture

In RBAC, permissions are associated with roles, and users are assigned to appropriate roles. This simplifies management of permissions. Roles are created for the various job functions in an organization and users are assigned roles based on their responsibilities and qualifications. Users can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as needed [4].

A *role* is properly viewed as a semantic construct around which access control policy is formulated. In general, a role is a stable construct because an organization's activities or functions usually change infrequently, whereas a particular collection of users and permissions brought together by a role is often transitory. A role can represent competency to do specific tasks, such as a physician or a pharmacist, or specific duty assignments that are rotated through multiple users, e.g., a duty physician or shift manager.

RBAC needs enhancements for open and decentralized multi-centric systems, such as when transforming an enterprise into a Cloud Computing environment, where the user population is dynamic and the identity of all users are not known in advance [5]. Assigning appropriate roles to dynamic users in such systems requires credential-based RBAC models. Credentials implement a notion of binary trust. Here the user has to produce a predetermined set of credentials (for example, cryptographic keys) to gain specific access privileges [6]. Credentials provide information about the rights, qualifications, responsibilities and other characteristics attributable to its bearer by one or more trusted authorities. In addition, the credentials provide trust information about the authorities themselves.

The following figure illustrates example components of a credential-based RBAC, as follows [7]:

- The *Administration Tool* creates key pairs and their public key certificates, manages roles in the RBAC system, and inserts access control policy into policy attribute certificates and binds them to the public key certificates of the roles. The Administration tool also creates role attribute certificates and assigns them to users by binding them to the public key certificates. Certificates are stored in the *Support DBMS* servers.
- The *Control Policy* component maps policy attributes to access specific resources. The component is also responsible for ensuring the integrity of binding the roles to the policy attributes.
- The *User* uploads authentication credentials to obtain the roles assigned to the User.
- The *Delegation* component enables users to delegate roles to other users, if allowed by policy.
- The *Access Control Engine* component is responsible for authenticating users, enforcing policies, and accessing data resources.
- The *Data Resources* component contains the servers and file systems.

In a hospital setting, for example, a user can have one of the following roles [8]:

- *Qualified Entity*: A Qualified Entity is an entity that has been recognized as having certain training or experience or other characteristics that would make the entity an appropriate performer for a certain activity. A user is assigned a Qualified Entity role by an organization that educates or qualifies entities.
- *Licensed Entity*: A user is assigned a Licensed Entity role (e.g., a medical care giver, a medical device, or a provider organization) to perform certain activities

by an organization (e.g., a health authority licensing healthcare providers or a medical quality authority certifying healthcare professionals) that has jurisdiction over these activities.

- *Employee*: The purpose of the role is to identify the type of relationship the employee has to an employer, rather than the nature of the work actually performed. This relationship is between a person or organization and a person or organization for the purpose of exchanging work for compensation.
- *Access*: In this role, a user provides medication to another user.
- *Patient*: This is the role of a living subject who is a recipient of health care services from a healthcare provider.

9.2.1.2 Implementation of Credential-Based RBACs in Cloud Infrastructure

Entire organizations may store all of their data in the Cloud. Therefore, intruders who succeed in circumventing security in the Cloud can gain access to vast amounts of data, thus possibly causing far greater damage than would have been the case had the intruder gained access to only a few computers within the organization [9].

One of the challenges in role-based security for Cloud s is for the Cloud provider to enable role-based security for client enterprises while ensuring the separation of role-based architecture for one enterprise from the architecture for another enterprise.

Effective and secure management of credentials dictates the overall security of RBAC-based Cloud infrastructures. Users identify themselves to the infrastructure via some secure mechanism and request authorization to access data or resources as part of the privileges associated with roles. Upon authentication of the users by the Cloud security infrastructure and authorization by the Cloud administrative tools, the security infrastructure issues credentials to the users to access Cloud data and services. Since the components of an RBAC-based architecture can reside in separate locations within the Cloud, and communication between the components can occur over publicly accessible paths, the processes used by the Cloud infrastructure take into account several security threats that do not traditionally exist in enterprise networks, such as man-in-the-middle attacks.

Users can identify themselves to the Cloud via a variety of well-known secure methods. There are three main techniques: what a user knows, what a user has, and who the user is [10]. Passwords are examples of what a user knows, smartcards are examples of what a user has, and biometric devices are examples of who a user is.

After a user authenticates its identity with the Cloud infrastructure, the Cloud infrastructure issues a *session key* to the user by using the Diffie-Hellman public key cryptosystem. The public key cryptosystem is shown as PKI in Fig. 9.2. The session key is valid only for the session in which the user is logged into the Cloud and is not re-used for other sessions with the user. This restriction on the session key prevents replay attacks, where a malicious entity tries to obtain access to the Cloud resources by attempting to re-use the keys. To defend against man-in-the-middle attacks when establishing the session key, the user knows a priori a public key of the PKI, and the PKI knows a priori a public key associated with the user.

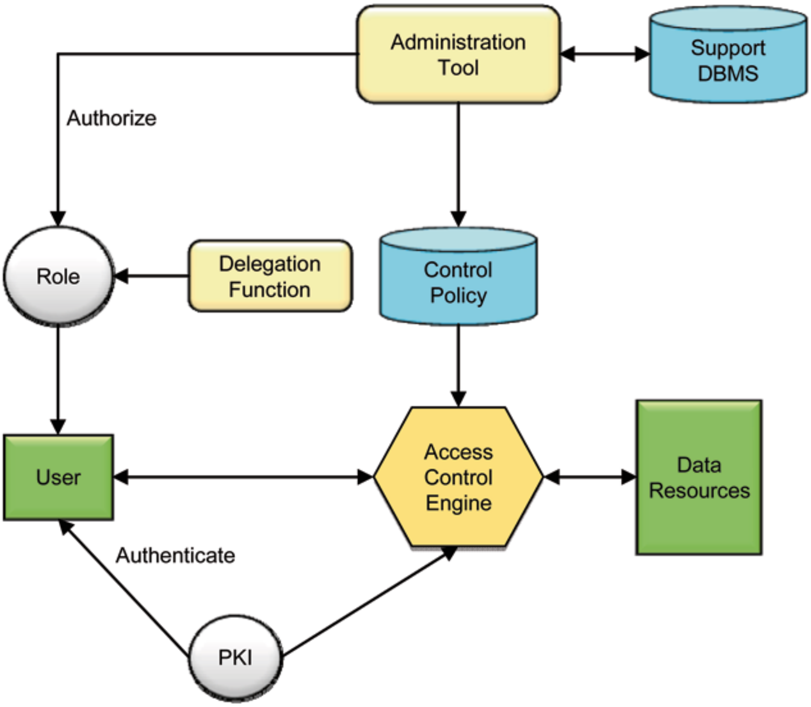


Fig. 9.2 Example components of a credential-based RBAC

The PKI registers the session key, along with the user identity, with the Access Control Engine. Unlike users, the PKI is expected to be in frequent communication with the Access Control Engine, so, rather than use session keys, a secure IPSec tunnel is established between the PKI and Access Control Engine. The Administration Tool binds specific roles to users by issuing public key certificates that contain the user identifiers and the roles assigned to the users. The certificates may reside with the users or may reside in the Cloud databases. The certificate would be signed with the private key of the Administration Tool and verified by the Administration Tool public key known to the Access Control Engine. The Access Control Engine consults the public key certificates to ensure that the user can claim the role that the user wants. The Access Control Engine also consults the control policies for the role. For example, it may be the case that there is a limit on how many concurrent actors with a particular role are allowed on the Cloud infrastructure. The Access Control Engine then uses the session key assigned to the user by the PKI to inform the user of the status of the user request. In addition, depending on the control policies, the Access Control Engine may request that the Administration Tool generate additional certificates for the user for particular roles. The user then can use the session keys and permissions received from the Access Control Engine to access the data resources and services of the Cloud.

The model described above is an example of a *server-pull* RBAC model. The server-pull architecture requires the RBAC servers to cooperate to obtain a user's role information from the role server for each session. This increases the freshness of the roles, so the information update (e.g., role revocation) is efficient, because all the roles are stored in the role server and pulled by the RBAC servers on demand [11].

In a *user-pull* RBAC model, users pull the roles from the role server, and then present the role information to the RBAC servers along with authentication information. Once users obtain the roles, they can use them in several sessions and with many servers until the roles expire. This increases reusability of role information. However, the longevity of the roles decreases the freshness of the roles. For instance, if users already pulled their roles, updated versions in the role server would not become effective instantly, so additional synchronization processes would be required to push the status change of the users' roles to the RBAC servers.

9.2.2 Cloud Security Services

The enterprise users of the Cloud need to have the flexibility to trade-off security with speed and efficiency, regardless of the underlying security of the Cloud infrastructure. Therefore, Cloud networks should offer an array of security services and tools to enable enterprises to tailor the security services to individual needs. In particular, the Cloud security services can include the basic cryptography building blocks for confidentiality, authentication, and integrity. Examples of the building blocks include public key algorithms such as RSA, Diffie-Hellman, and elliptic curve cryptography; secret key algorithms such as AES, 3DES, and DES; and hash and message digest algorithms such as MD5 and SHA-1. Making these services available to enterprises requires Cloud providers to have the necessary supporting policies and infrastructure, as follows.

9.2.2.1 Export Control Policies

Some of the security building blocks may be restricted by export control laws of some countries. The restrictions can cover the use of a building block itself, e.g., the use of AES or 3DES [12], or the cryptographic key lengths that can be used, e.g., the use of symmetric key lengths greater than 64-bits [13]. Therefore, a Cloud SP must ensure that the policies applied to the enterprises that use the Cloud conform to the export control policies imposed by the copyright owner of the building block.

Because of the distributed nature of Clouds, enforcing the policies in a Cloud infrastructure requires the Cloud providers to create a hierarchy of *Certificate Authorities* (CAs). A CA generates certificates for enterprises, which are signed messages that specify, among other things, the policies applied to the enterprises. The root CA is expected to be implemented on special hardware designed so that it is tamper-resistant and managed by systems administrators who have been certified for security by the Cloud provider. The root CA issues certificates to the next level CAs. Since the next level CAs are not expected to be added or modified frequently, the root CA does not need to be online. This helps provide physical security for the managed assets and also helps design the secure hardware.

The sub-CAs, i.e., the next level CAs, issue certificates for enterprises. Hence, they are expected to be online. The CAs can be dedicated to the different policies. In particular, some of the sub-CAs can be dedicated to issuing certificates to third party enterprise CAs, so that enterprise providers can issue their own certificates to their users. By issuing their own certificates to users, enterprises can reduce the costs associated with certificates, since Cloud services often charge a fee for each certificate issued. Another advantage of issuing certificates is that enterprises can continue to use the certificate infrastructure that already exists in the enterprise. Yet another advantage is to separate the security policies within the enterprise from the security policies of the Cloud providers, so that the business model for the enterprise can remain separate from the business model for the Cloud provider.

9.2.2.2 Cloud Infrastructure to Support Public Key Algorithms

The Cloud provider can make software modules, called *Cryptographic Service Providers* (CSPs) [14], available for use by enterprises. CSPs contain implementations of cryptographic standards and algorithms [15]. The Cloud provider must digitally sign every CSP and verify the signature when the enterprises load the CSPs. In addition, after being loaded, the Cloud provider periodically re-scans the CSPs to detect tampering, either by malicious software such as computer viruses or by the users themselves trying to circumvent restrictions (for example on cryptographic key length) that might be built into the CSP's code.

An enterprise can use the Cloud infrastructure that supports public key algorithms to augment the enterprise's own PKI. The enterprise then uses secure APIs to access the CSPs, as we will discuss in Sect. 9.5.1. Many of the infrastructure elements that support public key algorithms in Cloud s were discussed in Sect. 9.2.2.1. In particular, public key algorithms require the use of CAs to verify the authenticity of the certificates. In addition, the Cloud infrastructure needs to implement a mechanism to distribute *Certificate Revocation Lists* (CRLs). A CRL lists the numbers of certificates that should not be honored. Typically, CRLs are posted periodically by the CAs. CRLs have issue times, so enterprises need to ensure that they have the latest CRLs. To reduce the overhead on enterprises, Cloud providers can publish the CRLs to online revocation servers. The enterprises can use authenticated communication to query these servers over the Internet about the revocation status of certificates.

9.2.2.3 Cloud Infrastructure to Support Secret Key Algorithms

Cloud providers can make secret key algorithms available in CSPs, in the same way that public key algorithms are available in CSPs.

The infrastructure to support secret key algorithms requires the use of *Key Distribution Centers* (KDCs) [16]. A KDC creates secret keys with which two communicating parties can encrypt the communication. This allows the two parties to verify each other's identities, because the KDC verifies the identities of the two parties before issuing the communication key to them.

KDCs are replicated in the Cloud to ensure that a single KDC does not become a single point of failure or a performance bottleneck [10]. All replicas of the KDC must be interchangeable with all other KDCs, in the sense that they all would have identical databases. This is done by having one KDC hold the master copy to which all updates, such as adding a user, deleting a user, and changing a user key must be made. Having a single master copy avoids problems such as combining updates made at different KDCs and resolving conflicting updates. All other KDCs download the database periodically from the master KDC. Having a single master copy can cause a single point of failure. Fortunately, the Kerberos network authentication service described in the IETF RFC 4120 is designed so that all critical operations, and most of the non-critical operations, are read-only operations of the KDC database.

9.2.2.4 Cloud Infrastructure to Support Hash and Message Digest Algorithms

The infrastructure to support hashes and message digests is simpler than the infrastructure for secret key algorithms. No KDC is required; only CSPs that implement the hash and message digest algorithms. It is assumed that, if secret keys are needed, e.g., to perform keyed hashes, then the secret key is sent to the communicating parties either via some secret key infrastructure or via an out-of-band mechanism.

9.2.3 Integration of Role-Based Architecture in the Web

RBAC is an architecture implementation for use by WWW (Web) servers. Because RBAC for the Web (RBAC/Web) places no requirements on a browser, any browser that can be used with a particular Web server can be used with that server enhanced with RBAC/Web. [17] RBAC/Web is implemented for both UNIX (e.g., for Netscape, NCSA, CERN, or Apache servers) and Windows NT (e.g., for Internet Information Server, WebSite, or Purveyor) environments.

Table 9.1 shows the components of RBAC/Web. RBAC/Web for UNIX uses all of the components in Table 9.1, whereas the NT version uses only the Database, Session Manager, and Admin Tool components. With RBAC/Web for UNIX, there are two ways to use RBAC/Web with a UNIX Web server. The simplest way is by means of the RBAC/Web *Computer Generated Interface* (CGI). The RBAC/Web CGI can be used with any existing UNIX server without modifying its source code. RBAC URLs are passed through the Web server and processed by the RBAC/Web

Table 9.1 RBAC/web components

Database	Files that specify the relationship between users and roles, the role hierarchy, the constraints on user/role relationships, current active roles, and the relationship between roles and operations
Database server	Hosts the authoritative copies of the files which define relationships between users and roles, the role hierarchy, and the constraints on user/role relationships. These files are created and maintained by the Admin Tool. When changes are made to these files, the Database Server notifies the Web Servers to update their cached copies
API library	A specification which may be used by Web servers and CGIs to access the RBAC/Web Database. The API is the means by which RBAC may be added to any Web server implementation. The API Library is a C and Perl library which implements the RBAC/Web API
CGI	Implements RBAC as a CGI for use with any currently existing Web server without having to modify the server. The RBAC/Web CGI uses the RBAC/Web API
Session manager	Manages the RBAC Session. The RBAC/Web Session Manager creates and removes a user's current active role set
Admin tool	Allows server administrators to create users, roles, and permitted operations, associate users with roles and roles with permitted operations, specify constraints on user/role relationships, and maintain the RBAC Database. Administrators access the RBAC/Web Admin tool with a Web browser

CGI. RBAC/Web configuration files map URLs to file names, while providing access control based on the user's roles. Installation of the RBAC/Web CGI is similar to the installation of the Web server.

While the RBAC/Web CGI is relatively simple to install and use, it is not as efficient as performing access control directly in the Web server. So, the other way to use RBAC/Web is to modify the UNIX Web server to call the RBAC/Web API to determine RBAC access. A URL is configured as an RBAC-controlled URL by means of the Web Server configuration files that map URLs to file names.

Some Web servers for a UNIX environment, such as Netscape and Apache, divide their operation into steps and provide the capability for each step to be enhanced or replaced by means of configuration parameters. This allows Web server operation to be modified without having to change the server's source code. For these Web servers, the RBAC/Web API can be integrated by simply providing the appropriate calling sequence and modifying configuration parameters.

9.3 Security for Enterprises that Use Cloud Services

This section describes several methods to provide security for enterprises that use Cloud services. In particular, the section discusses federated identity management and methods to counteract potential attacks that could arise from the integration of enterprise services with Cloud offerings.

9.3.1 Federated Identity Management Architecture

There are several possible federated identity management architectures [18]. *Federated identity management* aims to unify, share, and link digital identities of users among different security domains.

A *Federated Identity Architecture* (FIA) is a group of organizations that have built trust relationships among each other in order to exchange digital identity information in a safe way, preserving the integrity and confidentiality of the user personal information. The FIA involves *Identity Providers* (IdPs) and SPs in a structure of trust by means of secured communication channels and business agreements. IdPs manage the identity information of users and do the authentication processes in order to validate their identities. SPs provide one or more services to users within a federation.

Two architectures compete to implement FIAs in Clouds—Liberty Alliance architecture [19] and WS-Federation architecture [20]. The Liberty Alliance architecture defines a *Circle of Trust* (CoT) to which SPs and IdPs adhere to by signing a business agreement, in order to support secure transactions among CoT members. Each CoT member may know a user under distinct identities. All identities are related or federated in such a way that the authentication process can be performed by any CoT member. Any IdP within the CoT may authenticate a user.

Figure 9.3 shows an example CoT. For a user to access any service inside the CoT (Step 1 in the figure), an SP asks the user to select an IdP, and the user is redirected to this IdP for authentication (Step 2). The IdP authenticates the user and as-

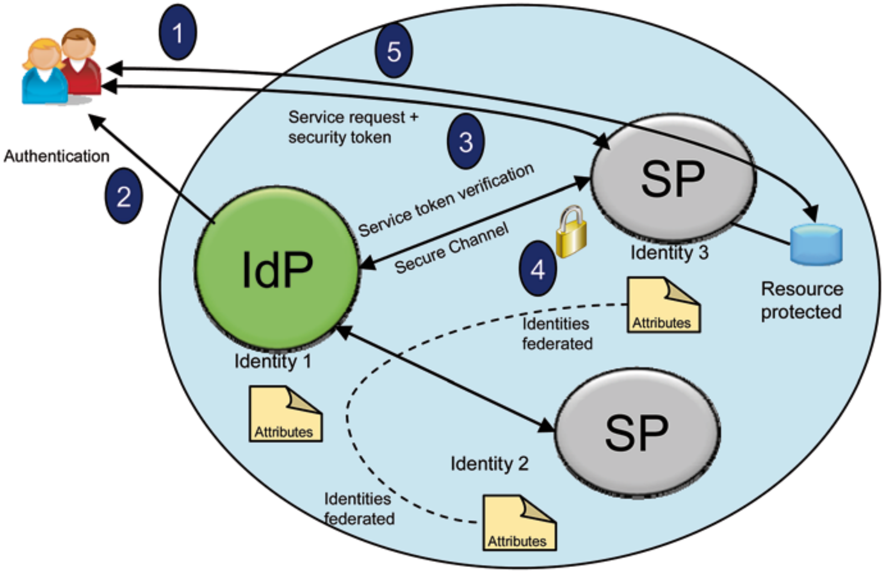


Fig. 9.3 Circle of trust

signs a “security token” with identity information which is next forwarded to the SP (Step 3); the “security token” is verified between the SP and IdP in a back secured channel (Step 4), and in case of validity, access to the service is granted (Step 5). If the SP requires additional attributes, then they are requested to the IdP through the secure channel. The CoT model requires that SPs trust IdPs. Therefore, it requires a secure communication infrastructure that guarantees the integrity, confidentiality and non-repudiation of the interchanged messages. The security mechanisms in the specification of Liberty Alliance include security in communication channels as well as security in message exchanges [19]. The secure communication can be implemented by means of current standard protocols such as TLS, SSL and IPsec. These protocols implement authentication mechanisms among SPs, IdPs and users before initiating message exchanges.

In transforming an enterprise into a Cloud environment, the IdP that the user selects is one of the IdPs that correspond to the user’s enterprise network. Therefore, the token that the IdP issues contains attributes that the enterprise network is allowed to request from the SPs. This enables enterprises to provide their users services from SPs that the Cloud itself provides or that are provided by other enterprises in the Cloud.

Note that the need for establishing secure tunnels and business agreements among the IdPs, and SPs may not offer enough flexibility for building large CoTs. On the other hand, WS-Federation can support large numbers of users, IdPs and SPs due to the flexibility of Web services so that they can be programmed to behave as IdPs or SPs.

The WS-Federation model includes three elements: the *Requestor* (RQ), that is, an application requiring access to Web services; the IdP or *Security Token Server* (STS) whose function is to carry out the authentication process and to transmit security tokens with relevant attributes; and the *Resource Provider* (RP), which is formed by one or more Web services that provide the resources required by the Requestor [20].

Figure 9.4 shows the interactions among the different components of the architecture. When RQ in security domain A requests a Web service located in another security domain (B in the figure), it is first authenticated by its IdP and obtains a security token with its identity information (Step 1 in the figure). Depending on the requested Web service, an additional access token may be obtained from the STS in security domain B with the necessary attributes to request the resource (Step 2). Finally, the security token is presented to the Web service (RP), which evaluates the security token and then applies its access control policy in order to grant access to the protected resource (Step 3).

In transforming an enterprise into a Cloud environment, security domain A corresponds to the user’s enterprise network. Therefore, the token that the IdP in domain A issues contains attributes that the enterprise network is allowed to request from the SPs. This enables enterprises to provide their users services from SPs that the Cloud itself provides or that are provided by other enterprises in the Cloud.

The WS-Federation architecture assumes that Web applications would be accessed only via other Web applications. On the other hand, the Liberty Alliance

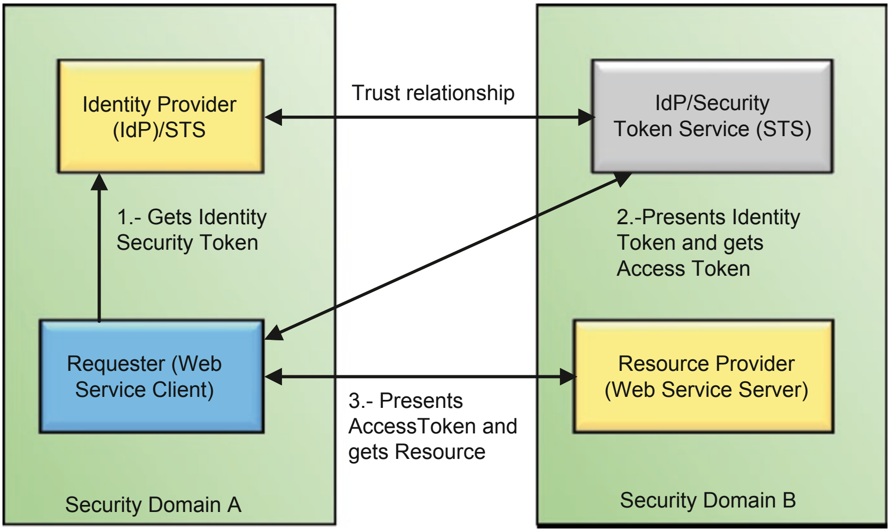

Identity Provider (IdP)/STS
Trust relationship
IdP/Security Token Service (STS)
1.- Gets Identity Security Token
2.-Presents Identity Token and gets Access Token
Requester (Web Service Client)
Resource Provider (Web Service Server)
3.- Presents AccessToken and gets Resource
Security Domain A
Security Domain B

Fig. 9.4 Relationship among components of web service architecture

architecture allows access to Web applications from Web browsers or from other Web application.

9.3.2 *Side Channel Attacks and Counter-Measures*

Cloud infrastructure can introduce side channel vulnerabilities in the security of Cloud Computing [21]. A *side channel attack* is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the cryptography algorithms [22]. It is possible to map internal Cloud infrastructure, identify where a particular target Cloud VM is likely to reside, and then instantiate new VMs until one is placed co-resident with the target [21]. Such placement can then be used to mount cross-VM side channel attacks to extract information from a target VM on the same machine. The following sections discuss these issues and show counter-measures to this threat.

9.3.2.1 Threat Model

To maximize efficiency, Cloud providers may simultaneously assign multiple VMs to execute on the same physical server. Moreover, Cloud providers may allow *multi-tenancy*, i.e., multiplexing the VMs of disjointed customers upon the same physical hardware. Thus, it is conceivable that customers' VMs can be assigned to the same physical server as their adversaries. This, in turn, engenders the threat that an adversary may penetrate the isolation between VMs (e.g., via a vulnerability that allows an "escape" to the hypervisor or via sidechannels between VMs) and violate customer confidentiality. Therefore, attacks require two main steps: *placement* and *extraction*. *Placement* refers to adversaries arranging to place their malicious VMs on the same physical machine as that of a target customer. This is further described in Sect. 9.3.2.2 below. Having managed to place a VM co-resident with the target, *extraction* refers to adversaries managing to obtain confidential information via a cross-VM attack. While there are a number of avenues for such an attack, the focus is on information leakage due to the sharing of physical resources (e.g., the CPU's data caches) [21]. There are some building blocks, such as cache load measurements, and coarse-grained attacks, such as measuring activity burst timing, that enable practical side-channel attacks when transforming enterprises into Cloud-computing environments. This is further described in Sect. 9.3.2.3 below.

