# CONTENTS

PREFACE XV
ACKNOWLEDGMENTS XIX
CONTRIBUTORS XXI

PART I FOUNDATIONS 1

**1 Introduction to Cloud Computing** **3**
*William Voorsluys, James Broberg, and Rajkumar Buyya*

1.1 Cloud Computing in a Nutshell / 3
1.2 Roots of Cloud Computing / 5
1.3 Layers and Types of Clouds / 13
1.4 Desired Features of a Cloud / 16
1.5 Cloud Infrastructure Management / 17
1.6 Infrastructure as a Service Providers / 26
1.7 Platform as a Service Providers / 31
1.8 Challenges and Risks / 34
1.9 Summary / 37
References / 37

**2 Migrating into a Cloud** **43**
*T. S. Mohan*

2.1 Introduction / 43
2.2 Broad Approaches to Migrating into the Cloud / 48
2.3 The Seven-Step Model of Migration into a Cloud / 51
2.4 Conclusions / 54
Acknowledgments / 55
References / 55

**3 Enriching the 'Integration as a Service' Paradigm for the Cloud Era** **57**
*Pethuru Raj*

3.1 An Introduction / 57
3.2 The Onset of Knowledge Era / 59
3.3 The Evolution of SaaS / 59
3.4 The Challenges of SaaS Paradigm / 61
3.5 Approaching the SaaS Integration Enigma / 63
3.6 New Integration Scenarios / 67
3.7 The Integration Methodologies / 69
3.8 SaaS Integration Products and Platforms / 72
3.9 SaaS Integration Services / 80
3.10 Businesses-to-Business Integration (B2Bi) Services / 84
3.11 A Framework of Sensor—Cloud Integration [3] / 89
3.12 SaaS Integration Appliances / 94
3.13 Conclusion / 95
References / 95

**4 The Enterprise Cloud Computing Paradigm** **97**
*Tariq Ellahi, Benoit Hudzia, Hui Li, Maik A. Lindner, and Philip Robinson*

4.1 Introduction / 97
4.2 Background / 98
4.3 Issues for Enterprise Applications on the Cloud / 103
4.4 Transition Challenges / 106
4.5 Enterprise Cloud Technology and Market Evolution / 108
4.6 Business Drivers Toward a Marketplace for Enterprise Cloud Computing / 112
4.7 The Cloud Supply Chain / 115
4.8 Summary / 117
Acknowledgments / 117
References / 118

PART II INFRASTRUCTURE AS A SERVICE (IAAS) 121

**5 Virtual Machines Provisioning and Migration Services** **123**
*Mohamed El-Refaey*

5.1 Introduction and Inspiration / 123

5.2 Background and Related Work / 124
5.3 Virtual Machines Provisioning and Manageability / 130
5.4 Virtual Machine Migration Services / 132
5.5 VM Provisioning and Migration in Action / 136
5.6 Provisioning in the Cloud Context / 145
5.7 Future Research Directions / 151
5.8 Conclusion / 154
References / 154

**6 On the Management of Virtual Machines for Cloud Infrastructures 157**
*Ignacio M. Llorente, Rubén S. Montero, Borja Sotomayor, David Breitgand, Alessandro Maraschini, Eliezer Levy, and Benny Rochwerger*

6.1 The Anatomy of Cloud Infrastructures / 158
6.2 Distributed Management of Virtual Infrastructures / 161
6.3 Scheduling Techniques for Advance Reservation of Capacity / 166
6.4 Capacity Management to meet SLA Commitments / 172
6.5 Conclusions and Future Work / 185
Acknowledgments / 186
References / 187

**7 Enhancing Cloud Computing Environments Using a Cluster as a Service 193**
*Michael Brock and Andrzej Goscinski*

7.1 Introduction / 193
7.2 Related Work / 194
7.3 RVWS Design / 197
7.4 Cluster as a Service: The Logical Design / 202
7.5 Proof of Concept / 212
7.6 Future Research Directions / 218
7.7 Conclusion / 219
References / 219

**8 Secure Distributed Data Storage in Cloud Computing 221**
*Yu Chen, Wei-Shinn Ku, Jun Feng, Pu Liu, and Zhou Su*

8.1 Introduction / 221
8.2 Cloud Storage: from LANs TO WANs / 222
8.3 Technologies for Data Security in Cloud Computing / 232

8.4 Open Questions and Challenges / 242
8.5 Summary / 246
References / 246

PART III PLATFORM AND SOFTWARE AS A SERVICE (PaaS/IaaS) 249

**9 Aneka—Integration of Private and Public Clouds 251**
*Christian Vecchiola, Xingchen Chu, Michael Mattess, and Rajkumar Buyya*

9.1 Introduction / 251
9.2 Technologies and Tools for Cloud Computing / 254
9.3 Aneka Cloud Platform / 257
9.4 Aneka Resource Provisioning Service / 259
9.5 Hybrid Cloud Implementation / 262
9.6 Visionary thoughts for Practitioners / 269
9.7 Summary and Conclusions / 271
Acknowledgments / 272
References / 273

**10 CometCloud: An Autonomic Cloud Engine 275**
*Hyunjoo Kim and Manish Parashar*

10.1 Introduction / 275
10.2 CometCloud Architecture / 276
10.3 Autonomic Behavior of CometCloud / 280
10.4 Overview of CometCloud-based Applications / 286
10.5 Implementation and Evaluation / 287
10.6 Conclusion and Future Research Directions / 295
Acknowledgments / 295
References / 296

**11 T-Systems' Cloud-Based Solutions for Business Applications 299**
*Michael Pauly*

11.1 Introduction / 299
11.2 What Enterprises Demand of Cloud Computing / 300
11.3 Dynamic ICT Services / 302
11.4 Importance of Quality and Security in Clouds / 305

11.5 Dynamic Data Center—Producing Business-ready, Dynamic ICT Services / 307
11.6 Case Studies / 314
11.7 Summary: Cloud Computing offers much more than Traditional Outsourcing / 318
Acknowledgments / 319
References / 319

**12 Workflow Engine for Clouds 321**
*Suraj Pandey, Dileban Karunamoorthy, and Rajkumar Buyya*

12.1 Introduction / 321
12.2 Background / 322
12.3 Workflow Management Systems and Clouds / 323
12.4 Architecture of Workflow Management Systems / 326
12.5 Utilizing Clouds for Workflow Execution / 328
12.6 Case Study: Evolutionary Multiobjective Optimizations / 334
12.7 Visionary thoughts for Practitioners / 340
12.8 Future Research Directions / 341
12.9 Summary and Conclusions / 341
Acknowledgments / 342
References / 342

**13 Understanding Scientific Applications for Cloud Environments 345**
*Shantenu Jha, Daniel S. Katz, Andre Luckow, Andre Merzky, and Katerina Stamou*

13.1 Introduction / 345
13.2 A Classification of Scientific Applications and Services in the Cloud / 350
13.3 SAGA-based Scientific Applications that Utilize Clouds / 354
13.4 Discussion / 363
13.5 Conclusions / 367
References / 368

**14 The MapReduce Programming Model and Implementations 373**
*Hai Jin, Shadi Ibrahim, Li Qi, Haijun Cao, Song Wu, and Xuanhua Shi*

14.1 Introduction / 373
14.2 MapReduce Programming Model / 375
14.3 Major MapReduce Implementations for the Cloud / 379

14.4 MapReduce Impacts and Research Directions / 385
14.5 Conclusion / 387
Acknowledgments / 387
References / 387

PART IV MONITORING AND MANAGEMENT 391

**15 An Architecture for Federated Cloud Computing** **393**
*Benny Rochwerger, Constantino Vázquez, David Breitgand, David Hadas, Massimo Villari, Philippe Massonet, Eliezer Levy, Alex Galis, Ignacio M. Llorente, Rubén S. Montero, Yaron Wolfsthal, Kenneth Nagin, Lars Larsson, and Fermín Galán*

15.1 Introduction / 393
15.2 A Typical Use Case / 394
15.3 The Basic Principles of Cloud Computing / 398
15.4 A Model for Federated Cloud Computing / 400
15.5 Security Considerations / 407
15.6 Summary and Conclusions / 410
Acknowledgments / 410
References / 410

**16 SLA Management in Cloud Computing: A Service Provider's Perspective** **413**
*Sumit Bose, Anjaneyulu Pasala, Dheepak R. A, Sridhar Murthy and Ganesan Malaiyandisamy*

16.1 Inspiration / 413
16.2 Traditional Approaches to SLO Management / 418
16.3 Types of SLA / 421
16.4 Life Cycle of SLA / 424
16.5 SLA Management in Cloud / 425
16.6 Automated Policy-based Management / 429
16.7 Conclusion / 435
References / 435

**17 Performance Prediction for HPC on Clouds** **437**
*Rocco Aversa, Beniamino Di Martino, Massimiliano Rak, Salvatore Venticinque, and Umberto Villano*

17.1 Introduction / 437
17.2 Background / 440

17.3 Grid and Cloud / 442
17.4 HPC in the Cloud: Performance-related Issues / 445
17.5 Summary and Conclusions / 453
References / 454

PART V APPLICATIONS 457

**18 Best Practices in Architecting Cloud Applications in the AWS Cloud 459**
*Jinesh Varia*

18.1 Introduction / 459
18.2 Background / 459
18.3 Cloud Concepts / 463
18.4 Cloud Best Practices / 468
18.5 GrepTheWeb Case Study / 479
18.6 Future Research Directions / 486
18.7 Conclusion / 487
Acknowledgments / 487
References / 487

**19 Massively Multiplayer Online Game Hosting on Cloud Resources 491**
*Vlad Nae, Radu Prodan, and Alexandru Iosup*

19.1 Introduction / 491
19.2 Background / 492
19.3 Related Work / 494
19.4 Model / 495
19.5 Experiments / 500
19.6 Future Research Directions / 507
19.7 Conclusions / 507
Acknowledgments / 507
References / 507

**20 Building Content Delivery Networks Using Clouds 511**
*James Broberg*

20.1 Introduction / 511
20.2 Background/Related Work / 512

20.3 MetaCDN: Harnessing Storage Clouds for Low-Cost, High-Performance Content Delivery / 516
20.4 Performance of the MetaCDN Overlay / 525
20.5 Future Directions / 527
20.6 Conclusion / 528
Acknowledgments / 529
References / 529

**21 Resource Cloud Mashups** **533**
*Lutz Schubert, Matthias Assel, Alexander Kipp, and Stefan Wesner*

21.1 Introduction / 533
21.2 Concepts of a Cloud Mashup / 536
21.3 Realizing Resource Mashups / 542
21.4 Conclusions / 545
References / 546

PART VI GOVERNANCE AND CASE STUDIES 549

**22 Organizational Readiness and Change Management in the Cloud Age** **551**
*Robert Lam*

22.1 Introduction / 551
22.2 Basic Concept of Organizational Readiness / 552
22.3 Drivers for Changes: A Framework to Comprehend the Competitive Environment / 555
22.4 Common Change Management Models / 559
22.5 Change Management Maturity Model (CMMM) / 563
22.6 Organizational Readiness Self-Assessment: (Who, When, Where, and How) / 565
22.7 Discussion / 567
22.8 Conclusion / 570
Acknowledgments / 571
References / 572

**23 Data Security in the Cloud** **573**
*Susan Morrow*

23.1 An Introduction to the Idea of Data Security / 573
23.2 The Current State of Data Security in the Cloud / 574

23.3 Homo Sapiens and Digital Information / 575
23.4 Cloud Computing and Data Security Risk / 576
23.5 Cloud Computing and Identity / 578
23.6 The Cloud, Digital Identity, and Data Security / 584
23.7 Content Level Security—Pros and Cons / 586
23.8 Future Research Directions / 588
23.9 Conclusion / 590
Acknowledgments / 591
Further Reading / 591
References / 591

**24 Legal Issues in Cloud Computing** **593**
*Janine Anthony Bowen*

24.1 Introduction / 593
24.2 Data Privacy and Security Issues / 596
24.3 Cloud Contracting models / 601
24.4 Jurisdictional Issues Raised by Virtualization and Data Location / 603
24.5 Commercial and Business Considerations—A Cloud User's Viewpoint / 606
24.6 Special Topics / 610
24.7 Conclusion / 611
24.8 Epilogue / 611
References / 612

**25 Achieving Production Readiness for Cloud Services** **615**
*Wai-Kit Cheah and Henry Kasim*

25.1 Introduction / 615
25.2 Service Management / 615
25.3 Producer–Consumer Relationship / 616
25.4 Cloud Service Life Cycle / 620
25.5 Production Readiness / 626
25.6 Assessing Production Readiness / 626
25.7 Summary / 634
References / 634

**Index** **635**

# PREFACE

Cloud computing has recently emerged as one of the buzzwords in the ICT industry. Numerous IT vendors are promising to offer computation, storage, and application hosting services and to provide coverage in several continents, offering service-level agreements (SLA)-backed performance and uptime promises for their services. While these "clouds" are the natural evolution of traditional data centers, they are distinguished by exposing resources (computation, data/storage, and applications) as standards-based Web services and following a "utility" pricing model where customers are charged based on their utilization of computational resources, storage, and transfer of data. They offer subscription-based access to infrastructure, platforms, and applications that are popularly referred to as IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). While these emerging services have increased interoperability and usability and reduced the cost of computation, application hosting, and content storage and delivery by several orders of magnitude, there is significant complexity involved in ensuring that applications and services can scale as needed to achieve consistent and reliable operation under peak loads.

Currently, expert developers are required to implement cloud services. Cloud vendors, researchers, and practitioners alike are working to ensure that potential users are educated about the benefits of cloud computing and the best way to harness the full potential of the cloud. However, being a new and popular paradigm, the very definition of cloud computing depends on which computing expert is asked. So, while the realization of true utility computing appears closer than ever, its acceptance is currently restricted to cloud experts due to the perceived complexities of interacting with cloud computing providers.

This book illuminates these issues by introducing the reader with the cloud computing paradigm. The book provides case studies of numerous existing compute, storage, and application cloud services and illustrates capabilities and limitations of current providers of cloud computing services. This allows the reader to understand the mechanisms needed to harness cloud computing in their own respective endeavors. Finally, many open research problems that have arisen from the rapid uptake of cloud computing are detailed. We hope that this motivates the reader to address these in their own future research and

development. We believe the book to serve as a reference for larger audience such as systems architects, practitioners, developers, new researchers, and graduate-level students. This book also comes with an associated Web site (hosted at http://www.manjrasoft.com/CloudBook/) containing pointers to advanced on-line resources.

## ORGANIZATION OF THE BOOK

This book contains chapters authored by several leading experts in the field of cloud computing. The book is presented in a coordinated and integrated manner starting with the fundamentals and followed by the technologies that implement them.

The content of the book is organized into six parts:

I. Foundations
II. Infrastructure as a Service (IaaS )
III. Platform and Software as a Service (PaaS/SaaS)
IV. Monitoring and Management
V. Applications
VI. Governance and Case Studies

Part I presents fundamental concepts of cloud computing, charting their evolution from mainframe, cluster, grid, and utility computing. Delivery models such as Infrastructure as a Service, Platform as a Service, and Software as a Service are detailed, as well as deployment models such as Public, Private, and Hybrid Clouds. It also presents models for migrating applications to cloud environments.

Part II covers Infrastructure as a Service (IaaS), from enabling technologies such as virtual machines and virtualized storage, to sophisticated mechanisms for securely storing data in the cloud and managing virtual clusters.

Part III introduces Platform and Software as a Service (PaaS/IaaS), detailing the delivery of cloud hosted software and applications. The design and operation of sophisticated, auto-scaling applications and environments are explored.

Part IV presents monitoring and management mechanisms for cloud computing, which becomes critical as cloud environments become more complex and interoperable. Architectures for federating cloud computing resources are explored, as well as service level agreement (SLA) management and performance prediction.

Part V details some novel applications that have been made possible by the rapid emergence of cloud computing resources. Best practices for architecting cloud applications are covered, describing how to harness the power of loosely coupled cloud resources. The design and execution of applications that leverage

cloud resources such as massively multiplayer online game hosting, content delivery and mashups are explored.

Part VI outlines the organizational, structural, regulatory and legal issues that are commonly encountered in cloud computing environments. Details on how companies can successfully prepare and transition to cloud environments are explored, as well as achieving production readiness once such a transition is completed. Data security and legal concerns are explored in detail, as users reconcile moving their sensitive data and computation to cloud computing providers.

Rajkumar Buyya
*The University of Melbourne and Manjrasoft Pty Ltd., Australia*

James Broberg
*The University of Melbourne, Australia*

Andrzej Goscinski
*Deakin University, Australia*

# ACKNOWLEDGMENTS

First and foremost, we are grateful to all the contributing authors for their time, effort, and understanding during the preparation of the book.

We thank Professor Albert Zomaya, editor of the Wiley book series on parallel and distributed computing, for his enthusiastic support and guidance during the preparation of book and enabling us to easily navigate through Wiley's publication process.

We would like to thank members of the book Editorial Advisory Board for their guidance during the preparation of the book. The board members are: Dr. Geng Lin (CISCO Systems, USA), Prof. Manish Parashar (Rutgers: The State University of New Jersey, USA), Dr. Wolfgang Gentzsch (Max-Planck-Gesellschaft, München, Germany), Prof. Omer Rana (Cardiff University, UK), Prof. Hai Jin (Huazhong University of Science and Technology, China), Dr. Simon See (Sun Microsystems, Singapore), Dr. Greg Pfister (IBM, USA (retired)), Prof. Ignacio M. Llorente (Universidad Complutense de Madrid, Spain), Prof. Geoffrey Fox (Indiana University, USA), and Dr. Walfredo Cirne (Google, USA).

All chapters were reviewed and authors have updated their chapters to address review comments. We thank members of the Melbourne CLOUDS Lab for their time and effort in peer reviewing of chapters.

Raj would like to thank his family members, especially Smrithi, Soumya, and Radha Buyya, for their love, understanding, and support during the preparation of the book. James would like to thank his wife, Amy, for her love and support. Andrzej would like to thank his wife, Teresa, for her love and support.

Finally, we would like to thank the staff at Wiley, particularly, Simone Taylor (Senior Editor, Wiley), Michael Christian (Editorial Assistant, Wiley), and S. Nalini (MPS Limited, a Macmillan Company, Chennai, India). They were wonderful to work with!

R.B.

J.B.

A.G.

# CONTRIBUTORS


Matthias Assel, High Performance Computing Center Stuttgart (HLRS), University of Stuttgart, 70550 Stuttgart, Germany

Rocco Aversa, Department of Information Engineering, Second University of Naples, 81031 Aversa (CE), Italy

Sumit Bose, Unisys Research Center, Bangalore, India - 560025

Janine Anthony Bowen, Esq., McKenna Long & Aldridge LLP, Atlanta, GA 30308, USA

David Breitgand, IBM Haifa Research Lab, Haifa University Campus, 31095, Haifa, Israel

James Broberg, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia

Michael Brock, School of Information Technology, Deakin University, Geelong, Victoria 3217, Australia

Rajkumar Buyya, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia

Haijun Cao, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

Wai-Kit Cheah, Advanced Customer Services, Oracle Corporation (S) Pte Ltd., Singapore 038986

Yu Chen, Department of Electrical and Computer Engineering, State University of New York—Binghamton, Binghamton, NY 13902

Xingchen Chu, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia

Beniamino Di Martino, Department of Information Engineering, Second University of Naples, 81031 Aversa (CE), Italy

Tariq Ellahi, SAP Research Belfast, BT3 9DT, Belfast, United Kingdom

Mohamed A. El-Refaey, Arab Academy for Science, Technology and Maritime Transport, College of Computing and Information Technology, Cairo, Egypt

Jun Feng, Department of Electrical and Computer Engineering, State University of New York—Binghamton, Binghamton, NY 13902

Fermín Galán, Telefónica I + D, Emilio Vargas, 6. 28043 Madrid, Spain

Alex Galis, University College London, Department of Electronic and Electrical Engineering, Torrington Place, London WC1E 7JE, United Kingdom

Andrzej Goscinski, School of Information Technology, Deakin University, Geelong, Victoria 3217, Australia

David Hadas, IBM Haifa Research Lab, Haifa University Campus, 31095, Haifa, Israel

Benoit Hudzia, SAP Research Belfast, BT3 9DT, Belfast, United Kingdom

Shadi Ibrahim, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

Alexandru Iosup, Electrical Engineering, Mathematics and Computer Science Department, Delft University of Technology, 2628 CD, Delft, The Netherlands

Shantenu Jha, Center for Computation and Technology and Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803

Hai Jin, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

Dileban Karunamoorthy, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia

Henry Kasim, HPC and Cloud Computing Center, Oracle Corporation (S) Pte Ltd, #18-01 Suntec Tower Four, Singapore 038986

Daniel S. Katz, Computation Institute, University of Chicago, Chicago, Illinois 60637

Hyunjoo Kim, Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, New Brunswick, NJ

Alexander Kipp, High Performance Computing Center Stuttgart (HLRS), University of Stuttgart, 70550 Stuttgart, Germany

Wei-Shinn Ku, Department of Computer Science and Software Engineering, Auburn University, AL 36849

ROBERT LAM, School of Information and Communication Technologies SAIT Polytechnic, Calgary, Canada T2M 0L4

LARS LARSSON, Department of Computing Science, University Umea, Sweden

ELIEZER LEVY, SAP Research SRC Ra'anana, Ra'anana 43665; Israel

HUI LI, SAP Research Karlsruhe, Vincenz-Priessnitz-Strasse, 176131 Karlsruhe, Germany

MAIK A. LINDNER, SAP Research Belfast, BT3 9DT, Belfast, United Kingdom

PU LIU, IBM Endicott Center, New York, NY

IGNACIO M. LLORENTE, Distributed Systems Architecture Research Group, Departmento de Arquitectura de Computadores y Automática, Facultad de Informática, Universidad Complutense de Madrid, 28040 Madrid, Spain

ANDRE LUCKOW, Center for Computation and Technology, Louisiana State University, Baton Rouge, LA, 70803

GANESAN MALAIYANDISAMY, SETLabs, Infosys Technologies Limited, Electronics City, Bangalore, India, 560100

ALESSANDRO MARASCHINI, ElsagDatamat spa, Rome, Italy

PHILIPPE MASSONET, CETIC, B-6041 Charleroi, Belgium

MICHAEL MATTESS, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia

ANDRE MERZKY, Center for Computation and Technology, Louisiana State University, Baton Rouge, LA, 70803

T. S. MOHAN, Infosys Technologies Limited, Electronics City, Bangalore, India, 560100

RUBÉN S. MONTERO, Distributed Systems Architecture Research Group, Departmento de Arquitectura de Computadores, y Automática, Facultad de Informatica, Universidad Complutense de Madrid, 28040 Madrid, Spain

SUSAN MORROW, Avoco Secure, London W1S 2LQ, United Kingdom

SRIDHAR MURTHY, Infosys Technologies Limited, Electronics City, Bangalore, India, 560100

VLAD NAE, Institute of Computer Science, University of Innsbruck, Technikerstraβe 21a, A-6020 Innsbruck, Austria

KENNETH NAGIN, IBM Haifa Research Lab, Haifa University Campus, 31095, Haifa, Israel

Suraj Pandey, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia

Manish Parashar, Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, New Jersey, USA.