9.3.2.2 Exploiting Placement Locality

Many Cloud providers keep their placement algorithms secret. This secrecy does not prevent attackers from collecting observations about the behavior of the placement algorithms and then exploiting these observations. An empirical measurement study [21] was done to understand VM placement in the EC2 system and how to achieve co-resident placement for an adversary. In this study, network probing was used both to identify public services hosted on EC2 and to provide evidence of co-residence, i.e., that two instances share the same physical server. In particular, TCP connect probes and SYN traceroutes were used. Next, this study mapped the EC2 service to understand where potential targets were located in the Cloud and the instance creation parameters needed to attempt establishing co-residence of an adversarial instance. This speeded up adversarial strategies for placing a malicious VM on the same machine as a target. To map EC2, this study made several hypotheses regarding the assignment of internal IP address ranges to parts of the Cloud infrastructure. This mapping allowed an adversary to determine which IP addresses corresponded to which creation parameters, thereby dramatically reducing the number of instances needed before a co-resident placement was achieved. To confirm the hypotheses, the study used data from several instances launched under several accounts. The following are some of the typical placement behavior observations [21]:

- A single account is not likely to have two instances simultaneously running on the same physical machine, so running instances in parallel under a single account results in placement on separate machines.
- The number of instances that each physical machine supports is limited, and this number can be readily known or estimated.
- While a machine is full, i.e., assigned its maximum number of instances, an attacker has no chance of being assigned to it.
- Launched instances exhibit both strong sequential and strong parallel locality. Sequential placement locality exists when two instances run sequentially, i.e.,

the first is terminated before launching the second, and both are often assigned to the same machine. Parallel placement locality exists when two instances run (from distinct accounts) at roughly the same time are often assigned to the same machine.

- There is a strong correlation among instance density, the number of instances assigned to a machine, and a machine's affinity for having a new instance assigned to it. In other words, there is a bias in placement towards machines with fewer instances already assigned. This would make sense from an operational viewpoint under the hypothesis that Cloud providers balance loads across running machines.

By using the hypotheses, this study offered two adversarial strategies to achieve co-residence with target victims, saying the attacker was successful if the attacker achieved good coverage (co-residence with a notable fraction of the target set). The brute-force strategy had an attacker simply launch many instances over a relatively long period of time. Such a naive strategy already achieved reasonable success rates (though for relatively large target sets). A more refined strategy had the attacker target recently-launched instances. This took advantage of the tendency for Cloud networks to assign fresh instances to the same small set of machines. Measurements show that the strategy achieves co-residence with a specific instance almost half the time.

9.3.2.3 Cross-Virtual Machine Information Leakage

By placing attack VMs on a specific physical machine, an attacker can perform several malicious actions, namely create covert channels, detect the rate of Web traffic a co-resident site receives, and time keystrokes by an honest user of a co-resident instance. Creating covert channels between two cooperating processes running in different VMs may not seem like a major threat for current deployments, since in most cases the cooperating processes can simply talk to each other over a network. However, covert channels become significant when communication is (supposedly) forbidden by *Information Flow Control* (IFC) mechanisms such as sandboxing and IFC kernels [23]. The latter are a promising emerging approach to improving security, and creating covert channels highlights a caveat to their effectiveness. In addition, note that the same resources can also be used to mount cross-VM performance degradation and DOS attacks, analogously to those demonstrated for non-virtualized multiprocessing [24].

Detecting the rate of Web traffic a co-resident site receives could be damaging if, for example, the co-resident Web server is operated by a enterprise competitor. In keystroke timing attacks [25], the adversary's goal is to measure the time between keystrokes made by a victim typing a password (or other sensitive information) into, for example, an SSH terminal. The gathered inter-keystroke times (if measured with sufficient resolution) can then be used to perform recovery of the password.

9.3.2.4 Counter-Measures

A Cloud provider could likely render network-based co-residence checks, for example, a provider may have the physical infrastructure machines not respond in traceroutes, may randomly assign internal IP addresses at the time of instance launch, or may use virtual LANs to isolate accounts [21]. If such precautions are taken, attackers can turn to co-residence checks that do not rely on network measurements. Even so, inhibiting network-based, co-residence checks would impede attackers to some degree, and so determining the most efficient means of obfuscating internal Cloud infrastructure from adversaries is a good potential avenue for defense.

Regardless of these mentioned defense mechanisms, SPs can have a straightforward option to "patch" all placement vulnerabilities: offload choice to users [21]. Namely, let users request placement of their VMs on machines that can only be populated by VMs from their (or other trusted) accounts. In exchange, the users can pay the opportunity cost of leaving some of these machines under-utilized. In an optimal assignment policy (for any particular instance type), this additional overhead should never need to exceed the cost of a single physical machine.

One may focus defenses against cross-VM attacks on preventing the side channel vulnerabilities themselves. This might be accomplished via blinding techniques to minimize the information that can be leaked (e.g., cache wiping, random delay insertion, adjusting each machine's perception of time [26], etc.). Countermeasures for covert channels (which appear to be particularly conducive to attacks) are extensively discussed in the literature [27]. These countermeasures suffer from two drawbacks. First, they are typically impractical, e.g., have high overhead or nonstandard hardware, are application-specific, or are insufficient for fully mitigating the risk. Second, these solutions ultimately require being confident that all possible side channels have been anticipated and disabled—itself a tall order, especially in light of the deluge of side channels observed in recent years. Thus, for unconditional security against cross-VM attacks, one must resort to avoiding co-residence.

9.4 Intrusion Detection in Cloud Computing

There are several classes of intruders that can occur in a Cloud environment. Three most common classes are listed below [28]:

- *Masquerader*: An individual who is not an authorized user of a system and who penetrates a system's access controls to exploit a legitimate user's account
- *Misfeasor*: A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses the privileges
- *Clandestine user*: An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

The objective of an intruder is to gain access to a system or to increase the range of privileges accessible on a system [29]. Generally, this requires the intruder to acquire information that should have been protected. In some cases, this information is in the form of a user password. With knowledge of some other user's password, an intruder can log in to a system and exercise all the privileges accorded to the legitimate user.

Inevitably, the best intrusion prevention system fails. A system's second line of defense is intrusion detection, and this has been the focus of much research in recent years. Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified. Of course, enterprises cannot expect that there will be a crisp, exact distinction between an attack by an intruder and the normal use of resources by an authorized user. Rather, we must expect that there will be some overlap [29].

Figure 9.5 suggests, in very abstract terms, the nature of the task confronting the designer of an intrusion detection system. Although the typical behavior of an intruder differs from the typical behavior of an authorized user, there is an overlap in these behaviors. Thus, a loose interpretation of intruder behavior, which will catch more intruders, will also lead to a number of "false positives," or authorized users identified as intruders. On the other hand, an attempt to limit false positives by a tight interpretation of intruder behavior will lead to an increase in false negatives, or intruders not identified as intruders. Thus, there is an element of compromise and art in the practice of intrusion detection [29].

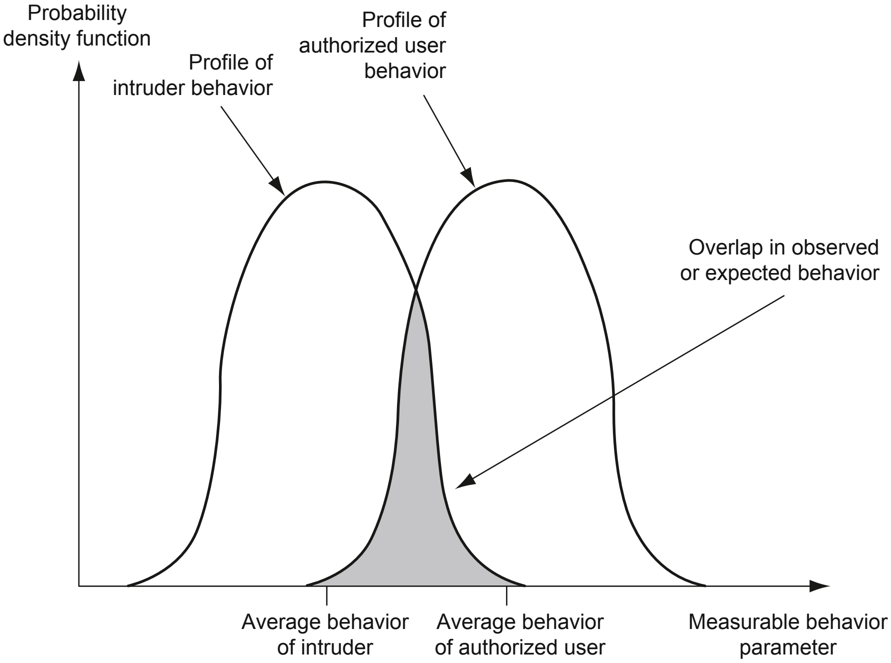

Probability density function
Profile of intruder behavior
Profile of authorized user behavior
Overlap in observed or expected behavior
Average behavior of intruder
Average behavior of authorized user
Measurable behavior parameter

Fig. 9.5 Profiles of behavior of intruders and authorized users

Intrusion detection can be implemented in different forms, depending upon the service types, implementation mechanisms, and attack types. Two common approaches are listed below [29]:

- *Statistical anomaly detection*: This approach involves the collection of data relating to the behavior of legitimate users over a period of time. Statistical tests are the applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior. The following are two examples of statistical anomaly detection:
  - *Threshold detection*: This approach involves defining thresholds, independent of users, for the frequency of occurrence of various events.
  - *Profile-based*: A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.
- *Rule-based detection*: This involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder. The following are two examples of rule-based detection:
  - *Anomaly detection*: Rules are developed to detect deviations from previous usage patterns.
  - *Penetration identification*: An expert system approach that searches for suspicious behavior.

In a nutshell, statistical approaches attempt to define normal or expected behavior, whereas rule-based approaches attempt to define proper behavior. In terms of the types of attackers listed earlier, statistical anomaly detection is effective against masqueraders, who are unlikely to mimic the behavior patterns of the accounts they appropriate. On the other hand, such techniques may be unable to deal with misfeasors. For such attacks, rule-based approaches may be able to recognize events and sequences that, in context, reveal penetration. In practice, a system may exhibit a combination of both approaches to be effective against a broad range of attacks.

Until recently, work on *Intrusion Detection Systems* (IDSs) focused on single-system, stand-alone facilities. Cloud providers, however, need to defend a distributed collection of enterprises. Although it is possible to mount a defense by using stand-alone IDSs on each host, a more effective defense can be achieved by coordination and cooperation among IDSs across the network.

There are some major and known issues in the design of a distributed intrusion detection system [29]:

- A distributed intrusion detection system may need to deal with different audit record formats, as discussed in Sect. 9.4.1 below. In a Cloud environment, different enterprises employ different native audit collection systems and, if using intrusion detection, may employ different formats for security-related audit records.
- One or more nodes in the network serve as collection and analysis points for the data from the systems on the network. Thus, either raw audit data or summary

data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality of these data. Integrity is required to prevent intruders from masking their activities by altering the transmitted audit information. Confidentiality is required because the transmitted audit information could be valuable.

- Either a centralized or decentralized architecture can be used. With a centralized architecture, there is a single, central point of collection and analysis for all audit data. This eases the task of correlating incoming reports, but creates a potential bottleneck and single point of failure. With a decentralized architecture, there is more than one analysis center. These centers must coordinate their activities and exchange information. Sect. 9.4.2 below discusses a distributed intrusion detection architecture.

9.4.1 Types of Raw Data Collected

A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system. Stallings discusses two plans [29]:

- *Native audit records*: Virtually all multiuser OS include accounting software that collects information on user activity. The advantage of using this information is that no additional collection software is needed. The disadvantage is that the native audit records may not contain the needed information or may not contain it in a convenient form.
- *Detection-specific audit records*: A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system. One advantage of such an approach is that it could be made vendor-independent and ported to a variety of systems. The disadvantage is the extra overhead involved in having, in effect, two accounting packages running on a machine.
    - An example of detection-specific audit records is one developed in [30]. Each audit record contains the following fields:
        - *Subject*: A subject is an initiator of actions. A subject is typically a terminal user but might also be a process acting on behalf of users or groups of users. All activity arises through commands issued by subjects. Subjects may be grouped into different access classes, and these classes may overlap.
        - *Action*: An action is an operation performed by the subject on or with an object; for example, login, read, perform I/O, execute.
        - *Object*: An object is a receptor of actions. Examples include files, programs, messages, records, terminals, printers, and user- or program-created

structures. When a subject is the recipient of an action, such as electronic mail, then that subject is considered an object. Objects may be grouped by type. Object granularity may vary by object type and by environment. For example, database actions may be audited for the database as a whole or at the record level.

  - ○ *Exception-Condition*: An exception-condition denotes which, if any, exception-condition is raised on return.
  - ○ *Resource-Usage*: This is a list of quantitative elements in which each element gives the amount used of some resource, e.g., the number of lines printed or displayed, the number of records read or written, processor time, I/O units used, session elapsed time.
  - ○ *Time-Stamp*: This is a unique time-and-date stamp identifying when the action took place.

9.4.2 Distributed Intrusion Detection Architecture

IDSs logically consist of three functional components [31]:

- *Sensors*: They are responsible for collecting data
- *Analyzers*: They are responsible for analyzing data and determining if an intrusion has occurred
- *UI*: It enables a user to view the output or control the behavior of the system

Figure 9.6 depicts a DHT-based overlay architecture as the backbone for a distributed intrusion detection system [32]. As a virtual communication structure lay logically on top of physical networks, the overlay network maintains a robust virtual inter-networking topology. Through this topology, trusted, direct, application-level functionalities facilitate inter-site policy negotiation and management functions such as authentication, authorization, delegation, policy exchange, malicious node control, job scheduling, resource discovery and management, etc.

The system functions as a *Cooperative Anomaly and Intrusion Detection System* (CAIDS). Intrusion information is exchanged by the overlay topology with confidentiality and integrity. Each local IDS is autonomous, and new algorithms can be added easily due to the high scalability of the overlay. Each node may work as an agent for others and various security models/policies can be implemented.

In Fig. 9.6, available functional blocks include the WormShield [33], CAIDS [34], and DDoS pushback scheme [35].

The CAIDS is supported by alert correlation sensors. These sensors are scattered around the Cloud infrastructure. They generate a large amount of low-level alerts. These alerts are transmitted to the alert correlation modules to generate high-level intrusion reports, which can provide a broader detection coverage and lower more false alarm rates than the localized alerts generated by a single IDS. Figure 9.7 shows the alert operations performed by various functional modules locally and globally.

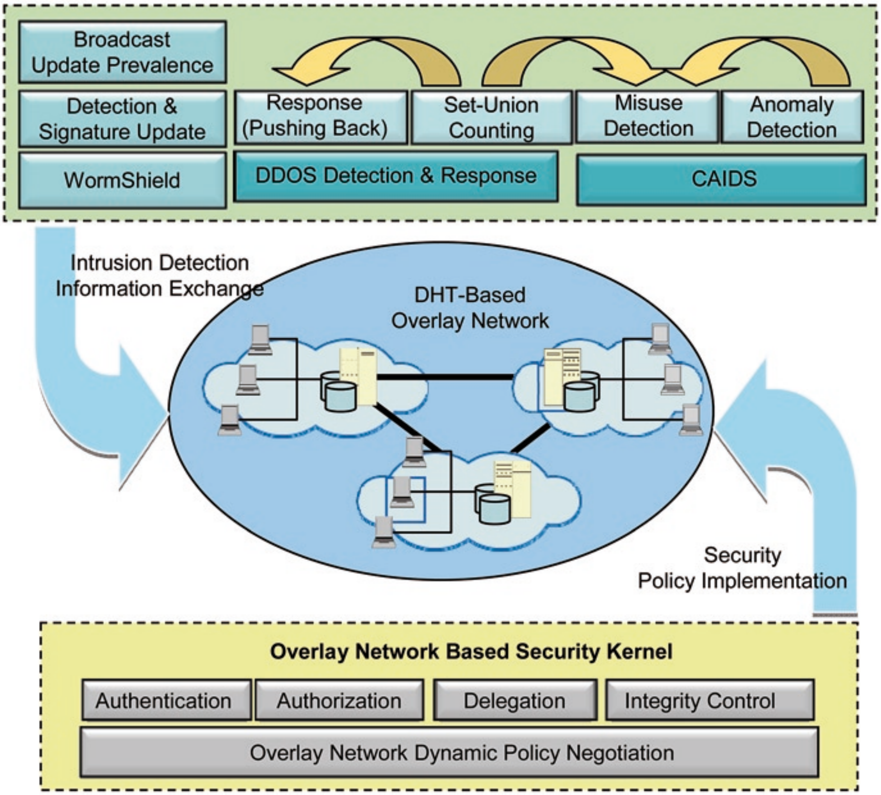

Broadcast Update Prevalence
Detection & Signature Update
WormShield
Response (Pushing Back)
Set-Union Counting
Misuse Detection
Anomaly Detection
DDOS Detection & Response
CAIDS
Intrusion Detection Information Exchange
DHT-Based Overlay Network
Security Policy Implementation
Overlay Network Based Security Kernel
Authentication
Authorization
Delegation
Integrity Control
Overlay Network Dynamic Policy Negotiation

Fig. 9.6 DHT-based distributed intrusion detection system

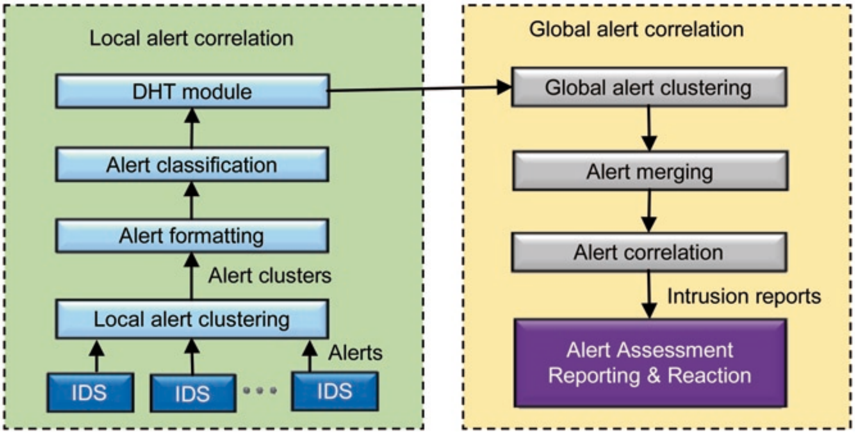


Fig. 9.7 Alert operations performed locally and correlated globally

9.4.3 Fusion-Based Intrusion Detection Systems

Multi-sensor data fusion in distributed IDS, such as in a Cloud infrastructure, is based on the concept that combination of data from multiple sensors can enhance the quality of the resulting information [31]. Data fusion enables the combination of, and intelligent reasoning with, the output of different types of IDSs. By making inferences from the combined data, a multiple level-of-abstraction situation description emerges. The Intrusion Detection Data Fusion model is shown in Fig. 9.8. The Level 1 fusion results in a collection of objects representing the observed data, the object base. This object base is further analyzed by the Level 2 and Level 3 processes to form the situation base. At the lowest level of inference, a fusion-based IDS indicates the existence of an intrusion. At the highest level of inference, such an IDS presents an analysis of the threat of the current situation.

9.4.3.1 Functional Data Fusion Process Model

The Functional Data Fusion Process Model, depicted in Fig. 9.9, consists of eight components [31]:

- *Sources provide input to the data fusion system*: Possible sources are local sensors, distributed sensors, human input and a priori information from databases. Multiple sensors that are from the same type are called commensurate sensors, as opposed to non-commensurate sensors that are of different types.
- Source pre-processing is sometimes referred to as "*Level 0 Processing*" or "*Process Assignment*:" It covers initial signal processing and allocates data to appropriate processes. It enables the data fusion process to focus on data that applies most to the current situation and reduces the data fusion system load.
- *Level 1 Processing*: Object Refinement fuses sensor information to achieve a refined representation of an individual entity. It usually consists of four functions:
  - *Data Alignment*, which aligns data received from multiple sensors to a common reference frame
  - *Association*, which combines, sorts, or correlates observations from multiple sensors that relate to a single entity
  - *Tracking*, which involves the combination of multiple observations of positional data to estimate the position and velocity of an entity
  - *Identification*, which combines data related to identity to refine the estimation of an entity's identity or classification

    Level 1 fusion benefits from the use of heterogeneous sensors, the employment of spatially distributed sensors, and the application of non-sensor derived information.
- *Level 2 Processing*: Situation Refinement develops a contextual description of relations between entities. It focuses on relational information to determine the meaning of a group of entities. It consists of object aggregation, event and

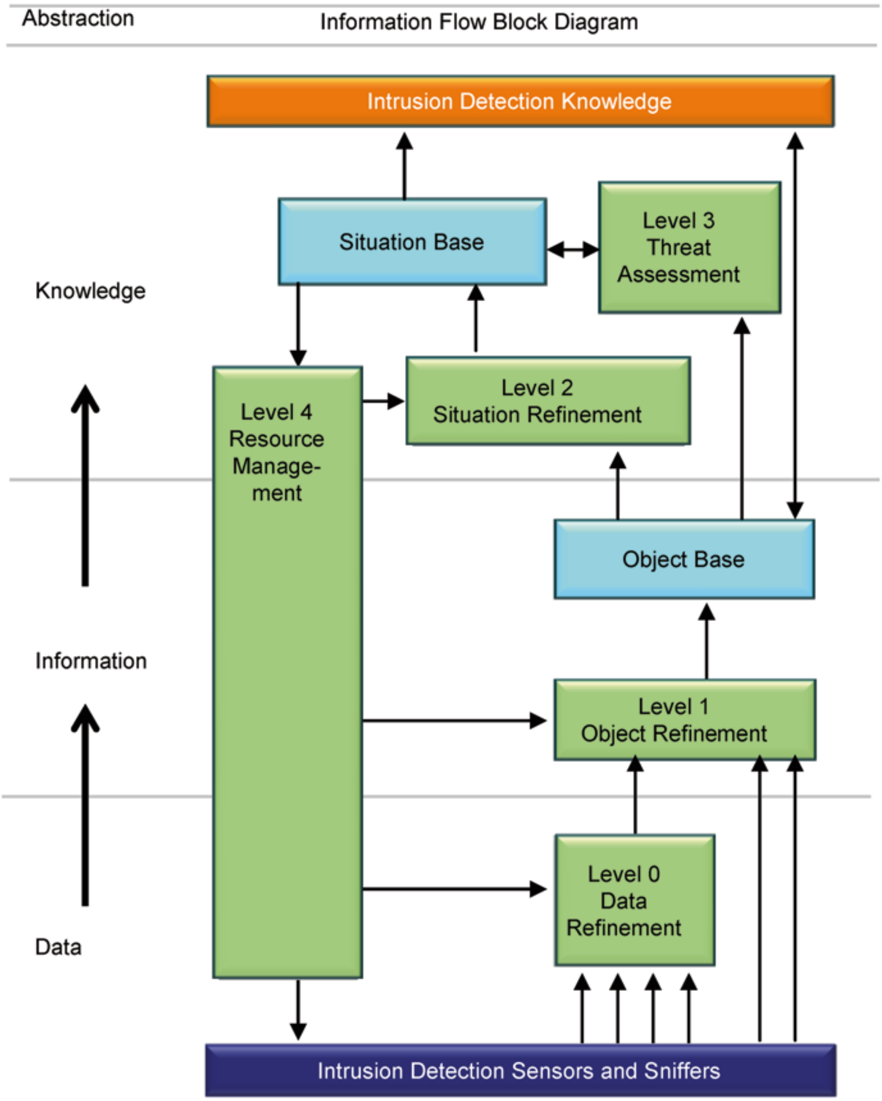

Abstraction
Information Flow Block Diagram
Intrusion Detection Knowledge
Situation Base
Level 3 Threat Assessment
Knowledge
Level 4 Resource Manage-ment
Level 2 Situation Refinement
Object Base
Information
Level 1 Object Refinement
Level 0 Data Refinement
Data
Intrusion Detection Sensors and Sniffers

Fig. 9.8 IDS data fusion model

activity interpretation, and eventually contextual interpretation. Its results are indicative of hostile behavior patterns. It effectively extends and enhances the completeness, consistency, and level of abstraction of the situation description produced by Object Refinement.

- *Level 3 Processing*: Threat Refinement analyzes the current situation and projects it into the future to draw inferences about possible outcomes. It identifies potential enemy intent and friendly force vulnerabilities. Threat refinement focuses on intent, lethality, and opportunity.

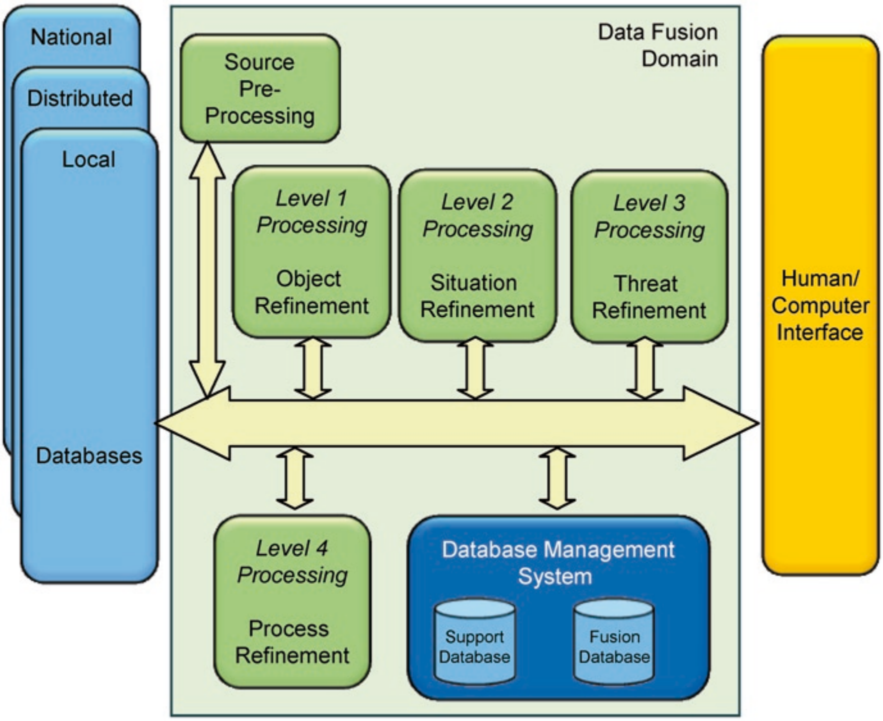


Fig. 9.9 Functional data fusion process model

- *Level 4 Processing*: Process Refinement is a meta-process that aims to optimize the overall performance of the fusion system. It consists of four key functions:
  - Performance evaluation, which provides information about real-time control and long-term performance
  - Process control, which identifies the information needed to improve the multilevel fusion product
  - Source requirements determination, which determines the source-specific requirements to collect relevant information
  - Mission management, which allocates and directs sources to achieve mission goals

Part of the process refinement, in particular mission management, may be outside the domain of specific data fusion functions. It is therefore partially placed outside the fusion process in Fig. 9.9 [31]:

- The Database Management System provides access to, and management of, data fusion databases. It is the most extensive support function for data fusion processing. Its functions include data retrieval, storage, archiving, compression, relational queries, and data protection.