Anjaneyulu Pasala, SETLabs, Infosys Technologies Limited, Electronics City, Bangalore, India, 560100

Michael Pauly, T-Systems, Aachen, Germany

Radu Prodan, Institute of Computer Science, University of Innsbruck, A-6020 Innsbruck, Austria

Li Qi, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

Dheepak R A, SETLabs, Infosys Technologies Limited, Electronics City, Bangalore, India, 560100

Pethuru Raj, Robert Bosch India, Bangalore 560068, India

Massimiliano Rak, Department of Information Engineering, Second University of Naples, 81031 Aversa (CE), Italy

Philip Robinson, SAP Research Belfast, BT3 9DT, Belfast, United Kingdom

Benny Rochwerger, IBM Haifa Research Lab, Haifa University Campus, 31095, Haifa, Israel

Lutz Schubert, High Performance Computing Center Stuttgart (HLRS), University of Stuttgart, 70550 Stuttgart, Germany

Xuanhua Shi, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

Borja Sotomayor, Department of Computer Science, University of Chicago, Chicago, IL

Katerina Stamou, Department of Computer Science, Louisiana State University, Baton Rouge, LA, 70803

Zhou Su, Department of Computer Science, Graduate School of Science and Engineering, Waseda University, Japan

Jinesh Varia, Amazon Web Services, Seattle, WA 98109

Constantino Vázquez, Facultad de Informática, Universidad Complutense de Madrid, 28040 Madrid, Spain

Christian Vecchiola, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia

SALVATORE VENTICINQUE, Department of Information Engineering, Second University of Naples, 81031 Aversa (CE), Italy

UMBERTO VILLANO, Department of Engineering, University of Sannio, 82100 Benevento, Italy

MASSIMO VILLARI, Department. of Mathematics Faculty of Engineering, University of Messina, 98166 Messina, Italy

WILLIAM VOORSLUYS, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia

STEFAN WESNER, High Performance Computing Center Stuttgart (HLRS), University of Stuttgart, 70550 Stuttgart, Germany

YARON WOLFSTHAL, IBM Haifa Research Lab, Haifa University Campus, 31095, Haifa, Israel

SONG WU, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

**PART I**

# FOUNDATIONS

# CHAPTER 1

# INTRODUCTION TO CLOUD COMPUTING

WILLIAM VOORSLUYS, JAMES BROBERG, and RAJKUMAR BUYYA

## 1.1 CLOUD COMPUTING IN A NUTSHELL

When plugging an electric appliance into an outlet, we care neither how electric power is generated nor how it gets to that outlet. This is possible because electricity is virtualized; that is, it is readily available from a wall socket that hides power generation stations and a huge distribution grid. When extended to information technologies, this concept means delivering useful functions while hiding how their internals work. Computing itself, to be considered fully virtualized, must allow computers to be built from distributed components such as processing, storage, data, and software resources [1].

Technologies such as *cluster*, *grid*, and now, *cloud* computing, have all aimed at allowing access to large amounts of computing power in a fully virtualized manner, by aggregating resources and offering a single system view. In addition, an important aim of these technologies has been delivering computing as a utility. Utility computing describes a business model for on-demand delivery of computing power; consumers pay providers based on usage ("pay-as-you-go"), similar to the way in which we currently obtain services from traditional public utility services such as water, electricity, gas, and telephony.

Cloud computing has been coined as an umbrella term to describe a category of sophisticated on-demand computing services initially offered by commercial providers, such as Amazon, Google, and Microsoft. It denotes a model on which a computing infrastructure is viewed as a "cloud," from which businesses and individuals access applications from anywhere in the world on demand [2]. The main principle behind this model is offering computing, storage, and software "as a service."

*Cloud Computing: Principles and Paradigms*, Edited by Rajkumar Buyya, James Broberg and Andrzej Goscinski 

Many practitioners in the commercial and academic spheres have attempted to define exactly what "cloud computing" is and what unique characteristics it presents. Buyya et al. [2] have defined it as follows: "Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualised computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers."

Vaquero et al. [3] have stated "clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service Level Agreements."

A recent McKinsey and Co. report [4] claims that "Clouds are hardware-based services offering compute, network, and storage capacity where: Hardware management is highly abstracted from the buyer, buyers incur infrastructure costs as variable OPEX, and infrastructure capacity is highly elastic."

A report from the University of California Berkeley [5] summarized the key characteristics of cloud computing as: "(1) the illusion of infinite computing resources; (2) the elimination of an up-front commitment by cloud users; and (3) the ability to pay for use ... as needed ..."

The National Institute of Standards and Technology (NIST) [6] characterizes cloud computing as "... a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

In a more generic definition, Armbrust et al. [5] define cloud as the "data center hardware and software that provide services." Similarly, Sotomayor et al. [7] point out that "cloud" is more often used to refer to the IT infrastructure deployed on an Infrastructure as a Service provider data center.

While there are countless other definitions, there seems to be common characteristics between the most notable ones listed above, which a cloud should have: (i) pay-per-use (no ongoing commitment, utility prices); (ii) elastic capacity and the illusion of infinite resources; (iii) self-service interface; and (iv) resources that are abstracted or virtualised.

In addition to raw computing and storage, cloud computing providers usually offer a broad range of software services. They also include APIs and development tools that allow developers to build seamlessly scalable applications upon their services. The ultimate goal is allowing customers to run their everyday IT infrastructure "in the cloud."

A lot of hype has surrounded the cloud computing area in its infancy, often considered the most significant switch in the IT world since the advent of the

Internet [8]. In midst of such hype, a great deal of confusion arises when trying to define what cloud computing is and which computing infrastructures can be termed as "clouds."

Indeed, the long-held dream of delivering computing as a utility has been realized with the advent of cloud computing [5]. However, over the years, several technologies have matured and significantly contributed to make cloud computing viable. In this direction, this introduction tracks the roots of cloud computing by surveying the main technological advancements that significantly contributed to the advent of this emerging field. It also explains concepts and developments by categorizing and comparing the most relevant R&D efforts in cloud computing, especially public clouds, management tools, and development frameworks. The most significant practical cloud computing realizations are listed, with special focus on architectural aspects and innovative technical features.

## 1.2 ROOTS OF CLOUD COMPUTING

We can track the roots of clouds computing by observing the advancement of several technologies, especially in hardware (virtualization, multi-core chips), Internet technologies (Web services, service-oriented architectures, Web 2.0), distributed computing (clusters, grids), and systems management (autonomic computing, data center automation). Figure 1.1 shows the convergence of technology fields that significantly advanced and contributed to the advent of cloud computing.

Some of these technologies have been tagged as hype in their early stages of development; however, they later received significant attention from academia and were sanctioned by major industry players. Consequently, a specification and standardization process followed, leading to maturity and wide adoption. The emergence of cloud computing itself is closely linked to the maturity of such technologies. We present a closer look at the technologies that form the base of cloud computing, with the aim of providing a clearer picture of the cloud ecosystem as a whole.

### 1.2.1 From Mainframes to Clouds

We are currently experiencing a switch in the IT world, from in-house generated computing power into utility-supplied computing resources delivered over the Internet as Web services. This trend is similar to what occurred about a century ago when factories, which used to generate their own electric power, realized that it is was cheaper just plugging their machines into the newly formed electric power grid [8].

Computing delivered as a utility can be defined as "on demand delivery of infrastructure, applications, and business processes in a security-rich, shared, scalable, and based computer environment over the Internet for a fee" [9].

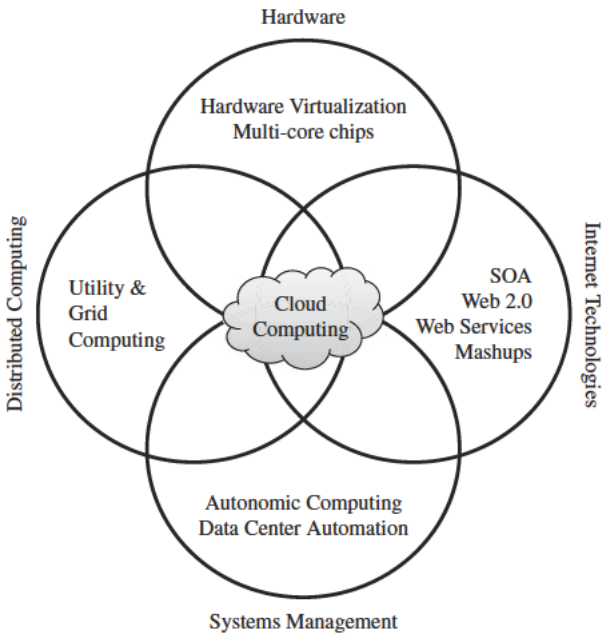



**FIGURE 1.1.** Convergence of various advances leading to the advent of cloud computing.

This model brings benefits to both consumers and providers of IT services. Consumers can attain reduction on IT-related costs by choosing to obtain cheaper services from external providers as opposed to heavily investing on IT infrastructure and personnel hiring. The "on-demand" component of this model allows consumers to adapt their IT usage to rapidly increasing or unpredictable computing needs.

Providers of IT services achieve better operational costs; hardware and software infrastructures are built to provide multiple solutions and serve many users, thus increasing efficiency and ultimately leading to faster return on investment (ROI) as well as lower total cost of ownership (TCO) [10].

Several technologies have in some way aimed at turning the utility computing concept into reality. In the 1970s, companies who offered common data processing tasks, such as payroll automation, operated time-shared mainframes as utilities, which could serve dozens of applications and often operated close to 100% of their capacity. In fact, mainframes had to operate at very high utilization rates simply because they were very expensive and costs should be justified by efficient usage [8].

The mainframe era collapsed with the advent of fast and inexpensive microprocessors and IT data centers moved to collections of commodity servers. Apart from its clear advantages, this new model inevitably led to isolation of workload into dedicated servers, mainly due to incompatibilities

between software stacks and operating systems [11]. In addition, the unavailability of efficient computer networks meant that IT infrastructure should be hosted in proximity to where it would be consumed. Altogether, these facts have prevented the utility computing reality of taking place on modern computer systems.

Similar to old electricity generation stations, which used to power individual factories, computing servers and desktop computers in a modern organization are often underutilized, since IT infrastructure is configured to handle theoretical demand peaks. In addition, in the early stages of electricity generation, electric current could not travel long distances without significant voltage losses. However, new paradigms emerged culminating on transmission systems able to make electricity available hundreds of kilometers far off from where it is generated. Likewise, the advent of increasingly fast fiber-optics networks has relit the fire, and new technologies for enabling sharing of computing power over great distances have appeared.

These facts reveal the potential of delivering computing services with the speed and reliability that businesses enjoy with their local machines. The benefits of economies of scale and high utilization allow providers to offer computing services for a fraction of what it costs for a typical company that generates its own computing power [8].

### 1.2.2 SOA, Web Services, Web 2.0, and Mashups

The emergence of Web services (WS) open standards has significantly contributed to advances in the domain of software integration [12]. Web services can glue together applications running on different messaging product platforms, enabling information from one application to be made available to others, and enabling internal applications to be made available over the Internet.

Over the years a rich WS software stack has been specified and standardized, resulting in a multitude of technologies to describe, compose, and orchestrate services, package and transport messages between services, publish and discover services, represent quality of service (QoS) parameters, and ensure security in service access [13].

WS standards have been created on top of existing ubiquitous technologies such as HTTP and XML, thus providing a common mechanism for delivering services, making them ideal for implementing a service-oriented architecture (SOA). The purpose of a SOA is to address requirements of loosely coupled, standards-based, and protocol-independent distributed computing. In a SOA, software resources are packaged as "services," which are well-defined, self-contained modules that provide standard business functionality and are independent of the state or context of other services. Services are described in a standard definition language and have a published interface [12].

The maturity of WS has enabled the creation of powerful services that can be accessed on-demand, in a uniform way. While some WS are published with the

intent of serving end-user applications, their true power resides in its interface being accessible by other services. An enterprise application that follows the SOA paradigm is a collection of services that together perform complex business logic [12].

This concept of gluing services initially focused on the enterprise Web, but gained space in the consumer realm as well, especially with the advent of Web 2.0. In the consumer Web, information and services may be programmatically aggregated, acting as building blocks of complex compositions, called *service mashups*. Many service providers, such as Amazon, del.icio.us, Facebook, and Google, make their service APIs publicly accessible using standard protocols such as SOAP and REST [14]. Consequently, one can put an idea of a fully functional Web application into practice just by gluing pieces with few lines of code.

In the Software as a Service (SaaS) domain, cloud applications can be built as compositions of other services from the same or different providers. Services such user authentication, e-mail, payroll management, and calendars are examples of building blocks that can be reused and combined in a business solution in case a single, ready-made system does not provide all those features. Many building blocks and solutions are now available in public marketplaces. For example, Programmable Web[1] is a public repository of service APIs and mashups currently listing thousands of APIs and mashups. Popular APIs such as Google Maps, Flickr, YouTube, Amazon eCommerce, and Twitter, when combined, produce a variety of interesting solutions, from finding video game retailers to weather maps. Similarly, Salesforce.com's offers AppExchange,[2] which enables the sharing of solutions developed by third-party developers on top of Salesforce.com components.

### 1.2.3 Grid Computing

Grid computing enables aggregation of distributed resources and transparently access to them. Most production grids such as TeraGrid [15] and EGEE [16] seek to share compute and storage resources distributed across different administrative domains, with their main focus being speeding up a broad range of scientific applications, such as climate modeling, drug design, and protein analysis.

A key aspect of the grid vision realization has been building standard Web services-based protocols that allow distributed resources to be "discovered, accessed, allocated, monitored, accounted for, and billed for, etc., and in general managed as a single virtual system." The Open Grid Services Architecture (OGSA) addresses this need for standardization by defining a set of core capabilities and behaviors that address key concerns in grid systems.

[1] http://www.programmableweb.com

[2] http://sites.force.com/appexchange

Globus Toolkit [18] is a middleware that implements several standard Grid services and over the years has aided the deployment of several service-oriented Grid infrastructures and applications. An ecosystem of tools is available to interact with service grids, including grid brokers, which facilitate user interaction with multiple middleware and implement policies to meet QoS needs.

The development of standardized protocols for several grid computing activities has contributed—theoretically—to allow delivery of on-demand computing services over the Internet. However, ensuring QoS in grids has been perceived as a difficult endeavor [19]. Lack of performance isolation has prevented grids adoption in a variety of scenarios, especially on environments where resources are oversubscribed or users are uncooperative. Activities associated with one user or virtual organization (VO) can influence, in an uncontrollable way, the performance perceived by other users using the same platform. Therefore, the impossibility of enforcing QoS and guaranteeing execution time became a problem, especially for time-critical applications [20].

Another issue that has lead to frustration when using grids is the availability of resources with diverse software configurations, including disparate operating systems, libraries, compilers, runtime environments, and so forth. At the same time, user applications would often run only on specially customized environments. Consequently, a portability barrier has often been present on most grid infrastructures, inhibiting users of adopting grids as utility computing environments [20].

Virtualization technology has been identified as the perfect fit to issues that have caused frustration when using grids, such as hosting many dissimilar software applications on a single physical platform. In this direction, some research projects (e.g., Globus Virtual Workspaces [20]) aimed at evolving grids to support an additional layer to virtualize computation, storage, and network resources.

### 1.2.4 Utility Computing

With increasing popularity and usage, large grid installations have faced new problems, such as excessive spikes in demand for resources coupled with strategic and adversarial behavior by users. Initially, grid resource management techniques did not ensure fair and equitable access to resources in many systems. Traditional metrics (throughput, waiting time, and slowdown) failed to capture the more subtle requirements of users. There were no real incentives for users to be flexible about resource requirements or job deadlines, nor provisions to accommodate users with urgent work.

In utility computing environments, users assign a "utility" value to their jobs, where utility is a fixed or time-varying valuation that captures various QoS constraints (deadline, importance, satisfaction). The valuation is the amount they are willing to pay a service provider to satisfy their demands. The service providers then attempt to maximize their own utility, where said utility may directly correlate with their profit. Providers can choose to prioritize

high yield (i.e., profit per unit of resource) user jobs, leading to a scenario where shared systems are viewed as a marketplace, where users compete for resources based on the perceived utility or value of their jobs. Further information and comparison of these utility computing environments are available in an extensive survey of these platforms [17].

### 1.2.5 Hardware Virtualization

Cloud computing services are usually backed by large-scale data centers composed of thousands of computers. Such data centers are built to serve many users and host many disparate applications. For this purpose, hardware virtualization can be considered as a perfect fit to overcome most operational issues of data center building and maintenance.

The idea of virtualizing a computer system's resources, including processors, memory, and I/O devices, has been well established for decades, aiming at improving sharing and utilization of computer systems [21]. Hardware virtualization allows running multiple operating systems and software stacks on a single physical platform. As depicted in Figure 1.2, a software layer, the virtual machine monitor (VMM), also called a hypervisor, mediates access to the physical hardware presenting to each guest operating system a virtual machine (VM), which is a set of virtual platform interfaces [22].

The advent of several innovative technologies—multi-core chips, paravirtualization, hardware-assisted virtualization, and live migration of VMs—has contributed to an increasing adoption of virtualization on server systems. Traditionally, perceived benefits were improvements on sharing and utilization, better manageability, and higher reliability. More recently, with the adoption of virtualization on a broad range of server and client systems, researchers and practitioners have been emphasizing three basic capabilities regarding

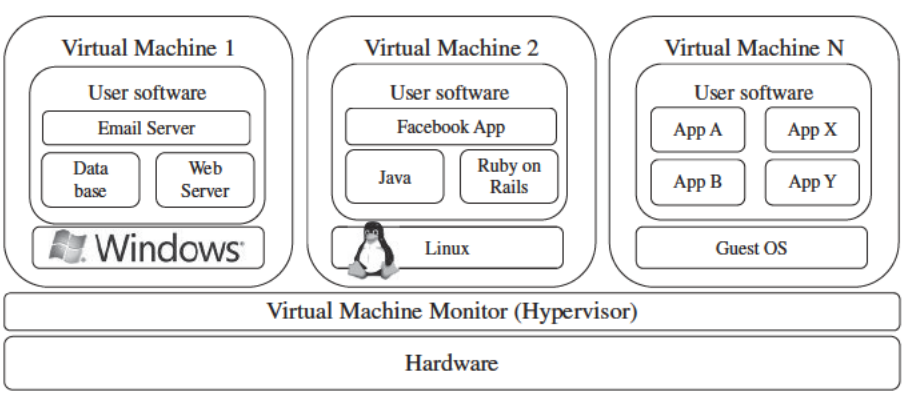



**FIGURE 1.2.** A hardware virtualized server hosting three virtual machines, each one running distinct operating system and user level software stack.

management of workload in a virtualized system, namely isolation, consolidation, and migration [23].

Workload isolation is achieved since all program instructions are fully confined inside a VM, which leads to improvements in security. Better reliability is also achieved because software failures inside one VM do not affect others [22]. Moreover, better performance control is attained since execution of one VM should not affect the performance of another VM [23].

The consolidation of several individual and heterogeneous workloads onto a single physical platform leads to better system utilization. This practice is also employed for overcoming potential software and hardware incompatibilities in case of upgrades, given that it is possible to run legacy and new operation systems concurrently [22].

Workload migration, also referred to as application mobility [23], targets at facilitating hardware maintenance, load balancing, and disaster recovery. It is done by encapsulating a guest OS state within a VM and allowing it to be suspended, fully serialized, migrated to a different platform, and resumed immediately or preserved to be restored at a later date [22]. A VM's state includes a full disk or partition image, configuration files, and an image of its RAM [20].

A number of VMM platforms exist that are the basis of many utility or cloud computing environments. The most notable ones, VMWare, Xen, and KVM, are outlined in the following sections.

***VMWare ESXi.*** VMware is a pioneer in the virtualization market. Its ecosystem of tools ranges from server and desktop virtualization to high-level management tools [24]. ESXi is a VMM from VMWare. It is a bare-metal hypervisor, meaning that it installs directly on the physical server, whereas others may require a host operating system. It provides advanced virtualization techniques of processor, memory, and I/O. Especially, through memory ballooning and page sharing, it can overcommit memory, thus increasing the density of VMs inside a single physical server.

***Xen.*** The Xen hypervisor started as an open-source project and has served as a base to other virtualization products, both commercial and open-source. It has pioneered the para-virtualization concept, on which the guest operating system, by means of a specialized kernel, can interact with the hypervisor, thus significantly improving performance. In addition to an open-source distribution [25], Xen currently forms the base of commercial hypervisors of a number of vendors, most notably Citrix XenServer [26] and Oracle VM [27].

***KVM.*** The kernel-based virtual machine (KVM) is a Linux virtualization subsystem. Is has been part of the mainline Linux kernel since version 2.6.20, thus being natively supported by several distributions. In addition, activities such as memory management and scheduling are carried out by existing kernel

features, thus making KVM simpler and smaller than hypervisors that take control of the entire machine [28].

KVM leverages hardware-assisted virtualization, which improves performance and allows it to support unmodified guest operating systems [29]; currently, it supports several versions of Windows, Linux, and UNIX [28].

### 1.2.6 Virtual Appliances and the Open Virtualization Format

An application combined with the environment needed to run it (operating system, libraries, compilers, databases, application containers, and so forth) is referred to as a "virtual appliance." Packaging application environments in the shape of virtual appliances eases software customization, configuration, and patching and improves portability. Most commonly, an appliance is shaped as a VM disk image associated with hardware requirements, and it can be readily deployed in a hypervisor.

On-line marketplaces have been set up to allow the exchange of ready-made appliances containing popular operating systems and useful software combinations, both commercial and open-source. Most notably, the VMWare virtual appliance marketplace allows users to deploy appliances on VMWare hypervisors or on partners public clouds [30], and Amazon allows developers to share specialized Amazon Machine Images (AMI) and monetize their usage on Amazon EC2 [31].

In a multitude of hypervisors, where each one supports a different VM image format and the formats are incompatible with one another, a great deal of interoperability issues arises. For instance, Amazon has its Amazon machine image (AMI) format, made popular on the Amazon EC2 public cloud. Other formats are used by Citrix XenServer, several Linux distributions that ship with KVM, Microsoft Hyper-V, and VMware ESX.

In order to facilitate packing and distribution of software to be run on VMs several vendors, including VMware, IBM, Citrix, Cisco, Microsoft, Dell, and HP, have devised the Open Virtualization Format (OVF). It aims at being "open, secure, portable, efficient and extensible" [32]. An OVF package consists of a file, or set of files, describing the VM hardware characteristics (e.g., memory, network cards, and disks), operating system details, startup, and shutdown actions, the virtual disks themselves, and other metadata containing product and licensing information. OVF also supports complex packages composed of multiple VMs (e.g., multi-tier applications) [32].