- The Human-Computer Interface allows human input into the data fusion process. It is also a means of communicating data fusion results to a human operator.

9.4.3.2 Data Fusion Architectures

A fundamental issue regarding data fusion systems is the question where in the data flow the fusion must take place, or in other words, the choice of architecture. There are three architectural approaches to the fusion of information at Level 1 in the fusion model [31]. The first approach involves the fusion of raw sensor data, and is called centralized fusion (with raw data), or data-level fusion, as shown in Fig. 9.10. It is the most accurate way of fusing data, but may also require much communication bandwidth, since all raw data must be transmitted from the sensors to a central processing facility. The fusion of raw data is possible if commensurate sensors are available.

Another possible architecture is centralized fusion with feature vector data, as shown in Fig. 9.11. This is also called feature-level fusion. In this architecture, feature vectors rather than raw data are transmitted to the central fusion process. The feature vectors are extracted from the raw data by the sensors. Since the feature vectors are a representation of the raw data, this approach inherently results in data loss. In practice, this is often less problematic than it sounds. Compared with data-level fusion, feature-level fusion has advantages that might outweigh the disadvantage of data loss. Although there is some data loss, feature-level fusion enables the fusion

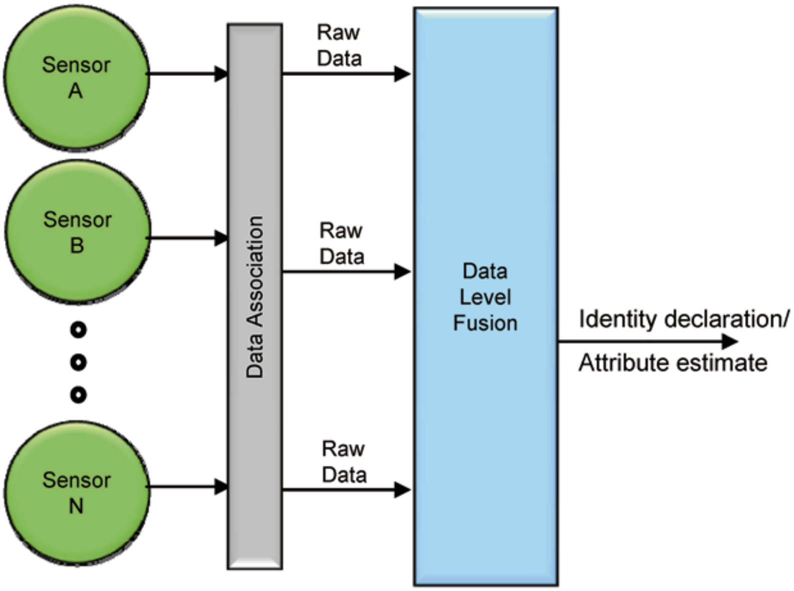


Fig. 9.10 Data level fusion

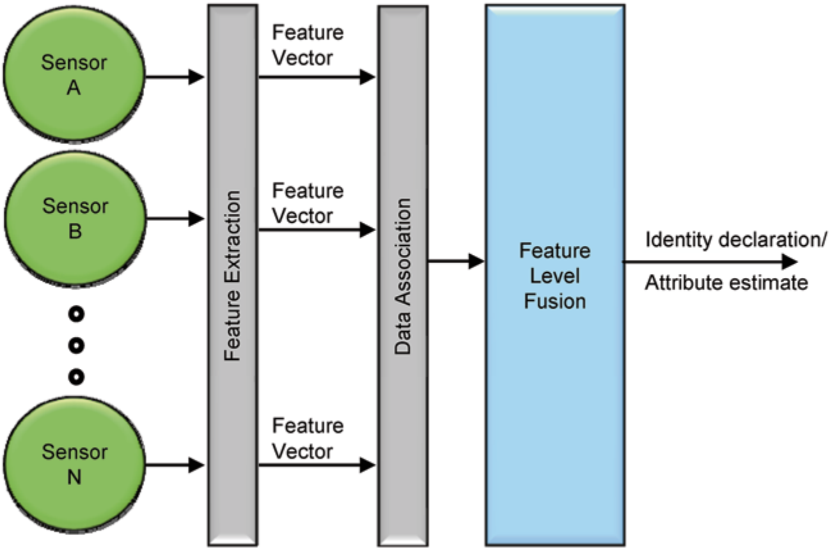


Fig. 9.11 Feature-level fusion

of data from non-commensurate sensors and reduces the required communication bandwidth.

Autonomous fusion, or decision-level fusion, is the third possible Level 1 fusion architecture, as shown in Fig. 9.12. Instead of outputting raw data or feature vectors, a sensor makes a decision based on its own single-source data. This decision, a

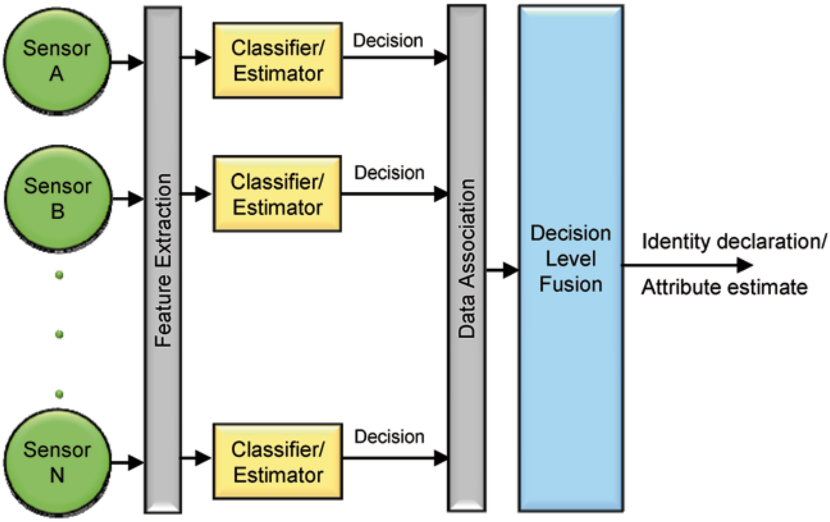


Fig. 9.12 Decision-level fusion

declaration of identity or estimation of position and/or velocity, is the input for the fusion process. This allows data from non-commensurate sensors to be fused. There is significant data loss compared with raw data fusion, and decision-level fusion may result in a local rather than a general optimized solution.

There is no ‘best’ architecture in general. The choice of architecture depends on requirements and constraints, such as the available communications bandwidth, sensor characteristics, etc. For each application, the architectural advantages and disadvantages must be weighed against each other.

The multiple level-of-abstraction situational view that a fusion-based IDS maintains is summarized in Fig. 9.13. This figure shows what kind of information is associated with the abstraction levels represented by Levels 1, 2, and 3 of the data fusion model. It also shows the relationships between the different levels of abstraction. The collection of alerts forms the lowest level of situation description. Analysis reveals two types of objects at a higher abstraction level: *attacks* and *attackers*. There is a close relationship between attacks and attackers in that there is always

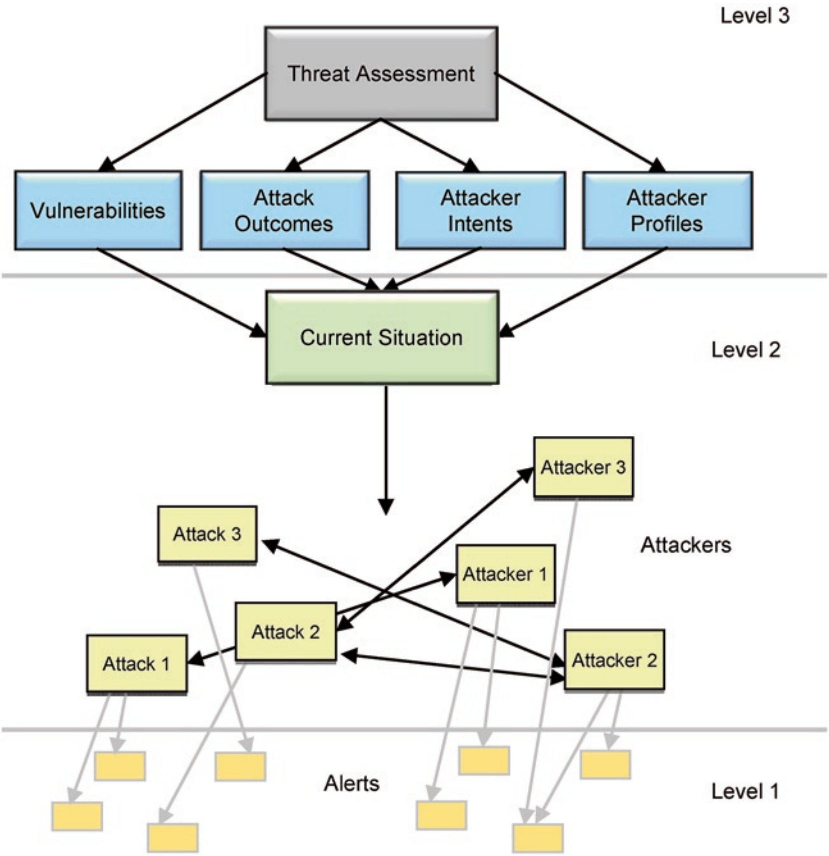


Fig. 9.13 Multiple levels of abstraction

at least one attacker involved in an attack, and there is at least one attack that an attacker is involved in. At yet another level higher, the impact of the current situation is analyzed by taking into account the capabilities and intents of the attackers, and the vulnerabilities of the monitored systems. The multiple level-of-abstraction view is attained in two different steps: combination, or fusion, of data originating from multiple sensors (Level 1) and further fusion of, and reasoning with, the fused Level 1 data (Levels 2 and 3).

9.5 Security for Cloud Service Management

In this section, we will discuss methods to ensure that enterprises use secure APIs to access Cloud services. The section also explains how to secure VM resources.

9.5.1 Security for APIs

Cloud APIs are APIs into Cloud services and are specific to how applications and their source code interact with the Cloud infrastructure. A Cloud API is a mechanism by which software can request information from one or more Cloud Computing platforms through a direct or indirect interface. Cloud APIs are most commonly written to expose their interfaces as REST [36] or SOAP [37]. Since most Public Clouds are Web-based, power Web applications, and provide interfaces and management utilities that are Web-based, most use an open REST style architecture for their Cloud APIs. There are many examples of Cloud APIs including both Cloud provider based APIs and cross-platform-based Cloud APIs. Cloud provider-based APIs commonly provide an abstraction from the Cloud provider's internal APIs, but still require API calls specific to their infrastructure implementation. Cross-platform-based Cloud APIs attempt to abstract the details of Cloud provider implementations so that an application or developer writing an application only has to call a single API to get a response regardless of the back-end Cloud.

CSA published a report that listed insecure interfaces and APIs as a top threat to Cloud Computing [38]. Cloud Computing providers expose a set of software interfaces or APIs that customers use to manage and interact with Cloud services. Provisioning, management, orchestration, and monitoring are all performed using these interfaces. The security and availability of general Cloud services is dependent upon the security of these basic APIs. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy. Furthermore, organizations and third parties often build upon these interfaces to offer value-added services to their customers. This introduces the complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency.

While most providers strive to ensure security is well integrated into their service models, it is critical for consumers of those services to understand the security implications associated with the usage, management, orchestration, and monitoring of Cloud services. Reliance on a weak set of interfaces and APIs exposes organizations to a variety of security issues related to confidentiality, integrity, availability, and accountability.

Enterprises often provide abstract APIs for security operations [39]. Applications use these APIs to access multiple keystores, such as OpenSSL files and security tokens, and multiple validation modules, such as on-line revocation servers and CRL checking. Often, the APIs can be extended by third parties for proprietary and legacy implementations.

The APIs enable an application to manage security keys, such as create and manage public/private key pairs, certificates, certificate validation, and storage and retrieval of keys. The APIs also enable common cryptographic operations, such sign, verify, encrypt, and decrypt.

To ensure security for APIs, Cloud users need to sign API calls to launch and terminate instances, change firewall parameters, or perform other functions with the users' private keys or secret keys. Without access to the customer's keys, API calls cannot be made on their behalf. In addition, API calls can be encrypted in transit with SSL to maintain confidentiality. By using SSL-protected endpoints for API calls, SSL can provide server authentication. New SSH host keys should be created on first boot and logged. Users can then use the secure APIs to obtain the logs and access the host keys before logging into the instance for the first time.

9.5.2 Security for Service Containers

OS, command interpreters, and application environments provide a way for software instructions to be executed [40] when transforming an enterprise into a Cloud environment. The concept of *execution containers* is an architectural abstraction used to describe virtual compute resources. Sun Microsystems defines a *secure execution container* as a special class of secure component that provides a safe environment within which applications, jobs, or services can be run. Execution containers are frequently used within the context of OS: OS instances (real or virtual) can themselves be run on physical, logical, or virtual hardware platforms. Execution containers can also be environments in which applications, services, or other components are executed, such as *Java 2 Enterprise Edition* (J2EE) Containers.

A secure execution container typically protects itself from unauthorized access or use by the services running within it, protects any service running within the container from unauthorized external influence, protects the infrastructure environment outside of the container if a running service becomes compromised, and provides an audit log of events occurring within the container.

Secure execution containers are charged with exposing (to their running services) only the interfaces that are specifically needed to support their successful operation and use. This is particularly crucial for transforming enterprises into dynamic Cloud environments, in which services are provisioned into, and executed within, secure execution containers. Secure execution containers should also restrict the activities of the users and services running on the system based upon business and technical requirements.

The methods used to deploy a secure execution container vary based on organizational requirements, product capabilities, and the threat profile for a given service or application. Some organizations or services may require physical separations, while others may employ virtualization at the electrical, logical, or resource level to achieve similar goals. Secure execution containers can be instantiated at the platform and OS layer using a variety of methods, such as separate platforms to enforce physical separation and separate dynamic system domains to provide electrical isolation.

Although there is nothing inherent in the secure execution container building block that precludes running multiple services within a container, organizations must assess and determine whether the expected rewards of running multiple services in a single container outweigh the potential risks. When deploying multiple services into a single secure execution container, additional protections must be implemented that protect each service from the others running in the same container. It is critical that controls be implemented to ensure that the compromise of one service does not lead to the immediate or effective compromise of the remaining services running in the container. Further, resource controls should also be implemented that limit the exposure of services to resource exhaustion attacks.

Immutable Service Containers (ISCs) are an architectural deployment pattern used to describe a platform for highly secure service delivery. Building upon concepts and functionality enabled by OS, hypervisors, virtualization, and networking, ISCs provide a security-reinforced container into which a service or set of services is deployed. By expressing core design principles, along with functional and non-functional requirements, ISCs are not constrained to a particular product or technology, but rather can be implemented using a variety of ways [41].

An ISC node provides a security-reinforced environment within which a single application, job, or service can be run. The intent of the use of ISC nodes is to develop a set of well-defined and verifiable security metrics that can be used in the creation and validation of security-reinforced service containers. By providing a set of core guiding principles, implementers can deal with their applications, jobs, or services in a controllable environment, thus reduces the complexity of cross-application security coordinations. The structure is shown in Fig. 9.14. ISC nodes extend upon the core principles of the secure execution container pattern in the following ways:

- Ensures that only a single, logical application or service is implemented per node
- Activates and exposes only those network services required for operation
- Restricts the initiation of outbound communication to those required for operation

- Uses immutable files and directories for critical, read-only items
- Uses encrypted storage for critical, sensitive items
- Operates with unique credentials and least privilege for all of its operations
- Monitors and audits all security relevant operations
- Operates within a resource-controlled environment

The components shown in Fig. 9.14 are as follows:

- A *service* is an object that is installed and executed within the ISC node. It can be a composite application, a single service, or a scheduled job. There are no restrictions on the type of service that can be used within the node. The actual service used will determine what security protections are needed in terms of the service itself, the node, and an ISC dock, all of which work in concert to provide a strong security boundary.
- The ISC node is installed within an ISC dock and is used to execute a given service.
- The ISC is managed by an ISC dock. The dock offers a way of aggregating one or more ISC nodes on a single, logical system. The dock also provides additional security protections, such as a centralized log and audit collection, resource con-

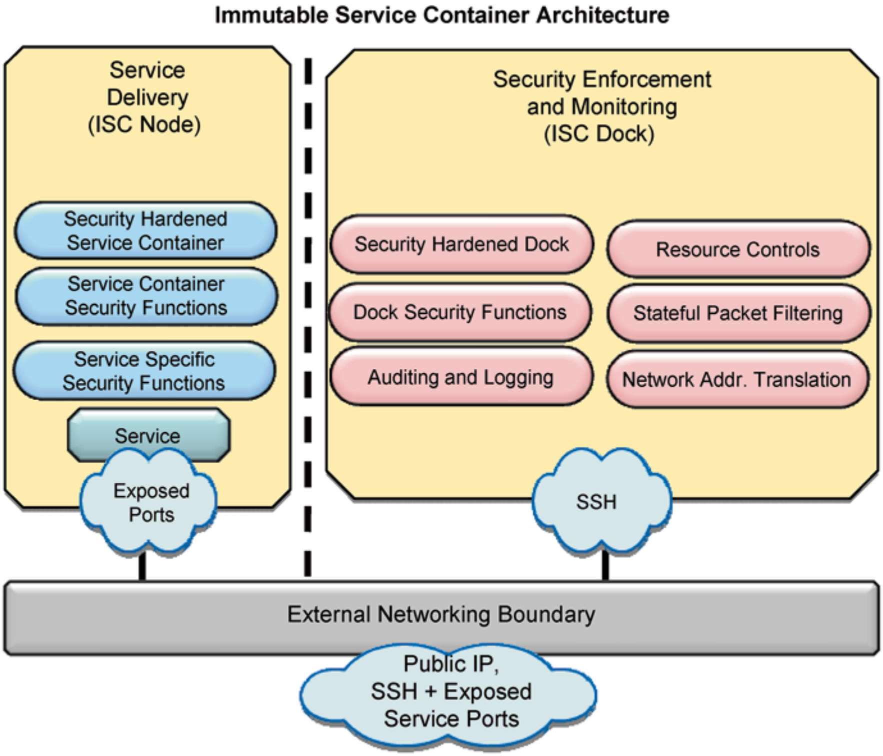

Immutable Service Container Architecture
Service Delivery (ISC Node)
Security Hardened Service Container
Service Container Security Functions
Service Specific Security Functions
Service
Exposed Ports
Security Enforcement and Monitoring (ISC Dock)
Security Hardened Dock
Resource Controls
Dock Security Functions
Stateful Packet Filtering
Auditing and Logging
Network Addr. Translation
SSH
External Networking Boundary
Public IP, SSH + Exposed Service Ports

Fig. 9.14 Immutable service container node structure

trol (e.g., compute, storage, and memory capacity, network bandwidth, etc.), and network-level protections to ensure that ISC nodes only communicate in accordance with a defined policy.
- The external networking boundary is actually a component of the ISC dock. Only for the sake of clarity was it shown as a separate entity in the above diagrams. It should be considered as a functional element of the ISC dock.
- There are a variety of security controls that can be implemented by the ISC node and ISC dock. The nature of these controls varies based upon the implementation model chosen. In addition to, or instead of, the controls shown in Fig. 9.14, additional security controls can be selected as appropriate.

9.6 Measures for Cross-Virtual Machine Security

In this section, we will discuss how to secure VMs against insider attacks and security risks introduced by sharing a VM image. In addition, the section discusses a method to *Secure File Systems* (SFS) that separate key management from file system security.

9.6.1 Virtual Machine Security

IaaS providers allow their customers to have access to all VMs hosted by the provider. The providers manage one or more clusters whose nodes run a hypervisor, i.e., a VM monitor to host customers' VMs. A VM is launched from *a* VMI. Once a VM is launched, users can log in to it using normal tools such as SSH. The hypervis or exports services that can be used to perform administrative tasks such as adding and removing VMIs or users. In addition, some hypervisors support live migration, i.e., allowing a VM to shift its physical host while still running, in a way that is transparent to the user. Migration can be useful for resource consolidation or load balancing within the cluster [42].

A system administrator working for the Cloud provider who has privileged control over the backend can perpetrate many attacks in order to access the memory of a customer's VM. With root privileges at each machine, the system administrator can install or execute many types of software to perform an attack. Furthermore, with physical access to the machine, a system administrator can perform sophisticated attacks like cold boot attacks and even tamper with the hardware. It is likely that no single person accumulates all these privileges. Moreover, providers already deploy stringent security devices, restricted access control policies, and surveillance mechanisms to protect the physical integrity of the hardware. Thus, we assume that, by enforcing a security perimeter, the provider itself can prevent attacks that require physical access to the machines. Nevertheless, system administrators still have privileged permissions at the cluster's machines in order to manage the software they run. Due to this, system administrators can login remotely to any machine with root privileges at any point in time and gain physical access to a node running a customer's VM. System administrators can accomplish this by diverting the VM to a machine under the administrators' control, perhaps located outside the IaaS's security perimeter. Therefore, a secure implementation must be able to confine the VM execution to occur inside the security perimeter, and be able to guarantee that at any point a system administrator with root privileges remotely logged to a machine hosting a VM cannot access its memory.

A possible implementation to address security for VMIs is to build on the techniques proposed by the Trusted Computing Group [43]. Two components can be used: a *trusted VM monitor* and a *trusted coordinator*. Each node runs a trusted VM monitor that hosts the customers' VMs and prevents privileged users from inspecting or modifying them. The trusted VM monitor protects its own integrity over time by using one of the techniques discussed in Sect. 9.2.2 above. Nodes must go through a secure boot process to install the trusted VM monitor. On the other hand, the trusted coordinator manages the set of nodes that can run a customer's VM securely. The nodes must be located within the security perimeter and run the trusted VM monitor. To meet these conditions, the trusted coordinator maintains a record of the nodes located in the security perimeter and attests to the nodes' platforms to verify that the node is running a trusted VM monitor. The trusted coordinator can cope with the occurrence of events such as adding or removing nodes from a cluster or shutting down nodes temporarily for maintenance or upgrades. A user can verify whether the IaaS service secures its computation by attesting to the trusted coordinator.

To secure the VMs, each trusted VM monitor running at each node cooperates with the trusted coordinator in order to confine the execution of a VM to a trusted node and to protect the VM state against inspection or modification when it is in transit on the network. The trusted coordinator is expected to be hosted on an external trusted entity that securely updates the information provided to the trusted coordinator about the set of nodes deployed within the IaaS perimeter and the set of trusted configurations. Intruders and system administrators that manage the IaaS have no privileges inside the external trusted entity and, therefore, cannot tamper with the trusted coordinator. The external trusted entity is likely to be maintained by a third party with little or no incentive to collude with the IaaS provider.

9.6.2 File System Security Management

Efficient instantiation of VMs across distributed resources requires middleware support for the transfer of large VM state files (e.g., memory, disks, etc.) and thus poses challenges to data management infrastructures [44]. The *Hadoop Distributed File System* (HDFS) is an example of a distributed file system that is used in large-scale Cloud infrastructures [45]. Nevertheless, security currently is limited to simple file permissions and network authentication protocols, like Kerberos, for user authentication. Encryption of data transfers is not supported.

For effectiveness consideration, SFS can separate its key management from file system security [46]. SFS file names themselves effectively contain public keys, making them *self-certifying pathnames*, as discussed in Sect. 9.6.2.1 below. Thus, key management in SFS occurs outside of the file system, in whatever procedure users choose to generate file names. SFS decouples user authentication from the file system through a modular architecture. External programs authenticate users with protocols opaque to the file system software itself. These programs communicate with the file system software through RPC interfaces. SFS splits overall security into two pieces: *file system security* and *key management*. *File system security* means attackers cannot read or modify the file system without permission, and programs get the correct content of whatever files they ask for. SFS assumes that users trust the clients they use. For instance, clients must actually run the real SFS software to get its benefits. Attackers can intercept packets, tamper with them, and inject new packets onto the network. Under these assumptions, SFS ensures that attackers can do no worse than delay the file system's operation or conceal the existence of servers until reliable network communication is reestablished. SFS cryptographically enforces all file access control. Users cannot read, modify, delete, or otherwise tamper with files without possessing an appropriate secret key, unless anonymous access is explicitly permitted. SFS also cryptographically guarantees that results of file system operations come from the appropriate server or private key owner. Clients and read-write servers always communicate over a low-level secure channel that guarantees secrecy, data integrity, freshness (including replay prevention), and forward secrecy (secrecy of previously recorded encrypted transmissions in the face of a subsequent compromise). The encryption keys for these channels cannot be shortened to insecure lengths without breaking compatibility.

File system security in itself does not usually satisfy a user's overall security needs. *Key management* lets the user harness file system security to meet higher-level security goals. The right key management mechanism depends on the details of a user's higher-level goals. A user may want to access a file server authenticated by virtue of a pre-arranged secret password, a file system of a well-known company, or even a catalog of any reputable merchant selling a particular product. No key management mechanism satisfies all needs. Thus, SFS provides primitives from which users can build a range of key management mechanisms [46].

9.6.2.1 Self-certifying Pathnames

SFS cryptographically guarantees the content of remote files without relying on external information. SFS therefore introduces *self-certifying pathnames*; file names that inherently specify all information necessary to communicate securely with remote file servers, namely a network address and a public key. Every SFS file system is accessible under a pathname in the form /sfs/Location:HostID. *Location* tells an SFS client where to look for the file system's server, while *HostID* tells the client how to certify a secure channel to that server. Location can be, for example, a DNS hostname or an IP address. To achieve secure communication, every SFS server has a public key. HostID is a cryptographic hash of that key and the server's Location. HostIDs let clients ask servers for their public keys and verify the authenticity of the reply. Knowing the public key of a server lets a client communicate securely with it [46].

SFS clients do not need to know about file systems before users access them. When a user references a non-existent self-certifying pathname in /sfs, a client attempts to contact the machine named by Location. If that machine exists, runs SFS, and can prove possession of a private key corresponding to HostID, then the client transparently creates the referenced pathname and mounts the remote file system there. Given an Internet address or domain name to use as a Location, anyone can generate a public key, determine the corresponding HostID, run the SFS server software, and immediately reference that server by its self-certifying pathname on any client in the Cloud.

Key management policy in SFS results from the names of the files users decide to access. Some users can retrieve self-certifying pathnames with their passwords. Others can get the same paths from a certification authority. Yet others may obtain the paths from an untrusted source, but want to peruse the file system anyway. In other words, SFS delivers cryptographic file system security to whatever file system the users actually name [46].

9.6.2.2 Server Key Management

Most users do not want to manipulate raw, self-certifying pathnames. Thus, server key management techniques can be built on SFS so that ordinary users need not concern themselves with raw HostIDs. Examples include manual key distribution. If the administrators of a site want to install some server's public key on the local hard disk of every client, they can create a symbolic link to the appropriate self-certifying pathname.

Similarly, SFS CAs can be implemented as file systems serving symbolic links. Unlike traditional CAs, SFS CAs get queried interactively. This places high integrity, availability, and performance needs on the servers. On-the-fly symbolic link creation can be used to exploit existing PKIs. For example, one might want to use SSL certificates to authenticate SFS servers. An agent that generates self-certifying pathnames from SSL certificates can be built. The agent intercepts every request for a file name and contacts the hostname's secure Web server, downloads and checks the server's certificate, and constructs from the certificate a self-certifying pathname to which to redirect the user.

9.6.3 Virtual Machine Image Security

The reduction of management costs, in both hardware and software, constitutes one of the value propositions of Cloud Computing. This cost reduction comes from sharing the knowledge of how to manage a piece of IT assets via VMIs. Nevertheless, VMI sharing unavoidably introduces security risks [47]. A user of Cloud services risks running vulnerable or malicious images introduced into the Cloud repository by a publisher. While running a vulnerable VM lowers the overall security level of a virtual network of machines in the Cloud, running a malicious VM is similar to moving the attacker's machine directly into the network, bypassing any firewall or IDS around the network. VMI sharing provides a straightforward way of developing and propagating Trojan horses. Traditionally, a trojan horse program can only be developed and tested on a hacker's machine and runs on a victim's machine only if the victim's software stack satisfies its dependencies. Therefore, to target a wide range of victims, the hacker must develop and test variances of the trojan horse on various software stacks and make sure that the right version is delivered to the right victim. Using a VMI as a carrier for the trojan horse makes the hacker's job easier than before, because the VMI encapsulates all software dependencies of the Trojan horse. In other words, the dependency on the victim's software stack is eliminated. Users of Cloud services also risk running illegal software, e.g., unlicensed or with expired licenses, contained in an image.

A Cloud provider risks hosting and distributing images that contain malicious or illegal content. In addition, security attributes of dormant images are not constant. Typically, the security level of a dormant VMI degrades over time, because a vulnerability may be unknown when the VMI is initially published, but becomes known and exploitable at a later time. If dormant VMIs are not managed, e.g., scanned periodically for worms, a virtual environment may never converge to a steady state, because worm-carrying VMIs can sporadically run, infect other machines, and disappear before they can be detected. The same idea holds for software licenses. Administrators thus carry a latent security risk that stems from long-lived, but inactive, images. This risk is often over-looked by administrators due to the high-maintenance costs of keeping those images up to date with regard to security patches and software licenses. As the number of VMIs grows, so does the risk and along with it the cost of maintenance.

The security concerns discussed above is possible to be addressed by using the image management system [47]. Figure 9.15 shows the overall architecture of such system, with an emphasis on its security capabilities. It consists of four major components that implement four features:

- An access control framework that regulates the sharing of VMIs. This reduces the risk of unauthorized access to images.
- Image filters that are applied to an image at publish and retrieve times to remove unwanted information in the image. Unwanted information can be information that is private to the user, such as passwords; malicious, such as malware; or il-

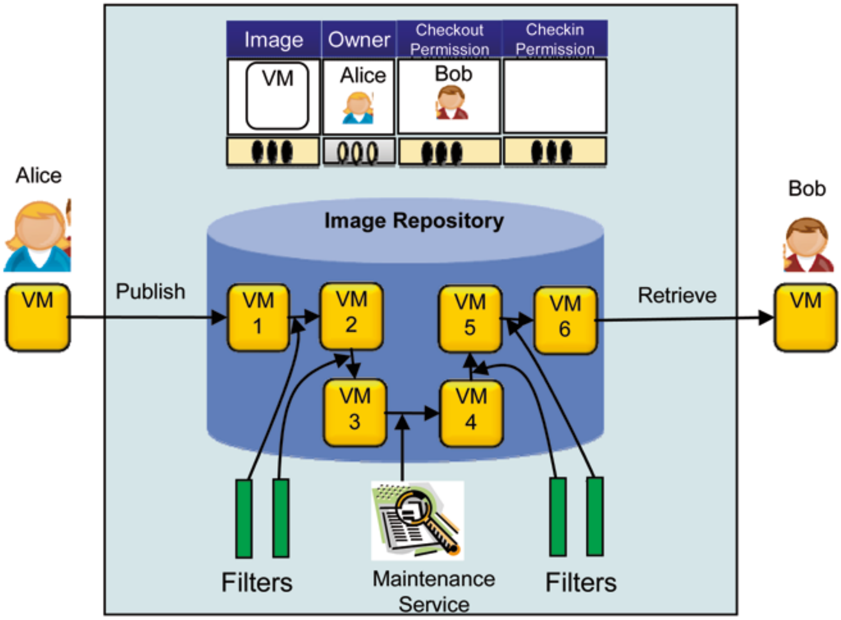


Fig. 9.15 Security features of the management system

legal, such as pirated software. Filters reduce users' risk of consuming illegal or harmful content.

- A provenance tracking mechanism that tracks the derivation history of an image and the associated operations that have been performed on the image through an image repository API. Security functionalities like auditing can be built on top of this provenance tracking layer. Provenance tracking provides accountability and discourages the intentional introduction of malicious or illegal content, which in turn reduces a Cloud provider's risk of hosting images that contain such content. The provenance mechanism also tracks modifications to images that result from applying filters.
- A set of repository maintenance services, such as periodic virus scanning of the entire repository, that detect and fix vulnerabilities discovered after images are published. These reduce users' risk of running malicious or illegal software and the risk of hosting them.

Each image in the repository in Fig. 9.15 has a unique owner, who can share images with trusted parties by granting access permissions. Examples of access permissions are *checkout* and *checkin*. A checkin permission implies a checkout permission, whereas retrieving and running an image requires a checkout permission. Revising an image and storing the revised image in the repository requires a checkin permission. Note that, even without a checkin permission for an image, a user can retrieve the image, modify it, and publish it as a new image; however, the provenance-tracking system would not consider the new image to be a revision of the original. All other operations on an image, such as granting and revoking access to the image, require the operator to be the owner (or the repository administrator). By default, an image is private, meaning that no one but the owner and the administrator can access the image.

Filters at publish time can remove or hide sensitive information from the publisher's original image. For example, a remove-filter excludes a file from the original image, and a hide-filter keeps the file but replaces its content with a safer version, e.g., replacing credit card numbers with invalid numbers.

The image management system tracks the derivation history of an image by recording the parent image information when a new image is deposited into the repository, along with the information about the operation that resulted in the creation of the new image. For example, if Bob in Fig. 9.15 checks out image A owned by Alice, modifies the image, and later checks it back in as a new image B, the system records that image B derives from image A via the *checkin* method. As another example, if the system discovers a vulnerability in image C and applies the latest security patch for it that results in a new image D, the system records the fact that image D derives from image C using the *maintenance* method, as well as the specific patch that was applied. This *provenance* information is used in two ways. It can be used by an audit system to trace the introduction of illegal or malicious content. It can also be used to alert the owners of derived images when the parent image is patched, e.g., because a vulnerability is discovered and fixed, so that the derived images can be patched as well.

As mentioned previously, dormant images are more than just static data. They should be regularly checked for compliance, scanned for malware, and patched with the latest security fixes.

As an alternative to the techniques discussed above, the security and management functionalities can be performed at the client's side instead of at the repository. For instance, users can employ software tools to remove traces of personal information from the user's hard drive. Users can also schedule a periodic task to scan the dormant images for viruses and expired licenses. However, doing it only at the client side is potentially less effective and less efficient than when combined with security management operations at the repository. It is less effective because not all users are aware of privacy protection tools or have access to them to protect sensitive information or to cleanse downloaded VMIs. Implementing security at the client side may also miss many performance optimization opportunities that are only present in a centralized image repository system. Having a large set of images provides the opportunity to explore data mining techniques to automatically discover sensitive or malicious data that might have been missed by off-the-shelf tools [47].

9.7 Conclusion

This chapter discussed the transformation of enterprise security into Cloud services and infrastructure security. The security concerns for Cloud providers differ from the concerns of the Cloud users. Cloud providers want to ensure that only authorized

Cloud provider personnel can modify the basic services provided by the Cloud. On the other hand, the Cloud providers want to enable authorized users of the Cloud to use the IaaS and SaaS functions to customize the services of the Cloud to suit the enterprise users' security needs. Thus, Cloud providers need to balance two seemingly conflicting objectives: prevent users of the Cloud from modifying the basic services that the Cloud provides to the users while enabling the users to customize the services to suit individual enterprise needs. In addition, the Cloud providers need to allow the individual enterprises to protect their security services from use by or disclosure to other enterprises. In addition, Cloud providers must ensure that any security vulnerabilities introduced by the security practices of individual enterprises do not affect the security of the Cloud itself and the security of other enterprise users of the Cloud.

The enterprise users of the Cloud want to ensure that the security services provided by the enterprise meet the security expectations of the enterprise users. In some cases, the security expectations may be more stringent than those offered by the underlying Cloud services. In other cases, the enterprise security requirements may be less stringent than the security provided by the underlying Cloud services. The enterprise users of the Cloud need to have the flexibility to trade-off security with speed and efficiency, regardless of the limits imposed by the Cloud services.

The data from competing enterprises may reside alongside one another in the same Cloud servers. In fact, data from one enterprise may reside in the Cloud servers of a competing enterprise. Hence, Cloud providers need to have services that ensure the anonymity of the sources of the data and the randomization of the location of the data.

Managing security for Cloud Computing requires RBAC architectures in the Cloud that can integrate well with customer systems. Security management comprises security for the Cloud network itself and security for customer data and infrastructure hosted in the Cloud. Security for the Cloud network itself requires secure APIs so that users of the Cloud are assured of the security of the services the Cloud offers. Security for data and infrastructure hosted in the Cloud requires that VMs for different customers operate autonomously so that the hardware and software resources used by one VM are securely protected from other VMs. RBAC needs enhancements for open and decentralized multi-centric systems, such as when transforming an enterprise into a Cloud Computing environment, where the user population is dynamic and the identity of all users is not known in advance [5, 6].

Federated identity management aims to unify, share, and link digital identities of users among different security domains. A FIA is a group of organizations that have built trust relationships among each other in order to exchange digital identity information in a safe way, preserving the integrity and confidentiality of the user's personal information. The FIA involves IdPs and SPs in a structure of trust by means of secured communication channels and business agreements. IdPs manage the identity information of users and do the authentication processes in order to validate their identities. SPs provide one or more services to users within a federation. In transforming an enterprise into a Cloud environment, the tokens that IdPs issue contain attributes that the enterprise network is allowed to request from the SPs. This enables enterprises to provide their users services from SPs that the Cloud itself provides or that are provided by other enterprises in the Cloud. There are some building blocks, such as cache load measurements, and coarse-grained attacks, such as measuring activity burst timing, that enable practical side-channel attacks when transforming enterprises into Cloud Computing environments. One may focus defenses against cross-VM attacks on preventing the side channel vulnerabilities themselves. This might be accomplished via blinding techniques to minimize the information that can be leaked (e.g., cache wiping, random delay insertion, adjusting each machine's perception of time, etc.). Countermeasures for covert channels (which appear to be particularly conducive to attacks) are extensively discussed in the literature. These countermeasures suffer from two drawbacks. First, they are typically either impractical, e.g., high overhead or nonstandard hardware, application-specific, or insufficient for fully mitigating the risk. Second, these solutions ultimately require being confident that all possible side channels have been anticipated and disabled—itself a tall order, especially in light of the deluge of side channels observed in recent years. Thus, for unconditional security against cross-VM attacks one must resort to avoiding co-residence [18, 27].

Until recently, work on IDSs focused on single-system, stand-alone facilities. Cloud providers, however, need to defend a distributed collection of enterprises. Although it is possible to mount a defense by using stand-alone IDSs on each host, a more effective defense can be achieved by coordination and cooperation among IDSs across the network. There are major issues in the design of a distributed IDS. A distributed intrusion detection system may need to deal with different audit record formats. In a Cloud environment, different enterprises employ different native audit collection systems and, if using intrusion detection, may employ different formats for security-related audit records. In addition, one or more nodes in the network serve as collection and analysis points for the data from the systems in the network. Thus, either raw audit data or summary data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality of these data. In addition, either a centralized or decentralized architecture can be used. With a centralized architecture, there is a single central point of collection and analysis of all audit data. This eases the task of correlating incoming reports but creates a potential bottleneck and single point of failure. With a decentralized architecture, there is more than one analysis center. These centers must coordinate their activities and exchange information. The main idea behind multi-sensor data fusion in distributed IDSs, such as the ones for Cloud infrastructure, is that the combination of data from multiple sensors enhances the quality of the resulting information. Data fusion enables the combination of, and intelligent reasoning with, the output of different types of IDSs. By making inferences from the combined data, a multiple level-of-abstraction situation description emerges [29, 31].

CSA published a report that listed insecure interfaces and APIs as a top threat to Cloud Computing. Cloud Computing providers expose a set of software interfaces or APIs that customers use to manage and interact with Cloud services. Provisioning, management, orchestration, and monitoring are all performed using these interfaces. The security and availability of general Cloud services is dependent upon the security of these basic APIs. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy. Furthermore, organizations and third parties often build upon these interfaces to offer value-added services to their customers. This introduces the complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency. To ensure security for APIs, Cloud users need to sign API calls to launch and terminate instances, change firewall parameters, or perform other functions with the users' private keys or secret keys.

OS, command interpreters, and application environments provide a way for software instructions to be executed when transforming an enterprise into a Cloud environment. The concept of execution containers is an architectural abstraction used to describe virtual compute resources. Sun Microsystems defines a secure execution container as a special class of secure components that provides a safe environment within which applications, jobs, or services can be run. Execution containers are frequently used within the context of OS: OS instances (real or virtual) can themselves be run on physical, logical, or virtual hardware platforms. Execution containers can also be environments in which applications, services, or other components are executed, such as J2EE Containers [40].

IaaS providers allow their customers to have access to all VMs hosted by the provider. The providers manage one or more clusters whose nodes run a hypervisor, i.e., a VM monitor to host customers' VMs. A system administrator working for the Cloud provider who has privileged control over the backend can perpetrate many attacks in order to access the memory of a customer's VM. With root privileges at each machine, the system administrator can install or execute all sorts of software to perform an attack. Furthermore, with physical access to the machine, a system administrator can perform sophisticated attacks like cold boot attacks and even tamper with the hardware. A possible implementation to address security for VM images is to build on the techniques proposed by the Trusted Computing Group. Two components can be used: a *trusted VM monitor*, and a *trusted coordinator*.

Efficient instantiation of VMs across distributed resources requires middleware support for the transfer of large VM state files (e.g., memory, disks, etc.) and thus poses challenges to data management infrastructures. Nevertheless, security currently is limited to simple file permissions and network authentication protocols, like Kerberos, for user authentication. Encryption of data transfers is not supported. A secure file system that separates key management from file system security can be implemented. In this implementation, file names themselves effectively contain public keys, making them self-certifying pathnames. Thus, key management occurs outside of the file system, in whatever procedure users choose to generate file names. This decouples user authentication from the file system through a modular architecture. External programs authenticate users with protocols opaque to the file system software itself. These programs communicate with the file system software through RPC interfaces.

The reduction of management costs, in both hardware and software, constitutes one of the value propositions of Cloud Computing. This cost reduction comes from sharing the knowledge of how to manage a piece of IT assets via VMIs. Nevertheless, VMI sharing unavoidably introduces security risks. A user of Cloud services risks running vulnerable or malicious images introduced into the Cloud repository by a publisher. While running a vulnerable VM lowers the overall security level of a virtual network of machines in the Cloud, running a malicious VM is similar to moving the attacker's machine directly into the network, bypassing any firewall or IDS around the network. VMI sharing provides a straightforward way of developing and propagating Trojan horses. Using a VMI as a carrier for the trojan horse makes the hacker's job easier than before, because the VMI encapsulates all software dependencies of the Trojan horse. In other words, the dependency on the victim's software stack is eliminated. Users of Cloud services also risk running illegal software, e.g., unlicensed or with expired licenses, contained in an image.

A Cloud provider risks hosting and distributing images that contain malicious or illegal content. In addition, security attributes of dormant images are not constant. If dormant VMIs are not managed, e.g., scanned periodically for worms, a virtual environment may never converge to a steady state, because worm-carrying VMIs can sporadically run, infect other machines, and disappear before they can be detected. The same idea holds for software licenses. As the number of VMIs grows, so does the risk, and along with it, the cost of maintenance. An image management system that addresses these security concerns can be implemented. The implementation consists of four major components that implement four features. The first feature is an access control framework that regulates the sharing of VMIs. This reduces the risk of unauthorized access to images. The second feature is an image filter that is applied to an image at publish and retrieve times to remove unwanted information in the image. Unwanted information can be information that is private to the user, such as passwords; or malicious, such as malware; or illegal, such as pirated software. Filters reduce users' risk of consuming illegal or harmful content. The third feature is a provenance tracking mechanism that tracks the derivation history of an image and the associated operations that have been performed on the image through an image repository API. Provenance tracking provides accountability and discourages the intentional introduction of malicious or illegal content, which in turn reduces a Cloud provider's risk of hosting images that contain such content. The provenance mechanism also tracks modifications to images that result from applying filters. The fourth feature is a set of repository maintenance services, such as periodic virus scanning of the entire repository, that detect and fix vulnerabilities discovered after images are published. These reduce users' risk of running or hosting malicious or illegal software.

References

1. The Committee on National Security Systems: National Information Assurance (IA) glossary, CNSS Instruction No. 4009. CNSS. June 2006
2. National Institute of Standards and Technology: Role Based Access Control (RBAC) and Role-Based Security, http://csrc.nist.gov/groups/SNS/rbac/, July/Aug 2010

3. American National Standards Institute: American national standard for information technology—role based access control, ANSI INCITS 359-2004. ANSI. Feb 2004
4. Sandhu, R.S., Coynek, E.J., Feinsteink, H.L., Youmank, C.E.: Role-based access control models. IEEE Comput. **29**(2), 38–47 (Feb 1996)
5. Chakraborty, S., Ray, I.: TrustBAC—integrating trust relationships into the RBAC model for access control in open systems. Proceedings of the ACM Symposium on Access Control Models and Technologies (SACMAT), ACM, Lake Tahoe, 7–9 June 2006
6. Otenko, D.C.A., Ball, E.: Role-based access control with X.509 attribute certificates. IEEE Internet Comput. **7**(2), 62–69 (March/April 2003)
7. Zhou, W., Meinel, C.: Implement Role-Based Access Control with Attribute Certificates. Forschungsgruppe Institut für Telematik, Universität Trier, 54286, Trier (n.d.)
8. Health Level Seven International, *HL7 Vocabulary,* http://www.hl7.org/ and http://www.hl7.org/v3ballot/html/infrastructure/vocabulary/vocabulary.htm, Version 1058-20100815, 22 Aug 2010
9. Spalding, R.S., III: Net-centric warfare 2.0: Cloud Computing and the new age of war. Air War College, Air University (22 Feb 2009)
10. Kaufman, C., Perlman, R., Speciner, M.: Network Security: Private Communication a Public World. Prentice Hall, Upper Saddle River (2002)
11. Park, J.S., Ahn, G.-J., Sandhu, R.: RBAC on the Web using LDAP. Proceedings of the 15th IFIP WG 11.3 Working Conference on Database and Application Security, IFIP, Ontario, 15–18 July 2001
12. Cisco Systems, *Public Export Product Data,* http://tools.cisco.com/legal/export/pepd/Search.do, 2006
13. COMMERCIAL ENCRYPTION EXPORT CONTROLS. http://www.bis.doc.gov/encryption/guidance.htm
14. Java Cryptography Architecture—cryptographic service provider. http://java.sun.com/j2se/1.4.2/docs/guide/security/CryptoSpec.html#ProviderArch
15. Microsoft Developer Network page about CSPs. http://msdn.microsoft.com/en-us/library/aa380245(VS.85).aspx
16. Neuman, C., Yu, T., Hartman, S., Raeburn, K.: The Kerberos network authentication service (V5). Internet Engineering Task Force Request for Comments (IETF RFC) 4120. July 2005
17. Barkley, J.F., Kuhn, D.R., Rosenthal, L.S., Skall, M.W., Cincotta, A.V.: Role-Based access control for the web. National Institute of Standards and Technology. http://csrc.nist.gov/rbac/cals-paper.html
18. Fragoso-Rodriguez, U., Laurent-Maknaviciu, M., Incera-Dieguez, J.: Federated identity architectures. 1st Mexican Conference on Informatics Security 2006 (MCIS 2006), IEEE Computer Society, Oaxaca, Nov 2006
19. Liberty Alliance. http://www.projectliberty.org/liberty/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates/
20. Goodner, M., Hondo, M., Nadalin, A., McIntosh, M., Schmidt, D.: Understanding WS-Federation. May 2007. http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fed/WS-FederationSpec05282007.pdf?S_TACT=105AGX04&S_CMP=LP
21. Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, you, get off of my cloud: exploring information leakage in third-party compute Clouds. Proceedings of 16th ACM Conference on Computer and Communications Security, ACM, Chicago, 9–13 Nov 2009
22. Side channel attack. Wikipedia. http://en.wikipedia.org/wiki/Side_channel_attack
23. Krohn, M., Tromer, E.: Non-interference for a practical DIFC-based operating system. Proceedings of the 2009 30th IEEE Symposium on Security and Privacy, IEEE, Oakland, 17–20 May 2009
24. Moscibroda, T., Mutlu, O.: Memory performance attacks: denial of memory service in multicore systems. Proceedings of 16th USENIX Security Symposium, USENIX, Boston, 6–10, Aug 2007
25. Song, D.X., Wagner, D., Tian, X.: Timing analysis of keystrokes and SSH timing attacks. Proceedings of 10th USENIX Security Symposium, USENIX, Washington, 13–17 Aug 2001

26. Hu, W.-M.: Reducing timing channels with fuzzy time. Proceedings of IEEE Symposium on Security and Privacy, IEEE, Oakland, 20–22 May 1991
27. Tromer, E., Osvik, D.A., Shamir, A.: Efficient cache attacks on AES, and countermeasures. J. Cryptol. **23**(1), 37–71 (2010)
28. Anderson, J.P.: Computer Security Threat Monitoring and Surveillance. James P. Anderson Co., Fort Washington (1980)
29. Stallings, W.: Cryptography and Network Security Principles and Practices, 4th edn. Prentice Hall, Upper Saddle River (2005)
30. Denning, D.E.: An intrusion-detection model. IEEE Trans. Software Eng. **13**(2), 222–232 (1987)
31. de Boer, R.C.: A generic architecture for fusion-based intrusion detection systems. Master Thesis, Erasmus University Rotterdam (Oct 2002)
32. Hwang, K., Kwok, Y.-K., Song, S., Chen, M.C.Y., Chen, Y., Zhou, R., Lou, X.: GridSec: trusted Grid Computing with security binding and self-defense against network worms and DDoS attacks. International Workshop on Grid Computing Security and Resource Management (GSRM'05), in conjunction with the International Conference on Computational Science (ICCS 2005), Emory University, Atlanta, 22–25 May 2005
33. Cai, M., Hwang, K., Kwok, Y.-K., Chen, Y., Song, S.: Collaborative internet worm containment. IEEE Secur. Priv. **3**(3), 25–33 (2005)
34. Hwang, K., Chen, Y., Liu, H.: Protecting network-centric computing system from intrusive and anomalous attacks. Proceedings of 1st IEEE International Workshop on Security in Systems and Networks (SSN'05), in conjunction with IEEE/ACM IPDPS, Denver, IEEE/ACM, 8 April 2005
35. Cai, M., Kwok, Y.-K., Hwang, K.: Inferring network anomalies from mices: a low-complexity traffic monitoring approach. ACM SIGCOMM Workshop on Mining Network Data, Philadelphia, Pennsylvania, 26 Aug 2005
36. RESTful Web services: the basics. IBM. Nov 2008. http://www.ibm.com/developerworks/webservices/library/ws-restful
37. Simple Object Access Protocol (SOAP). W3C, http://www.w3.org/TR/soap/
38. Top threats to Cloud Computing V1.0. Cloud Security Alliance. March 2010. http://www.cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf
39. Solaris security for developers guide. Sun Microsystems. Nov 2009
40. Toward systemically secure IT architectures. Sun Microsystems. Feb 2006
41. Immutable service containers. Open Solaris. 2010. http://hub.opensolaris.org/bin/view/Project+isc/Architecture
42. Santos, N., Gummadi, K.P., Rodrigues, R.: Towards trusted Cloud Computing. http://www.mpi-sws.org/~rodrigo/tccp-houtcloud09.pdf
43. The Trusted Computing Group. https://www.trustedcomputinggroup.org
44. Zhao, M., Zhang, J., Figueiredo, R.: Distributed file system support for virtual machines in Grid Computing. Proceedings of the 13th IEEE International Symposium on high performance distributed computing, IEEE, Honolulu, 4–6 June 2004
45. Hadoop Distributed File System (HDFS). Apache. http://hadoop.apache.org/common/docs/current/hdfs_design.html
46. Mazières, D., Kaminsky, M., Kaashoek, M.F., Witchel, E.: Separating key management from file system security. Operat. Sys. Rev. **34**(5), 124–139 (Dec 1999)
47. Wei, J., Zhang, X., Ammons, G., Bala, V., Ning, P.: Managing security of virtual machine images in a Cloud environment. Proceedings of the 2009 ACM Workshop on Cloud Computing Security, ACM, Chicago, 13 Nov 2009

Chapter 10
Enterprise Cloud Service Applications and Transformations[1,2]

The introduction of Cloud services forever changes the way end users use paid/free services. New expectations are on the rise, with customers demanding more control of their data. The tactical decision on the part of the service user to trust third parties with data access, management, and security is no longer acceptable, and users are quickly realizing the price of freedom in terms of the loss of rights and identity protection.