OVF's extensibility has encouraged additions relevant to management of data centers and clouds. Mathews et al. [33] have devised virtual machine contracts (VMC) as an extension to OVF. A VMC aids in communicating and managing the complex expectations that VMs have of their runtime environment and vice versa. A simple example of a VMC is when a cloud consumer wants to specify minimum and maximum amounts of a resource that a VM needs to function; similarly the cloud provider could express resource limits as a way to bound resource consumption and costs.

### 1.2.7 Autonomic Computing

The increasing complexity of computing systems has motivated research on autonomic computing, which seeks to improve systems by decreasing human involvement in their operation. In other words, systems should manage themselves, with high-level guidance from humans [34].

Autonomic, or self-managing, systems rely on monitoring probes and gauges (sensors), on an adaptation engine (autonomic manager) for computing optimizations based on monitoring data, and on effectors to carry out changes on the system. IBM's Autonomic Computing Initiative has contributed to define the four properties of autonomic systems: self-configuration, self-optimization, self-healing, and self-protection. IBM has also suggested a reference model for autonomic control loops of autonomic managers, called MAPE-K (Monitor Analyze Plan Execute—Knowledge) [34, 35].

The large data centers of cloud computing providers must be managed in an efficient way. In this sense, the concepts of autonomic computing inspire software technologies for data center automation, which may perform tasks such as: management of service levels of running applications; management of data center capacity; proactive disaster recovery; and automation of VM provisioning [36].

## 1.3 LAYERS AND TYPES OF CLOUDS

Cloud computing services are divided into three classes, according to the abstraction level of the capability provided and the service model of providers, namely: (1) Infrastructure as a Service, (2) Platform as a Service, and (3) Software as a Service [6]. Figure 1.3 depicts the layered organization of the cloud stack from physical infrastructure to applications.

These abstraction levels can also be viewed as a layered architecture where services of a higher layer can be composed from services of the underlying layer [37]. The reference model of Buyya et al. [38] explains the role of each layer in an integrated architecture. A core middleware manages physical resources and the VMs deployed on top of them; in addition, it provides the required features (e.g., accounting and billing) to offer multi-tenant pay-as-you-go services. Cloud development environments are built on top of infrastructure services to offer application development and deployment capabilities; in this level, various programming models, libraries, APIs, and mashup editors enable the creation of a range of business, Web, and scientific applications. Once deployed in the cloud, these applications can be consumed by end users.

### 1.3.1 Infrastructure as a Service

Offering virtualized resources (computation, storage, and communication) on demand is known as Infrastructure as a Service (IaaS) [7]. A *cloud infrastructure*

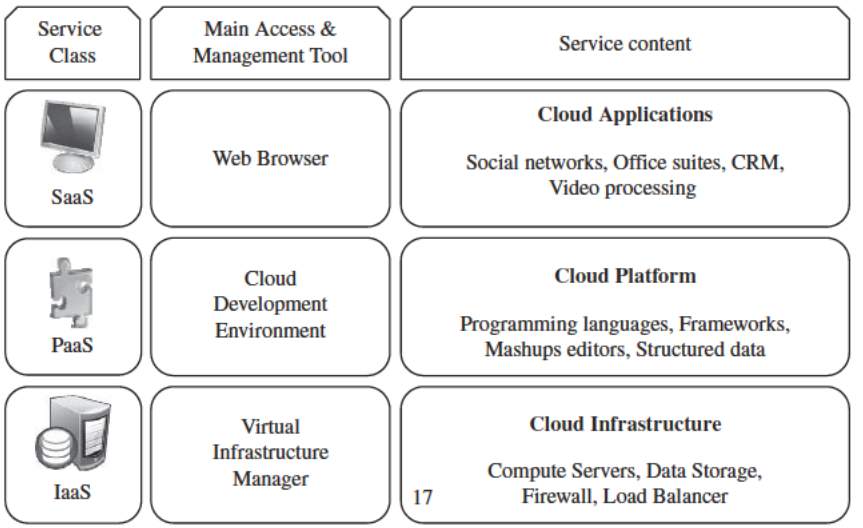



**FIGURE 1.3.** The cloud computing stack.

enables on-demand provisioning of servers running several choices of operating systems and a customized software stack. Infrastructure services are considered to be the bottom layer of cloud computing systems [39].

Amazon Web Services mainly offers IaaS, which in the case of its EC2 service means offering VMs with a software stack that can be customized similar to how an ordinary physical server would be customized. Users are given privileges to perform numerous activities to the server, such as: starting and stopping it, customizing it by installing software packages, attaching virtual disks to it, and configuring access permissions and firewalls rules.

### 1.3.2 Platform as a Service

In addition to infrastructure-oriented clouds that provide raw computing and storage services, another approach is to offer a higher level of abstraction to make a cloud easily programmable, known as Platform as a Service (PaaS). A *cloud platform* offers an environment on which developers create and deploy applications and do not necessarily need to know how many processors or how much memory that applications will be using. In addition, multiple programming models and specialized services (e.g., data access, authentication, and payments) are offered as building blocks to new applications [40].

Google AppEngine, an example of Platform as a Service, offers a scalable environment for developing and hosting Web applications, which should be written in specific programming languages such as Python or Java, and use the services' own proprietary structured object data store. Building blocks

include an in-memory object cache (memcache), mail service, instant messaging service (XMPP), an image manipulation service, and integration with Google Accounts authentication service.

### 1.3.3 Software as a Service

Applications reside on the top of the cloud stack. Services provided by this layer can be accessed by end users through Web portals. Therefore, consumers are increasingly shifting from locally installed computer programs to on-line software services that offer the same functionally. Traditional desktop applications such as word processing and spreadsheet can now be accessed as a service in the Web. This model of delivering applications, known as Software as a Service (SaaS), alleviates the burden of software maintenance for customers and simplifies development and testing for providers [37, 41].

Salesforce.com, which relies on the SaaS model, offers business productivity applications (CRM) that reside completely on their servers, allowing costumers to customize and access applications on demand.

### 1.3.4 Deployment Models

Although cloud computing has emerged mainly from the appearance of public computing utilities, other deployment models, with variations in physical location and distribution, have been adopted. In this sense, regardless of its service class, a cloud can be classified as public, private, community, or hybrid [6] based on model of deployment as shown in Figure 1.4.

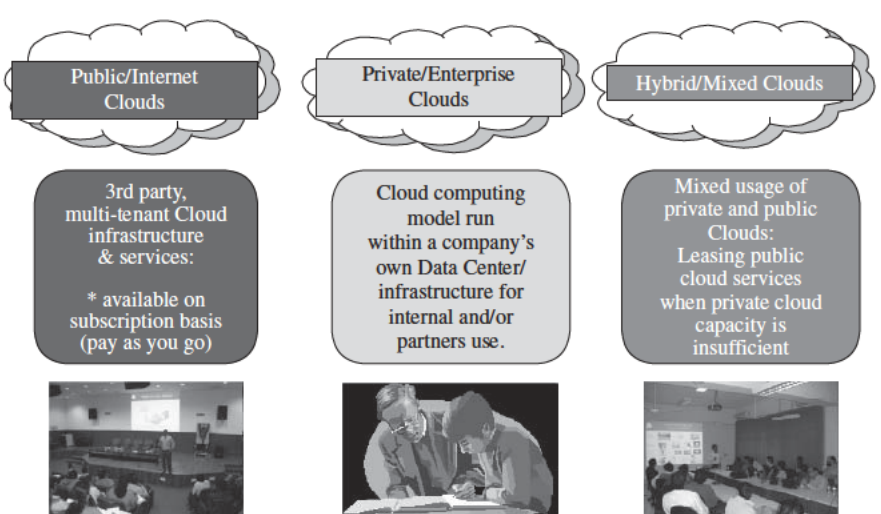



**FIGURE 1.4.** Types of clouds based on deployment models.

Armbrust et al. [5] propose definitions for *public cloud* as a "cloud made available in a pay-as-you-go manner to the general public" and *private cloud* as "internal data center of a business or other organization, not made available to the general public."

In most cases, establishing a private cloud means restructuring an existing infrastructure by adding virtualization and cloud-like interfaces. This allows users to interact with the local data center while experiencing the same advantages of public clouds, most notably self-service interface, privileged access to virtual servers, and per-usage metering and billing.

A *community cloud* is "shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations) [6]."

A *hybrid cloud* takes shape when a private cloud is supplemented with computing capacity from public clouds [7]. The approach of temporarily renting capacity to handle spikes in load is known as "cloud-bursting" [42].

## 1.4 DESIRED FEATURES OF A CLOUD

Certain features of a cloud are essential to enable services that truly represent the cloud computing model and satisfy expectations of consumers, and cloud offerings must be (i) self-service, (ii) per-usage metered and billed, (iii) elastic, and (iv) customizable.

### 1.4.1 Self-Service

Consumers of cloud computing services expect on-demand, nearly instant access to resources. To support this expectation, clouds must allow self-service access so that customers can request, customize, pay, and use services without intervention of human operators [6].

### 1.4.2 Per-Usage Metering and Billing

Cloud computing eliminates up-front commitment by users, allowing them to request and use only the necessary amount. Services must be priced on a short-term basis (e.g., by the hour), allowing users to release (and not pay for) resources as soon as they are not needed [5]. For these reasons, clouds must implement features to allow efficient trading of service such as pricing, accounting, and billing [2]. Metering should be done accordingly for different types of service (e.g., storage, processing, and bandwidth) and usage promptly reported, thus providing greater transparency [6].

### 1.4.3 Elasticity

Cloud computing gives the illusion of infinite computing resources available on demand [5]. Therefore users expect clouds to rapidly provide resources in any

quantity at any time. In particular, it is expected that the additional resources can be (a) provisioned, possibly automatically, when an application load increases and (b) released when load decreases (scale up and down) [6].

### 1.4.4 Customization

In a multi-tenant cloud a great disparity between user needs is often the case. Thus, resources rented from the cloud must be highly customizable. In the case of infrastructure services, customization means allowing users to deploy specialized virtual appliances and to be given privileged (root) access to the virtual servers. Other service classes (PaaS and SaaS) offer less flexibility and are not suitable for general-purpose computing [5], but still are expected to provide a certain level of customization.

## 1.5 CLOUD INFRASTRUCTURE MANAGEMENT

A key challenge IaaS providers face when building a cloud infrastructure is managing physical and virtual resources, namely servers, storage, and networks, in a holistic fashion [43]. The orchestration of resources must be performed in a way to rapidly and dynamically provision resources to applications [7].

The software toolkit responsible for this orchestration is called a virtual infrastructure manager (VIM) [7]. This type of software resembles a traditional operating system—but instead of dealing with a single computer, it aggregates resources from multiple computers, presenting a uniform view to user and applications. The term "cloud operating system" is also used to refer to it [43]. Other terms include "infrastructure sharing software [44]" and "virtual infrastructure engine [45]."

Sotomayor et al. [7], in their description of the cloud ecosystem of software tools, propose a differentiation between two categories of tools used to manage clouds. The first category—cloud toolkits—includes those that "expose a remote and secure interface for creating, controlling and monitoring virtualize resources," but do not specialize in VI management. Tools in the second category—the virtual infrastructure managers—provide advanced features such as automatic load balancing and server consolidation, but do not expose remote cloud-like interfaces. However, the authors point out that there is a superposition between the categories; cloud toolkits can also manage virtual infrastructures, although they usually provide less sophisticated features than specialized VI managers do.

The availability of a remote cloud-like interface and the ability of managing many users and their permissions are the primary features that would distinguish "cloud toolkits" from "VIMs." However, in this chapter, we place both categories of tools under the same group (of the VIMs) and, when applicable, we highlight the availability of a remote interface as a feature.

Virtually all VIMs we investigated present a set of basic features related to managing the life cycle of VMs, including networking groups of VMs together and setting up virtual disks for VMs. These basic features pretty much define whether a tool can be used in practical cloud deployments or not. On the other hand, only a handful of software present advanced features (e.g., high availability) which allow them to be used in large-scale production clouds.

### 1.5.1 Features

We now present a list of both basic and advanced features that are usually available in VIMs.

***Virtualization Support.*** The multi-tenancy aspect of clouds requires multiple customers with disparate requirements to be served by a single hardware infrastructure. Virtualized resources (CPUs, memory, etc.) can be sized and resized with certain flexibility. These features make hardware virtualization, the ideal technology to create a virtual infrastructure that partitions a data center among multiple tenants.

***Self-Service, On-Demand Resource Provisioning.*** Self-service access to resources has been perceived as one the most attractive features of clouds. This feature enables users to directly obtain services from clouds, such as spawning the creation of a server and tailoring its software, configurations, and security policies, without interacting with a human system administrator. This capability "eliminates the need for more time-consuming, labor-intensive, human-driven procurement processes familiar to many in IT" [46]. Therefore, exposing a self-service interface, through which users can easily interact with the system, is a highly desirable feature of a VI manager.

***Multiple Backend Hypervisors.*** Different virtualization models and tools offer different benefits, drawbacks, and limitations. Thus, some VI managers provide a uniform management layer regardless of the virtualization technology used. This characteristic is more visible in open-source VI managers, which usually provide pluggable drivers to interact with multiple hypervisors [7]. In this direction, the aim of libvirt [47] is to provide a uniform API that VI managers can use to manage domains (a VM or container running an instance of an operating system) in virtualized nodes using standard operations that abstract hypervisor specific calls.

***Storage Virtualization.*** Virtualizing storage means abstracting logical storage from physical storage. By consolidating all available storage devices in a data center, it allows creating virtual disks independent from device and location. Storage devices are commonly organized in a storage area network (SAN) and attached to servers via protocols such as Fibre Channel, iSCSI, and

NFS; a storage controller provides the layer of abstraction between virtual and physical storage [48].

In the VI management sphere, storage virtualization support is often restricted to commercial products of companies such as VMWare and Citrix. Other products feature ways of pooling and managing storage devices, but administrators are still aware of each individual device.

***Interface to Public Clouds.*** Researchers have perceived that extending the capacity of a local in-house computing infrastructure by borrowing resources from public clouds is advantageous. In this fashion, institutions can make good use of their available resources and, in case of spikes in demand, extra load can be offloaded to rented resources [45].

A VI manager can be used in a hybrid cloud setup if it offers a driver to manage the life cycle of virtualized resources obtained from external cloud providers. To the applications, the use of leased resources must ideally be transparent.

***Virtual Networking.*** Virtual networks allow creating an isolated network on top of a physical infrastructure independently from physical topology and locations [49]. A virtual LAN (VLAN) allows isolating traffic that shares a switched network, allowing VMs to be grouped into the same broadcast domain. Additionally, a VLAN can be configured to block traffic originated from VMs from other networks. Similarly, the VPN (virtual private network) concept is used to describe a secure and private overlay network on top of a public network (most commonly the public Internet) [50].

Support for creating and configuring virtual networks to group VMs placed throughout a data center is provided by most VI managers. Additionally, VI managers that interface with public clouds often support secure VPNs connecting local and remote VMs.

***Dynamic Resource Allocation.*** Increased awareness of energy consumption in data centers has encouraged the practice of dynamic consolidating VMs in a fewer number of servers. In cloud infrastructures, where applications have variable and dynamic needs, capacity management and demand prediction are especially complicated. This fact triggers the need for dynamic resource allocation aiming at obtaining a timely match of supply and demand [51].

Energy consumption reduction and better management of SLAs can be achieved by dynamically remapping VMs to physical machines at regular intervals. Machines that are not assigned any VM can be turned off or put on a low power state. In the same fashion, overheating can be avoided by moving load away from hotspots [52].

A number of VI managers include a dynamic resource allocation feature that continuously monitors utilization across resource pools and reallocates available resources among VMs according to application needs.

***Virtual Clusters.*** Several VI managers can holistically manage groups of VMs. This feature is useful for provisioning computing *virtual clusters on demand*, and interconnected VMs for multi-tier Internet applications [53].

***Reservation and Negotiation Mechanism.*** When users request computational resources to available at a specific time, requests are termed advance reservations (AR), in contrast to best-effort requests, when users request resources whenever available [54]. To support complex requests, such as AR, a VI manager must allow users to "lease" resources expressing more complex terms (e.g., the period of time of a reservation). This is especially useful in clouds on which resources are scarce; since not all requests may be satisfied immediately, they can benefit of VM placement strategies that support queues, priorities, and advance reservations [55].

Additionally, leases may be negotiated and renegotiated, allowing provider and consumer to modify a lease or present counter proposals until an agreement is reached. This feature is illustrated by the case in which an AR request for a given slot cannot be satisfied, but the provider can offer a distinct slot that is still satisfactory to the user. This problem has been addressed in OpenPEX, which incorporates a bilateral negotiation protocol that allows users and providers to come to an alternative agreement by exchanging offers and counter offers [56].

***High Availability and Data Recovery.*** The high availability (HA) feature of VI managers aims at minimizing application downtime and preventing business disruption. A few VI managers accomplish this by providing a failover mechanism, which detects failure of both physical and virtual servers and restarts VMs on healthy physical servers. This style of HA protects from host, but not VM, failures [57, 58].

For mission critical applications, when a failover solution involving restarting VMs does not suffice, additional levels of fault tolerance that rely on redundancy of VMs are implemented. In this style, redundant and synchronized VMs (running or in standby) are kept in a secondary physical server. The HA solution monitors failures of system components such as servers, VMs, disks, and network and ensures that a duplicate VM serves the application in case of failures [58].

Data backup in clouds should take into account the high data volume involved in VM management. Frequent backup of a large number of VMs, each one with multiple virtual disks attached, should be done with minimal interference in the systems performance. In this sense, some VI managers offer data protection mechanisms that perform incremental backups of VM images. The backup workload is often assigned to proxies, thus offloading production server and reducing network overhead [59].

### 1.5.2 Case Studies

In this section, we describe the main features of the most popular VI managers available. Only the most prominent and distinguishing features of each tool are discussed in detail. A detailed side-by-side feature comparison of VI managers is presented in Table 1.1.

***Apache VCL.*** The Virtual Computing Lab [60, 61] project has been incepted in 2004 by researchers at the North Carolina State University as a way to provide customized environments to computer lab users. The software components that support NCSU's initiative have been released as open-source and incorporated by the Apache Foundation.

Since its inception, the main objective of VCL has been providing desktop (virtual lab) and HPC computing environments anytime, in a flexible cost-effective way and with minimal intervention of IT staff. In this sense, VCL was one of the first projects to create a tool with features such as: self-service Web portal, to reduce administrative burden; advance reservation of capacity, to provide resources during classes; and deployment of customized machine images on multiple computers, to provide clusters on demand.

In summary, Apache VCL provides the following features: (i) multi-platform controller, based on Apache/PHP; (ii) Web portal and XML-RPC interfaces; (iii) support for VMware hypervisors (ESX, ESXi, and Server); (iv) virtual networks; (v) virtual clusters; and (vi) advance reservation of capacity.

***AppLogic.*** AppLogic [62] is a commercial VI manager, the flagship product of 3tera Inc. from California, USA. The company has labeled this product as a Grid Operating System.

AppLogic provides a fabric to manage clusters of virtualized servers, focusing on managing multi-tier Web applications. It views an entire application as a collection of components that must be managed as a single entity. Several components such as firewalls, load balancers, Web servers, application servers, and database servers can be set up and linked together. Whenever the application is started, the system manufactures and assembles the virtual infrastructure required to run it. Once the application is stopped, AppLogic tears down the infrastructure built for it [63].

AppLogic offers dynamic appliances to add functionality such as Disaster Recovery and Power optimization to applications [62]. The key differential of this approach is that additional functionalities are implemented as another pluggable appliance instead of being added as a core functionality of the VI manager.

In summary, 3tera AppLogic provides the following features: Linux-based controller; CLI and GUI interfaces; Xen backend; Global Volume Store (GVS) storage virtualization; virtual networks; virtual clusters; dynamic resource allocation; high availability; and data protection.

**TABLE 1.1. Feature Comparison of Virtual Infrastructure Managers**

| | License | Installation Platform of Controller | Client UI, API, Language Bindings | Backend Hypervisor(s) | Storage Virtualization | Interface to Public Cloud | Virtual Networks | Dynamic Resource Allocation | Advance Reservation of Capacity | High Availability | Data Protection |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Apache VCL | Apache v2 | Multi-platform (Apache/ PHP) | Portal, XML-RPC | VMware ESX, ESXi, Server | No | No | Yes | No | Yes | No | No |
| AppLogic | Proprietary | Linux | GUI, CLI | Xen | Global Volume Store (GVS) | No | Yes | Yes | No | Yes | Yes |
| Citrix Essentials | Proprietary | Windows | GUI, CLI, Portal, XML-RPC | XenServer, Hyper-V | Citrix Storage Link | No | Yes | Yes | No | Yes | Yes |
| Enomaly ECP | GPL v3 | Linux | Portal, WS | Xen | No | Amazon EC2 | Yes | No | No | No | No |
| Eucalyptus | BSD | Linux | EC2 WS, CLI | Xen, KVM | No | EC2 | Yes | No | No | No | No |
| Nimbus | Apache v2 | Linux | EC2 WS, WSRF, CLI | Xen, KVM | No | EC2 | Yes | Via integration with OpenNebula | Yes (via integration with OpenNebula) | No | No |
| OpenNEbula | Apache v2 | Linux | XML-RPC, CLI, Java | Xen, KVM | No | Amazon EC2, Elastic Hosts | Yes | Yes | Yes (via Haizea) | No | No |
| OpenPEX | GPL v2 | Multiplatform (Java) | Portal, WS | XenServer | No | No | No | No | Yes | No | No |
| oVirt | GPL v2 | Fedora Linux | Portal | KVM | No | No | No | No | No | No | No |
| Platform ISF | Proprietary | Linux | Portal | Hyper-V XenServer, VMWare ESX | No | EC2, IBM CoD, HP Enterprise Services | Yes | Yes | Yes | Unclear | Unclear |
| Platform VMO | Proprietary | Linux, Windows | Portal | XenServer | No | No | Yes | Yes | No | Yes | No |
| VMWare vSphere | Proprietary | Linux, Windows | CLI, GUI, Portal, WS | VMware ESX, ESXi | VMware vStorage VMFS | VMware vCloud partners | Yes | VMware DRM | No | Yes | Yes |

***Citrix Essentials.*** The Citrix Essentials suite is one the most feature complete VI management software available, focusing on management and automation of data centers. It is essentially a hypervisor-agnostic solution, currently supporting Citrix XenServer and Microsoft Hyper-V [64].

By providing several access interfaces, it facilitates both human and programmatic interaction with the controller. Automation of tasks is also aided by a workflow orchestration mechanism.

In summary, Citrix Essentials provides the following features: Windows-based controller; GUI, CLI, Web portal, and XML-RPC interfaces; support for XenServer and Hyper-V hypervisors; Citrix Storage Link storage virtualization; virtual networks; dynamic resource allocation; three-level high availability (i.e., recovery by VM restart, recovery by activating paused duplicate VM, and running duplicate VM continuously) [58]; data protection with Citrix Consolidated Backup.