To meet these new requirements and challenges, enterprises are gearing up with deeper levels of on-demand computing. Vendors are pushing the envelope to move up towards platforms and applications by adding features for scalability and load balancing and by integrating other functions from content deliveries. Yet even with all this, the authors believe the changes are still in their infancy. Changes in these enterprises will introduce new functions. New experiences from improved functions and services will stimulate new expectations, and the cycle will last until full maturity of the Cloud technology is reached.

It is widely accepted that Cloud services will shift the emphasis from current business practices to closer value-chain relationships with a new level of commitment from the SPs. The content and definitions of services and SLAs are expected to be altered as the supplier and consumer relationship continues to evolve. When we look at some of the advances in the Cloud service industry and how the technology ecosystem is changing due to the Cloud paradigm, the answers may not be so unclear. In this final chapter, the authors will derive discussions from previous chapters with respect to business and technology transformation. Based on different enterprises' needs and the current trends of paradigm development, this chapter intends to conclude a viable enterprise transformation framework. It will also explain how enterprises can establish, explore, activate, and apply the new Cloud model to make them more competitive and profitable.

W. Y. Chang et al., *Transforming Enterprise Cloud Services,*
DOI 10.1007/978-90-481-9846-7_10, © Springer Science+Business Media B.V. 2010

10.1 Overview

Given the complexity of the current IT status and technology-based enterprises' market situation, privately held IT departments are becoming more complex. This is due to the increasing dependencies upon the enterprises' technology providers, supplier-chain partners, and computer managed customer relationships. Over time, enterprises' systems and their management procedures will become too massive and complex for even the most skilled system integrators to install, configure, optimize, maintain, and merge. The burden can also discount the enterprises' ability to make timely, decisive responses to the rapid stream of changing and conflicting business demands.

The Cloud service model, as seen in Chap. 2, addresses tactical problems with which IT continually deals, such as resource availability and reliability, datacenter costs, and operational process standardization. Most of these near-term objectives represent sufficient justification for enterprises to adopt Clouds, despite the fact that there may not be a need to improve their applications, platforms, or infrastructures. However, there are some imperative, longer-term business drivers for enterprises to improve business agility as well. These include the flexibility to integrate their capabilities with their partners by creating a distributed framework that can aggressively deliver or expand their existing products or services.

In Chaps. 2 and 4, the authors show that improving enterprise agility is no longer a hidden secret, as many Cloud vendors and integrators are advocating specific reasoning for adaptation. As a result, enterprise collaboration is no longer a competitive business edge, but a needed feature to stay in business. Many vendors have shown the value of real-time collaboration that is seamlessly integrated with business applications, the results put any standalone enterprise collaboration offerings in a disadvantaged position.

Larger enterprises require refined control of their processes and technologies due to scale and leadership. A unified framework to deal with their services that enables targeted community participants to coordinate their efforts is a key for success. From this perspective, the selling and adoption processes must be relevant and adaptable to specific issues. On the other hand, smaller enterprises drive their leadership positions in innovation and sharing, but are typically more resource constrained. The limitations can come either from operations or from their marketing development efforts. Regardless of the constraints, transforming into a Cloud can assist them in reaching out to a broader market and help them look bigger, faster, and stronger. These smaller enterprises can also demonstrate their competitiveness by significantly employing efficiency and flexibility in their Cloud approaches, with fast implementation and ease of administration.

With respect to the relationships between technologies and customers or among business partners, a trusted framework to tightly link supply chain participants can expedite the exchanges between members in the CoI. These exchanges can quickly deliver customers' needs to the suppliers, precisely capture and refine the requirements, and effectively deliver the requested products. During this service management process, communications services as a part of the Cloud are important to emerging vertical industry solutions, like connected cars enabling trust-based partnerships.

These advantages are not unique to client enterprises, they are also essential to SPs. For instance, although long being in the telecommunication SPs' core service domain, infrastructure alone may not drive lasting competitive advantage. This is because excellence in technology does not necessarily derive excellence in services. Realizing that Cloud technology can increasingly serve the business relationship, these providers are now proactively incorporating it into their business processes, in order to implement appropriate business priorities in their market. Other good examples are content and applications providers. These providers are increasingly needed for a set of processes and systems that can monetize and grow their subscriber base. An effective community-based platform for the creation, delivery, execution, and operations of the service will be a key factor for their Cloud transformations. For these providers, and various sizes of enterprises, one of their initiatives include the SDFs, as discussed in Chaps. 5 and 7.

In the following sections, the authors will recap the transformation steps from earlier chapters that include how enterprises can effectively establish, explore, act on, and apply full advantages of Cloud services. Although the majority of Cloud features and advantages are common across Cloud users, businesses, and IT Environments, we will differentiate domain-specific subjects as necessary throughout this chapter.

Regardless of whether enterprises are seeking transformation for their current datacenters, expanding their current service offerings to Cloud environments, or taking advantage of publically available Cloud products, general considerations can be seen in Fig. 10.1. This figure details a general transformation path and its focus areas. Theses actions of interest will be concluded based on different service categories and potential Cloud applications.

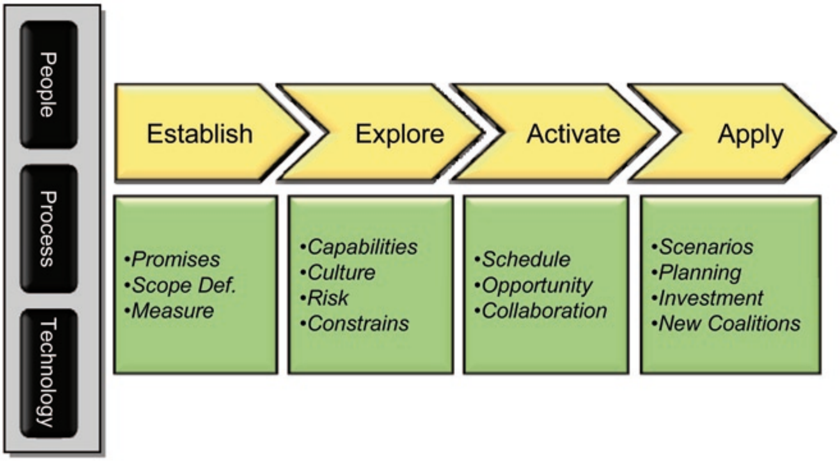


Fig. 10.1 Cloud transformation in enterprise architecture

10.2 Business and Technology Transformation

The introduction of the Cloud concept changes the ICT-related landscape dramatically, as we see a significant increase in the opportunities available to influence all types of SPs. Business opportunities for providers, datacenters, and organizations presented in the form of services are offered through a three-layer hierarchy across different Clouds, as seen in Fig. 10.2.

For instance, Web hosting SPs, telecommunication operators, *Internet service providers* (ISPs), ISVs, online services companies, systems integrators, and VARs are only a few of the players that are becoming Cloud-based or cost-plus SPs. With providers and consumers in mind, enterprises must take the right standpoint and the appropriate steps for a successful transformation.

In the following subsections, we will see some important concepts that are essential for the first step of adaptation.

10.2.1 Establish Strategic Promises

The first step for an enterprise to take when adopting a new technology to improve their business is to establish strategic promises to their organizations. Through this process, the enterprise leadership shows a clear vision and commitment and can enforce execution of the plan. After setting the goal, it will become dramatically easier for the enterprise to do business with Cloud providers. Meanwhile, new business models will emerge to make the Cloud more consumable. Furthermore, enterprises must include the following strategies in their adoption plan:

1. *Determine Larger Market Involvement*: Figure 10.3 is a simplified version of Gartner's Hype Cycle of Emerging Technologies, a research result published in

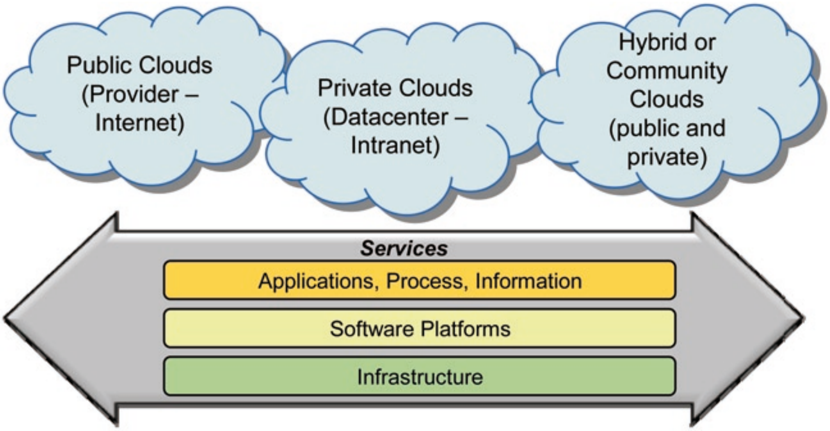

Public Clouds (Provider – Internet)
Private Clouds (Datacenter – Intranet)
Hybrid or Community Clouds (public and private)
Services
Applications, Process, Information
Software Platforms
Infrastructure

Fig. 10.2 Cloud deployments and services

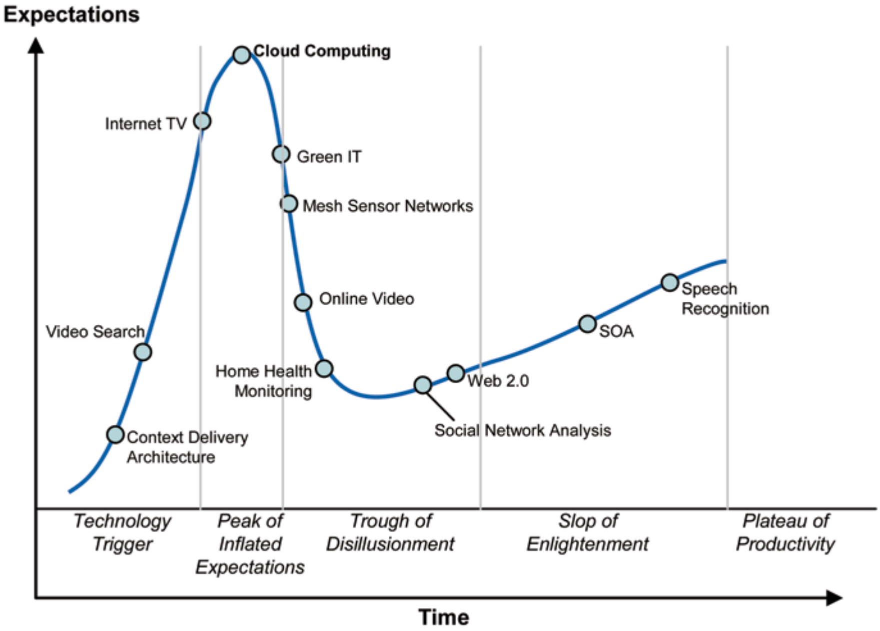


Fig. 10.3 Gartner's Hype Cycle of emerging technologies

August 2009. It shows that Cloud Computing is situated at the peak of inflated expectations after a series of technology triggers, the expectation level then gradually reduces until it reaches its lowest point and starts a new development cycle. Although this research does not necessarily prove that all the following technologies are dependent on the Cloud, it does show a sequence of potential continuity. In other words, global growth in technology demands will increase the importance of high-leverage application development. It enables more rapid development of higher-quality products for the market, similar to what we saw in the dot com era. Enterprises must be ready to deal with the trend of large vendors' and major international development enterprises' entrance into the Cloud business. If the case is clear, the enterprise should also be open to the option of outsourcing IT to the Cloud—allowing applications enabled by ubiquitous Internet access to move some of their on-premise IT infrastructure to an off-premise location. Due to ROI motivators, for instance, enterprises can eliminate significant capital expenditures of ongoing IT operations if they outsource the IT infrastructure to a specialist organization when developing, expanding, or replacing their ICT functionalities. The decision makers also need to consider the changes to the functions of the *Project Management Office* (PMO) when adapting to an evolved IT department in order to fit into the organizational value equation [1].

2. *Consider the Stage Growth Approach*: The level of adaptation to Cloud technologies varies, depending upon the enterprises' business objectives and operational goals. For enterprises that have a long-term goal to embrace a fully functional

Cloud environment, they can take steps toward the goal using different transformation paths. While adopting a partial Cloud can be the catalyst for a potentially more sound IT and financial strategy, it may not be able to address some key IT challenges. For instance, database intensive environments may not be conducive to partially residing within the Cloud due to their integrity requirements. In this case, the applications or infrastructure must remain in a private datacenter or run on dedicated servers in a Private Cloud, managed centrally in a virtualized environment by third parties or enterprise IT staff. Although the Private Cloud can meet the needs of an application system by any combination of Public and Internal Cloud resources, it is only affordable to large and medium sized enterprises. As aforementioned, the Private Cloud can support the degree of trust and address scalability for enterprises, allowing their users to consume the services from internal VPN or through a Public Cloud provider. As seen in Chap. 2, most large enterprises or organizations that use the Private Cloud continue to operate with the perception that everything is running and fully controlled in their own datacenters. However, regardless of whether enterprises choose a Private or Public Cloud as their first catalyst, they all can transition to a Hybrid Cloud without any major technical barrier. In a Hybrid Cloud, the only requirement is that individually consumed resources must be separately managed through the interfaces provided by their respective owners, connections can be made through dedicated services or Cloud Bursting. Enterprise decision makers can closely examine up-to-date Cloud standards for the most appropriate option that best fits their business [2, 3].

3. *Get Ready for Partnerships Galore*: To maintain competitive edges, some enterprises can secure their sustainable market share and/or establish new vehicles for unexplored business opportunities via new technology. One example focuses on sales initiatives and related technology investments to improve sales process efficiencies for profitable sales growth. With the high speed of information exchanges in modern market places, strategic alliances and partnerships are critical to any business success. In such event, the Cloud acts as an effective enabler for value-chain and market community partners to exchange their interests and business insights more closely and efficiently than ever. In a community-based marketplace, a Cloud-based solution not only increases the exposure of enterprises' products to other target audiences, but also provides innovative and robust frameworks for cross-domain business interactions. In previous chapters, the authors showed how enterprises can take advantage of Cloud Aggregators and Integrators to integrate their Clouds and management services. By doing so, enterprises can broaden their capabilities in order to deepen their involvement in community-based activities. Aggregators and Integrators typically have much deeper insights into the workings of each of their partners and thus should have a better position and interest in driving standards and interoperability. However, when a Cloud Aggregator or Integrator suggests vendor-dependent solutions, enterprises should cross-reference the corresponding industrial standards and guidance to ensure the adopted solution remains open and adoptive for future developments.

4. *Invest in Long Term Viability*: If driving long-term business growth is one of the enterprises' direct transformation goals, the project managers must incorporate inputs from their customers, providers, and integrators from relevant business domains in the earlier stages of the project. For instance, mainstream consumers are becoming more aggressive in lowering their cost of both personal and business computing devices that are lightweight, free running, and open-sourced OS and applications. Thus, enterprises must consider cross-broad infrastructures to support this trend. For enterprises whose target is to eliminate high *capital expenditures* (CapEx) and shift to *operating expenditures* (OpEx) by a pay-for-what-you-use and use-only-what-you-need model, the functions of their existing IT support must be altered. Finally, competitions in the Cloud market have prevented Cloud Aggregators and Integrators from charging their clients based purely on their services without having to introduce or invest other distinguished values. Therefore, enterprise customers should take advantage of this situation and work through their Aggregators or Integrators to either contribute to or influence their affiliated industrial communities or standard bodies to gain visibility or even market direction. Table 10.1 highlights some key business values of Public and Private Clouds from the previous chapters for enterprises to review. It also provides some sample business drivers as a reference for their transformation plan.

Table 10.1 Key business outcomes

Public Cloud	Private Cloud
Shift non-mission critical workloads out of expensive Datacenter Environments, offload non-mission critical applications out of the datacenter onto a low cost Public Cloud, and allow the datacenter to focus on core applications	Lower Costs through increased virtualization and automation
Increase application standardization through SaaS deployment and lower application support costs by standardizing key applications (such as e-mail, collaboration, office suites, CRM) through standardized SaaS offerings	Utilize advanced CloudWare, which allows for effective, automated management of highly virtualized Cloud Farms, providing greater utilization, lower management overhead, and significantly lower infrastructure costs (70–80% virtualization)
Transition Costs from Capital Expense to Operating Expense	Use efficient self-service model for provisioning & de-provisioning
Manage new provisioning in the Cloud and pay for it as an operating expense, eliminating expensive up-front capital costs	Use self-service provisioning to lower labor costs and provide faster, more effective service for users of datacenter services
Offload capacity spikes onto pay-as-you-need-it infrastructure	Eliminate underutilized DC assets via rapid repurposing to easily repurpose servers, storage, and software licenses (environmental and application) across a broad array of users via advanced provisioning tools
Allow Cloudbursting technology to load balance capacity spikes onto pay-as-you-go Cloud services, reducing infrastructure over-provisioning	Simplify and lower costs associated with Disaster Recovery
	Provide cheaper disaster recovery warm site implementation through the use of Hybrid Cloud infrastructure models

10.2.2 Plan for New Business Models

After firmly establishing enterprises' Cloud transformation strategies, the next step is to explore and activate their plans. With thousands of vendors and providers in active roles, the Cloud market is under the pressure of consolidation, although some activities are not obvious in certain functional domains. Meanwhile, as some technologies become mature, vendors gain further hands-on experience from various use-cases and business scenarios, resulting in higher degrees of readiness in different integration solutions. As discussed in Chaps. 3 and 7, the frameworks used by SPs to create, deploy, execute, and manage their services can be categorized as SDF. SDF is made up of applications, process integration software, and process functions. The SDF can be a leading capability in allowing SPs to support enterprises' service lifecycle management and to orchestrate or control the associated systems, processes, and partners, a useful enabler for transforming to the new business model.

1. *Realignment of Enterprises' Solutions Architectures*: Based on the new enterprise directions, their solution architectures must begin to align with what is offered by the Cloud. Their process orchestration can take in more Cloud-based software and integration elements to enhance the level of efficiency. It is also important to incorporate consumers' preferences, especially in the areas of connectivity, technology, and interactive experiences. While Cloud SPs are clearly evolving their strategies to show that technology is their core business, enterprises must be realistic in making sure the solution is feasible. For instance, the SDF is increasingly a core process that can link to the development of a technology strategy. As more end users select Cloud applications for their core business, practical requirements, such as meeting integration and operations needs from the Cloud, move from systems providers to SPs, leading to customization being in high demand. The drivers for these requirements are to reduce costs and strengthen system survivability due to constant changes on the underlying application systems. The emergence of the SDF capability as a business platform gives a clear trend and opportunity for enterprises to strengthen their commitment and services to their customers. Through the SDF framework, enterprises can integrate their processes, expertise, customer intimacy, and employee satisfaction in a common business framework to be more agile and competitive in the market. With SDF, enterprises can create billable service offerings on a unified platform, provide various subscriber data points, and establish critical real-time features to help install new services. It even includes the features of advertising, marketing, and storefront modules [4].
2. *New Functions of IT Departments*: It is inevitable that the IT department will be the center of some transformation projects, not only from the technology perspective, but also from the management process and resource management perspectives. As the role changes, IT managers will no longer be limited to oversee rollout, integration, and development projects. The IT functions of the new enterprise strategy focus on extracting the most business value from new

technologies. As Cloud technologies help IT departments shed the burden of technological implementations and assist enterprises in concentrating on business processes, employees in the new IT department must be equipped with new domains of knowledge, including project management, quality assurance testing, business analysis, and other high-level abstract thinking such as integration, collaboration, and standards. Also, the new IT environment broadens its array of participants outside the traditional domain and encourages business units and even individual, non-IT employees to control the processing of information directly, without the need for legions of technical specialists. These are the new requirements the enterprises involving a Cloud business model must consider.

3. *Future of Datacenters*: One of the primary benefits of future datacenters is the speed at which customers can bypass traditional IT departments to procure services. With Cloud technologies, datacenters can now offer an abstracted, fabric-based infrastructure that enables dynamic movement, growth, and protection of services that is billed like a utility. Services are based on consumption and the technology infrastructure and are optimized for hosting several customers. The resiliency of the growing number of systems and increasing amount of data can be improved. New technologies and delivery models can make mission-critical practices less burdensome. Figure 10.4 portrays the evolution path of service hosting, showing the hosting service from an old fashion ISP configuration, through collocation, to the dynamic Cloud architecture. Table 10.2 provides a review summary of the major differences between old datacenters and transformed Cloud-based datacenters [5, 6].

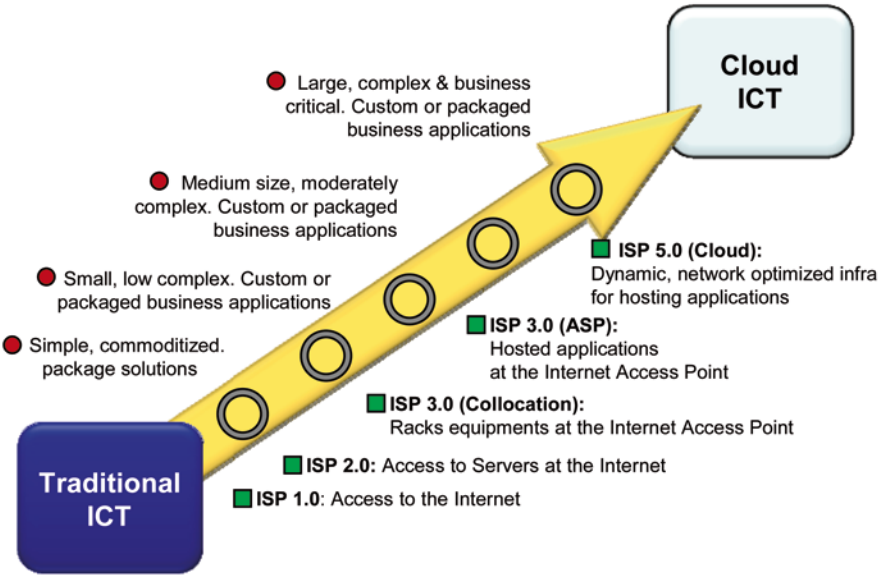

Cloud ICT
Large, complex & business critical. Custom or packaged business applications
Medium size, moderately complex. Custom or packaged business applications
Small, low complex. Custom or packaged business applications
Simple, commoditized. package solutions
ISP 5.0 (Cloud): Dynamic, network optimized infra for hosting applications
ISP 3.0 (ASP): Hosted applications at the Internet Access Point
ISP 3.0 (Collocation): Racks equipments at the Internet Access Point
ISP 2.0: Access to Servers at the Internet
ISP 1.0: Access to the Internet
Traditional ICT

Fig. 10.4 The latest evolution of applications and hosting

Table 10.2 Summary of accomplishments to date

Old datacenter	Transformed datacenter
Difficult to repurpose resources quickly for changing engineering requirements	Dynamic computing infrastructure is standardized, scalable, portable, and highly available
Complex process with all requests whether VMs or physical servers—manual work required	*Self-Service*: Intuitive, easy to use, and self-provisions resources
All aspects of the datacenter are manual with little configuration management	*Minimal or Self-managed*: Automation, self-scheduling resources, and configuration management
Repurposing servers to new tasks is time consuming or not possible	*High Utilization*: Quickly and easily repurposes an instance of a new environment in production or test
Server Virtualization is stuck at 30% due to virtual server management overhead	Highly dynamic utilization capability server virtualization exceeds 75% and servers easily share across production, DR, and Test environments

4. *Part-Time, Cloud-Computing Vendors*: Some large enterprises take a straight approach to implement certain degrees of a Private Cloud for minimizing impact to their existing IT operations. The built-up experiences and the new assets can eventually lead to different business opportunities, this is evidenced by Amazon's transformation from an e-retailer to a Cloud SP. Their operation experiences in thin margins, coupled with advanced IT technology, were proven to be a very successful model of Cloud transformation. Similarly, any enterprises that go through the transformation can become candidates for offering IT-based services. This is because enterprises typically maintain huge IT infrastructures to meet their potentially highest operational limitations, often with excess capacity. During less-busy periods, enterprises can release these extra resources to external customers or suppliers, thus making the enterprise a part-time or full-time Cloud-service vendor. If this is a desirable option for an enterprise, the supporting business process and model must be considered during the planning and design phases.