***Enomaly ECP.*** The Enomaly Elastic Computing Platform, in its most complete edition, offers most features a service provider needs to build an IaaS cloud.

Most notably, ECP Service Provider Edition offers a Web-based customer dashboard that allows users to fully control the life cycle of VMs. Usage accounting is performed in real time and can be viewed by users. Similar to the functionality of virtual appliance marketplaces, ECP allows providers and users to package and exchange applications.

In summary, Enomaly ECP provides the following features: Linux-based controller; Web portal and Web services (REST) interfaces; Xen back-end; interface to the Amazon EC2 public cloud; virtual networks; virtual clusters (ElasticValet).

***Eucalyptus.*** The Eucalyptus [39] framework was one of the first open-source projects to focus on building IaaS clouds. It has been developed with the intent of providing an open-source implementation nearly identical in functionality to Amazon Web Services APIs. Therefore, users can interact with a Eucalyptus cloud using the same tools they use to access Amazon EC2. It also distinguishes itself from other tools because it provides a storage cloud API—emulating the Amazon S3 API—for storing general user data and VM images.

In summary, Eucalyptus provides the following features: Linux-based controller with administration Web portal; EC2-compatible (SOAP, Query) and S3-compatible (SOAP, REST) CLI and Web portal interfaces; Xen, KVM, and VMWare backends; Amazon EBS-compatible virtual storage devices; interface to the Amazon EC2 public cloud; virtual networks.

***Nimbus3.*** The Nimbus toolkit [20] is built on top of the Globus framework. Nimbus provides most features in common with other open-source VI managers, such as an EC2-compatible front-end API, support to Xen, and a backend interface to Amazon EC2. However, it distinguishes from others by

providing a Globus Web Services Resource Framework (WSRF) interface. It also provides a backend service, named Pilot, which spawns VMs on clusters managed by a local resource manager (LRM) such as PBS and SGE.

Nimbus' core was engineered around the Spring framework to be easily extensible, thus allowing several internal components to be replaced and also eases the integration with other systems.

In summary, Nimbus provides the following features: Linux-based controller; EC2-compatible (SOAP) and WSRF interfaces; Xen and KVM backend and a Pilot program to spawn VMs through an LRM; interface to the Amazon EC2 public cloud; virtual networks; one-click virtual clusters.

***OpenNebula.*** OpenNebula is one of the most feature-rich open-source VI managers. It was initially conceived to manage local virtual infrastructure, but has also included remote interfaces that make it viable to build public clouds. Altogether, four programming APIs are available: XML-RPC and libvirt [47] for local interaction; a subset of EC2 (Query) APIs and the OpenNebula Cloud API (OCA) for public access [7, 65].

Its architecture is modular, encompassing several specialized pluggable components. The *Core* module orchestrates physical servers and their hypervisors, storage nodes, and network fabric. Management operations are performed through pluggable *Drivers*, which interact with APIs of hypervisors, storage and network technologies, and public clouds. The *Scheduler* module, which is in charge of assigning pending VM requests to physical hosts, offers dynamic resource allocation features. Administrators can choose between different scheduling objectives such as packing VMs in fewer hosts or keeping the load balanced. Via integration with the Haizea lease scheduler [66], OpenNebula also supports advance reservation of capacity and queuing of best-effort leases [7].

In summary, OpenNebula provides the following features: Linux-based controller; CLI, XML-RPC, EC2-compatible Query and OCA interfaces; Xen, KVM, and VMware backend; interface to public clouds (Amazon EC2, ElasticHosts); virtual networks; dynamic resource allocation; advance reservation of capacity.

***OpenPEX.*** OpenPEX (Open Provisioning and EXecution Environment) was constructed around the notion of using advance reservations as the primary method for allocating VM instances. It distinguishes from other VI managers by its leases negotiation mechanism, which incorporates a bilateral negotiation protocol that allows users and providers to come to an agreement by exchanging offers and counter offers when their original requests cannot be satisfied.

In summary, OpenPEX provides the following features: multi-platform (Java) controller; Web portal and Web services (REST) interfaces; Citrix XenServer backend; advance reservation of capacity with negotiation [56].

***oVirt.*** oVirt is an open-source VI manager, sponsored by Red Hat's Emergent Technology group. It provides most of the basic features of other VI managers,

including support for managing physical server pools, storage pools, user accounts, and VMs. All features are accessible through a Web interface [67].

The oVirt admin node, which is also a VM, provides a Web server, secure authentication services based on freeIPA, and provisioning services to manage VM image and their transfer to the managed nodes. Each managed node libvirt, which interfaces with the hypervisor.

In summary, oVirt provides the following features: Fedora Linux-based controller packaged as a virtual appliance; Web portal interface; KVM backend.

***Platform ISF.*** Infrastructure Sharing Facility (ISF) is the VI manager offering from Platform Computing [68]. The company, mainly through its LSF family of products, has been serving the HPC market for several years.

ISF's architecture is divided into three layers. The top most *Service Delivery* layer includes the user interfaces (i.e., self-service portal and APIs); the *Allocation Engine* provides reservation and allocation policies; and the bottom layer—*Resource Integrations*—provides adapters to interact with hypervisors, provisioning tools, and other systems (i.e., external public clouds). The Allocation Engine also provides policies to address several objectives, such as minimizing energy consumption, reducing impact of failures, and maximizing application performance [44].

ISF is built upon Platform's VM Orchestrator, which, as a standalone product, aims at speeding up delivery of VMs to end users. It also provides high availability by restarting VMs when hosts fail and duplicating the VM that hosts the VMO controller [69].

In summary, ISF provides the following features: Linux-based controller packaged as a virtual appliance; Web portal interface; dynamic resource allocation; advance reservation of capacity; high availability.

***VMWare vSphere and vCloud.*** vSphere is VMware's suite of tools aimed at transforming IT infrastructures into private clouds [36, 43]. It distinguishes from other VI managers as one of the most feature-rich, due to the company's several offerings in all levels the architecture.

In the vSphere architecture, servers run on the ESXi platform. A separate server runs vCenter Server, which centralizes control over the entire virtual infrastructure. Through the vSphere Client software, administrators connect to vCenter Server to perform various tasks.

The Distributed Resource Scheduler (DRS) makes allocation decisions based on predefined rules and policies. It continuously monitors the amount of resources available to VMs and, if necessary, makes allocation changes to meet VM requirements. In the storage virtualization realm, vStorage VMFS is a cluster file system to provide aggregate several disks in a single volume. VMFS is especially optimized to store VM images and virtual disks. It supports storage equipment that use Fibre Channel or iSCSI SAN.

In its basic setup, vSphere is essentially a private administration suite. Self-service VM provisioning to end users is provided via the vCloud API, which

interfaces with vCenter Server. In this configuration, vSphere can be used by service providers to build public clouds. In terms of interfacing with public clouds, vSphere interfaces with the vCloud API, thus enabling cloud-bursting into external clouds.

In summary, vSphere provides the following features: Windows-based controller (vCenter Server); CLI, GUI, Web portal, and Web services interfaces; VMware ESX, ESXi backend; VMware vStorage VMFS storage virtualization; interface to external clouds (VMware vCloud partners); virtual networks (VMWare Distributed Switch); dynamic resource allocation (VMware DRM); high availability; data protection (VMWare Consolidated Backup).

## 1.6 INFRASTRUCTURE AS A SERVICE PROVIDERS

Public Infrastructure as a Service providers commonly offer virtual servers containing one or more CPUs, running several choices of operating systems and a customized software stack. In addition, storage space and communication facilities are often provided.

### 1.6.1 Features

In spite of being based on a common set of features, IaaS offerings can be distinguished by the availability of specialized features that influence the cost–benefit ratio to be experienced by user applications when moved to the cloud. The most relevant features are: (i) geographic distribution of data centers; (ii) variety of user interfaces and APIs to access the system; (iii) specialized components and services that aid particular applications (e.g., load-balancers, firewalls); (iv) choice of virtualization platform and operating systems; and (v) different billing methods and period (e.g., prepaid vs. post-paid, hourly vs. monthly).

***Geographic Presence.*** To improve availability and responsiveness, a provider of worldwide services would typically build several data centers distributed around the world. For example, Amazon Web Services presents the concept of "availability zones" and "regions" for its EC2 service. Availability zones are "distinct locations that are engineered to be insulated from failures in other availability zones and provide inexpensive, low-latency network connectivity to other availability zones in the same region." Regions, in turn, "are geographically dispersed and will be in separate geographic areas or countries [70]."

***User Interfaces and Access to Servers.*** Ideally, a public IaaS provider must provide multiple access means to its cloud, thus catering for various users and their preferences. Different types of user interfaces (UI) provide different levels of abstraction, the most common being graphical user interfaces (GUI), command-line tools (CLI), and Web service (WS) APIs.

GUIs are preferred by end users who need to launch, customize, and monitor a few virtual servers and do not necessary need to repeat the process several times. On the other hand, CLIs offer more flexibility and the possibility of automating repetitive tasks via scripts (e.g., start and shutdown a number of virtual servers at regular intervals). WS APIs offer programmatic access to a cloud using standard HTTP requests, thus allowing complex services to be built on top of IaaS clouds.

***Advance Reservation of Capacity.*** Advance reservations allow users to request for an IaaS provider to reserve resources for a specific time frame in the future, thus ensuring that cloud resources will be available at that time. However, most clouds only support best-effort requests; that is, users requests are server whenever resources are available [54].

Amazon Reserved Instances is a form of advance reservation of capacity, allowing users to pay a fixed amount of money in advance to guarantee resource availability at anytime during an agreed period and then paying a discounted hourly rate when resources are in use. However, only long periods of 1 to 3 years are offered; therefore, users cannot express their reservations in finer granularities—for example, hours or days.

***Automatic Scaling and Load Balancing.*** As mentioned earlier in this chapter, elasticity is a key characteristic of the cloud computing model. Applications often need to scale up and down to meet varying load conditions. Automatic scaling is a highly desirable feature of IaaS clouds. It allow users to set conditions for when they want their applications to scale up and down, based on application-specific metrics such as transactions per second, number of simultaneous users, request latency, and so forth.

When the number of virtual servers is increased by automatic scaling, incoming traffic must be automatically distributed among the available servers. This activity enables applications to promptly respond to traffic increase while also achieving greater fault tolerance.

***Service-Level Agreement.*** Service-level agreements (SLAs) are offered by IaaS providers to express their commitment to delivery of a certain QoS. To customers it serves as a warranty. An SLA usually include availability and performance guarantees. Additionally, metrics must be agreed upon by all parties as well as penalties for violating these expectations.

Most IaaS providers focus their SLA terms on availability guarantees, specifying the minimum percentage of time the system will be available during a certain period. For instance, Amazon EC2 states that "if the annual uptime Percentage for a customer drops below 99.95% for the service year, that customer is eligible to receive a service credit equal to 10% of their bill.[3]"

[3] http://aws.amazon.com/ec2 sla

***Hypervisor and Operating System Choice.*** Traditionally, IaaS offerings have been based on heavily customized open-source Xen deployments. IaaS providers needed expertise in Linux, networking, virtualization, metering, resource management, and many other low-level aspects to successfully deploy and maintain their cloud offerings. More recently, there has been an emergence of turnkey IaaS platforms such as VMWare vCloud and Citrix Cloud Center (C3) which have lowered the barrier of entry for IaaS competitors, leading to a rapid expansion in the IaaS marketplace.

### 1.6.2 Case Studies

In this section, we describe the main features of the most popular public IaaS clouds. Only the most prominent and distinguishing features of each one are discussed in detail. A detailed side-by-side feature comparison of IaaS offerings is presented in Table 1.2.

***Amazon Web Services.*** Amazon WS[4] (AWS) is one of the major players in the cloud computing market. It pioneered the introduction of IaaS clouds in 2006. It offers a variety cloud services, most notably: S3 (storage), EC2 (virtual servers), Cloudfront (content delivery), Cloudfront Streaming (video streaming), SimpleDB (structured datastore), RDS (Relational Database), SQS (reliable messaging), and Elastic MapReduce (data processing).

The Elastic Compute Cloud (EC2) offers Xen-based virtual servers (instances) that can be instantiated from Amazon Machine Images (AMIs). Instances are available in a variety of sizes, operating systems, architectures, and price. CPU capacity of instances is measured in Amazon Compute Units and, although fixed for each instance, vary among instance types from 1 (small instance) to 20 (high CPU instance). Each instance provides a certain amount of nonpersistent disk space; a persistence disk service (Elastic Block Storage) allows attaching virtual disks to instances with space up to 1TB.

Elasticity can be achieved by combining the CloudWatch, Auto Scaling, and Elastic Load Balancing features, which allow the number of instances to scale up and down automatically based on a set of customizable rules, and traffic to be distributed across available instances. Fixed IP address (Elastic IPs) are not available by default, but can be obtained at an additional cost.

In summary, Amazon EC2 provides the following features: multiple data centers available in the United States (East and West) and Europe; CLI, Web services (SOAP and Query), Web-based console user interfaces; access to instance mainly via SSH (Linux) and Remote Desktop (Windows); advanced reservation of capacity (aka reserved instances) that guarantees availability for periods of 1 and 3 years; 99.5% availability SLA; per hour pricing; Linux and Windows operating systems; automatic scaling; load balancing.

[4] http://aws.amazon.com

**TABLE 1.2. Feature Comparison Public Cloud Offerings (Infrastructure as a Service)**

| | Geographic Presence | Client UI API Language Bindings | Primary Access to Server | Advance Reservation of Capacity | SLA Uptime | Smallest Billing Unit | Hypervisor | Guest Operating Systems | Automated Horizontal Scaling | Load Balancing | Runtime Server Resizing/ Vertical Scaling | Instance Hardware Capacity | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | Processor | Memory | Storage |
| Amazon EC2 | US East, Europe | CLI, WS, Portal | SSH (Linux), Remote Desktop (Windows) | Amazon reserved instances (Available in 1 or 3 years terms, starting from reservation time) | 99.95% | Hour | Xen | Linux, Windows | Available with Amazon CloudWatch | Elastic Load Balancing | No | 1–20 EC2 compute units | 1.7–15 GB | 160–1690 GB 1 GB–1 TB (per EBS volume) |
| Flexiscale | UK | Web Console | SSH | No | 100% | Hour | Xen | Linux, Windows | No | Zeus software loadbalancing | Processors, memory (requires reboot) | 1–4 CPUs | 0.5–16 GB | 20–270 GB |
| GoGrid | | REST, Java, PHP, Python, Ruby | SSH | No | 100% | Hour | Xen | Linux, Windows | No | Hardware (F5) | No | 1–6 CPUs | 0.5–8 GB | 30–480 GB |
| Joyent Cloud | US (Emeryville, CA; San Diego, CA; Andover, MA; Dallas, TX) | | SSH, VirtualMin (Web-based system administration) | No | 100% | Month | OS Level (Solaris Containers) | OpenSolaris | No | Both hardware (F5 networks) and software (Zeus) | Automatic CPU bursting (up to 8 CPUs) | 1/16–8 CPUs | 0.25–32 GB | 5–100 GB |
| Rackspace Cloud Servers | US (Dallas, TX) | Portal, REST, Python, PHP, Java, C#/.NET | SSH | No | 100% | Hour | Xen | Linux | No | No | Memory, disk (requires reboot) Automatic CPU bursting (up to 100% of available CPU power of physical host) | Quad-core CPU (CPU power is weighed proportionally to memory size) | 0.25–16 GB | 10–620 GB |

***Flexiscale.*** Flexiscale is a UK-based provider offering services similar in nature to Amazon Web Services. However, its virtual servers offer some distinct features, most notably: persistent storage by default, fixed IP addresses, dedicated VLAN, a wider range of server sizes, and runtime adjustment of CPU capacity (aka CPU bursting/vertical scaling). Similar to the clouds, this service is also priced by the hour.

In summary, the Flexiscale cloud provides the following features: available in UK; Web services (SOAP), Web-based user interfaces; access to virtual server mainly via SSH (Linux) and Remote Desktop (Windows); 100% availability SLA with automatic recovery of VMs in case of hardware failure; per hour pricing; Linux and Windows operating systems; automatic scaling (horizontal/vertical).

***Joyent.*** Joyent's Public Cloud offers servers based on Solaris containers virtualization technology. These servers, dubbed accelerators, allow deploying various specialized software-stack based on a customized version of OpenSolaris operating system, which include by default a Web-based configuration tool and several pre-installed software, such as Apache, MySQL, PHP, Ruby on Rails, and Java. Software load balancing is available as an accelerator in addition to hardware load balancers.

A notable feature of Joyent's virtual servers is automatic vertical scaling of CPU cores, which means a virtual server can make use of additional CPUs automatically up to the maximum number of cores available in the physical host.

In summary, the Joyent public cloud offers the following features: multiple geographic locations in the United States; Web-based user interface; access to virtual server via SSH and Web-based administration tool; 100% availability SLA; per month pricing; OS-level virtualization Solaris containers; OpenSolaris operating systems; automatic scaling (vertical).

***GoGrid.*** GoGrid, like many other IaaS providers, allows its customers to utilize a range of pre-made Windows and Linux images, in a range of fixed instance sizes. GoGrid also offers "value-added" stacks on top for applications such as high-volume Web serving, e-Commerce, and database stores.

It offers some notable features, such as a "hybrid hosting" facility, which combines traditional dedicated hosts with auto-scaling cloud server infrastructure. In this approach, users can take advantage of dedicated hosting (which may be required due to specific performance, security or legal compliance reasons) and combine it with on-demand cloud infrastructure as appropriate, taking the benefits of each style of computing.

As part of its core IaaS offerings, GoGrid also provides free hardware load balancing, auto-scaling capabilities, and persistent storage, features that typically add an additional cost for most other IaaS providers.

***Rackspace Cloud Servers.*** Rackspace Cloud Servers is an IaaS solution that provides fixed size instances in the cloud. Cloud Servers offers a range of Linux-based pre-made images. A user can request different-sized images, where the size is measured by requested RAM, not CPU.

Like GoGrid, Cloud Servers also offers hybrid approach where dedicated and cloud server infrastructures can be combined to take the best aspects of both styles of hosting as required. Cloud Servers, as part of its default offering, enables fixed (static) IP addresses, persistent storage, and load balancing (via A-DNS) at no additional cost.

## 1.7 PLATFORM AS A SERVICE PROVIDERS

Public Platform as a Service providers commonly offer a development and deployment environment that allow users to create and run their applications with little or no concern to low-level details of the platform. In addition, specific programming languages and frameworks are made available in the platform, as well as other services such as persistent data storage and in-memory caches.

### 1.7.1 Features

***Programming Models, Languages, and Frameworks.*** Programming models made available by IaaS providers define how users can express their applications using higher levels of abstraction and efficiently run them on the cloud platform. Each model aims at efficiently solving a particular problem. In the cloud computing domain, the most common activities that require specialized models are: processing of large dataset in clusters of computers (MapReduce model), development of request-based Web services and applications; definition and orchestration of business processes in the form of workflows (Workflow model); and high-performance distributed execution of various computational tasks.

For user convenience, PaaS providers usually support multiple programming languages. Most commonly used languages in platforms include Python and Java (e.g., Google AppEngine), .NET languages (e.g., Microsoft Azure), and Ruby (e.g., Heroku). Force.com has devised its own programming language (Apex) and an Excel-like query language, which provide higher levels of abstraction to key platform functionalities.

A variety of software frameworks are usually made available to PaaS developers, depending on application focus. Providers that focus on Web and enterprise application hosting offer popular frameworks such as Ruby on Rails, Spring, Java EE, and .NET.

***Persistence Options.*** A persistence layer is essential to allow applications to record their state and recover it in case of crashes, as well as to store user data.

Traditionally, Web and enterprise application developers have chosen relational databases as the preferred persistence method. These databases offer fast and reliable structured data storage and transaction processing, but may lack scalability to handle several petabytes of data stored in commodity computers [71].

In the cloud computing domain, distributed storage technologies have emerged, which seek to be robust and highly scalable, at the expense of relational structure and convenient query languages. For example, Amazon SimpleDB and Google AppEngine datastore offer schema-less, automatically indexed database services [70]. Data queries can be performed only on individual tables; that is, join operations are unsupported for the sake of scalability.

### 1.7.2 Case Studies

In this section, we describe the main features of some Platform as Service (PaaS) offerings. A more detailed side-by-side feature comparison of VI managers is presented in Table 1.3.

***Aneka.*** Aneka [72] is a .NET-based service-oriented resource management and development platform. Each server in an Aneka deployment (dubbed Aneka cloud node) hosts the Aneka container, which provides the base infrastructure that consists of services for persistence, security (authorization, authentication and auditing), and communication (message handling and dispatching). Cloud nodes can be either physical server, virtual machines (XenServer and VMware are supported), and instances rented from Amazon EC2.

The Aneka container can also host any number of optional services that can be added by developers to augment the capabilities of an Aneka Cloud node, thus providing a single, extensible framework for orchestrating various application models.

Several programming models are supported by such task models to enable execution of legacy HPC applications and MapReduce, which enables a variety of data-mining and search applications.

Users request resources via a client to a reservation services manager of the Aneka master node, which manages all cloud nodes and contains scheduling service to distribute request to cloud nodes.

***App Engine.*** Google App Engine lets you run your Python and Java Web applications on elastic infrastructure supplied by Google. App Engine allows your applications to scale dynamically as your traffic and data storage requirements increase or decrease. It gives developers a choice between a Python stack and Java. The App Engine serving architecture is notable in that it allows real-time auto-scaling without virtualization for many common types of Web applications. However, such auto-scaling is dependent on the

**TABLE 1.3. Feature Comparison of Platform-as-a-Service Cloud Offerings**

| | Target Use | Programming Language, Frameworks | Developer Tools | Programming Models | Persistence Options | Automatic Scaling | Backend Infrastructure Providers |
|---|---|---|---|---|---|---|---|
| Aneka | .Net enterprise applications, HPC | .NET | Standalone SDK | Threads, Task, MapReduce | Flat files, RDBMS, HDFS | No | Amazon EC2 |
| AppEngine | Web applications | Python, Java | Eclipse-based IDE | Request-based Web programming | BigTable | Yes | Own data centers |
| Force.com | Enterprise applications (esp. CRM) | Apex | Eclipse-based IDE, Web-based wizard | Workflow, Excel-like formula language, Request-based web programming | Own object database | Unclear | Own data centers |
| Microsoft Windows Azure | Enterprise and Web applications | .NET | Azure tools for Microsoft Visual Studio | Unrestricted | Table/BLOB/ queue storage, SQL services | Yes | Own data centers |
| Heroku | Web applications | Ruby on Rails | Command-line tools | Request-based web programming | PostgreSQL, Amazon RDS | Yes | Amazon EC2 |
| Amazon Elastic MapReduce | Data processing | Hive and Pig, Cascading, Java, Ruby, Perl, Python, PHP, R, C++ | Karmasphere Studio for Hadoop (NetBeans-based) | MapReduce | Amazon S3 | No | Amazon EC2 |

application developer using a limited subset of the native APIs on each platform, and in some instances you need to use specific Google APIs such as URLFetch, Datastore, and memcache in place of certain native API calls. For example, a deployed App Engine application cannot write to the file system directly (you must use the Google Datastore) or open a socket or access another host directly (you must use Google URL fetch service). A Java application cannot create a new Thread either.

***Microsoft Azure.*** Microsoft Azure Cloud Services offers developers a hosted .NET Stack (C#, VB.Net, ASP.NET). In addition, a Java & Ruby SDK for .NET Services is also available. The Azure system consists of a number of elements. The Windows Azure Fabric Controller provides auto-scaling and reliability, and it manages memory resources and load balancing. The .NET Service Bus registers and connects applications together. The .NET Access Control identity providers include enterprise directories and Windows LiveID. Finally, the .NET Workflow allows construction and execution of workflow instances.

***Force.com.*** In conjunction with the Salesforce.com service, the Force.com PaaS allows developers to create add-on functionality that integrates into main Salesforce CRM SaaS application.

Force.com offers developers two approaches to create applications that can be deployed on its SaaS plaform: a hosted Apex or Visualforce application. Apex is a proprietary Java-like language that can be used to create Salesforce applications. Visualforce is an XML-like syntax for building UIs in HTML, AJAX, or Flex to overlay over the Salesforce hosted CRM system. An application store called AppExchange is also provided, which offers a paid & free application directory.

***Heroku.*** Heroku is a platform for instant deployment of Ruby on Rails Web applications. In the Heroku system, servers are invisibly managed by the platform and are never exposed to users. Applications are automatically dispersed across different CPU cores and servers, maximizing performance and minimizing contention. Heroku has an advanced logic layer than can automatically route around failures, ensuring seamless and uninterrupted service at all times.

## 1.8 CHALLENGES AND RISKS

Despite the initial success and popularity of the cloud computing paradigm and the extensive availability of providers and tools, a significant number of challenges and risks are inherent to this new model of computing. Providers, developers, and end users must consider these challenges and risks to take good advantage of cloud computing. Issues to be faced include user privacy, data

security, data lock-in, availability of service, disaster recovery, performance, scalability, energy-efficiency, and programmability.

### 1.8.1 Security, Privacy, and Trust

Ambrust et al. [5] cite information security as a main issue: "current cloud offerings are essentially public . . . exposing the system to more attacks." For this reason there are potentially additional challenges to make cloud computing environments as secure as in-house IT systems. At the same time, existing, well-understood technologies can be leveraged, such as data encryption, VLANs, and firewalls.

Security and privacy affect the entire cloud computing stack, since there is a massive use of third-party services and infrastructures that are used to host important data or to perform critical operations. In this scenario, the trust toward providers is fundamental to ensure the desired level of privacy for applications hosted in the cloud [38].

Legal and regulatory issues also need attention. When data are moved into the Cloud, providers may choose to locate them anywhere on the planet. The physical location of data centers determines the set of laws that can be applied to the management of data. For example, specific cryptography techniques could not be used because they are not allowed in some countries. Similarly, country laws can impose that sensitive data, such as patient health records, are to be stored within national borders.

### 1.8.2 Data Lock-In and Standardization

A major concern of cloud computing users is about having their data locked-in by a certain provider. Users may want to move data and applications out from a provider that does not meet their requirements. However, in their current form, cloud computing infrastructures and platforms do not employ standard methods of storing user data and applications. Consequently, they do not interoperate and user data are not portable.

The answer to this concern is standardization. In this direction, there are efforts to create open standards for cloud computing.

The Cloud Computing Interoperability Forum (CCIF) was formed by organizations such as Intel, Sun, and Cisco in order to "enable a global cloud computing ecosystem whereby organizations are able to seamlessly work together for the purposes for wider industry adoption of cloud computing technology." The development of the Unified Cloud Interface (UCI) by CCIF aims at creating a standard programmatic point of access to an entire cloud infrastructure.

In the hardware virtualization sphere, the Open Virtual Format (OVF) aims at facilitating packing and distribution of software to be run on VMs so that virtual appliances can be made portable—that is, seamlessly run on hypervisor of different vendors.

### 1.8.3 Availability, Fault-Tolerance, and Disaster Recovery

It is expected that users will have certain expectations about the service level to be provided once their applications are moved to the cloud. These expectations include availability of the service, its overall performance, and what measures are to be taken when something goes wrong in the system or its components. In summary, users seek for a warranty before they can comfortably move their business to the cloud.

SLAs, which include QoS requirements, must be ideally set up between customers and cloud computing providers to act as warranty. An SLA specifies the details of the service to be provided, including availability and performance guarantees. Additionally, metrics must be agreed upon by all parties, and penalties for violating the expectations must also be approved.

### 1.8.4 Resource Management and Energy-Efficiency

One important challenge faced by providers of cloud computing services is the efficient management of virtualized resource pools. Physical resources such as CPU cores, disk space, and network bandwidth must be sliced and shared among virtual machines running potentially heterogeneous workloads.

The multi-dimensional nature of virtual machines complicates the activity of finding a good mapping of VMs onto available physical hosts while maximizing user utility. Dimensions to be considered include: number of CPUs, amount of memory, size of virtual disks, and network bandwidth. Dynamic VM mapping policies may leverage the ability to suspend, migrate, and resume VMs as an easy way of preempting low-priority allocations in favor of higher-priority ones. Migration of VMs also brings additional challenges such as detecting when to initiate a migration, which VM to migrate, and where to migrate. In addition, policies may take advantage of live migration of virtual machines to relocate data center load without significantly disrupting running services. In this case, an additional concern is the trade-off between the negative impact of a live migration on the performance and stability of a service and the benefits to be achieved with that migration [73].

Another challenge concerns the outstanding amount of data to be managed in various VM management activities. Such data amount is a result of particular abilities of virtual machines, including the ability of traveling through space (i.e., migration) and time (i.e., checkpointing and rewinding) [74], operations that may be required in load balancing, backup, and recovery scenarios. In addition, dynamic provisioning of new VMs and replicating existing VMs require efficient mechanisms to make VM block storage devices (e.g., image files) quickly available at selected hosts.

Data centers consumer large amounts of electricity. According to a data published by HP [4], 100 server racks can consume 1.3 MW of power and another 1.3 MW are required by the cooling system, thus costing USD 2.6 million per

year. Besides the monetary cost, data centers significantly impact the environment in terms of $CO_2$ emissions from the cooling systems [52].

In addition to optimize application performance, dynamic resource management can also improve utilization and consequently minimize energy consumption in data centers. This can be done by judiciously consolidating workload onto smaller number of servers and turning off idle resources.

## 1.9 SUMMARY

Cloud computing is a new computing paradigm that offers a huge amount of compute and storage resources to the masses. Individuals (e.g., scientists) and enterprises (e.g., startup companies) can have access to these resources by paying a small amount of money just for what is really needed.

This introductory chapter has surveyed many technologies that have led to the advent of cloud computing, concluding that this new paradigm has been a result of an evolution rather than a revolution.

In their various shapes and flavors, clouds aim at offering compute, storage, network, software, or a combination of those "as a service." Infrastructure-, Platform-, and Software-as-a-service are the three most common nomenclatures for the levels of abstraction of cloud computing services, ranging from "raw" virtual servers to elaborate hosted applications.

A great popularity and apparent success have been visible in this area. However, as discussed in this chapter, significant challenges and risks need to be tackled by industry and academia in order to guarantee the long-term success of cloud computing. Visible trends in this sphere include the emergence of standards; the creation of value-added services by augmenting, combining, and brokering existing compute, storage, and software services; and the availability of more providers in all levels, thus increasing competiveness and innovation. In this sense, numerous opportunities exist for practitioners seeking to create solutions for cloud computing.

## REFERENCES

1. I. Foster, The grid: Computing without bounds, *Scientific American*, vol. 288, No. 4, (April 2003), pp. 78–85.
2. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation Computer Systems*, **25**:599–616, 2009.
3. L. M. Vaquero, L. Rodero Merino, J. Caceres, and M. Lindner, A break in the clouds: Towards a cloud definition, *SIGCOMM Computer Communications Review*, **39**:50–55, 2009.
4. McKinsey & Co., Clearing the Air on Cloud Computing, *Technical Report*, 2009.

5. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, and R. Katz, Above the Clouds: A Berkeley View of Cloud Computing, *UC Berkeley Reliable Adaptive Distributed Systems Laboratory White Paper*, 2009.
6. P. Mell and T. Grance, The NIST Definition of Cloud Computing, National Institute of Standards and Technology, Information Technology Laboratory, *Technical Report Version 15*, 2009.
7. B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, Virtual infrastructure management in private and hybrid clouds, *IEEE Internet Computing*, **13**(5):14–22, September/October, 2009.
8. N. Carr, *The Big Switch: Rewiring the World*, from Edison to Google.W. W. Norton & Co., New York, 2008.
9. M. A. Rappa, The utility business model and the future of computing systems, *IBM Systems Journal*, **43**(1):32–42, 2004.
10. C. S. Yeo et al., Utility computing on global grids, Chapter 143, Hossein Bidgoli (ed.), *The Handbook of Computer Networks*, ISBN: 978 0 471 78461 6, John Wiley & Sons, New York, USA, 2007.
11. I. Foster and S. Tuecke, Describing the elephant: The different faces of IT as service, *ACM Queue*, **3**(6):26–29, 2005.
12. M. P. Papazoglou and W. J. van den Heuvel, Service oriented architectures: Approaches, technologies and research issues, *The VLDB Journal*, **16**:389–415, 2007.
13. H. Kreger, Fulfilling the Web services promise, *Communications of the ACM*, **46**(6):29, 2003.
14. B. Blau, D. Neumann, C. Weinhardt, and S. Lamparter, Planning and pricing of service mashups, in *Proceedings of the 2008 10th IEEE Conference on E Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E Commerce and E Services*, Crystal City, Washington, DC, 2008, pp.19–26.
15. C. Catlett, The philosophy of TeraGrid: Building an open, extensible, distributed TeraScale facility, in *Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, Berlin, Germany, 2002, p. 8.
16. F. Gagliardi, B. Jones, F. Grey, M. E. Begin, and M. Heikkurinen, Building an infrastructure for scientific grid computing: Status and goals of the EGEE project, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **363**(1833):1729, 2005.
17. J. Broberg, S. Venugopal, and R. Buyya, Market oriented Grid and utility computing: The state of the art and future directions, *Journal of Grid Computing*, **6**:255–276, 2008.
18. I. Foster, Globus toolkit version 4: Software for service oriented systems, *Journal of Computer Science and Technology*, **21**(513–520), 2006.
19. R. Buyya and S. Venugopal, *Market oriented computing and global Grids: An introduction*, in *Market Oriented Grid and Utility Computing*, R. Buyya and K. Bubendorfer (eds.), John Wiley & Sons, Hoboken, NJ, 2009, pp. 24–44.
20. K. Keahey, I. Foster, T. Freeman, and X. Zhang, Virtual workspaces: Achieving quality of service and quality of life in the grid, *Scientific Programming*, **13**(4):265–275, 2005.
21. R. P. Goldberg, Survey of virtual machine research, *IEEE Computer*, **7**(6):34–45, 1974.

22. R. Uhlig et al., Intel virtualization technology, *IEEE Computer*, **38**(5):48−56, 2005.
23. P. Barham et al., Xen and the art of virtualization, in *Proceedings of 19th ACM Symposium on Operation Systems Principles*, New York, 2003, pp. 164−177.
24. VMWare Inc., VMWare, *http://www.vmware.com*, 22/4/2010.
25. Xen.org Community, *http://www.xen.org*, 22/4/2010.
26. Citrix Systems Inc., XenServer, *http://www.citrix.com/XenServer*, 22/4/2010.
27. Oracle Corp., Oracle VM, *http://www.oracle.com/technology/products/vm*, 24/4/2010.
28. KVM Project, Kernel based virtual machine, *http://www.linux kvm.org*, 22/4/2010.
29. A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, KVM: The Linux virtual machine monitor, in *Proceedings of the Linux Symposium*, Ottawa, Canada, 2007, p. 225.
30. VMWare Inc., VMWare Virtual Appliance Marketplace, *http://www.vmware.com/appliances*, 22/4/2010.
31. Amazon Web Services Developer Community, Amazon Machine Images, *http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID = 171*, 22/4/2010.
32. Distributed Management Task Force Inc, Open Virtualization Format, *Specifica tion DSP0243 Version 1.0.0*, 2009.
33. J. Matthews, T. Garfinkel, C. Hoff, and J. Wheeler, Virtual machine contracts for datacenter and cloud computing environments, in *Proceedings of the 1st Workshop on Automated Control for Datacenters and Clouds*, 2009, pp. 25−30.
34. International Business Machines Corp., An architectural blueprint for autonomic computing, *White Paper Fourth Edition*, 2006.
35. M. C. Huebscher and J. A. McCann, A survey of autonomic computing degrees, models, and applications, *ACM Computing Surveys*, **40**:1−28, 2008.
36. VMWare Inc., VMware vSphere, *http://www.vmware.com/products/vsphere/*, 22/4/2010.
37. L. Youseff, M. Butrico, and D. Da Silva, Toward a unified ontology of cloud computing, in *Proceedings of the 2008 Grid Computing Environments Workshop*, 2008, pp. 1−10.
38. R. Buyya, S. Pandey, and C. Vecchiola, Cloudbus toolkit for market oriented cloud computing, in *Proceedings 1st International Conference on Cloud Computing (CloudCom 09)*, Beijing, 2009, pp. 3−27.
39. D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, The Eucalyptus open source cloud computing system, in *Proceedings of IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2009)*, Shanghai, China, pp. 124−131, University of California, Santa Barbara. (2009, Sep.) Eucalyptus [online]. http://open.eucalyptus.com.
40. Appistry Inc., Cloud Platforms vs. Cloud Infrastructure, *White Paper*, 2009.
41. B. Hayes, Cloud computing, *Communications of the ACM*, **51**:9−11, 2008.
42. P. T. Jaeger, J. Lin, J. M. Grimes, and S. N. Simmons, Where is the cloud? Geography, economics, environment, and jurisdiction in cloud computing, *First Monday*, **14**(4−5): 2009.
43. VMWare Inc., VMware vSphere, the First Cloud Operating, *White Paper*, 2009.
44. Platform Computing, Platform ISF Datasheet, *White Paper*, 2009.

45. M. D. de Assuncao, A. di Costanzo, and R. Buyya, Evaluating the cost–benefit of using cloud computing to extend the capacity of clusters, in *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing (HPDC 2009)*, Munich, Germany, 2009, pp. 141–150.
46. D. Amrhein, Websphere Journal, *http://websphere.sys con.com/node/1029500*, 22/4/2010.
47. Libvirt: The Virtualization API, Terminology and Goals, *http://libvirt.org/goals.html*, 22/4/2010.
48. A. Singh, M. Korupolu, and D. Mohapatra, Server storage virtualization: Inte gration and load balancing in data centers, in *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, 2008, pp. 1–12.
49. R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*, Addison Wesley Longman, Boston, MA, 1999.
50. A. S. Tanenbaum, *Computer Networks*, Prentice Hall, Upper Saddle River, NJ, 2002.
51. D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, Capacity management and demand prediction for next generation data centers, in *Proceedings of IEEE International Conference on Web Services*, 2007, pp. 43 50.
52. A. Verma, P. Ahuja, and A. Neogi, pMapper: Power and migration cost aware application placement in virtualized systems, in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, 2008, pp. 243 264.
53. K. Keahey and T. Freeman, Contextualization: Providing one click virtual clusters, in *Proceedings of IEEE Fourth International Conference on eScience*, 2008, pp. 301 308.
54. B. Sotomayor, K. Keahey, and I. Foster, Combining batch execution and leasing using virtual machines, in *Proceedings of the 17th International Symposium on High Performance Distributed Computing*, 2008, pp. 87 96.
55. B. Sotomayor, R. Montero, I. M. Llorente, and I. Foster, Capacity leasing in cloud systems using the opennebula engine, *Cloud Computing and Applications*, 2008.
56. S. Venugopal, J. Broberg, and R. Buyya, OpenPEX: An open provisioning and EXecution system for virtual machines, in *Proceedings of the 17th International Conference on Advanced Computing and Communications (ADCOM 2009)*, Ben galuru, India, 2009.
57. VMWare Inc., VMware High Availability (HA), *http://www.vmware.com/products/high availability/index.html*, 22/4/2010.
58. Citrix Systems Inc., The three levels of high availability Balancing priorities and cost, *White Paper*, 2008.
59. VMWare Inc., VMWare vStorage APIs for Data Protection, *http://www.vmware.com/products/vstorage apis for data protection*, 22/4/2010.
60. H. E. Schaffer et al., NCSUs Virtual Computing Lab: A cloud computing solution, *Computer*, **42**:94–97, 2009.
61. North Carolina State University, Virtual Computing Lab (VCL), *http://vcl.ncsu.edu*, 22/4/2010.
62. 3tera Inc., AppLogic Grid Operating System for Web Applications, *http://www.3tera.com/AppLogic*, 22/4/2010.
63. 3Tera Inc., The AppLogic Grid Operating System, *White Paper*, 2006.

64. Citrix Systems Inc., Citrix essentials for Hyper V, *http://www.citrix.com/ehv*, 22/4/2010.
65. Distributed Systems Architecture Group, OpenNebula: The open source toolkit for cloud computing, *http://www.opennebula.org*, 22/4/2010.
66. University of Chicago, Haizea An open source VM based lease manager, *http://haizea.cs.uchicago.edu*, 22/4/2010.
67. Red Hat's Emerging Technology group, oVirt, *http://ovirt.org*, 22/4/2010.
68. Platform Computing Corporation, Platform ISF. *http://www.platform.com/Products/platform isf*, 22/4/2010.
69. Platform Computing, Platform VM Orchestrator, *http://www.platform.com/Products/platform vm orchestrator*, 22/4/2010.
70. Amazon Inc., Amazon Web Services, *http://www.amazon.com*, 22/4/2010.
71. F. Chang et al., Bigtable: A distributed storage system for structured data, in *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI'06)*, 2006, pp. 205–218.
72. C. Vecchiola, X. Chu, and R. Buyya, Aneka: A software platform for .NET based cloud computing, in *High Speed and Large Scale Scientific Computing*, W. Gentzsch, L. Grandinetti, and G. Joubert (eds.), IOS Press, Amsterdam, Nether lands, 2009, pp. 267–295.
73. W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, Cost of virtual machine live migration in clouds: A performance evaluation, in *Proceedings 1st International Conference on Cloud Computing*, Beijing, 2009, pp. 254–265.
74. D. T. Meyer et al., Parallax: Virtual disks for virtual machines, in *Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems*, 2008, pp. 41–54.

# CHAPTER 2

# MIGRATING INTO A CLOUD

T. S. MOHAN

## 2.1 INTRODUCTION

The promise of cloud computing has raised the IT expectations of small and medium enterprises beyond measure. Large companies are deeply debating it. Cloud computing is a disruptive model of IT whose innovation is part technology and part business model—in short a "disruptive techno-commercial model" of IT. This tutorial chapter focuses on the key issues and associated dilemmas faced by decision makers, architects, and systems managers in trying to understand and leverage cloud computing for their IT needs. Questions asked and discussed in this chapter include: when and how to migrate one's application into a cloud; what part or component of the IT application to migrate into a cloud and what not to migrate into a cloud; what kind of customers really benefit from migrating their IT into the cloud; and so on. We describe the key factors underlying each of the above questions and share a Seven-Step Model of Migration into the Cloud.

Cloud computing has been a hotly debated and discussed topic amongst IT professionals and researchers both in the industry and in academia. There are intense discussions on several blogs, in Web sites, and in several research efforts [1–4]. This also resulted in several entrepreneurial efforts to help leverage and migrate into the cloud given the myriad issues, challenges, benefits, and limitations and lack of comprehensive understanding of what cloud computing can do. On the one hand, there were these large cloud computing IT vendors like Google, Amazon, and Microsoft, who had started offering cloud computing services on what seemed like a demonstration and trial basis though not explicitly mentioned. They were charging users fees that in certain contexts demonstrated very attractive pricing models. It demonstrated that cloud computing *per se* was for real and that the "techno-commerical disruptive

*Cloud Computing: Principles and Paradigms*, Edited by Rajkumar Buyya, James Broberg and Andrzej Goscinski 

business model" was indeed giving a greater return on investment (ROI) than traditional IT investment for a business. On the other hand, these initial cloud computing offerings were premature. The cloud computing service vendors were grappling real issues of distributed systems as well as business models and had a number open engineering and research problems [2] that indicated in multiple ways that the cloud computing services were yet to mature fully.

Several efforts have been made in the recent past to define the term "cloud computing" and many have not been able to provide a comprehensive one [2, 5, 6]. This has been more challenging given the scorching pace of the technological advances as well as the newer business model formulations for the cloud services being offered. We propose the following definition of cloud computing: "*It is a techno-business disruptive model of using distributed large-scale data centers either private or public or hybrid offering customers a scalable virtualized infrastructure or an abstracted set of services qualified by service-level agreements (SLAs) and charged only by the abstracted IT resources consumed.*" Most enterprises today are powered by captive data centers. In most large or small enterprises today, IT is the backbone of their operations. Invariably for these large enterprises, their data centers are distributed across various geographies. They comprise systems and software that span several generations of products sold by a variety of IT vendors. In order to meet varying loads, most of these data centers are provisioned with capacity beyond the peak loads experienced. If the enterprise is in a seasonal or cyclical business, then the load variation would be significant. Thus what is observed generally is that the provisioned capacity of IT resources is several times the average demand. This is indicative of significant degree of idle capacity. Many data center management teams have been continuously innovating their management practices and technologies deployed to possibly squeeze out the last possible usable computing resource cycle through appropriate programming, systems configurations, SLAs, and systems management. Cloud computing turned attractive to them because they could pass on the additional demand from their IT setups onto the cloud while paying only for the usage and being unencumbered by the load of operations and management.

### 2.1.1 The Promise of the Cloud

Most users of cloud computing services offered by some of the large-scale data centers are least bothered about the complexities of the underlying systems or their functioning. More so given the heterogeneity of either the systems or the software running on them. They were most impressed by the simplicity, uniformity, and ease of use of the Cloud Computing Service abstractions. In small and medium enterprises, cloud computing usage for all additional cyclical IT needs has yielded substantial and significant economic savings. Many such success stories have been documented and discussed on the Internet. This economics and the associated trade-offs, of leveraging the cloud computing services, now popularly called "cloudonomics," for satisfying enterprise's

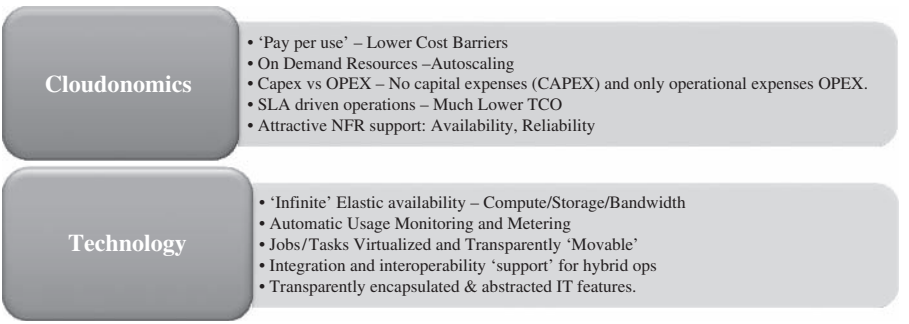



**FIGURE 2.1.** The promise of the cloud computing services.

seasonal IT loads has become a topic of deep interest amongst IT managers and technology architects.