10.2.3 Establish a Technical Innovation Culture

Enterprise ICT must have an outstanding strategy, vision, and long-term commitment to stay competitive or strike for success in new business frontiers. Before the transformation project starts, enterprises should prepare the staff who will be involved with the new wave of technology innovation. This new wave includes accelerating rates of change that will impact their traditional EA, more divisive interests from the broader stakeholder community, more complex interdependencies between business partners, and more vendors and providers with different options. These trends are continuing to challenge the already dynamic business operations in today's enterprises. Although impossible to clearly identify future changes, the following samples intend to capture some already established changes:

1. Virtualization helps make efficient use of idle Cloud resources. It can remove local constraints on energy costs and capacities, space requirements for IT infrastructure, and up-front costs.
2. Through Cloud-based operations, it is now feasible for IT departments to help enterprises sustain and grow their business with very thin margins by tightly managing IT costs, while supporting highly reliable and efficient technology and information services.
3. High margin, big-ticket software systems continue to have their market shares, but they will be challenged by new generations' application development environments, where open-source, standard-based software platforms are becoming the mainstream practice. Complex software system architectures are now divided into functional compartments.
4. Since Cloud services can be introduced to their markets in small increments, enterprises can first build their killer applications, then take the common layers of those applications and expose them as utilities for other services to take advantage. This technique can greatly empower the exiting business process, allowing enterprises to mobilize or influence a commercially viable ecosystem.
5. Many open source projects and standard forums will thrive in the area of service management, with new procedures and methods providing more accurate, automated, and simplified solutions to improve the current system administration, configuration, and management.

Revolutions or evolutions of the above changes will make business and technology integrations much more seamless. It will enable enterprises to go well beyond that of simple shared data applications. It is logical to see integrations take place in the lower levels of IT first. After Infrastructure integration, the enterprise can then help create new and unique IaaS and even PaaS offerings to the market.

10.3 The New Form of Software and Service

Enterprise software for multiple or single tenant applications expands to involve external business operations, whether through the Cloud provider's value-chain relationship or Cloud-based communities. The flexibility of SaaS expedites technology maturity as vendors see broadened adaptations and are willing to invest in reliability for their services. From an enterprise perspective, the risk of data portability and corporate business sensitive data (e.g., client profiles, employee records) being locked into third party servers require enterprises to strategize their transformation carefully.

10.3.1 End Users' Expectations

When dealing with SaaS, user demands and emerging interactive technologies deeply impact the principles of enterprise services that aim to offer access and leverage data across their business. Through Web technology, browsers have changed from a simple exchange and share-point replacement to a business front-end for enterprise applications with the following refreshed feature requirements:

- Customer service and customer experience continue to be the key determinants of enterprises' reputation, brand management, and customer loyalty.
- Desktop applications are in the form of Web services residing in hybrid online or offline applications, as a part of the Cloud. Web browsers represent the only user-facing function of the desktop software required to access the Cloud.
- Personal computing devices (e.g., PC, PDA, Smart phone) become slimmer, agile gateways to the Cloud. The client-server computing paradigm will return to business applications.
- New applications are based on completely interactive online scenarios. Commerce, community, and connection are merged together for the end consumer through the Cloud, very similar to the online gaming model today.
- Some services are free-to-play, encouraging clients to stay engaged. The providers can then tap into these service communities via virtual marketing programs. A common wallet system is available for purchasing services or products via micro-transactions.

While designing enterprise applications such as a CRM, the solution architects and software architects must ensure the system users fully appreciate the functional features supported by the software as well as the Cloud capability from the corresponding IaaS and PaaS if applicable.

10.3.2 Expanding Service Categories

In a simple cost-based view, Cloud technology is an effective way to help enterprises pay less for hardware and software. This, however, should not be the only business benefit enterprises envision.

When enterprises' business processes directly engage with their stockholder communities or value-chain partners, they actually reach out to unlimited channels of new opportunities. Such engagements not only provide new and diversified portals for enterprises to market their products, but also allow enterprises to bundle their products or services with commodity services from other providers. The direct benefit of such an arrangement is that enterprises can increase their footprints on the existing market as well as offer an opportunity to explore new territories without having to invest their own efforts.

For instance, many existing SaaS vendors are partnering with professional services firms to provide expertise that makes their applications more useful and attractive. The vendors can thus continue to focus on the development effort and leave the customer support and solution customizations to other providers, while keeping high customer satisfaction. With help from artificial intelligence and automation technologies, enterprises can further streamline their integration points with other community developers to introduce more features that can either tailor or even amplify the values of the enterprises' products.

10.3.3 More Destiny Sharing Interactions

As mentioned earlier, Web-based and customer-enabling applications are changing the relationship between enterprises and customers. Enterprise executives should remain abreast of Cloud features that can help them build up high customer loyalty by closely engaging with current customers and the associated value-chain participants. For instance, social networking creates a whole new way for software game publishers to monetize their online sales. As shown in an extremely high growth rate, community-based marketing and sales is proven to be one of the strongest mechanisms used in gaming to promote adoption and commerce.

From the enterprises' perspective, their SaaS solutions can engage their partners and customers in cooperative processes of product and service improvements in real-time, rather than developing inward-looking systems and finding out about market feedback many months after the service deployment. The challenges we discussed in Chap. 4 illustrated a progression of changes that most Cloud-based enterprises have already encountered, whether they are just starting up or are well established. Two major barriers that can prevent enterprises from establishing fully integrated supply-chains or value-chain networks with the customer community are the concerns of open interfaces and security. Enterprises that need to pursue implementation and management of a Cloud service architecture with the current maturity level of products will have to re-architect current platforms to leverage Cloud technology, as well as formalize the way that policy is used to manage security and collaboration within and across service boundaries, as discussed in Chaps. 6 and 7.

10.3.4 Evolving Web Applications

It is expected that future Web technology will be even more significant in its potential to create change and opportunities for the software/application industry. These changes may impact the technology itself, the usage of the technology, and/or the economics of how a Web system or feature is sold. The following list summarizes some recent developments:

- The next Web applications will be more open and collaborative than any previous technology and will serve as a critical force in future Cloud applications. Current usability needs some evolutionary changes to satisfy the new capability.
- The new Web uses more parallel processing with multiple cores per socket and threads to improve service performance. This is evidenced by new generation browsers that have built-in threading. This feature will reply on faster broadband networks.
- Virtualization enables users to tap into software and services stored in datacenters rather than on their own computers. Users will be able to experience the Web on a phone, or move from device to device, instead of being limited to a PC.
- It offers users a richer application tier with far more logic via SaaS. Providers can develop and deploy their services much more quickly and cost effectively. Cloud services are aggregated from the collections of providers with options for the end users to contribute new features. Recent content distribution and media hosting features from providers will encourage the speed of multimedia applications development.
- Current e-business models will be revisited by the commercial industry as SaaS has changed the traditional development, marketing, and sales approaches. For instance, the content delivery structure, customer profile, roles of SCs (e.g., provider, end user), and pay methods will be more dynamic and portable on the Web.

10.3.5 Integrating Enterprise SaaS with Cloud Services

Subscribing to a SaaS application means housing business data outside the controlled local network and within the Cloud infrastructure. An integration architecture specifies how to transform enterprises to bring this outside data into the logical enterprise infrastructure, so that internal and external infrastructure components can interoperate with one another to access needed data. In most cases, implementing a SaaS application involves transferring data from one or more existing applications or data repositories local to an enterprise into a transformed system that combines internal and external infrastructure components. A composition architecture makes composite applications possible. A composite application is where business functions and information can be integrated effectively for end users. Many vendors provide API that expose the applications data and functionality to developers for use in creating composite applications. Presenting information as a unified whole, instead of as isolated streams of data, carries benefits for users. It enables them to see relationships between data from different sources and apply their own "domain intelligence," i.e., their own preexisting knowledge of how the business and its processes work, to make informed decisions. The business benefits of a well-designed composite application include reduced redundant data entry, improved human collaboration, heightened awareness of outstanding tasks and their statuses, and improved visibility of interrelated business information. In a service-centric IT department, applications and other resources become ingredients that can be combined together to create task-focused composite applications. Creating a composite application involves integrating different applications, protocols, and technologies that were not necessarily designed to communicate with one another. Providers of SaaS applications organize data in architectures that enable either multi-tenancy or isolation of software. Multi-tenancy is a software architecture in which a single instance of the software runs on a SaaS vendor's servers, thus serving multiple client organizations (tenants). By contrast, complete isolation refers to architectures where separate software instances or hardware systems are set up for different client organizations.

10.4 Platform Integrations and Collaborations

Evolved software development platforms, due to the introduction of PaaS, will result in new philosophies of software architecture, deployment, and operations. PaaS focuses on the middleware adaptation for enterprise application developments and the impacts of Cloud application frameworks. As more scripting languages and Cloud-based APIs are pushed into platforms on Clouds, developers can benefit from such dynamic development environments and be more influential to future SaaS revolutions.

For enterprises that consider using PaaS, one essential factor for platform selection is the potential implication of vendor lock-in. This is when a claimed open-source or standard-based platform does not guarantee enterprises that their developed applications will be portable to other vendors with similar claims. Therefore, the availability of porting tools for future migration to other vendors' solutions should be one of the key selection criteria for the overall transformation plan.

10.4.1 New Applications Development Functions

As PaaS will interact with many Web technologies and other Cloud-driven business frameworks, the software built on PaaS will inherit many of the other entities' features and become more resilient, adaptive, and reliable. PaaS not only provides design, development, testing, installation, and deployment tools, it also supports collaboration among application stockholders, co-development communities, value-chain partners, and even users. As a result, it will provide significant savings for enterprises in variable and on-going costs related to software upgrades, support, maintenance, and management due to much less effort needed to install, maintain, and repair the software. Software image management now becomes more dynamic and flexible. It is operated at the "file level" and no longer requires starting an image for a software upgrade. Figure 10.5 illustrates this innovative change. Some key features specifically from Cloud technology include:

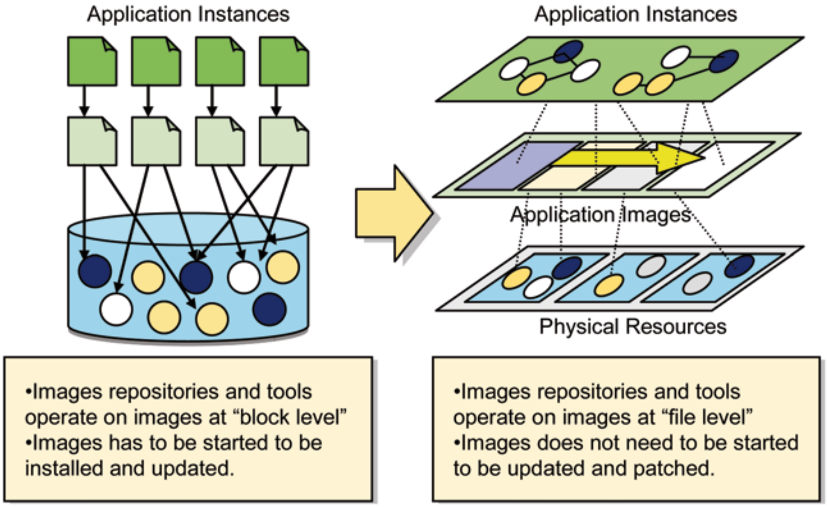

Application Instances
Application Instances
Application Images
Physical Resources
•Images repositories and tools operate on images at "block level"
•Images has to be started to be installed and updated.
•Images repositories and tools operate on images at "file level"
•Images does not need to be started to be updated and patched.

Fig. 10.5 Software image management innovation

- The enabler for applications to grow and shrink based on SLAs in order to support the utility model such as pay-per-use.
- Considerable savings by leveraging shared service models for custom applications and related services.
- More open source software and products continue to expand in the PaaS market and make software development faster and easier.
- The Cloud developer community grows even faster than the open-source community does. These communities will be complemented by standards forums and common-interest working groups to jointly define the general discipline of building applications on the Cloud. The population of Cloud developers will grow faster than before.
- Traditional application servers will give away their roles to the next generation of applications that are hardware independent, more portable, and easy to access.

10.4.2 Software Development Standards

The principle of PaaS will benefit enterprise developers with high-level agility based on the ways they can rapidly iterate over the write-build-test cycle; thus giving datacenter services the ability to be only a credit card away. However, without the appropriate level of details in standardized specifications, the claimed best practices and even basic interoperability will be challenged. The pace of Cloud technology innovation is so rapid that the emergence of truly open Cloud standards are slow in comparison. An immediate negative result could be enterprises' hesitation in participating, thus slowing down their adoption process. Fortunately, we have seen a lot of promising progress in defining low-level specifications on Cloud infrastructure as well as some market-focused (e.g., TM Forum for telecommunications) standards, as discussed in Chap. 3.

Nevertheless, to achieve a fully functional Cloud service framework, the development of standards and interoperability between the varying levels of Clouds for vertical integration is inevitable. It is also crucial for the industry to specify the guidance and protocols for horizontal integration, examples include cross-vendor portability and interoperability.

When needed specifications are not yet fully standardized and enterprises must move forward in developing their applications without standard protocols, they should be careful of false claims from some traditional vendors.

10.4.3 New Software Packaging Focus

Previously, the focus of system deployment has been on the server, not the application. This means the design and implementation of enterprise software systems are bound by hardware and resource restrictions, as specified in solution or system architectures. Thus, changes to software must first be compliant with the hardware architecture. When an enterprise acquires software products from a vendor, the IT department is billed based on the size and type of servers (e.g., number of processes, CPUs) that host that software. After that, the entire package is managed and monitored largely from the perspective of the hosting servers. This model is no longer true in highly portable Cloud applications.

With virtualization technology, some OS functionalities are wrapped inside application containers as part of the deliverables; some are integrated onto the hardware circuitry and thus detach from the software platform (becomes firmware platform). Software functions are now treated as elements of a bundled compartment, thus the concept of packaging can be now defined in application terms. Furthermore, package monitoring and management are now conducted through software services and interfaces, making the packaging model purely application-oriented instead of server-oriented. This change enhances the portability of software packaging and delivery. Any new packages can be moved around within datacenters, or even among them. The ease of moving software around reduces the complexity of the old fashion packaging and monitoring logic, which directly streamlined the software market cycles and accelerated software releases.

10.4.4 New Relationship with Hardware Resources

The agile programming and project management methods from PaaS make sense because Cloud-based applications are detached from their hardware infrastructure.

The new service-oriented approach to software and systems architecture allows developers to focus on building their logic beyond the boundaries of the computing community directly into user and application communities. When software platforms are no longer divided by servers, storage, and networks, the software focus can be much more flexible in many aspects of enterprise services, e.g., simplified processes, less resource dependent, etc. The changing relationship will result in new organizational structures within the IT department.

Furthermore, the QoS in server-centric platforms is relatively more predictable and measureable in a static deployment model. When dealing with application-focus software in the Cloud, enterprises need special expertise to tailor their application performance for different runtime options with varying workload intensities and system configurations. For resource management, the enterprise should focus on administration tasks of virtualized resources and their run-time implications for PaaS applications to meet high-level performance objectives.

10.4.5 Integrating Enterprise and Cloud PaaS

PaaS generally refers to internet-based software delivery platforms for which third-party ISVs or custom application developers can create multi-tenant, Web-based applications that are hosted on the PaaS provider's infrastructure and offered as a service to customers. The main premise of PaaS is providing software developers and vendors with an integrated environment for development, hosting, delivery, collaboration, and support for their on-demand software applications. Like other software platforms, PaaS aims to be a foundation for a broad, interdependent ecosystem of users and businesses. It can support tasks from code editing to deployment, runtime, and management. The current PaaS ecosystem shows a wide range of different levels of service. Some platforms offer little more than a set of APIs on top of an elastic infrastructure, while others offer fully functional Web-based IDEs or fourth-generation programming language environments allowing an easy creation of metadata-level mash-ups. Additionally, a PaaS could support built-in backend functionalities of applications like billing, metering, advertising, etc.

10.5 Infrastructure Transformations

The flexibility feature of virtualization technologies has changed the face of infrastructure offerings. Physical resources are no longer required to collocate with enterprises, whether for mission-critical applications or not. As more Cloud infrastructures scatter throughout the world, enterprises in a relevant value-chain have to trade-off their completive edges with the new lean IT infrastructure. The decision is not whether they should go for the Cloud or not, rather, it is deciding how much of their resources should be distributed between the Public and Private Clouds. If a hybrid scenario is preferable, what level of resources should be retained in-house?

10.5.1 Customizable Service Resources

Using virtualization technology, Cloud infrastructure offers a dynamic runtime environment, allowing enterprises to move their resources around or even out of their datacenters. It also provides great potential for the enterprises' IT departments to customize their resources. However, without common rules or a standard framework to implement best practices, a cross-vendor solution will be difficult. It is especially important for enterprises to establish a common theme to address cross-domain security and cross-company service management. Both of these challenges have been slowing the adaptation of running mission critical applications on the Public Cloud or Hybrid Cloud. On the other hand, many complex custom applications that provide competitive advantages for enterprises are transitioning to the Private Cloud to take advantage of the Cloud's rigidity. As seen in Chap. 2, many large enterprises and government organizations have successfully installed their Private, and sometimes standalone, IaaS ready for their users.

To achieve profitable business goals, enterprises require their IT operators to provide tangible and measurable performance from their IaaS. In a value chain business model, cross-provider SLA negotiation relays upon common QoS agreements in the form of performance metrics and its context in business operations, such as the charging structure and billing mechanism. SPs on both ends must be able to honor the agreement with a unified framework to ensure the enterprise-caliber SLAs are sensible for their infrastructure clients. As the IaaS clients can directly access the IT resources from their personal devices, a multiple layer SLA should be in place to accommodate financial treatments in different segments of the supply chain.

10.5.2 Improved Infrastructure

Cloud technologies offer integrated IT infrastructure for optimal resource utilization and enable enterprises to offer expanded, differentiated, on-demand services with increased value and longevity. They offer the following improved features to existing IT hardware services:

- As seen in Chaps. 2 and 3, IaaS helps enterprises move ICT resource attributes from the physical level to the logical level, thus enabling great management flexibility.
- The portability offered by the Cloud helps IT managers achieve significant savings in fixed IT infrastructure costs.

- With the option to outsource to other providers, enterprises can acquire IaaS to obtain significant savings in variable and on-going costs related to upgrades, support, maintenance, monitoring, and administration.
- When there is new hardware equipment in interest, enterprises have the flexibility to choose a hardware services provider without having to deal with the hardware and infrastructure lock-in issue.
- Since the infrastructure is outsourced to external providers, the enterprise IT department can pass-on the liability of previous SLAs to the providers. For their premium customers, enterprises can have the option to offer more stringent penalty clauses and risk-reward models, adhering to pre-negotiated SLA or OLA from IaaS providers. These SLA or OLA may include QoS metrics such as performance, scalability, and availability.
- Enterprises can now offer more complex product or service profiles from their existing LoB by leveraging their providers' different levels of service pricing models, such as pay-per-use, pay-for-capability, and pay-for storage.

10.5.3 Customer Portal and Rapid Provisioning

Combining the ability of online access and direct acquisition from virtual datacenters, IaaS users can manage and monitor the enterprise infrastructure offerings via tools such as network snapshots, fault alarms, performance graphs, and VM controls. More experienced users can even use tools from the providers to conduct rapid provisioning of computing resources and software packages. Through these technologies, service customers have the freedom to customize their purchased infrastructures to improve Cloud use and tailor its performance to make it the most suitable for their business interest.

For some IT professionals, there may be a need to automate resource management in their managed domain. In this case, they can develop programmatic controls through the IaaS provider's API, in addition to the GUI, to build a management application that is a part of the larger management services. These functions can build up experiences and expertise so individuals or enterprises can focus on the integration aspect of the services and leave the adaptation of datacenter technologies of Cloud environments to the providers.

10.5.4 Integrating Enterprise and Cloud IaaS

The fundamental building block of an infrastructure is a workload. Workloads can be thought of as the amount of work that a single server or application container can provide given the amount of resources allocated to it. IaaS providers publish APIs that allow enterprise administrators to build their own solutions on top of the IaaS services. Usually, the APIs support a programming style based on the principles of REST or SOAP. Enterprises can use the APIs to perform such operations as browsing, where the enterprises discover the contents of a container that has an application or a virtual media image, and provisioning, where the enterprises can populate a container with entities such as virtual media ISO images. The OVF is an open, portable, efficient and extensible format for the packaging and distribution of software to be run in VMs. OVF was developed by the DMTF, a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. A virtual application or VM is typically made up of one or more virtual disk files that contain the OS and applications that run on the VM, and a configuration file containing metadata that describe how the V is configured and deployed. An OVF package includes these components, as well as optional certificate and manifest files. [C5::12,13,14]

10.6 Cloud Management and Operational Framework

As application, platform, and infrastructure environments become increasingly dynamic and virtualized, the "virtual datacenter" will emerge as the new enterprise service platform. In Chap. 1, we saw the concept of the "orchestration layer" that sits between Cloud operators and the various Cloud services they manage. This layer assists operators in determining the best Cloud service for a particular job based on lowest cost, highest performance, and other requirements. Such an approach makes it possible for Cloud operators to maintain a common method while optimizing service usage. Additionally, enterprise service assurance requires information governance as a central tenet of governance, risk, and compliance planning. Effective information governance, in turn, also depends on a better orchestration approach.

10.6.1 Management Paradigms

Figure 10.6 shows how service customers, designers, and providers can access the process and service, application, and information sources from the multiple-layer Cloud architecture. Successful and seamless service transactions rely on effective management platforms, as discussed in Chaps. 7–9. As first seen in Chap. 1, there are two general approaches for an enterprise to harmonize its management policy and procedures in order to assure its ICT applications can deliver the designed values across the designated business boundaries and different Cloud service layers. There is the SoS approach and the FoS approach.

- The *SoS Management Approach* relies on a uniform management abstraction layer that enforces the managed resources to conform to common framework interfaces or well defined management interfaces. This allows the management layer to apply global management policies down to the contents of resource containers. The advantages of such a design include the efficiency of management

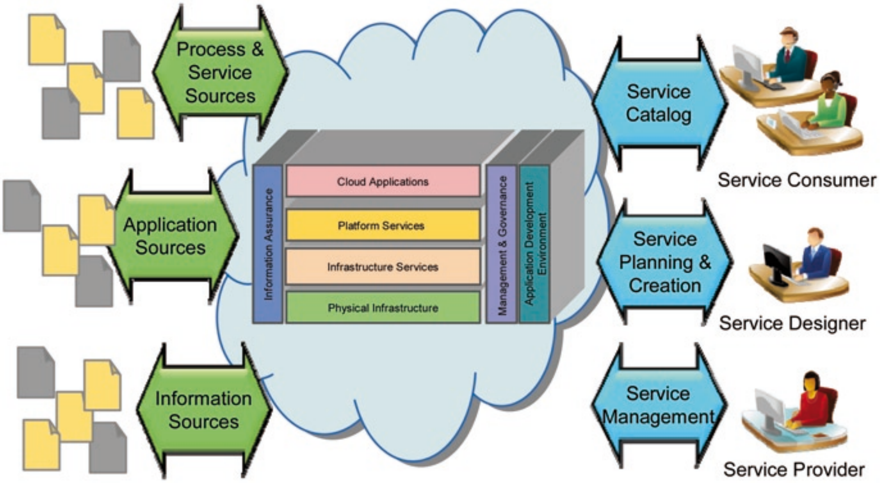


Fig. 10.6 The managed Cloud

over heterogeneous Cloud resources with a uniform rule set and the effectiveness of policy execution because of standardized frameworks. Its disadvantages include lacking standards for the Clouse-based policy framework and potential implementation efforts required for the existing applications.

- The *FoS Management Approach* is in contrast to the SoS approach, which is more virtualization-based. In this approach, the Cloud resources are not directly manageable by the elastic computing infrastructure. For instance, some vendors manage their resources with a set of standard management metrics. Different management systems or OSS correlate business resources to system resources through systems metrics for managed service containers. This makes cross-vendor service SA possible. Because the content of each container is implementation–specific, systems need to use the technique described in Chap. 6 to facilitate service federations. This approach has less dependency on providers' individual implementations, but needs more complex interpretation logic at each node.

10.6.2 Service Management Automation

In Cloud communities, the new management platforms must have the ability to apply configuration, SLA, and policy across thousands of transient servers, fluid storage pools, and dynamically allocated networking resources. At the application level, there are variable workloads and transactions that often need to be quickly exchanged within the enterprise datacenters or among different supply-chain partners. Furthermore, the platform must support IT admin to perform other service management such as provisioning, failure recovery, scaling, trouble ticket management, and so on. Without the ability to automate such dynamics in the management domain, enterprises will fall short in their strategy to achieve true "Cloud economics."

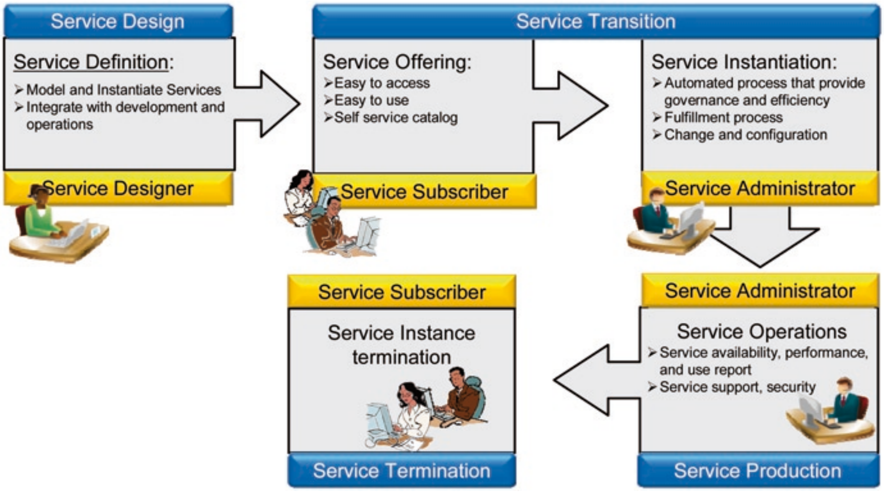


Fig. 10.7 Lifecycle of a Cloud service

When dealing with the scalability of Clouds from the management perspective, the structure of the Cloud must be addressed. Here, an autonomic management framework requires the corresponding service management to be scalable and autonomic. For instance, if two Clouds are autonomic and support the same management interfaces, they can be composited into a larger Cloud while preserving their original identities. The concept of Cloud SDF gives enterprises a solution to consistently and clearly determine which collaboration points can be provided for managing a joint Cloud to support their customers, developers, and others. When applying this to partner relationship management, for instance, the Cloud SDF provides a mission-critical foundation, allowing enterprises to scale partnering efforts with a manageable degree of staff and infrastructure investments. Figure 10.7 depicts the high-level lifecycle of a Cloud service, the actual implementation varies depending upon the nature of service types and different customer needs.