As shown in Figure 2.1, the promise of the cloud both on the business front (the attractive cloudonomics) and the technology front widely aided the CxOs to spawn out several non-mission critical IT needs from the ambit of their captive traditional data centers to the appropriate cloud service. Invariably, these IT needs had some common features: They were typically Web-oriented; they represented seasonal IT demands; they were amenable to parallel batch processing; they were non-mission critical and therefore did not have high security demands. They included scientific applications too [7]. Several small and medium business enterprises, however, leveraged the cloud much beyond the cautious user. Many startups opened their IT departments exclusively using cloud services—very successfully and with high ROI. Having observed these successes, several large enterprises have started successfully running pilots for leveraging the cloud. Many large enterprises run SAP to manage their operations. SAP itself is experimenting with running its suite of products: SAP Business One as well as SAP Netweaver on Amazon cloud offerings. Gartner, Forrester, and other industry research analysts predict that a substantially significant percentage of the top enterprises in the world would have migrated a majority of their IT needs to the cloud offerings by 2012, thereby demonstrating the widespread impact and benefits from cloud computing. Indeed the promise of the cloud has been significant in its impact.

### 2.1.2 The Cloud Service Offerings and Deployment Models

Cloud computing has been an attractive proposition both for the CFO and the CTO of an enterprise primarily due its ease of usage. This has been achieved by large data center service vendors or now better known as cloud service vendors again primarily due to their scale of operations. Google,[1] Amazon,[2]

[1] http://appengine.google.com

[2] http://aws.amazon.com

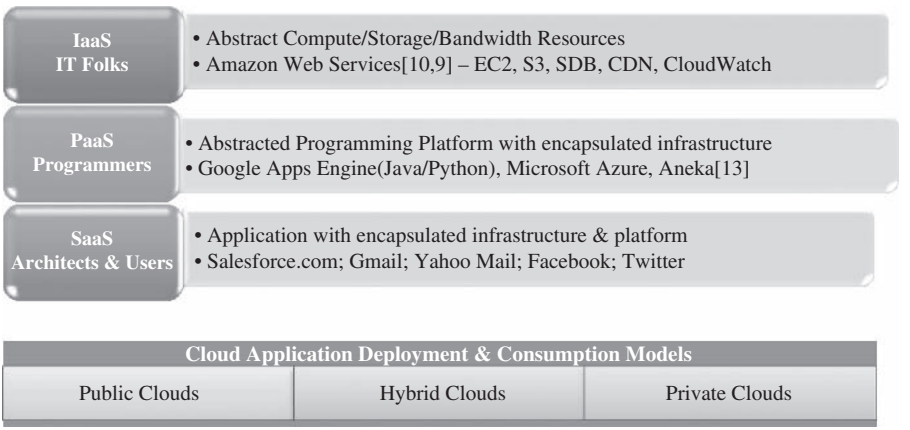



**FIGURE 2.2.** The cloud computing service offering and deployment models.

Microsoft,[3] and a few others have been the key players apart from open source Hadoop[4] built around the Apache ecosystem. As shown in Figure 2.2, the cloud service offerings from these vendors can broadly be classified into three major streams: the *Infrastructure as a Service* (IaaS), the *Platform as a Service* (PaaS), and the *Software as a Service* (SaaS). While IT managers and system administrators preferred IaaS as offered by Amazon for many of their virtualized IT needs, the programmers preferred PaaS offerings like Google AppEngine (Java/Python programming) or Microsoft Azure (.Net programming). Users of large-scale enterprise software invariably found that if they had been using the cloud, it was because their usage of the specific software package was available as a service—it was, in essence, a SaaS offering. Salesforce.com was an exemplary SaaS offering on the Internet.

From a technology viewpoint, as of today, the IaaS type of cloud offerings have been the most successful and widespread in usage. However, the potential of PaaS has been high: All new cloud-oriented application development initiatives are based on the PaaS model. The significant impact of enterprises leveraging IaaS and PaaS has been in the form of services whose usage is representative of SaaS on the Cloud. Be it search (Google/Yahoo/Bing, etc.) or email (Gmail/Yahoomail/Hotmail, etc.) or social networking (Facebook/Twitter/Orkut, etc.), most users are unaware that much of their on-line activities has been supported in one form or the other by the cloud.

The cloud application deployment and consumption was modeled at three levels: the public cloud offerings from cloud vendors; the private cloud initiatives within large enterprises; and the hybrid cloud initiatives that leverage both the public cloud and the private cloud or managed services data centers.

[3] http://azure.microsoft.com

[4] http://hadoop.apache.org

The IaaS–oriented services offered abstracted (or virtualized and scalable) hardware—like compute power or storage or bandwidth. For example, as seen from its pricing tariffs webpage for 2009, Amazon[5] offered six levels of abstracted *elastic cloud compute* (EC2) server power: the "small-instance," "large-instance," "extra-large instance," "high-cpu instance," "high-cpu medium instance," or "high-cpu extra-large instance." Each of these are accompanied by appropriate RAM, storage, performance guarantees, and bandwidth support. The PaaS offerings are focused on supporting programming platforms whose runtime implicitly use's cloud services offered by their respective vendors. As of today, these highly vendor-locked PaaS technologies have been leveraged to develop new applications by many startups. Compared to IaaS offerings, applications riding on PaaS deliver better performance due to the intrinsic cloud support for the programming platform. The SaaS on Cloud offerings are focused on supporting large software package usage leveraging cloud benefits. Most users of these packages are invariably ignorant of the underlying cloud support—in fact most, if not all, do not care. Indeed, a significant degree of the features of the software package invariably reflect the support of the cloud computing platform under the hood. For example, in gmail, users hardly bother about either the storage space taken up or whether an email needs to be deleted or its storage location. Invariably these reflect the cloud underneath, where storage (most do not know on which system it is) is easily scalable or for that matter where it is stored or located.

### 2.1.3 Challenges in the Cloud

While the cloud service offerings present a simplistic view of IT in case of IaaS or a simplistic view of programming in case PaaS or a simplistic view of resources usage in case of SaaS, the underlying systems level support challenges are huge and highly complex. These stem from the need to offer a uniformly consistent and robustly simplistic view of computing while the underlying systems are highly failure-prone, heterogeneous, resource hogging, and exhibiting serious security shortcomings. As observed in Figure 2.3, the promise of the cloud seems very similar to the typical distributed systems properties that most would prefer to have. Invariably either in the IaaS or PaaS or SaaS cloud services, one is proffered features that smack of full network reliability; or having "instant" or "zero" network latency; or perhaps supporting "infinite" bandwidth; and so on. But then robust distributed systems are built while keeping mind that are these fallacies[6] that must be studiously avoided at design time as well as during implementations and deployments. Cloud computing has the ironical role of projecting this idealized view of its services while ensuring that the underlying systems are managed realistically. In fact the challenges in implementing cloud computing services are plenty: Many

[5] http://aws.amazon.com/ec2

[6] http://blogs.sun.com/jag/resource/Fallacies.html

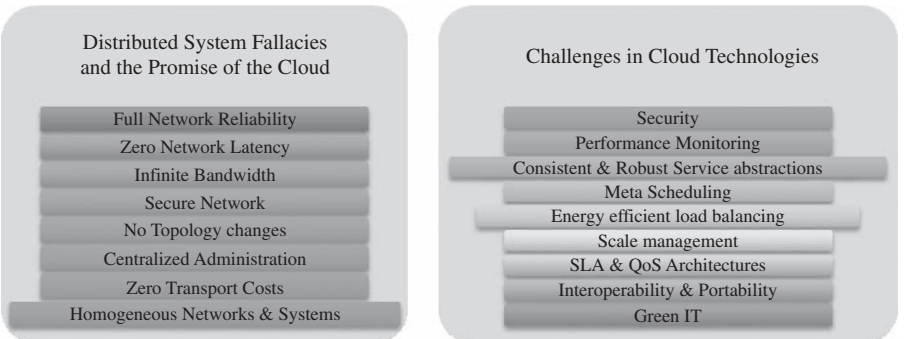



**FIGURE 2.3.** 'Under the hood' challenges of the cloud computing services implementations.

of them are listed in Figure 2.3. Prime amongst these are the challenges of security. The Cloud Security Alliance seeks to address many of these issues [8].

## 2.2 BROAD APPROACHES TO MIGRATING INTO THE CLOUD

Given that cloud computing is a "techno-business disruptive model" and is on the top of the top 10 strategic technologies to watch for 2010 according to Gartner,[7] migrating into the cloud is poised to become a large-scale effort in leveraging the cloud in several enterprises. "Cloudonomics" deals with the economic rationale for leveraging the cloud and is central to the success of cloud-based enterprise usage. At what IT costs—both short term and long term—would one want to migrate into the cloud? While all capital expenses are eliminated and only operational expenses incurred by leveraging the cloud, does this satisfy all strategic parameters for enterprise IT? Does the total cost of ownership (TCO) become significantly less as compared to that incurred when running one's own private data center? Decision-makers, IT managers, and software architects are faced with several dilemmas when planning for new Enterprise IT initiatives.

### 2.2.1 Why Migrate?

There are economic and business reasons why an enterprise application can be migrated into the cloud, and there are also a number of technological reasons. Many of these efforts come up as initiatives in adoption of cloud technologies in the enterprise, resulting in integration of enterprise applications running off the captive data centers with the new ones that have been developed on the cloud. Adoption of or integration with cloud computing services is a use case of migration.

[7] http://www.gartner.com/it/page.jsp?id 1210613

At the core, migration of an application into the cloud can happen in one of several ways: Either the application is clean and independent, so it runs as is; or perhaps some degree of code needs to be modified and adapted; or the design (and therefore the code) needs to be first migrated into the cloud computing service environment; or finally perhaps the migration results in the core architecture being migrated for a cloud computing service setting, this resulting in a new architecture being developed, along with the accompanying design and code implementation. Or perhaps while the application is migrated as is, it is the usage of the application that needs to be migrated and therefore adapted and modified. In brief, migration can happen at one of the five levels of application, code, design, architecture, and usage.

With due simplification, the migration of an enterprise application is best captured by the following:

$$P \rightarrow P'_C + P'_l \rightarrow P'_{\text{OFC}} + \text{P}'_l$$

where $P$ is the application before migration running in captive data center, $P'_C$ is the application part after migration either into a (hybrid) cloud, $P'_l$ is the part of application being run in the captive local data center, and $P'_{\text{OFC}}$ is the application part *optimized for cloud*. If an enterprise application cannot be migrated fully, it could result in some parts being run on the captive local data center while the rest are being migrated into the cloud—essentially a case of a hybrid cloud usage. However, when the entire application is migrated onto the cloud, then $P'_l$ is null. Indeed, the migration of the enterprise application $P$ can happen at the five levels of application, code, design, architecture, and usage. It can be that the $P'_C$ migration happens at any of the five levels without any $P'_l$ component. Compound this with the kind of cloud computing service offering being applied—the IaaS model or PaaS or SaaS model—and we have a variety of migration use cases that need to be thought through thoroughly by the migration architects. To capture this situation succinctly, on enumeration, we have the following migration scenario use-case numbers: For migrating into an IaaS offering, there are 30 use-case scenarios. For migrating into a PaaS offering, there are 20 use-case scenarios. For migrating into a SaaS offering, it is purely a case of migration of usage, with no accompanying enterprise application migration—like the case of migrating from an existing local ERP system to SAP already being offered on a cloud. Of course, for each of these migration use-case scenarios, detailed approaches exist while for many commonly applicable scenarios, enterprises have consolidated their migration strategy best practices. In fact, the migration industry thrives on these custom and proprietary best practices. Many of these best practices are specialized at the level of the components of an enterprise application—like migrating application servers or the enterprise databases.

***Cloudonomics.*** Invariably, migrating into the cloud is driven by economic reasons of cost cutting in both the IT capital expenses (Capex) as well as

operational expenses (Opex). There are both the short-term benefits of opportunistic migration to offset seasonal and highly variable IT loads as well as the long-term benefits to leverage the cloud. For the long-term sustained usage, as of 2009, several impediments and shortcomings of the cloud computing services need to be addressed.

At the core of the cloudonomics, as articulated in Ambrust et al. [2], is the expression of when a migration can be economically feasible or tenable. If the average costs of using an enterprise application on a cloud is substantially lower than the costs of using it in one's captive data center and if the cost of migration does not add to the burden on ROI, then the case for migration into the cloud is strong.

Apart from these costs, other factors that play a major role in the cloudonomics of migration are the licensing issues (for perhaps parts of the enterprise application), the SLA compliances, and the pricing of the cloud service offerings. Most cloud service vendors, at a broad level, have tariffs for the kind of elastic compute, the elastic storage, or the elastic bandwidth. Of course these pricing tariffs can be variable too, and therefore the cloudonomics of migration should be soundly meaningful accommodating the pricing variability.

### 2.2.2 Deciding on the Cloud Migration

In fact, several proof of concepts and prototypes of the enterprise application are experimented on the cloud to take help in making a sound decision on migrating into the cloud. Post migration, the ROI on the migration should be positive for a broad range of pricing variability. Arriving at a decision for undertaking migration demands that either the compelling factors be clearly understood or the pragmatic approach of consulting a group of experts be constituted. In the latter case, much like software estimation, one applies Wide-Band Delphi Techniques [9] to make decisions. We use the following technique: A questionnaire with several classes of key questions that impact the IT due to the migration of the enterprise application is posed to a select audience chosen for their technology and business expertise. Assume that there are $M$ such classes. Each class of questions is assigned a certain relative weightage $B_i$ in the context of the entire questionnaire. Assume that in the $M$ classes of questions, there was a class with a maximum of $N$ questions. We can then model the weightage-based decision making as $M \times \mathrm{N}$ weightage matrix as follows:

$$C_l \leq \sum_{i=1}^{M} B_i \left( \sum_{j=1}^{N} A_{ij} X_{ij} \right) \leq C_h$$

where $C_l$ is the lower weightage threshold and $C_h$ is the higher weightage threshold while $A_{ij}$ is the specific constant assigned for a question and $X_{ij}$ is the fraction between 0 and 1 that represents the degree to which that answer to the question is relevant and applicable. Since all except one class of questions do not have all $N$ questions, the corresponding has a null value. The lower

and higher thresholds are defined to rule out trivial cases of migration. A simplified variant of this method can be presented as a balanced scorecard-oriented decision making. An example of that approach to the adoption of cloud is found in Dargha [10].

## 2.3 THE SEVEN-STEP MODEL OF MIGRATION INTO A CLOUD

Typically migration initiatives into the cloud are implemented in phases or in stages. A structured and process-oriented approach to migration into a cloud has several advantages of capturing within itself the best practices of many migration projects. While migration has been a difficult and vague subject—of not much interest to the academics and left to the industry practitioners—not many efforts across the industry have been put in to consolidate what has been found to be both a top revenue earner and a long standing customer pain. After due study and practice, we share the *Seven-Step Model of Migration into the Cloud* as part of our efforts in understanding and leveraging the cloud computing service offerings in the enterprise context. In a succinct way, Figure 2.4 captures the essence of the steps in the model of migration into the cloud, while Figure 2.5 captures the iterative process of the seven-step migration into the cloud.

Cloud migration assessments comprise assessments to understand the issues involved in the specific case of migration at the application level or the code, the design, the architecture, or usage levels. In addition, migration assessments are done for the tools being used, the test cases as well as configurations, functionalities, and NFRs of the enterprise application. This results in a meaningful formulation of a comprehensive migration strategy. The first step of the iterative process of the seven-step model of migration is basically at the assessment level. Proof of concepts or prototypes for various approaches to the migration along with the leveraging of pricing parameters enables one to make appropriate assessments.

These assessments are about the cost of migration as well as about the ROI that can be achieved in the case of production version. The next process step is in isolating all systemic and environmental dependencies of the enterprise

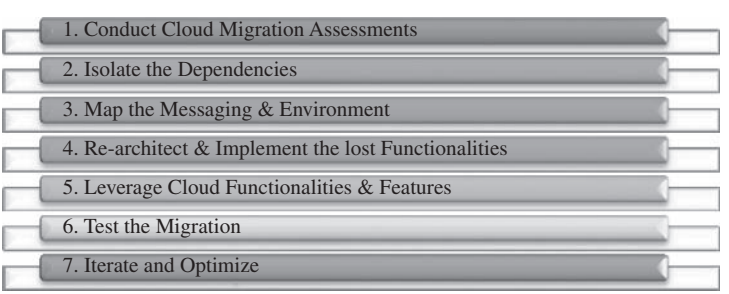



**FIGURE 2.4.** The Seven Step Model of Migration into the Cloud. (*Source:* Infosys Research.)

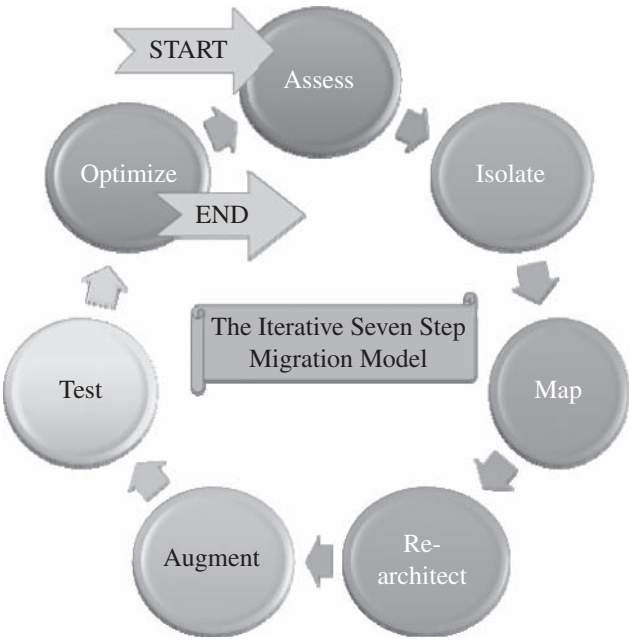



**FIGURE 2.5.** The iterative Seven step Model of Migration into the Cloud. (*Source:* Infosys Research.)

application components within the captive data center. This, in turn, yields a picture of the level of complexity of the migration. After isolation is complete, one then goes about generating the mapping constructs between what shall possibly remain in the local captive data center and what goes onto the cloud. Perhaps a substantial part of the enterprise application needs to be re-architected, redesigned, and reimplemented on the cloud. This gets in just about the functionality of the original enterprise application. Due to this migration, it is possible perhaps that some functionality is lost. In the next process step we leverage the intrinsic features of the cloud computing service to augment our enterprise application in its own small ways. Having done the augmentation, we validate and test the new form of the enterprise application with an extensive test suite that comprises testing the components of the enterprise application on the cloud as well. These test results could be positive or mixed. In the latter case, we iterate and optimize as appropriate. After several such optimizing iterations, the migration is deemed successful. Our best practices indicate that it is best to iterate through this Seven-Step Model process for optimizing and ensuring that the migration into the cloud is both robust and comprehensive. Figure 2.6 captures the typical components of the best practices accumulated in the practice of the Seven-Step Model of Migration into the Cloud. Though not comprehensive in enumeration, it is representative.

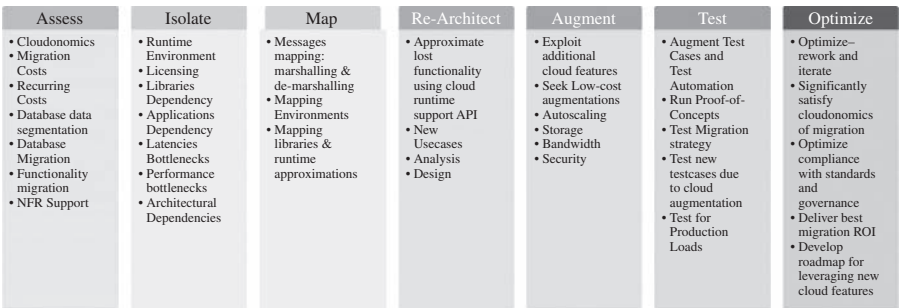



**FIGURE 2.6.** Some details of the iterative Seven Step Model of Migration into the Cloud.

Compared with the typical approach[8] to migration into the Amazon AWS, our Seven-step model is more generic, versatile, and comprehensive. The typical migration into the Amazon AWS is a phased over several steps. It is about six steps as discussed in several white papers in the Amazon website and is as follows: The first phase is the cloud migration assessment phase wherein dependencies are isolated and strategies worked out to handle these dependencies. The next phase is in trying out proof of concepts to build a reference migration architecture. The third phase is the data migration phase wherein database data segmentation and cleansing is completed. This phase also tries to leverage the various cloud storage options as best suited. The fourth phase comprises the application migration wherein either a "forklift strategy" of migrating the key enterprise application along with its dependencies (other applications) into the cloud is pursued. Or perhaps using the "hybrid migration strategy," the critical parts of the enterprise application are retained in the local captive data center while noncritical parts are moved into the cloud. The fifth phase comprises leveraging the various Amazon AWS features like elasticity, autoscaling, cloud storage, and so on. Finally in the sixth phase, the migration is optimized for the cloud. These phases are representative of how typical IT staff would like to migrate an enterprise application without touching its innards but only perhaps at the level of configurations—this perfectly matches with the typical IaaS cloud computing offerings. However, this is just a subset of our Seven-step Migration Model and is very specific and proprietary to Amazon cloud offering.

### 2.3.1 Migration Risks and Mitigation

The biggest challenge to any cloud migration project is how effectively the migration risks are identified and mitigated. In the Seven-Step Model of Migration into the Cloud, the process step of testing and validating includes

[8] http://aws.amazon.com

efforts to identify the key migration risks. In the optimization step, we address various approaches to mitigate the identified migration risks.

Migration risks for migrating into the cloud fall under two broad categories: the general migration risks and the security-related migration risks. In the former we address several issues including performance monitoring and tuning—essentially identifying all possible production level deviants; the business continuity and disaster recovery in the world of cloud computing service; the compliance with standards and governance issues; the IP and licensing issues; the quality of service (QoS) parameters as well as the corresponding SLAs committed to; the ownership, transfer, and storage of data in the application; the portability and interoperability issues which could help mitigate potential vendor lock-ins; the issues that result in trivializing and noncomprehending the complexities of migration that results in migration failure and loss of senior management's business confidence in these efforts.

On the security front, the cloud migration risks are plenty—as addressed in the guideline document published by the Cloud Security Alliance [8]. Issues include security at various levels of the enterprise application as applicable on the cloud in addition to issues of trust and issues of privacy. There are several legal compliances that a migration strategy and implementation has to fulfill, including obtaining the right execution logs as well as retaining the rights to all audit trails at a detailed level—which currently may not be fully available. On matters of governance, there are several shortcomings in the current cloud computing service vendors. Matters of multi-tenancy and the impact of IT data leakage in the cloud computing environments is acknowledged; however, the robustness of the solutions to prevent it is not fully validated. Key aspects of vulnerability management and incident responses quality are yet to be supported in a substantial way by the cloud service vendors. Finally there are issues of consistent identity management as well. These and several of the issues are discussed in Section 2.1. Issues and challenges listed in Figure 2.3 continue to be the persistent research and engineering challenges in coming up with appropriate cloud computing implementations.

## 2.4 CONCLUSIONS

While migrating into a cloud has a lot of challenges, many migration projects fail to fully comprehend the issues at stake—with the key sponsors and management either trivializing it or committing to migrating a piece of code and/or data into the cloud. There are significant opportunities and success factors for a well-designed cloud migration strategy leveraging the Seven-Step Model of Migration into the Cloud. Primary amongst them is a comprehensive understanding of the cloudonomics of the migration as well as the underlying technical challenges.

Developing the best practices in migrating to the cloud is unique to every class of enterprise applications and unique to every corporate practice group. Some of the key best practices include designing the migration as well as the

new application architecture or design or code for failures when in reality most assume that cloud computing service environments are failsafe. In fact most cloud computing data centers use commodity hardware and are routinely prone to failure. Approaches not reflecting this reality results in several performance penalties. Another best practice is the application and enforcement of loose-coupling between various parts of the target enterprise application. A key best practice has to been to build security at every level and layer of the migration. Finally the most important of the best practices has been to fully leverage the cloud computing service features while not being constrained by the baggage carried by the enterprise application in its traditional deployment in the captive data centers. Migrating into a cloud is a nontrivial activity. It is challenging given the complexity of comprehending the various factors involved for a successful migration. The proposed Seven-Step Model of Migration into the cloud helps structure and organize one's efforts in putting together a plan of action and process to successful complete the migration without problems. Of course best practices are accumulated through migration project executions, and the seven-step model of migration is reflective of this.

## ACKNOWLEDGMENTS

The author sincerely thanks S. V. Subrahmanya as well as the members of E-Com Research Labs, E&R, and Infosys for all the help and support.

## REFERENCES

1. J. Broberg, S. Venugopal, and R. Buyya, Market oriented Grids and utility computing: The state of the art and future directions, *Journal of Grid Computing*, **6**(3):255 276, 2008.
2. M. Ambrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, Above the Clouds: A Berkeley View of Cloud Computing, *UC Berkeley RAD Systems Labs*, Feb 2009.
3. G. Reese, *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*, O'Reilly, April 2007.
4. R. Buyya, C. S. Yeo, and S. Venugopal, Market oriented cloud computing: Vision, hype, and reality for delivering IT Services as Computing Utilities, in *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications*, September 25 27, 2008, pp. 5 13 Dalian, China.
5. Cloud Definitions: NIST, Gartner, Forrester in Cloud enterprise, August 2009. (Also available at: http://cloudenterprise.info/2009/08/04/cloud definitions nist gartner forrester/)
6. T. Velte, A. Velte, and R. Elsenpeter, *Cloud Computing, A Practical Approach*, McGraw Hill Computing, New York, 2009.

7. C. Vecchiola, S. Pandey, and R. Buyya, High performance cloud computing: A view of scientific applications, in *Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms and Networks (I SPAN 2009, IEEE CS Press, USA)*, Kaohsiung, Taiwan, December 14 16, 2009.
8. Security Guidance for Critical Areas of Focus in Cloud Computing, The Cloud Security Alliance April/November 2009. (Also available at: http://www.cloudsecurityalliance.org/csaguide.pdf )
9. A. Stellman and J. Greene, *Applied Software Project Management*, O'Reilly Media.
10. R. Dargha, Cloud Computing Key Considerations for Adoption, *Infosys Technol ogies Whitepaper*. 2009. (Also available at http://www.infosys.com/cloud computing/white papers/cloud computing.pdf)
11. C. Keene, I. Poddar, J. Nicke, and U. Budink, Cloud Quick Start A Roadmap for Adopting Cloud Computing IBM, WaveMaker and Rightscale, *WaveMaker Software Inc. Whitepaper*, November 2009. (Also available at: http://www.wavemaker.com/ibm quickstart.pdf)
12. A. Dubey, J. Mohiuddin, and A. Baijal, The Emerging Platform Wars in Enterprise Software, *A McKinsey & Company Whitepaper*, April 2008. (Available at http://www.mckinsey.com/clientservice/hightech/pdfs/Emerging_Platform_Wars.pdf )
13. C. Vecchiola, X. Chu, and R. Buyya, Aneka: A software platform for .NET based Cloud computing, *High Performance & Large Scale Computing, Advances in Parallel Computing*, W. Gentzsch, L. Grandinetti, and G. Joubert (eds.), IOS Press, 2009.
14. SMART The Service Oriented Migration and Reuse Technique. *Software Engineering Institute Tech Report*: CMU/SEI 2005 TN 029.

# CHAPTER 3

# ENRICHING THE 'INTEGRATION AS A SERVICE' PARADIGM FOR THE CLOUD ERA

PETHURU RAJ

## 3.1 AN INTRODUCTION

The trend-setting cloud paradigm actually represents the cool conglomeration of a number of proven and promising Web and enterprise technologies. Though the cloud idea is not conceptually new, practically it has brought in myriad tectonic shifts for the whole information and communication technology (ICT) industry. The cloud concepts have progressively and perceptibly impacted the IT and business domains on several critical aspects. The cloud computing has brought in series of novelty-packed deployment, delivery, consumption and pricing models whereas the service orientation prescribes a much simpler application design mechanism. The noteworthy contribution of the much-discoursed and deliberated cloud computing is the faster realization and proliferation of dynamic, converged, adaptive, on-demand, and online compute infrastructures, which are the key requirement for the future IT. The delightful distinctions here are that clouds guarantee most of the non-function requirements (Quality of Service (QoS) attributes) such as availability, high performance, on-demand scalability/elasticity, affordability, global-scale accessibility and usability, energy efficiency etc.

Having understood the exceptional properties of cloud infrastructures (hereafter will be described as just clouds), most of the global enterprises (small, medium and even large) are steadily moving their IT offerings such as business services and applications to clouds. This transition will facilitate a

*Cloud Computing: Principles and Paradigms*, Edited by Rajkumar Buyya, James Broberg and Andrzej Goscinski 

higher and deeper reach and richness in application delivery and consumability. Product vendors having found that the cloud style is a unique proposition are moving their platforms, databases, and middleware to clouds. Cloud Infrastructure providers are establishing cloud centers to host a variety of ICT services and platforms of worldwide individuals, innovators, and institutions. Cloud service providers (CSPs) are very aggressive in experimenting and embracing the cool cloud ideas and today every business and technical services are being hosted in clouds to be delivered to global customers, clients and consumers over the Internet communication infrastructure. For example, security as a service (SaaS) is a prominent cloud-hosted security service that can be subscribed by a spectrum of users of any connected device and the users just pay for the exact amount or time of usage. In a nutshell, on-premise and local applications are becoming online, remote, hosted, on-demand and off-premise applications. With the unprecedented advertisement, articulation and adoption of cloud concepts, the cloud movement is picking up fast as per leading market research reports. Besides the modernization of legacy applications and positing the updated and upgraded in clouds, fresh applications are being implemented and deployed on clouds to be delivered to millions of global users simultaneously affordably. It is hence clear that a number of strategic and significant movements happen silently in the hot field of cloud computing.

All these portend and predict that there is a new dimension to the integration scenario. Hitherto enterprise data and applications are being linked up via one or more standards-compliant integration platforms, brokers, engines, and containers within the corporate intranet. Business-to-business (B2B) integration is being attended via special data formats, message templates, and networks and even via the Internet. Enterprises consistently expand their operations to several parts of the world as they establish special partnerships with their partners or buy other companies in different geographies for enhancing the product and service portfolios. Business applications are finding their new residence in clouds. However most of the confidential and corporate data are still being maintained in enterprise servers for security reasons. The integration task gets just bigger with the addition of the cloud space and the integration complexity is getting murkier. Hence it is logical to take the integration middleware to clouds to simplify and streamline the enterprise-to-enterprise (E2E), enterprise-to-cloud (E2C) and cloud-to-cloud (C2C) integration.

In this chapter, we want you to walk through how cloud paradigm impacts the integration scene. That is, how cloud applications are being integrated with both enterprise as well as other cloud applications. Similarly how applications hosted in distributed clouds can find on another and share their functionality is also being given its share of attention. We have visualised and written about a few important integration scenarios wherein cloud-based middleware exceptionally contributes for simplifying and streamlining the increasingly complex integration goal. It is all about how integration becomes a cloud service.

## 3.2 THE ONSET OF KNOWLEDGE ERA

Having started its innings as the greatest business-enabler, today IT is tending towards the significant factor and the facilitator of every aspect of human lives. Path-breaking and people-centric technologies (miniaturization, virtualization, federation, composition, collaboration, etc.) are emerging and are being experimented, expounded, and established in order to empower the professional and the personal IT to be smart, simple, supple and sensitive towards users' situational needs and to significantly enhance peoples' comfort, care, convenience and choice. Novel computing paradigms (grid, on-demand, service, cloud, etc.) erupt and evolve relentlessly to be greatly and gracefully impactful and insightful. In the monolithic mainframe era, one centralized and large system performed millions of operations to respond to thousands of users (one-to-many), today everyone has his own compute machine (one-to-one), and tomorrow a multitude of smart objects and electronic devices (nomadic, wearable, portable, implantable etc.) will seamlessly and spontaneously co-exist, corroborate, correlate, and coordinate with one another dynamically with dexterity to understand one or more users' needs, conceive, construct, and deliver them at right time at right place (many-to-one). Anytime anywhere computing tends towards everywhere, every time and everything computing.

Ambient intelligence (AmI) is the newest buzzword today with ambient sensing, networking, perception, decision-making and actuation technologies. Multimedia and multimodal technologies are flourishing in order to be make human interaction more friendly and fruitful. Dynamic, virtualized and autonomic infrastructures, flexible, integrated and lean processes, constructive and contributive building-blocks (service, model, composite, agent, aspect etc.), slim and sleek devices and appliances, smart objects empowered by invisible tags and stickers, natural interfaces, ad-hoc and situational networking capabilities all combine adaptively together to accomplish the grandiose goals of the forthcoming ambient intelligence days and decades. In short, IT-sponsored and splurged smartness in every facet of our living in this world is the vision. Software engineering is on the right track with the maturity of service orientation concepts and software as a service (SaaS) model. Clouds chip in mightily in realizing the much-acclaimed knowledge era. Technologies form a dynamic cluster in real-time in order to contribute immensely and immeasurably for all the existing, evolving and exotic expectations of people.

## 3.3 THE EVOLUTION OF SaaS

SaaS paradigm is on fast track due to its innate powers and potentials. Executives, entrepreneurs, and end-users are ecstatic about the tactic as well as strategic success of the emerging and evolving SaaS paradigm. A number of positive and progressive developments started to grip this model. Newer resources and activities are being consistently readied to be delivered as a

service. Experts and evangelists are in unison that cloud is to rock the total IT community as the best possible infrastructural solution for effective service delivery. There are several ways clouds can be leveraged inspiringly and incredibly for diverse IT problems. Today there is a small list of services being delivered via the clouds and in future, many more critical applications will be deployed and consumed. In short, clouds are set to decimate all kinds of IT inflexibility and dawn a growing array of innovations to prepare the present day IT for sustainable prosperity.

**IT as a Service (ITaaS)** is the most recent and efficient delivery method in the decisive IT landscape. With the meteoric and mesmerizing rise of the service orientation principles, every single IT resource, activity and infrastructure is being viewed and visualized as a service that sets the tone for the grand unfolding of the dreamt service era. These days, systems are designed and engineered as elegant collections of enterprising and evolving services. Infrastructures are service-enabled to be actively participative and collaborative. In the same tenor, the much-maligned delivery aspect too has gone through several transformations and today the whole world has solidly settled for the green paradigm 'IT as a service (ITaaS)'. This is accentuated due to the pervasive Internet. Also we are bombarded with innumerable implementation technologies and methodologies. Clouds, as indicated above, is the most visible and viable infrastructure for realizing ITaaS. Another influential and impressive factor is the maturity obtained in the consumption-based metering and billing capability. HP even proclaims this evolving trend as 'everything as a service'.

**Integration as a service (IaaS)** is the budding and distinctive capability of clouds in fulfilling the business integration requirements. Increasingly business applications are deployed in clouds to reap the business and technical benefits. On the other hand, there are still innumerable applications and data sources locally stationed and sustained primarily due to the security reason. The question here is how to create a seamless connectivity between those hosted and on-premise applications to empower them to work together. IaaS overcomes these challenges by smartly utilizing the time-tested business-to-business (B2B) integration technology as the value-added bridge between SaaS solutions and in-house business applications.

B2B systems are capable of driving this new on-demand integration model because they are traditionally employed to automate business processes between manufacturers and their trading partners. That means they provide application-to-application connectivity along with the functionality that is very crucial for linking internal and external software securely. Unlike the conventional EAI solutions designed only for internal data sharing, B2B platforms have the ability to encrypt files for safe passage across the public network, manage large data volumes, transfer batch files, convert disparate file formats, and guarantee data delivery across multiple enterprises. IaaS just imitates this established communication and collaboration model to create reliable and durable linkage for ensuring smooth data passage between traditional and cloud systems over the Web infrastructure.

The use of hub & spoke (H&S) architecture further simplifies the implementation and avoids placing an excessive processing burden on the customer sides. The hub is installed at the SaaS provider's cloud center to do the heavy lifting such as reformatting files. A spoke unit at each user site typically acts as basic data transfer utility. With these pieces in place, SaaS providers can offer integration services under the same subscription / usage-based pricing model as their core offerings. This trend of moving all kinds of common and centralised services to clouds is gaining momentum these days. As resources are getting distributed and decentralised, linking and leveraging them for multiple purposes need a multifaceted infrastructure. Clouds, being the Web-based infrastructures are the best fit for hosting scores of unified and utility-like platforms to take care of all sorts of brokering needs among connected and distributed ICT systems.

1. The Web is the largest **digital information superhighway**
2. The Web is the largest repository of all kinds of resources such as **web pages, applications comprising enterprise components, business services, beans, POJOs, blogs, corporate data**, etc.
3. The Web is turning out to be the open, cost-effective and generic **business execution platform** (E-commerce, business, auction, etc. happen in the web for global users) comprising a wider variety of containers, adaptors, drivers, connectors, etc.
4. The Web is the global-scale **communication infrastructure (VoIP, Video conferencing, IP TV etc,)**
5. The Web is the next-generation **discovery, Connectivity, and integration middleware**

Thus the unprecedented absorption and adoption of the Internet is the key driver for the continued success of the cloud computing.

## 3.4 THE CHALLENGES OF SaaS PARADIGM

As with any new technology, SaaS and cloud concepts too suffer a number of limitations. These technologies are being diligently examined for specific situations and scenarios. The prickling and tricky issues in different layers and levels are being looked into. The overall views are listed out below. Loss or lack of the following features deters the massive adoption of clouds

1. Controllability
2. Visibility & flexibility
3. Security and Privacy
4. High Performance and Availability
5. Integration and Composition
6. Standards

A number of approaches are being investigated for resolving the identified issues and flaws. Private cloud, hybrid and the latest community cloud are being prescribed as the solution for most of these inefficiencies and deficiencies. As rightly pointed out by someone in his weblogs, still there are miles to go. There are several companies focusing on this issue. Boomi (http://www.dell.com/) is one among them. This company has published several well-written white papers elaborating the issues confronting those enterprises thinking and trying to embrace the third-party public clouds for hosting their services and applications.

***Integration Conundrum.*** While SaaS applications offer outstanding value in terms of features and functionalities relative to cost, they have introduced several challenges specific to integration. The first issue is that the majority of SaaS applications are point solutions and service one line of business. As a result, companies without a method of synchronizing data between multiple lines of businesses are at a serious disadvantage in terms of maintaining accurate data, forecasting, and automating key business processes. Real-time data and functionality sharing is an essential ingredient for clouds.

***APIs are Insufficient.*** Many SaaS providers have responded to the integration challenge by developing application programming interfaces (APIs). Unfortunately, accessing and managing data via an API requires a significant amount of coding as well as maintenance due to frequent API modifications and updates. Furthermore, despite the advent of web services, there is little to no standardization or consensus on the structure or format of SaaS APIs. As a result, the IT department expends an excess amount of time and resources developing and maintaining a unique method of communication for the API of each SaaS application deployed within the organization.

***Data Transmission Security.*** SaaS providers go to great length to ensure that customer data is secure within the hosted environment. However, the need to transfer data from on-premise systems or applications behind the firewall with SaaS applications hosted outside of the client's data center poses new challenges that need to be addressed by the integration solution of choice. It is critical that the integration solution is able to synchronize data bi-directionally from SaaS to on-premise without opening the firewall. Best-of-breed integration providers can offer the ability to do so by utilizing the same security as when a user is manually typing data into a web browser behind the firewall.

For any relocated application to provide the promised value for businesses and users, the minimum requirement is the interoperability between SaaS applications and on-premise enterprise packages. As SaaS applications were not initially designed keeping the interoperability requirement in mind, the integration process has become a little tougher assignment. There are other obstructions and barriers that come in the way of routing messages between on-demand applications and on-premise resources. Message, data and protocol

translations have to happen at end-points or at the middleware layer in order to decimate the blockade that is prohibiting the spontaneous sharing and purposeful collaboration among the participants. As applications and data are diverse, distributed and decentralized, versatile integration technologies and methods are very essential to smoothen the integration problem. Reflective middleware is an important necessity for enterprise-wide, real-time and synchronized view of information to benefit executives, decision-makers as well as users tactically as well as strategically. Data integrity, confidentiality, quality and value have to be preserved as services and applications are interlinked and saddled to work together.

***The Impacts of Clouds [1, 2].*** On the infrastructural front, in the recent past, the clouds have arrived onto the scene powerfully and have extended the horizon and the boundary of business applications, events and data. That is, business applications, development platforms etc. are getting moved to elastic, online and on-demand cloud infrastructures. Precisely speaking, increasingly for business, technical, financial and green reasons, applications and services are being readied and relocated to highly scalable and available clouds. The immediate implication and impact is that integration methodologies and middleware solutions have to take clouds too into account for establishing extended and integrated processes and views. Thus there is a clarion call for adaptive integration engines that seamlessly and spontaneously connect enterprise applications with cloud applications. Integration is being stretched further to the level of the expanding Internet and this is really a litmus test for system architects and integrators.

The perpetual integration puzzle has to be solved meticulously for the originally visualised success of SaaS style. Interoperability between SaaS and non-SaaS solutions remains the lead demand as integration leads to business-aware and people-centric composite systems and services. Boundaryless flow of information is necessary for enterprises to strategize to achieve greater successes, value and for delivering on the elusive goal of customer delight. Integration has been a big challenge for growing business behemoths, fortune 500 companies, and system integrators. Now with the availability, affordability and suitability of the cloud-sponsored and the state-of-the-art infrastructures for application deployment and delivery, the integration's scope, size, and scale is expanding and this beneficial extension however have put integration architects, specialists and consultants in deeper trouble.

## 3.5 APPROACHING THE SaaS INTEGRATION ENIGMA

Integration as a Service (IaaS) is all about the migration of the functionality of a typical enterprise application integration (EAI) hub / enterprise service bus (ESB) into the cloud for providing for smooth data transport between any enterprise and SaaS applications. Users subscribe to IaaS as they would do for

any other SaaS application. Cloud middleware is the next logical evolution of traditional middleware solutions. That is, cloud middleware will be made available as a service. Due to varying integration requirements and scenarios, there are a number of middleware technologies and products such as JMS-compliant message queues and integration backbones such as EAI, ESB, EII, EDB, CEP, etc. For performance sake, clusters, fabrics, grids, and federations of hubs, brokers, and buses are being leveraged.

For service integration, it is enterprise service bus (ESB) and for data integration, it is enterprise data bus (EDB). Besides there are message oriented middleware (MOM) and message brokers for integrating decoupled applications through message passing and pick up. Events are coming up fast and there are complex event processing (CEP) engines that receive a stream of diverse events from diverse sources, process them at real-time to extract and figure out the encapsulated knowledge, and accordingly select and activate one or more target applications thereby a kind of lighter connectivity and integration occurs between the initiating and the destination applications. Service orchestration and choreography enables process integration. Service interaction through ESB integrates loosely coupled systems whereas CEP connects decoupled systems. Besides data services, mashups perform and provide composite services, data and views. Thus at every layer or tier in the enterprise IT stack, there are competent integration modules and guidelines brewing for bringing up the much-anticipated dynamic integration.

With the unprecedented rise in cloud usage, all these integration software are bound to move to clouds. Amazon's Simple Queue Service (SQS) provides a straightforward way for applications to exchange messages via queues in the cloud. SQS is a classic example for understanding what happens when a familiar on-premise service is recast as a cloud service. However there are some problems with this. Because SQS replicates messages across multiple queues, an application reading from a queue is not guaranteed to see all messages from all queues on a particular read request. SQS also doesn't promise in-order and exactly-once delivery. These simplifications let Amazon make SQS more scalable, but they also mean that developers must use SQS differently from an on-premise message queuing technology.

Cloud infrastructure is not very useful without SaaS applications that run on top of them, and SaaS applications are not very valuable without access to the critical corporate data that is typically locked away in various corporate systems. So, for cloud applications to offer maximum value to their users, they need to provide a simple mechanism to import or load external data, export or replicate their data for reporting or analysis purposes, and finally keep their data synchronized with on-premise applications. That brings out the importance of SaaS integration subject.

As per one of the David Linthicum's white papers, approaching SaaS-to-enterprise integration is really a matter of making informed and intelligent choices. Choices are mainly around the integration approaches to leverage architectural patterns, the location of the integration engine, and, finally the

enabling technology. The unprecedented growth of SaaS means that more and more software components are migrated and made to reside in off-premise SaaS platforms. Hence the need for integration between remote cloud platforms with on-premise enterprise platforms, wherein the customer and corporate data are stored for ensuring unbreakable, impeccable and impenetrable security, has caught the serious and sincere attention and imagination of product vendors and SaaS providers.