Policy management plays an important role in automating service management. As seen in Chap. 6, externalization of policy management goes a long way toward making it possible to composite Clouds and manage policy compliance. In practice, policy extension points in the orchestration layer can enable Cloud resources to become more manageable, thus allowing them to participate in Cloud management more seamlessly.

10.6.3 Changing Process Management

Through the transformation practice of Cloud management, enterprises can improve their understanding of process and governance risk associated with the cost and inconsistency of on-premise IT. This helps them migrate their ICT strategy toward auditable and highly professional practices of the Cloud-service environment. Following the discussion from the last section, enterprises automate their processes and as much of their full lifecycle as possible in order to be sustainable and profitable through management efficiency. The result of automation is that enterprise customers are granted a potential ability to place orders and get fulfillment and basic support without human intervention. Moreover, all billing and usage accounting is processed completely automatically. All these are taking place without needing costly support calls or input from SP staff.

As seen in Chap. 2, larger enterprises may initially prioritize internal Private Cloud deployments in order to recognize the benefits of Cloud technology without compromising security or compliance concerns. In such an event, their systems administration function to manage this private operations center will remain needed. Although there could be a tactical arrangement before moving to a more virtual environment, these administrators have now extended their responsibilities to monitor the overall performance of applications running on the Cloud, as well as monitor the performance of the enterprises themselves.

10.6.4 Integrating Enterprise and Cloud Governance

Many Public Cloud services provide a deep stack of on-demand services, spanning the application, software platform, integration middleware, and hardware layers. By proliferating services deep into the stack, beyond the capabilities of today's SOA governance tools, Cloud environments make unified planning, design, provisioning, monitoring and control of all services difficult. One key area where Cloud governance differs from traditional SOA is in its focus on life-cycle governance of VMs. To facilitate automated provisioning of deep application and integration stacks on VMs, Cloud management environments can offer prepackaged server templates. These templates embed prepackaged policy definitions that govern important life-cycle service VM governance functions, including deployment, setup, booting, monitoring, control, optimization and scaling of VMs on one or more Public or Private Clouds. [C5::17]

Cloud governance encompasses the periodic need to decommission and throw away old VM instances, and launch new ones in their place. The problem of unchecked proliferation of VM instances across public and private virtualization infrastructures is sometimes known as VM sprawl. A growing range of commercial management tools provide the ability to control VM sprawl across disparate hypervisors. Preventing VM sprawl is referred to as instance management, and is a feature that is lacking from traditional SOA governance tools. The mass scale adoption of server virtualization in datacenters and Public or Private Cloud environments creates the need for high-speed, low latency and resilient Cloud networking. Building a combination of virtual and physical Cloud networks that are commensurate with virtual and physical servers demands an architectural approach to infrastructure build-out. The performance, latency and elasticity must be considered as well as the management of the networking infrastructure. Once architected and deployed, the solution can offer the services over a common shared infrastructure. [C5::17,19]

TM Forum's SDF supports contextual information of a service in relation to the business and operation environment, through the definition of SDF *Service Management Interface* (SMI), SDF service lifecycle metadata (or schema) associated with the service, and SDF support services. SDF is a framework and as such its role is to offer the appropriate artifacts to support the key operational processes and service management activities.

A SLA serves as the foundation for the expected level of service between a consumer or an enterprise and a Cloud services provider. QoS attributes, such as response time and throughput, usually form a part of an SLA. Since the QoS attributes change frequently over time and are based on traffic conditions, enterprises need to monitor these attributes. To monitor the QoS attributes, enterprises can demand that monitoring data, such as raw transaction count, be exposed by a SP without further refinement. Alternatively, enterprises can request that collected monitoring data be put into a meaningful context, such as statistical measures of average or standard deviation. This request requires that the Cloud SP create processes to collect data from several different sources and apply suitable algorithms for calculating meaningful results. A second alternative is for enterprises to request certain customized data be collected. Yet another alternative is for enterprises to dictate the way monitoring data is collected. [C5::21]

10.6.5 Integrating Enterprise and Cloud Quality Assurance

Achieving quality and performance targets for products or services may require an enterprise to establish and manage a number of SLAs. The complexity of global services brings together a myriad of services, suppliers, and technologies, all with potentially different performance requirements. Thus, the goal of enterprise SLAs is to improve the CE of the service or product for the enterprise clients, whether they are internal or external to the organization. CE is a collective term to form a measure of the quality of a service or product and includes all aspects of service: its performance, level of customer satisfaction in the total experience, pre and post sales, and the delivery of its products and services. Determining the CE provides a discriminator between various types of service or product that an enterprise provides, and leads to opportunities to balance the level of quality offered against price and customer expectation.

There has been much research in the development of standard equations that provide quality measurements from performance-related data. These equations can be used to model a network before it is deployed, assign values for an SLA contract, and perform analysis of data to predict the performance enhancement or degradation due to changes in the service such as the addition of a route controller or a move from narrowband to broadband connections. These equations can be used to determine thresholds and sensitivity analysis of PKI parameters for SLA monitoring and reporting.

A SQM framework needs to define a holistic framework for measuring and effectively managing service quality; key service quality metrics at each point along the service delivery network; service quality issues and the necessary accounting and rebating information, usage information, and problem resolution information; management capabilities to support each step in the service delivery network; and appropriate interfaces and API's to enable the interchange of such information electronically between the various providers in a service value chain.

Probe systems are a fundamental tool for network operators and SPs to monitor and manage the QoS. Probes can be placed at any point in the network, so they provide a greater flexibility than the systems based on network elements or other data sources. Active probes inject traffic in the network and send requests to services servers as an end user does. They are usually used to provide an end-to-end view. On the other hand, passive probes sniff packets from different services. They can only provide a view of a part of the network at several protocol levels.

Conformance with an SLA is ensured by using instruments in the system to provide appropriate KPI and KQI measures at required sample rates. It is important in the design process to ensure that the measurement process itself does not create or worsen system conditions by adding further load to the system, e.g., by using additional processing power or adding additional management traffic overhead. If a KQI for a first service is determined by correlating KPI or KQI data from a second service, the information from the second service may be required in real time so that true measurements can be made for proactive management of the first service. This allows for fault prevention rather than aggregate or stored information. Thus, an SLA should be monitored continually at a rate appropriate to the requirement for a service to assure that corrective actions can be taken and collated to form management reports.

Enterprises work towards high-level objectives that an SLA or collection of SLAs support. Business processes are judged against these high-level objectives. Conflicts may require modification to application objectives or requirements or, in some cases, changes to the enterprise objectives themselves.

The exact form of an SLA depends on the two entities that are entering into the agreement or contract. In particular, the form of the SLA will be different, especially in the area of penalties, among the cases when the SLA is between an enterprise and an external party, such as a Cloud provider, when the SLA is between internal enterprise parties, and when the SLA is between the enterprise and its customers. The SLA is a mutual agreement between two parties with expectations from both sides defined and defines the course of action to be taken when deviations from these expectations occur. An SLA is, in general, a legal contract between the parties, especially for SLAs between an enterprise and external parties, such as Cloud providers. It is therefore important to take legal advice as to the exact form of the contract and the language used. If the SLA is to span international boundaries, such as may occur in a Cloud environment, enterprises need legal advice that has an understanding of the differences in contract law, environmental, employment, and any relevant regulatory environment in the relevant countries. Even internal SLAs, where the SLA spans international boundaries, may have to take these issues into consideration.

Multiservice platforms present unique demands on event management systems because of the volume of traffic they process and the volume of alarms they can generate. An Event Manager component within a managed service can support event correlation and filtering to reduce the potential flood of events. Event filtering and correlation policies define the filtering and correlation performed. A correlation policy can be defined to link all associated events to a given root event, provided they arrive within the specified time interval. As a result, only the root event is forwarded, thus reducing the alarm overload on the management system.

10.7 Cloud Security and Information Assurance

Due to the fact that little critical information and few critical applications are shared across Public Clouds, many security concerns mentioned in the early chapters have not yet been put in the frontline of many Cloud SPs. However, as adoption expands and risks increase, security issues will soon get pushed down to every layer of virtualized services.

Based on the new Cloud paradigm, sensitive information may no longer reside on dedicated hardware resources. The protection technologies, processes, and procedures for enterprises' most sensitive information in the rapidly-evolving world of shared computing resources will continue to challenge enterprises, Cloud Integrators/Aggregators, Cloud SPs, and Cloud product vendors. In this section, we will list the final guidance for awareness and reference.

10.7.1 New Applications of Information Assurance

Management of security for Cloud Computing requires RBAC architectures in the Cloud that can integrate well with customer systems. Security management comprises security for the Cloud network itself and security for customer data and infrastructure hosted in the Cloud. Security for the Cloud network itself requires secure APIs so that users of the Cloud are assured of the security of the services the Cloud offers. Security for data and infrastructure hosted in the Cloud requires that VMs for different customers operate autonomously so that the hardware and software resources used by one VM are securely protected from other VMs. RBAC needs enhancements for open and decentralized multi-centric systems, such as when transforming an enterprise into a Cloud Computing environment, where the user population is dynamic and the identity of all users is not known in advance. [C9::5,6]

Federated identity management aims to unify, share, and link digital identities of users among different security domains. A FIA is a group of organizations that have built trust relationships among each other in order to exchange digital identity information in a safe way, preserving the integrity and confidentiality of the user personal information. The FIA involves IdPs and SPs in a structure of trust by means of secured communication channels and business agreements. IdPs manage the identity information of users and do the authentication processes in order to validate their identities. SPs provide one or more services to users within a federation. In transforming an enterprise into a Cloud environment, the tokens that IdPs issue contain attributes that the enterprise network is allowed to request from the SPs. This enables enterprises to provide their users services from SPs that the Cloud itself provides or that are provided by other enterprises in the Cloud. There are some building blocks, such as cache load measurements, and coarse-grained attacks, such as measuring activity burst timing, that enable practical side-channel attacks when transforming enterprises into Cloud-computing environments. One may focus defenses against cross-VM attacks on preventing the side channel vulnerabilities themselves. This might be accomplished via blinding techniques to minimize the information that can be leaked (e.g., cache wiping, random delay insertion, adjusting each machine's perception of time, etc.). Countermeasures for covert channels (which appear to be particularly conducive to attacks) are extensively discussed in the literature. These countermeasures suffer from two drawbacks. First, they are typically either impractical, e.g., high overhead or nonstandard hardware, application-specific, or insufficient for fully mitigating the risk. Second, these solutions ultimately require being confident that all possible side channels have been anticipated and disabled—itself a tall order, especially in light of the deluge of side channels observed in recent years. Thus, at the current state of the art, for unconditional security against cross-VM attacks one must resort to avoiding co-residence. [C9::18,27]

Until recently, work on IDSs focused on single-system stand-alone facilities. Cloud providers, however, need to defend a distributed collection of enterprises. Although it is possible to mount a defense by using stand-alone IDSs on each host, a more effective defense can be achieved by coordination and cooperation among IDSs across the network. There are major issues in the design of a distributed IDS. A distributed IDS may need to deal with different audit record formats. In a Cloud environment, different enterprises employ different native audit collection systems and, if using intrusion detection, may employ different formats for security-related audit records. In addition, one or more nodes in the network serve as collection and analysis points for the data from the systems on the network. Thus, either raw audit data or summary data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality of these data. In addition, either a centralized or decentralized architecture can be used. With a centralized architecture, there is a single central point of collection and analysis of all audit data. This eases the task of correlating incoming reports but creates a potential bottleneck and single point of failure. With a decentralized architecture, there is more than one analysis centers. These centers must coordinate their activities and exchange information. The main idea behind multi-sensor data fusion in distributed IDSs, such as the ones for Cloud infrastructure, is that the combination of data from multiple sensors enhances the quality of the resulting information. Data fusion enables the combination of, and intelligent reasoning with, the output of different types of IDSs. By making inferences from the combined data, a multiple level-of-abstraction situation description emerges. [C9::29,31]

CSA published a report that listed insecure interfaces and APIs as a top threat to Cloud Computing. Cloud Computing providers expose a set of software interfaces or APIs that customers use to manage and interact with Cloud services. Provisioning, management, orchestration, and monitoring are all performed using these interfaces. The security and availability of general Cloud services is dependent upon the security of these basic APIs. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy. Furthermore, organizations and third parties often build upon these interfaces to offer value-added services to their customers. This introduces the complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency. To ensure security for APIs, Cloud users need to sign API calls to launch and terminate instances, change firewall parameters, or perform other functions with the users' private keys or secret keys.

10.7.2 Security in Different Service Layers

OS, command interpreters, and application environments provide a way for software instructions to be executed when transforming an enterprise into a Cloud environment. The concept of execution containers is an architectural abstraction used to describe virtual compute resources. Sun Microsystems defines a secure execution container as a special class of secure components that provide a safe environment within which applications, jobs, or services can be run. Execution containers are frequently used within the context of OS: OS instances (real or virtual) can themselves be run on physical, logical, or virtual hardware platforms. Execution containers can also be environments in which applications, services, or other components are executed, such as J2EE Containers. [C9:40]

IaaS providers allow their customers to have access to all VMs hosted by the provider. The providers manage one or more clusters whose nodes run a hypervisor, i.e., a VM to host customers' VMs. A system administrator working for the Cloud provider who has privileged control over the backend can perpetrate many attacks in order to access the memory of a customer's VM. With root privileges at each machine, the system administrator can install or execute all sorts of software to perform an attack. Furthermore, with physical access to the machine, a system administrator can perform sophisticated attacks like cold boot attacks and even tamper with the hardware. A possible implementation to address security for VM images is to build on the techniques proposed by the Trusted Computing Group. Two components can be used: a trusted VM monitor and a trusted coordinator.

Efficient instantiation of VMs across distributed resources requires middleware support for the transfer of large VM state files (e.g., memory, disks) and thus poses challenges to data management infrastructures. Nevertheless, security currently is limited to simple file permissions and network authentication protocols like Kerberos for which both user authentication and encryption of data transfers are not supported. A secure file system that separates key management from file system security can be implemented. In this implementation, file names themselves effectively contain public keys, making them self-certifying pathnames. Thus, key management occurs outside of the file system, in whatever procedure users choose to generate file names. This decouples user authentication from the file system through a modular architecture. External programs authenticate users with protocols opaque to the file system software itself. These programs communicate with the file system software through RPC interfaces.

The reduction of management costs, in both hardware and software, constitutes one of the value propositions of Cloud Computing. This cost reduction comes from sharing the knowledge of how to manage a piece of IT assets via VMIs. Nevertheless, VMI sharing unavoidably introduces security risks. A user of Cloud services risks running vulnerable or malicious images introduced into the Cloud repository by a publisher. While running a vulnerable VM lowers the overall security level of a virtual network of machines in the Cloud, running a malicious VM is similar to moving the attacker's machine directly into the network, bypassing any firewall or IDS around the network. VMI sharing provides a straightforward way of developing and propagating Trojan horses. Using a VMI as a carrier for the trojan horse makes the hacker's job easier than before, because the VMI encapsulates all software dependencies of the Trojan horse. In other words, the dependency on the victim's software stack is eliminated. Users of Cloud services also risk running illegal software, e.g., unlicensed or with expired licenses, contained in an image.

A Cloud provider risks hosting and distributing images that contain malicious or illegal content. In addition, security attributes of dormant images are not constant. If dormant VMIs are not managed, e.g., scanned periodically for worms, a virtual environment may never converge to a steady state, because worm-carrying VMIs can sporadically run, infect other machines, and disappear before they can be detected. The same idea holds for software licenses. As the number of VMIs grows, so does the risk and along with it the cost of maintenance. An image management system that addresses these security concerns can be implemented. The implementation consists of four major components that implement four features. The first feature is an access control framework that regulates the sharing of VMIs. This reduces the risk of unauthorized access to images. The second feature is an image filter that is applied to an image at publish and retrieve times to remove unwanted information in the image. Unwanted information can be information that is private to the *user*, such as *passwords*; or *malicious*, such as *malware*; or *illegal*, such as *pirated* software. Filters reduce users' risk of consuming illegal or harmful content. The third feature is a provenance tracking mechanism that tracks the derivation history of an image and the associated operations that have been performed on the image through an image repository API. Provenance tracking provides accountability and discourages the intentional introduction of malicious or illegal content, which in turn reduces a Cloud provider's risk of hosting images that contain such content. The provenance mechanism also tracks modifications to images that result from applying filters. The fourth feature is a set of repository maintenance services, such as periodic virus scanning of the entire repository, that detect and fix vulnerabilities discovered after images are published. This reduces users' risk of running and hosting malicious or illegal software and the risk of hosting them.

10.8 Final Notes

The Cloud environment is a fascinating realm that makes it easier to deploy software and increase productivity. The Cloud presents a number of new challenges in data security, privacy control, compliance, application integration, and service quality. Enterprises should take small, incremental steps towards this new environment so they can reap the benefits for applicable business situations and learn to deal with the associated risks. In general, Cloud Computing will act as an accelerator for enterprises, enabling them to innovate and compete more effectively.

Businesses and enterprises should now take steps to experiment, learn, and reap some immediate business benefits by implementing Cloud Computing in their organizations when competing in today's increasingly multi-polar marketplace.

References

1. Predicts 2010: revised expectations for IT demand, supply and oversight. Gartner. Jan 2010
2. Urquhart, J.: The three routes to Cloud computing's future. CNET News. http://news.cnet.com/8301-19413_3-10196722-240.html (2009). 16 March 2009
3. Geelan, J.: The future of Cloud Computing. Cloud Comput. J. (18 Jan 2009). http://Cloudcomputing.sys-con.com/node/771947
4. Rainge, E.: SDPs can become more strategic than the network, IDC. TM Forum. Feb 2010. http://www.tmforum.org/ArticleSDPsCanBecome/8439/home.html
5. LaMonica, M.: Study: Cloud computing to brighten future of data centers. CNET News. http://news.cnet.com/8301-10784_3-9889947-7.html (2008). 10 March 2008
6. Jackson, W.: Cloud computing's future depends on securing it, industry execs say. Government Computer News (GCN). http://gcn.com/Articles/2010/03/02/RSA-Cloud-Keynotes.aspx (2010). 02 March 2010

Authors' Bios

William Y. Chang has over twenty years of consulting, engineering, and development experience in defense, telecommunications, and financial industries. He is the author of "Network Centric Service Oriented Enterprise" (2007, Springer) and co-author of "Service Assurance for Voice over WiFi and 3G Networks" (2005, Artech House). Mr. Chang's credentials reflect a career of blending leading-edge solutions with system engineering, deep-information management, telecommunications technologies, mobile-service creations and operations, and commercial software-product development. Currently, he serves as a lead consultant for NASA Jet Propulsion Laboratory and a Network Centricity Compliance program for US DoD's enterprise services. Previously, he acted as the project lead of the first TM Forum Defense Catalyst and a lead integration consultant for the U.S. Air Force's Transformational Satellite Communications System (TSAT). He also worked as a Lead System Integrator and Technical Consultant for the US Army's Future Combat System (FCS) Division at Boeing. His affiliations include: Booz Allen Hamilton (BAH), where he is Technical Fellow in Global Defense; Science Applications International Corporation (SAIC), where he was a Principal Systems Engineer with the Tactical Systems and Solutions business unit; AcuMaestro, as their Chief Technology Officer; Bell Communications Research (now Telcordia), where he was the Principal Systems Engineer of the Next Generation Network OSS Engineering Group; Bell Laboratories, as a software-architecture consultant for the AT&T Customer Network Management solution and Federal Government Network Management system (FTS2000); and Tandem Computer Corporation, where he provided engineering consultation for a distributed-banking solution.

Hosame Abu-Amara has over twenty years of consulting, engineering, and academic experience in defense, telecommunications, and computer engineering. His background includes academic as well as industrial experience, and he has extensive experience in enterprise, terrestrial, satellite, and wireless network operations, planning, and management and handheld, computer, router, and switch management and operations. He has over 15 granted patents and more than 40 technical publications in refereed journals and conferences. Currently, Mr. Abu-Amara is an Associate with Booz Allen Hamilton, Inc., where he provides consulting services in the areas

W. Y. Chang et al., *Transforming Enterprise Cloud Services,* 417

DOI 10.1007/978-90-481-9846-7 © Springer Science+Business Media B.V. 2010

of OSS systems integration and software development, mobile networking, network management, and satellite and terrestrial network performance modeling & simulation. He has worked on DoD Transformational Satellite (TSAT), Office of the Secretary of Defense—Network Information Integration (OSD-NII), Joint Program Executive Office Joint Tactical Radio System (JPEO JTRS), and Broadband Technology Opportunities Program (BTOP) projects. Previously, he was a Distinguished Member of Technical Staff with Motorola Mobile Devices, where he worked on standards for mobile devices security. Mr. Abu-Amara also was a Project Leader with Samsung Electronics, a Chief Technical Officer and founding member of the startup Lucrometrics, a Senior Advisor and Technical Manager at Nortel Networks, and an Assistant Professor of Electrical Engineering at Texas A&M University. He holds Ph.D. and M.S. degrees from the University of Illinois at Urbana-Champaign and a B.S. degree from the University of California, Berkeley, all in Electrical Engineering.

Jessica Feng Sanford has extensive expertise in next generation service-oriented network management technologies affiliated with TM Forum, systems engineering and integration, and wireless sensor networks. She is a chapter coauthor of "Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems" (CRC, 2004) and with more than 20 other conference and journal publications. She currently holds the position of Senior Consultant at Booz Allen Hamilton, where she supports SOA-based network management for space communications. In addition, Ms. Sanford supported the TSAT program in the areas of SLM and cross-system interface engineering. Prior to joining Booz Allen, Ms. Sanford held the position as an academic researcher at the University of California, Los Angeles. Her research focus included computational sensing, integrated practical optimization, statistical methods, and statistics-based algorithms for wireless ad-hoc sensor networks (WASNs). She holds Ph.D. and M.S. degrees in Computer Science from the University of California, Los Angeles.

Index

.NET, 215
2FA, 168

A
ABE, 236, 239, 289, 291
abstraction, 16, 23, 25, 28, 34, 39, 55, 148, 156, 238, 240, 332, 363, 367, 369, 380, 381, 405, 413
access control layer (ACL), 78
ADM, 4
Architecture change management phase, 6
Architecture vision phase, 5
  Business architecture phase, 5
  Implementation and Governance, 6
  Information systems architecture, 5
  Migration planning phase, 5
  Opportunities and solutions phase, 5
  Technology architecture phase, 5
Agent
  client-based monitor, 325
  managed, 337
  server-based monitor, 325
Aggregate Business Entities *See* ABE, 236
AJAX, 93, 117
Alarms, 324
Alerting, 202
Amazon Web Services *See* AWS, 17
Ames Internet Exchange, 78
API, 12, 18, 39, 57, 78, 107, 109, 118, 121, 122, 163, 211–213, 368, 380, 413
Application Program Interfaces *See* API, 12
Application resource measurement, 317
Architecture Development Method *See* ADM, 4
Area of Responsibility (AOR), 244
Argonne Leadership Computing Facility (ALCF), 76
ARM, 317, 318
Assets
  agent, 309, 310
  dynamic, 309
  enterprise, 309
  static, 309
Asynchronous Transfer Mode *See* ATM, 17
ATM, 17, 123
Atomicity, Consistency, Integrity, and Durability (ACID), 151
Authentication, 194, 198, 348
  system, 197
Authorization, 198, 215, 315, 316, 341, 356, 378, 379
Automation, 148, 157
Availability, 202, 206, 214, 315, 322
  management, 214
  zone, 206
AWS, 17, 18, 78, 112, 149
Azure, 17, 19, 96, 149, 150

B
Basic Execution Service (BES), 114
Basically Available, Soft State, Eventually Consistent (BASE), 152
Blogger, 124
BPM, 57, 218, 302
BSF, 332
  VBSF, 332
BSS, 62, 128, 129, 232, 234, 267, 296, 300
Business application platforms, 96
Business service fabric, 332
  virtual, 332
Business Support Systems *See* BSS, 62
Business to Business Integration (B2Bi), 14

C
C2, 72, 74
C2C2, 72

W. Y. Chang et al., *Transforming Enterprise Cloud Services*,
DOI 10.1007/978-90-481-9846-7 © Springer Science+Business Media B.V. 2010

CA service desk manager (CA Wily Introscope), 243
CA Spectrum Automation Manager (CA CMDB), 243
CAIDS, 360
Capacity management, 214
capital expenditures (CapEx), 391
Catalog, 198, 216
Center for Intelligent Information Retrieval (CIIR), 80
Certificate Authorities (CAs), 347
Certificate Revocation Lists (CRLs), 348
CIM, 102, 128, 240
CIO, 65
Circle of Trust (CoT), 351
CLI, 223
Closed Circuit Television (CCTV), 163
Cloud, 7
  ecosystem, 11
  services, 8
  technology, 9
Cloud Aggregators, 13
Cloud computing, 376
Cloud Computing, 7, 89, 215
Cloud Computing Program Management Office (CC PMO), 67
Cloud control system, 206
Cloud environment, 215
Cloud Infrastructure Vendors, 14
Cloud Integrators, 13
Cloud management environment, 220
Cloud Policy Extension Point *See* CPEP, 255
Cloud Security Alliance (CSA), 183
Cloud Users, 11
Cloud Virtual Resource, 81
Cluster Exploratory (CLuE), 80
CMDB, 281, 303, 304, 307
CoI, 31, 55, 72, 73, 231, 244, 253, 259, 386
Collaboration, 191, 198, 213–215
Colocation Games (CGs), 80
Command and Control *See* C2, 72
Command and Control Framework (NECC), 73
Command and Control of Command and Control *See* C2C2, 72
Command Line Interface *See* CLI, 223
Comma-Separated Values *See* CSV, 66
Commercial-off-the-Shelf *See* COTS, 241
Commoditization, 157
Commodity Futures Trading Commission (CFTC), 62
Common Information Model *See* CIM, 102
Community Cloud, 47, 273
Community of Interests *See* CoI, 31
Composite application, 193, 198, 371
Composition architecture, 198
Composition layer, 198
Computer Aided Design (CAD), 300
Confidentiality, 315, 341
Configuration Management Database *See* CMDB, 281
Connect time, 316
Content Monitoring and Filtering (CMF), 162
Content Providers, 14
Cooperative Anomaly and Intrusion Detection System *See* CAIDS, 360
COTS, 241, 254
CPEP, 255
Create Services, 337
CRM, 17, 19, 25, 54, 57, 59, 61, 319
Cryptographic Service Provider *See* CSP, 348
CSP, 348
CSV, 66
Customer Relationship Management *See* CRM, 17

D
DAS, 207
Data at Rest, 166
Data channel, 194
Data in Motion, 166
Data in Progress, 166
Data in Transit, 166
Data in Use, 166
Data Leak Protection *See* DLP, 171
Data Loss Prevention, 162
Data Movement Interface (DMI), 114
Data retention policy, 202
Data separation, 204
Data synchronization, 202
Data-aggregation, 198
Database Performance Manager (DPM), 243
datacenter, 12, 17, 53, 94, 96, 99, 105, 141, 212, 279, 294, 387, 390, 393, 403
Datacenter Managers, 12
Data-Intensive Computing, 76
Decryption, 194
Deep packet inspection (DPI), 162
De-feasible Policy Composition *See* DPC, 266
Defense Enterprise Computing Center (DECC), 74
Defense Information Systems Agency *See* DISA, 72
Delay, 315
DEN, 236, 237
Denial of Service *See* DOS, 144
DEN-ng, 237, 239

Department of Defense *See* DoD, 16
Department of Technology Services, 70
Deployment process, 212, 227
Diagnostics, 335
Digital Japan Creation Project, 82
Digital signature, 194
Direct-Attached Storage *See* DAS, 207
Directory Enabled Networks *See* DEN, 236
Directory service, 197
DISA, 72–75, 241
Disk Space, 316
Distributed Denial of Service (DDoS), 162
Distributed Management Task Force *See* DMTF, 19
DLP, 171
DMTF, 19, 64, 102, 208, 240, 269, 281, 305, 405
DNS Security Extensions (DNSSEC), 162
DoD, 16, 20, 21, 70, 72–75, 342
DOE, 76
Domain intelligence, 198
DOS, 144, 316
DPC, 266

E

EA, 3, 6, 34, 90, 100, 101, 103, 105–107, 189, 298, 394
EC2, 18, 78, 96, 110, 112, 243, 296, 354
ECBC, 19, 64
Elastic capacity, 215, 218, 219
Elastic Compute Cloud *See* EC2, 18
Elastic infrastructure, 213
Electronic Data Interchange For Administration, Commerce and Transport (EDIFACT), 195
Electronic Discovery Reference Model (EDRM), 122
Electronic Servicing Environment (ESE), 68
Enabling functions, 312
Encryption, 202
Enhanced Telecommunications Operations Map *See* eTOM, 4
Enterprise, 2, 34, 36, 59, 60, 72, 126, 138, 165, 174, 177, 191, 208, 214, 236, 273, 279, 303, 309, 369, 389, 394, 395
  Management, 6, 283, 287
Enterprise Architecture *See* EA, 3
Enterprise Cloud Buyers Council *See* ECBC, 19
Enterprise File Delivery (EFD), 74
Enterprise Key Management Infrastructure (EKMI), 122
Enterprise Resource Planning *See* ERP, 54
Enterprise Service Buses *See* ESB, 24
Entitlement, 202
ERP, 54, 57, 81
ESB, 24, 261
eTOM, 4, 6, 129, 282, 283, 288, 289, 298
Event
  correlation, 334
  manager, 334
Eventing, 199
Exception handling, 202
Exchange Commission (SEC), 62
eXtensible Access Control Markup Language (XACML), 122
Extensible Markup Language *See* XML, 12
Extensible Messaging and Presence Protocol (XMPP), 120

F

Family Education Rights and Privacy Act (FERBA), 171
Fault isolation, 335
Fault management, 335
Fault prevention, 324, 326
fault tolerance,
  service, 335
Fault, Configuration, Accounting, Performance, and Security (FCAPS), 253
Federal Chief Information Officer,
  Apps.gov, 65
  Data.gov, 66
  Federal EA, 65
  IT Dashboard, 67
Federal Chief Information Officer *See* CIO, 65
Federal Geographic Data Committee (FGDC), 67
Federated Identity Architecture (FIA), 351
Federation, 197, 201, 379
federation agreement, 220
Federation of Networks *See* FoN, 9
Federation of Systems *See* FoS, 253
File Transfer Protocol *See* FTP, 171
Financial management, 214
Firewall, 193, 196, 369, 381, 382
Flickr, 124
FoN, 10
Force.com, 17, 19, 96, 149
FoS, 253, 274, 405, 406
FP7, 82
  Engineering *See* Q-IMPRESS, 83
  Service Architectures *See* SOA4All, 83
  Service Front-ends *See* ServFace, 83
  Support Actions, 83

FP7, 82 *(cont.)*
Virtualized Architectures
See RESERVOIR, 83
Virtualized Architectures *See* SmartLM, 83
FTP, 171, 194

G
GCDS, 73
NCES Content Delivery, 74
NCES Content Discovery, 73
General Services Administration *See* GSA, 66
General-Purpose Computing on Graphics Processing Units (GPGPU), 206
GIG, 70, 73
GIG Content Delivery Service *See* GCDS, 73
Global Information Grid *See* GIG, 70
GLUE Schema, 114
Goodput, 315
Governance, 219
Government Information Technology Service (GITS), 84
Graceful degradation, 316
Gramm-Leach-Bliley Act (GLBA), 171
Graphical User Interface *See* GUI, 248
Grid Computing, 1, 17, 89, 105, 294
GridFTP, 114
GSA, 66, 67
GUI, 248

H
HaaS, 96, 97, 103
Hadoop Distributed File System (HDFS), 373
Hardware as a Service *See* HaaS, 96
Health Insurance Portability and Accountability Act (HIPAA), 171
Hewlett-Packard *See* HP, 17
High-Performance Computing (HPC), 75
High-Throughput Computing (HTC), 76
HMAC-based One Time Password (HOTP), 122
Host Intrusion Detection and Prevention System (HIDS/HIPS), 163
HP, 17, 18, 64, 74, 99, 242, 270
HRM, 6, 287
HTML, 29, 117, 120, 121, 304
HTML5 *See* HTML, 118
HTTP, 74, 116–118, 121, 125, 171, 210, 211
GET, 210, 211
POST, 211
HTTPS, 74
Human resources, 194
Human Resources Management *See* HRM, 6
Hybrid Cloud, 8, 13, 48, 77, 113, 159, 250, 259, 268, 273, 390, 403
HyperText Markup Language *See* HTML, 29
Hypertext Preprocessor *See* PHP, 74
Hypertext Transfer Protocol *See* HTTP, 74
Hypertext Transfer Protocol Secure *See* HTTPS, 74
Hypervisor, 216, 220, 332, 354, 370, 372, 381

I
I/O operation rate (IOPS), 77
IaaS, 19, 55, 57, 68, 69, 75, 78, 79, 81, 96, 98, 99, 103, 116, 121, 131, 155, 176, 205, 206, 208, 372, 373, 379, 381, 403, 404, 413
IAM, 70, 122, 162
IBM's WebSphere XD, 243
ICT, 81–83, 388, 389, 394, 403, 405, 407
IDE, 96, 213
Identification Management (IdM), 122
Identity, 201, 216, 219, 344, 351, 367, 379
Identity and Access Management *See* IAM, 70
Identity Federation Framework (ID-FF), 122
Identity-management component, 198
Idle time, 315
IDS, 358, 360, 362, 367, 380, 382, 412
IETF, 116, 119, 121, 125, 240, 242, 245, 269, 349
ILM, 170
IM, 120, 171
Immutable Service Containers (ISCs), 370
Independent Software Vendor *See* ISV, 14
Indicators, 202
Information and Communication Technology *See* ICT, 81
Information Assurance, 25, 40, 64, 88, 89, 161, 164, 172, 342, 343
Information Flow Control (IFC), 355
Information Lifecycle Management *See* ILM, 170
Information Technology *See* IT, 1
Information Technology Infrastructure Library *See* ITIL, 6
Infrastructures as a Service *See* IaaS, 19
Instance management, 220
Instant Messaging *See* IM, 120
Instrumentation, 325, 334
data, 325
Integrated Development Environments *See* IDE, 96
Integrated environment, 213
Integration architecture, 193
Integration broker, 194

Integration Markup Language (IML), 266
International Telecommunication Union
See ITU, 64
International Telecommunications Union, 314
Internet Engineering Task Force *See* IETF, 116
Internet Protocols (IP), 317
Internet service providers (ISPs), 388
Internet Small Computer System Interface
See iSCSI, 78
Interoperability, 316
Intrusion Detection System *See* IDS, 358
IP Security *See* IPSec, 112
IPsec, 146, 352
IPSec, 112, 346
iSCSI, 78
ISO
virtual media image, 208
Isolation
complete, 203
ISP, 393
ISV, 14, 92, 95, 213, 388
IT, 1, 50, 90, 97, 100, 105, 156, 173, 214, 231, 295, 304, 307, 320, 343, 376, 381, 386, 387, 392–395, 401, 407
assets, 4
Federal spending, 67
governance, 38
on-demand IT resources, 43
personnel, 27
roadmap and directions, 38
IT Service Continuity Management (ITCM), 214
ITIL, 6, 15, 129, 184, 275, 281, 306
itSMF, 64
ITU, 64, 314, 317

J
Java, 215
Java 2 Enterprise Edition (J2EE), 369
JavaScript and XML *See* AJAX, 93
JavaScript Object Notation (JSON), 117
Jitter, 315
Job Submission Description Language (JSDL), 114
Just Enough Operating System (JeOS), 28

K
Kasumigaseki Cloud, 82
Key Distribution Centers (KDCs), 349
Key Performance Indicator *See* KPI, 302
Key Quality Indicator *See* KQI, 302
Keyhole Markup Language *See* KML, 66
KML, 66
KPI, 302, 313, 314, 318
KQI, 302, 313, 314, 326

L
LAN, 27
LDAP, 114, 237, 240, 255
Leaf-spine architecture, 222
Learning Management System (LMS), 68
Liberty Alliance, 198, 351, 352
licensing agreements, 139
Lightweight Directory Access Protocol
See LDAP, 114
Line of Business *See* LoB, 68
LiveJournal, 124
LoB, 68
local area network *See* LAN, 27
Locking information, 315
Logical network name, 210
Logical unit number, 206
LUN, 206

M
Manageability, 336
Many-Task Computing, 76
Mash-ups, 213, 221
Massachusetts Institute of Technology (MIT), 80
Massive Graphs in Clusters, 81
Maturity model, 191
Mean Opinion Score *See* MOS, 317
Mean Time Between Failure (MTBF), 160
Mean Time To Repair (MTTR)., 160
Measurement, 326
Metadata, 194, 209, 210, 212
core, 212
execution engine, 202
service, 194
services, 202
Metering, 202
Metrics, 309, 315, 316, 370
client-side, 224
server-side, 224
Microsoft Operations Framework (MOF), 184
Ministry of Economic Development (MED), 84
Ministry of Internal Affairs and Communications (MIC), 82
Monitoring, 202, 334
agent, 309, 310
enterprise, 309
service, 309

Monthly Recurring Revenue (MRR), 61
MOS, 317
Multipurpose Internet Mail Extensions (MIME), 116
Multi-tenancy, 1, 9, 33, 92, 148, 156, 175, 203, 221, 280, 353, 399

N
NASA, 265
  Jet Propulsion Laboratory, 265
  Nebula, 78
NASA Ames Research Center, 77
National Business Center *See* NBC, 67
National Energy Research Scientific Computing Center (NERSC), 76
National Science Foundation (NSF), 76
NBC, 67
NBC Hybrid Cloud, 68
NBCApps, 68
NBCFiles, 68
NBCGrid, 68
NBCStage, 68
NCES, 72
  Collaboration, 73
  Core Enterprise Services (CES), 73
  Portal, 73
  Service-Oriented Architecture Foundation (SOAF), 73
NCO, 7, 16, 20, 21, 72, 73
Net-Centricity, 72
Network-Centric Warfare, 72
Network Interface Cards *See* NIC, 206
Network Intrusion Prevention and Detection Service (NIDS/NIPS), 162
Network-Centric Enterprise Services *See* NCES, 72
Network-Centric Operations *See* NCO, 7
Network-Centricity, 20
Networked service management, 189
New Generation Operations Systems and Software *See* NGOSS, 128
NGOSS, 128, 236, 237, 276, 289
NIC, 206, 222
  virtual, 210
NIPRNet, 74
NIST, 8, 69
Non-classified Internet Protocol Router Network *See* NIPRNet, 74
Non-repudiation, 316
non-SOA, 104–106, 108
Notifications, 199, 219

O
OATH Challenge Response Algorithms (OCRA), 122
OAuth, 122, 162
Object Store, 78
Office of Management and Budget (OMB), 65
OGF, 19, 114
OLA, 37, 252, 253
Open Cloud Computing Interface (OCCI), 19
Open Grid Forum *See* OGF, 19
Open Group, 317
Open Virtualization Format *See* OVF, 19, 208
OpenID, 40, 122, 123
Operability, 337
Operate Service, 337
operating expenditures (OpEx), 391
Operating Systems *See* OS, 1
Operational Level Agreement *See* OLA, 37
Operations Support Systems *See* OSS, 62
Opsware, 99
Orchestration, 202, 337
OS, 1, 9, 10, 14, 17, 27, 68, 107, 116, 312, 369, 381, 405, 413
OSI model, 311
OSS, 62, 63, 128, 129, 174, 232, 234, 254, 267, 296, 300, 406
OVF, 19, 115, 208, 211, 405
  descriptor, 209
  envelope, 210, 212
  package, 209–211
  virtual system, 210
OVF Specification, 114
OWL, 32

P
P2V, 27
PaaS, 54, 55, 68, 75, 79, 81, 94, 102, 117, 121, 150, 152, 154, 175, 213–217, 399–402
  provider, 215
Payable/Accounts Receivable/General Ledger (AP/AR/GL), 104
pay-as-you-go, 34, 134
pay-as-you-grow, 134
Payment Card Industry (PCI), 171
PBM, 40, 57, 99, 233, 234, 240, 243, 302
PDA, 17, 123
Performance, 202, 364
  events, 326
  parameters, 326
Personal Digital Assistant *See* PDA, 17
Phantom caller, 325
PHP, 215

Physical to Cloud (P2C), 154
Physical-to-Virtual *See* P2V, 27
PKI, 115, 122, 346, 348, 375, 409
Platform as a Service *See* PaaS, 54
PMO, 389
POD, 206, 207
Podding, 206
Policy Core Information Model (PCIM), 240
Policy-based Adaptive Network and Security Management (PAM), 265
Policy-Based Manag *See* PBM, 40
Primary functions, 312
Private Cloud, 8, 10, 48, 59, 62, 70, 77, 84, 111, 112, 180, 215, 217–220, 232, 250, 268, 273, 390, 391, 403
  internal Cloud, 48
Private Communications Transport (PCT), 124
Proactive management, 324
Probability distributions, 322
  Binomial, 322
  Kurtosis, 322
  Normal, 322
Probes, 322, 323
  active, 323, 324
  application, 320
  network, 320
  passive, 323, 324
Process intelligence, 198
Programming language environment, fourth-generation, 213
Project Management Office *See* PMO, 389
Provisioning of Symmetric Keys (KEYPROV), 122
Public Cloud, 8, 9, 46, 50, 59, 110, 113, 216, 250, 273, 368, 390, 403
Public Key Cryptography Standards, 122
Public Key Infrastructure *See* PKI, 115
Public Technology Institute (PTI), 70
Python, 215

Q
Q-IMPRESS, 83
QoS, 9, 20, 24, 62, 64, 157, 158, 161, 162, 181
  agreements, 264
  assurance, 26, 158
  attributes, 223
  database, 242
  management, 241
  metrics, 170, 266
  negotiation, 161
  parameters, 161, 253
  policies, 240, 242
  policy rule, 241
  reports, 324
  specification, 266
Quality management, 309, 318
Quality of service, 223
Quality of Service *See* QoS, 9

R
RACE, 74
Rack Unit (RU), 78
RAID, 78, 206
RAM, 68
Random-Access Memory *See* RAM, 68
Rapid Access Computing Environment *See* RACE, 74
Raw computing platforms, 96
RBAC, 201, 251, 343, 411
RBAC for the Web *See* RBAC/Web, 349
RBAC/Web, 349
RDF, 32, 119
  Schema *See* RDFS, 32
RDFS, 32
Really Simple Syndication (RSS), 119
Records and Information Management *See* RIM, 70
Redundant array of independent disks, 206
Redundant Array of Independent Disks *See* RAID, 78
Register Service, 337
Regulatory controls, 202
Relational Database Management System (RDBMS), 150
Remote Portlets, 24
Remote Procedure Call (RPC), 304
Reporting, 310, 311, 321, 336
Representational State Transfer *See* REST, 121
Requestor (RQ), 352
RESERVOIR, 83
Resilient network, 222
Resource Description Framework *See* RDF, 32
Resource management application, 320
Resource Provider (RP), 352
Response time, 315
REST, 121, 194, 208, 220, 368
return on investment *See* ROI, 22
Revocation, 316
Rich Internet Application (RIA), 118
RIM, 70, 122, 170
Risk Management, Compliance (GRC), 162
ROI, 22, 43, 61, 389
Role-Based Access Control *See* RBAC, 201
Roles, 332, 341, 344, 347

Routing module, 196
Ruby, 215
Running Object Table (ROT), 163

S
S3, 18, 78
SA, 20, 21, 133, 182, 252, 406
SaaS, 17, 19, 45, 54, 57, 68, 75, 79, 81, 91–94, 96, 101, 116–118, 120, 121, 143, 146, 176, 190, 193–196, 201, 203, 252, 379, 396–399
  ad-based, 191
  fixed-fee, 191
  policies, 197
  poll application, 196
  publish and subscribe application, 196
  push application, 196
  subscription-based, 191
  transaction-based, 191
  usage-based, 191
  value-based, 191
SAML, 40, 122, 168
SAN, 206, 207
Sarbanes-Oxley Act (SOX), 171
SC, 1
SDF, 189, 309, 319, 322, 332, 387, 392, 407, 409
  management, 189
SDK, 215
SDP, 189
Secret Internet Protocol Router Network *See* SIPRNet, 74
Secure File Systems (SFS), 372
Secure Internet Protocol *See* IPsec, 146
Secure Sockets Layer *See* SSL, 122
Security, 194, 219, 352, 377
  certification file, 209
  credentials, 197, 198, 337, 344, 345, 368, 371
  module, 194
  policies, 194, 348
  signed token, 197
Security Assertion Markup Language *See* SAML, 40
self-certifying pathnames, 374
Serial Advanced Technology Attachment (SATA), 78
Server template, 220
ServFace, 83
service, 3
Service Asset and Configuration Management (SACM), 306
Service Consumers *See* SC, 1
Service delivery, 214
Service Delivery Framework *See* SDF, 189
Service Delivery Platform *See* SDP, 189
Service delivery platforms, 189
Service Designers, 15
Service functions, 312
Service Level Agreement *See* SLA, 8
Service Level Management *See* SLM, 20
Service Level Objective *See* SLO, 223
Service Management Interface *See* SMI, 409
Service Providers *See* SP, 1
Service quality management, 231
Service Web 3.0, 83
Service-centric IT, 193, 198
Service-Oriented Architecture *See* SOA, 16
Seventh Framework Programme *See* FP7, 82
SGML, 117
SHA-1, 209, 211, 347
Shared Information and Data *See* SID, 128
SID, 128, 236, 289, 292, 297–299
Silo application, 191
Simple Object Access Protocol *See* SOAP, 220
Simple Storage Services *See* S3, 18
Single Sign-On *See* SSO, 169
SIPRNet, 74
Situational Awareness *See* SA, 20
SLA, 8, 25, 54, 61, 69, 87, 90, 128, 129, 146, 158, 175, 180, 184, 223, 228, 232, 252, 256, 309, 324, 327, 403
  Change Requests, 331
  Confidentiality, 332
  Customer Requirements, 330
  Details of Service, 330
  diagnostics, 321
  Dispute Resolution and Escalation, 331
  Exceptions, 330
  form, 329
  glossary, 329
  jeopardy events, 321
  legal advice, 329
  metrics reports, 321
  Overview of Service, 330
  parameters, 225
  Penalties, 331
  Relevant Law, 331
  Responsibilities, 330
  Sampling and Reporting, 331
  template, 330
  Term, 330
  trouble tickets, 321
  usage reports, 321
  violation events, 321
SLM, 20, 130, 173
SLO, 223

SmartLM, 83
SME, 141, 165
SMI, 409
SNIA, 19
SOA, 16, 22–25, 57, 72, 82, 83, 89, 102, 104–106, 108, 128, 134, 189, 261, 298, 299
SOA4All, 83
SOAP, 194, 208, 220, 368
Social application platforms, 95
Social networks, 214
SOE, 2, 37, 129
Software as a Service *See* SaaS, 17
Software Developers, 12
Software development, 213
  kit, 215
Software Development Kit (SDK), 242
Software-Platform-Infrastructure (SPI), 162
SoS, 21, 259, 274, 405
SP, 1, 12, 157, 160, 234, 388, 392, 394, 403, 410, 412
SQM, 318, 320
  Metrics, 322
SSL, 122, 124, 146, 352, 375
SSO, 169, 197, 201, 219
Standard Development Organizations (SDOs), 69
Standard Generalized Markup Language *See* SGML, 117
Storage Networking Industry Association *See* SNIA, 19
Storage Resource Manager Interface (SRM), 114
Subject Matter Expert *See* SME, 141
Supplier/Partner Relationship Management (S/PRM), 286
Support functions, 312
System of Systems *See* SoS, 21
Systems/Software Development Life Cycle (SDLC), 162

T
TAM, 299
Telecommunication Standardization Sector *See* ITU-T, 64
Telecommunications Management Forum *See* TM Forum, 4
Telecoms Application Map *See* TAM, 276
Termination, 316
Testing
  acceptance, 335
  comparison, 335
The Open Group Architectural Framework *See* TOGAF, 4
Third-Party Value-Added Providers, 14
Threshold
  warning, 314
Throughput, 315
Time to Close, 316
Time-based One Time Password (TOTP), 122
TLS, 122, 124, 146, 352
TM Forum, 4, 6, 15, 19, 63, 64, 128–130, 184, 189, 236, 237, 245, 275, 299, 401, 409
  eTOM, 276
  Integration Framework, 276
  SID, 276
  TAM, 276
TOGAF, 4, 7
Token Server (STS), 352
Tools
  fault management, 324
  troubleshooting, 324
Transaction rate, 315
Transformation, 189, 195
  organizations, 35
  people, 35
  process, 37
  reference architecture, 200
  Technology, 37
Transport Layer Security *See* TLS, 122
Two-Factor Authentication *See* 2FA, 168
Type of Service (ToS), 243

U
UDDI, 220, 227
UI, 14, 17, 24, 28, 55, 83, 122, 219, 360
Uniform Resource Locator *See* URL, 16
Universal Description Discovery and Integration *See* UDDI, 220
URL, 16, 121, 123, 125, 349
US Department of Energy *See* DOE, 76
Usage Record (UR), 114
User Interface *See* UI, 14
User profile, 202
User-centric layer, 199

V
Validation module, 195
Value-Added Reseller *See* VAR, 13
vApp, 28, 210–212
  body, 212
  template, 211, 212
VAR, 13, 60, 388

Ventilation, and Air Conditioning (HVAC), 312
Virtual application, 209
Virtual Application *See* vApp, 28
Virtual database, 28
Virtual datacenter, 212
Virtual disk, 209, 210
Virtual disk file, 210
Virtual disk files, 209, 210
Virtual Local Area Network *See* VLAN, 27
Virtual Machine *See* VM, 16
Virtual Machine Image *See* VMI, 69
virtual platform, 27
Virtual Private Cloud *See* VPC, 112
Virtual Private LAN Service *See* VPLS, 28
Virtual Private Network *See* VPN, 16
Virtual server, 206, 221
Virtual Switches *See* VSW, 206
Virtualization, 10, 16, 26, 52, 64, 115, 139, 148, 153, 154, 161, 216, 220, 222, 279, 280, 306, 370, 395, 398, 403
  Network virtualization, 27
  Service virtualization, 33
Viruses, 194, 348, 378
VLAN, 27, 222
VM, 16, 19, 27, 28, 78, 100, 206, 208, 209, 222, 332, 353–356, 368, 372, 373, 379, 380, 404, 405, 408, 412
  sprawl, 220
VMI, 69, 221, 372, 373, 376, 378, 381
VPC, 112
VPLS, 28
VPN, 16, 27, 50, 78, 112, 252, 390
VSW, 206, 221, 222

W
Web application platforms, 95
Web Hypertext Application Technology Working Group (WHATWG), 118
Web Ontology Language *See* OWL, 32
Web Service Definition Language *See* WSDL, 220
Web Service Level Agreement *See* WSLA, 223
Web Services Agreement Specification (WS-Agreement), 114
Web syndication, 118
Wordpress, 124
Workflow, 196, 198, 202, 211
  synchronization module, 196
Workload, 205, 206, 222
World Wide Web *See* WWW, 16
World Wide Web Consortium (W3C), 118
WSDL, 220
WS-Federation, 198, 351, 352
WSLA, 223, 224
WWW, 16, 29, 116, 349
  browser, 12, 14, 29, 30, 44, 47, 94, 107, 118, 120, 123, 125, 220, 304, 305, 349, 353
  Web 1.0, 29
  Web 2.0, 30
  Web 3.0, 32
  Web of data, 32

X
XML, 12, 32, 66, 117–121, 195, 209