***Why SaaS Integration is hard?.*** As indicated in the white paper, there is a mid-sized paper company that recently became a Salesforce.com CRM customer. The company currently leverages an on-premise custom system that uses an Oracle database to track inventory and sales. The use of the Salesforce.com system provides the company with a significant value in terms of customer and sales management. However, the information that persists within the Salesforce.com system is somewhat redundant with the information stored within the on-premise legacy system (e.g., customer data). Thus the "as is" state is in a fuzzy state and suffers from all kinds of costly inefficiencies including the need to enter and maintain data in two different locations, which ultimately costs more for the company. Another irritation is the loss of data quality which is endemic when considering this kind of dual operation. This includes data integrity issues, which are a natural phenomenon when data is being updated using different procedures, and there is no active synchronization between the SaaS and on-premise systems.

Having understood and defined the "to be" state, data synchronization technology is proposed as the best fit between the source, meaning Salesforce.com, and the target, meaning the existing legacy system that leverages Oracle. This technology is able to provide automatic mediation of the differences between the two systems, including application semantics, security, interfaces, protocols and native data formats. The end result is that information within the SaaS-delivered systems and the legacy systems are completely and compactly synchronized meaning that data entered into the CRM system would also exist in the legacy systems and vice versa, along with other operational data such as inventory, items sold, etc. The "to be" state thereby removes data quality and integrity issues fully. This directly and indirectly paves the way for saving thousands of dollars a month and producing a quick ROI from the integration technology that is studied and leveraged.

Integration has been the prominent subject of study and research by academic students and scholars for years as integration brings a sense of order to the chaos and mess created by heterogeneous systems, networks, and services. Integration technologies, tools, tips, best practices, guidelines, metrics, patterns, and platforms are varied and vast. Integration is not easier either to implement as successful untangling from the knotty situation is a big issue. The web of application and data silos really makes the integration task difficult and hence choosing a best-in class scheme for flexible and futuristic integration is insisted very frequently. First of all, we need to gain the insights about the

special traits and tenets of SaaS applications in order to arrive at a suitable integration route. The constraining attributes of SaaS applications are

- Dynamic nature of the SaaS interfaces that constantly change
- Dynamic nature of the metadata native to a SaaS provider such as Salesforce.com
- Managing assets that exist outside of the firewall
- Massive amounts of information that need to move between SaaS and on-premise systems daily and the need to maintain data quality and integrity.

As SaaS are being deposited in cloud infrastructures vigorously, we need to ponder about the obstructions being imposed by clouds and prescribe proven solutions. If we face difficulty with local integration, then the cloud integration is bound to be more complicated. The most probable reasons are

- New integration scenarios
- Access to the cloud may be limited
- Dynamic resources
- Performance

***Limited Access.*** Access to cloud resources (SaaS, PaaS, and the infrastructures) is more limited than local applications. Accessing local applications is quite simple and faster. Imbedding integration points in local as well as custom applications is easier. Even with the commercial applications, it is always possible to slip in database-triggers to raise events and provide hooks for integration access. Once applications move to the cloud, custom applications must be designed to support integration because there is no longer that low-level of access. Enterprises putting their applications in the cloud or those subscribers of cloud-based business services are dependent on the vendor to provide the integration hooks and APIs. For example, the SalesForce.com web services API does not support transactions against multiple records, which means integration code has to handle that logic. For PaaS, the platform might support integration for applications on the platform. However platform-to-platform integration is still an open question. There is an agreement that a limited set of APIs will improve the situation to an extent. But those APIs must be able to handle the integration required. Applications and data can be moved to public clouds but the application providers and data owners lose the much-needed controllability and flexibility, Most of the third-party cloud providers do not submit their infrastructures for third-party audit. Visibility is another vital factor lost out due to this transition.

***Dynamic Resources.*** Cloud resources are virtualized and service-oriented. That is, everything is expressed and exposed as a service. Due to the dynamism factor that is sweeping the whole could ecosystem, application versioning and

infrastructural changes are liable for dynamic changes. These would clearly impact the integration model. That is, the tightly coupled integration fails and falters at cloud. It is clear that the low-level interfaces ought to follow the Representational State Transfer (REST) route, which is a simple architectural style and subscribes to the standard methods of the Http protocol.

***Performance.*** Clouds support application scalability and resource elasticity. However the network distances between elements in the cloud are no longer under our control. Bandwidth is not the limiting factor in most integration scenarios but the round trip latency is an issue not to be sidestepped. Because of the latency aggravation, the cloud integration performance is bound to slow down.

## 3.6 NEW INTEGRATION SCENARIOS

Before the cloud model, we had to stitch and tie local systems together. With the shift to a cloud model is on the anvil, we now have to connect local applications to the cloud, and we also have to connect cloud applications to each other, which add new permutations to the complex integration channel matrix. It is unlikely that everything will move to a cloud model all at once, so even the simplest scenarios require some form of local / remote integration. It is also likely that we will have applications that *never* leave the building, due to regulatory constraints like HIPPA, GLBA, and general security issues. All of this means integration must criss-cross firewalls somewhere.

***Cloud Integration Scenarios.*** We have identified three major integration scenarios as discussed below.

***Within a Public Cloud (figure 3.1).*** Two different applications are hosted in a cloud. The role of the cloud integration middleware (say cloud-based ESB or internet service bus (ISB)) is to seamlessly enable these applications to talk to each other. The possible sub-scenarios include these applications can be owned


**FIGURE 3.1.** Within a Public Cloud.

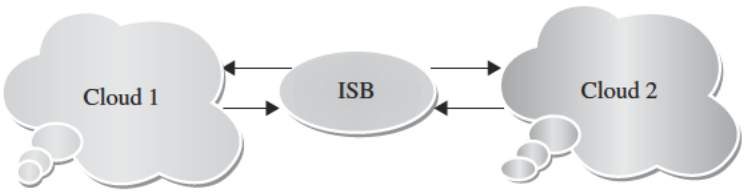



**FIGURE 3.2.** Across Homogeneous Clouds.

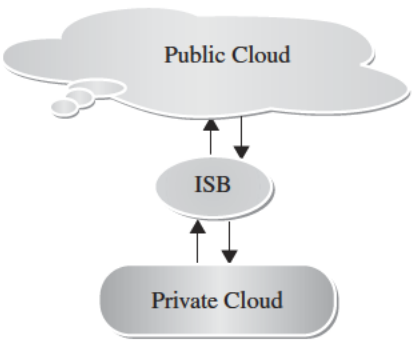



**FIGURE 3.3.** Across Heterogeneous Clouds.

by two different companies. They may live in a single physical server but run on different virtual machines.

***Homogeneous Clouds (figure 3.2).*** The applications to be integrated are posited in two geographically separated cloud infrastructures. The integration middleware can be in cloud 1 or 2 or in a separate cloud.

There is a need for data and protocol transformation and they get done by the ISB. The approach is more or less compatible to enterprise application integration procedure.

***Heterogeneous Clouds (figure 3.3).*** One application is in public cloud and the other application is private cloud.

As described above, this is the currently dominating scene for cloud integration. That is, businesses are subscribing to popular on-demand enterprise packages from established providers such as Salesforce.com and Ramco Systems (http://www.ramco.com/)'s customer relationship management (CRM), NetSuite's (http://www.netsuite.com) enterprise resource planning (ERP), etc. The first two scenarios will become prevalent once there are several commercial clouds and cloud services become pervasive. Then service integration and composition domains will become an important and incredible factor for global computing.

## 3.7 THE INTEGRATION METHODOLOGIES

Excluding the custom integration through hand-coding, there are three types for cloud integration

1. **Traditional Enterprise Integration Tools can be empowered with special connectors to access Cloud-located Applications**—This is the most likely approach for IT organizations, which have already invested a lot in integration suite for their application integration needs. With a persistent rise in the necessity towards accessing and integrating cloud applications, special drivers, connectors and adapters are being built and incorporated on the existing integration platforms to enable bidirectional connectivity with the participating cloud services. As indicated earlier, there are several popular and pioneering enterprise integration methods and platforms such as EAI/ESB, which are accordingly empowered, configured and customized in order to access and leverage the growing array of cloud applications too. For attaining an enhanced performance, integration appliances are very hot in the market.
2. **Traditional Enterprise Integration Tools are hosted in the Cloud**—This approach is similar to the first option except that the integration software suite is now hosted in any third-party cloud infrastructures so that the enterprise does not worry about procuring and managing the hardware or installing the integration software. This is a good fit for IT organizations that outsource the integration projects to IT service organizations and systems integrators, who have the skills and resources to create and deliver integrated systems. The IT divisions of business enterprises need not worry about the upfront investment of high-end computer machines, integration packages, and their maintenance with this approach. Similarly system integrators can just focus on their core competencies of designing, developing, testing, and deploying integrated systems. It is a good fit for cloud-to-cloud (C2C) integration, but requires a secure VPN tunnel to access on-premise corporate data. An example of a hosted integration technology is Informatica PowerCenter Cloud Edition on Amazon EC2.
3. **Integration-as-a-Service (IaaS) or On-Demand Integration Offerings**—These are SaaS applications that are designed to deliver the integration service securely over the Internet and are able to integrate cloud applications with the on-premise systems, cloud-to-cloud applications. Even on-premise systems can be integrated with other on-premise applications via this integration service. This approach is a good fit for companies who insist about the ease of use, ease of maintenance, time to deployment, and are on a tight budget. It is appealing to small and mid-sized companies, as well as large enterprises with departmental application deployments. It is also a good fit for companies who plan to use their

SaaS administrator or business analyst as the primary resource for managing and maintaining their integration work. A good example is **Informatica On-Demand Integration Services**.

In a nutshell, the integration requirements can be realised using any one of the following methods and middleware products.

1. Hosted and extended ESB (Internet service bus / cloud integration bus)
2. Online Message Queues, Brokers and Hubs
3. Wizard and configuration-based integration platforms (Niche integration solutions)
4. Integration Service Portfolio Approach
5. Appliance-based Integration (Standalone or Hosted)

With the emergence of the cloud space, the integration scope grows further and hence people are looking out for robust and resilient solutions and services that would speed up and simplify the whole process of integration.

***Characteristics of Integration Solutions and Products.*** The key attributes of integration platforms and backbones gleaned and gained from integration projects experience are connectivity, semantic mediation, Data mediation, integrity, security, governance etc

- **Connectivity** refers to the ability of the integration engine to engage with both the source and target systems using available native interfaces. This means leveraging the interface that each provides, which could vary from standards-based interfaces, such as Web services, to older and proprietary interfaces. Systems that are getting connected are very much responsible for the externalization of the correct information and the internalization of information once processed by the integration engine.
- **Semantic Mediation** refers to the ability to account for the differences between application semantics between two or more systems. Semantics means how information gets understood, interpreted and represented within information systems. When two different and distributed systems are linked, the differences between their own yet distinct semantics have to be covered.
- **Data Mediation** converts data from a source data format into destination data format. Coupled with semantic mediation, data mediation or data transformation is the process of converting data from one native format on the source system, to another data format for the target system.
- **Data Migration** is the process of transferring data between storage types, formats, or systems. Data migration means that the data in the old system is mapped to the new systems, typically leveraging data extraction and data loading technologies.

- **Data Security** means the ability to insure that information extracted from the source systems has to securely be placed into target systems. The integration method must leverage the native security systems of the source and target systems, mediate the differences, and provide the ability to transport the information safely between the connected systems.
- **Data Integrity** means data is complete and consistent. Thus, integrity has to be guaranteed when data is getting mapped and maintained during integration operations, such as data synchronization between on-premise and SaaS-based systems.
- **Governance** refers to the processes and technologies that surround a system or systems, which control how those systems are accessed and leveraged. Within the integration perspective, governance is about managing changes to core information resources, including data semantics, structure, and interfaces.

These are the prominent qualities carefully and critically analyzed for when deciding the cloud / SaaS integration providers.

***Data Integration Engineering Lifecycle.*** As business data are still stored and sustained in local and on-premise server and storage machines, it is imperative for a lean data integration lifecycle. The pivotal phases, as per Mr. David Linthicum, a world-renowned integration expert, are understanding, definition, design, implementation, and testing.

1. **Understanding** the existing problem domain means defining the metadata that is native within the source system (say Salesforce.com) and the target system (say an on-premise inventory system). By doing this, there is a complete semantic understanding of both source and target systems. If there are more systems for integration, the same practice has to be enacted.
2. **Definition** refers to the process of taking the information culled during the previous step and defining it at a high level including what the information represents, ownership, and physical attributes. This contributes a better perceptive of the data being dealt with beyond the simple metadata. This insures that the integration process proceeds in the right direction.
3. **Design** the integration solution around the movement of data from one point to another accounting for the differences in the semantics using the underlying data transformation and mediation layer by mapping one schema from the source to the schema of the target. This defines how the data is to be extracted from one system or systems, transformed so it appears to be native, and then updated in the target system or systems. This is increasingly done using visual-mapping technology. In addition,

there is a need to consider both security and governance and also consider these concepts within the design of the data integration solution.

4. **Implementation** refers to actually implementing the data integration solution within the selected technology. This means connecting the source and the target systems, implementing the integration flows as designed in the previous step, and then other steps required getting the data integration solution up-and-running
5. **Testing** refers to assuring that the integration is properly designed and implemented and that the data synchronizes properly between the involved systems. This means looking at known test data within the source system and monitoring how the information flows to the target system. We need to insure that the data mediation mechanisms function correctly as well as review the overall performance, durability, security, modifiability and sustainability of the integrated systems.

## 3.8 SaaS INTEGRATION PRODUCTS AND PLATFORMS

Cloud-centric integration solutions are being developed and demonstrated for showcasing their capabilities for integrating enterprise and cloud applications. The integration puzzle has been the toughest assignment for long due to heterogeneity and multiplicity-induced complexity. Now with the arrival and adoption of the transformative and disruptive paradigm of cloud computing, every ICT products are being converted into a collection of services to be delivered via the open Internet. In that line, the standards-compliant integration suites are being transitioned into services so that any integration need of any one from any part of the world can be easily, cheaply and rapidly met. At this point of time, primarily data integration products are highly visible as their need is greater compared to service or message-based integration of applications. But as the days go by, there will be a huge market for application and service integration. Interoperability will become the most fundamental thing. Composition and collaboration will become critical and crucial for the mass adoption of clouds, which are prescribed and proclaimed as the next-generation infrastructure for creating, deploying and delivering hordes of ambient, artistic, adaptive, and agile services. Cloud interoperability is the prime demand and the figure 3.4 for creating cloud peers, clusters, fabrics, and grids.

### 3.8.1 Jitterbit [4]

Force.com is a Platform as a Service (PaaS), enabling developers to create and deliver any kind of on-demand business application. However, in order to take advantage of this breakthrough cloud technology, there is a need for a flexible and robust integration solution to synchronize force.com with any on-demand or on-premise enterprise applications, databases, and legacy systems.

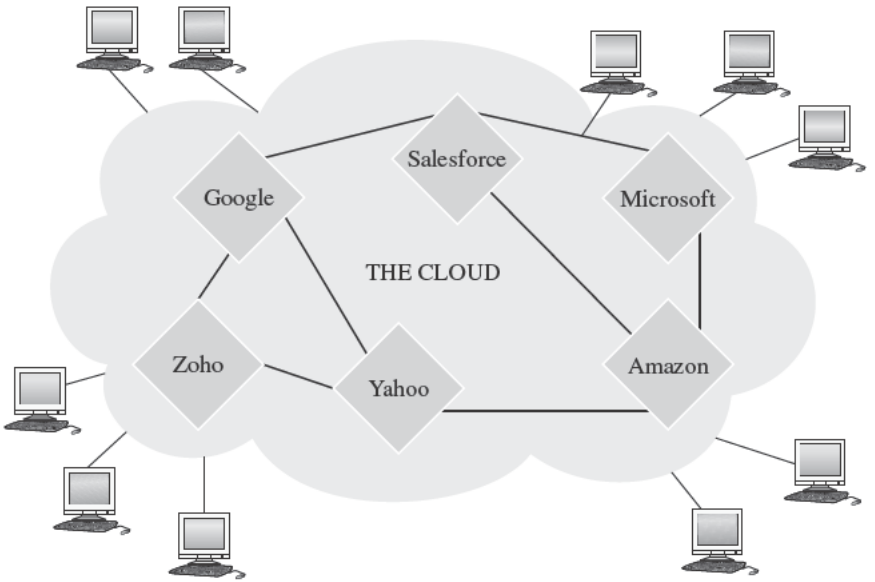



**FIGURE 3.4.** The Smooth and Spontaneous Cloud Interaction via Open Clouds.

Until now, integrating force.com applications with other on-demand applications and systems within an enterprise has seemed like a daunting and doughty task that required too much time, money, and expertise.

Jitterbit is a fully graphical integration solution that provides users a versatile platform and a suite of productivity tools to reduce the integration efforts sharply. Jitterbit can be used standalone or with existing EAI infrastructures, enabling users to create new projects or consume and modify existing ones offered by the open source community or service provider. The Jitterbit solution enables the cool integration among confidential and corporate data, enterprise applications, web services, XML data sources, legacy systems, simple and complex flat files. Apart from a scalable and secure server, Jitterbit provides a powerful graphical environment to help us quickly design, implement, test, deploy, and manage the integration projects. Jitterbit is comprised of two major components:

- **Jitterbit Integration Environment** An intuitive point-and-click graphical UI that enables to quickly configure, test, deploy and manage integration projects on the Jitterbit server.
- **Jitterbit Integration Server** A powerful and scalable run-time engine that processes all the integration operations, fully configurable and manageable from the Jitterbit application.

  Jitterbit is making integration easier, faster, and more affordable than ever before. Using Jitterbit, one can connect force.com with a wide variety

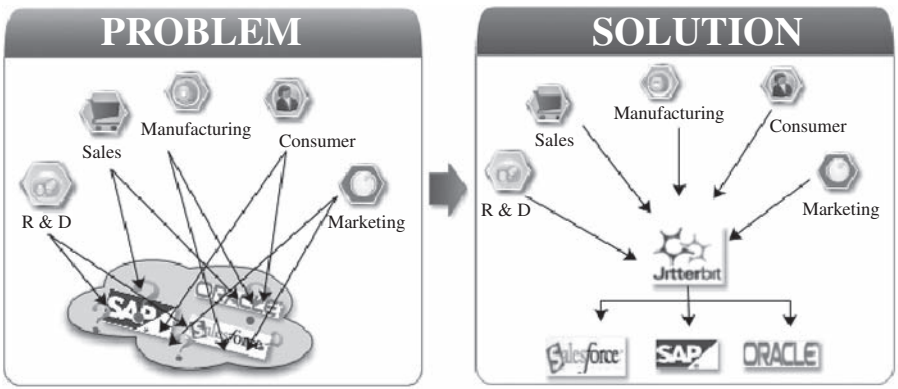



**FIGURE 3.5.** Linkage of On Premise with Online and On Demand Applications.

of on-premise systems including ERP, databases, flat files and custom applications. The figure 3.5 vividly illustrates how Jitterbit links a number of functional and vertical enterprise systems with on-demand applications

### 3.8.2 Boomi Software [5]

Has come out with an exciting and elegant SaaS integration product. It promises to fulfil the vision "**Integration on Demand".** While the popularity of SaaS applications rises dramatically, the integration task has been the "Achilles heel" of the SaaS mechanism. The integration challenge is real and unanimously cited by industry analysts as the leading barrier to overwhelming SaaS adoption.

Boomi AtomSphere is an integration service that is completely on-demand and connects any combination of SaaS, PaaS, cloud, and on-premise applications without the burden of installing and maintaining software packages or appliances. Anyone can securely build, deploy and manage simple to complex integration processes using only a web browser. Whether connecting SaaS applications found in various lines of business or integrating across geographic boundaries, AtomSphere is being presented as a centralized platform that could deliver integration with all the benefits one would expect from a SaaS solution. As new applications are connected to the AtomSphere, they become instantly accessible to the entire community with no adapters to purchase or upgrade to install. Boomi offers the "pure SaaS" integration solution that enables to quickly develop and deploy connections between applications, regardless of the delivery model.

### 3.8.3 Bungee Connect [6]

For professional developers, Bungee Connect enables cloud computing by offering an application development and deployment platform that enables

highly interactive applications integrating multiple data sources and facilitating instant deployment. Built specifically for cloud development, Bungee Connect reduces the efforts to integrate (mashup) multiple web services into a single application. Bungee automates the development of rich UI and eases the difficulty of deployment to multiple web browsers. Bungee Connect leverages the cloud development to bring an additional value to organizations committed to building applications for the cloud.

### 3.8.4 OpSource Connect [7]

Expands on the OpSource Services Bus (OSB) by providing the infrastructure for two-way web services interactions, allowing customers to consume and publish applications across a common web services infrastructure. OpSource Connect also addresses the problems of SaaS integration by unifying different SaaS applications in the "cloud" as well as legacy applications running behind a corporate firewall. By providing the platform to drive web services adoption and integration, OpSource helps its customers grow their SaaS application and increase customer retention.

***The Platform Architecture.*** OpSource Connect is made up of key features including

- OpSource Services Bus
- OpSource Service Connectors
- OpSource Connect Certified Integrator Program
- OpSource Connect ServiceXchange
- OpSource Web Services Enablement Program

The OpSource Services Bus (OSB) is the foundation for OpSource's turnkey development and delivery environment for SaaS and web companies. Based on SOA, it allows applications running on the OpSource On-Demand platform to quickly and easily tap web services. There is no longer a need to write code for these business functions, as OpSource has already invested in the upfront development. It is all about leveraging the OSB to quickly gain business functions and accelerate time-to-market.

### 3.8.5 SnapLogic [8]

SnapLogic is a capable, clean, and uncluttered solution for data integration that can be deployed in enterprise as well as in cloud landscapes. The free community edition can be used for the most common point-to-point data integration tasks, giving a huge productivity boost beyond custom code. SnapLogic professional edition is a seamless upgrade that extends the power of this solution with production management, increased capacity, and

multi-user features at a price that won't drain the budget, which is getting shrunk due to the economic slump across the globe. Even the much-expected "V" mode recovery did not happen; the craze for SaaS solutions is on the climb.

The web, SaaS applications, mobile devices, and cloud platforms have profoundly changed the requirements imposed on data integration technology. SnapLogic is a data integration platform designed for the changing landscape of data and applications. SnapLogic offers a solution that provides flexibility for today's data integration challenges.

- **Changing data sources.** SaaS and on-premise applications, Web APIs, and RSS feeds
- **Changing deployment options.** On-premise, hosted, private and public cloud platforms
- **Changing delivery needs.** Databases, files, and data services

Using a unique hybrid approach, SnapLogic delivers transparency and extensibility to adapt to new integration demands by combining the web principles and open source software with the traditional data integration capabilities.

***Transformation Engine and Repository.*** SnapLogic is a single data integration platform designed to meet data integration needs. The SnapLogic server is built on a core of connectivity and transformation components, which can be used to solve even the most complex data integration scenarios. The SnapLogic designer runs in any web browser and provides an efficient and productive environment for developing transformation logic. The entire system is repository based, with a single metadata store for all the definitions and transformation logic.

The SnapLogic designer provides an initial hint of the web principles at work behind the scenes. The SnapLogic server is based on the web architecture and exposes all its capabilities through web interfaces to outside world. Runtime control and monitoring, metadata access, and transformation logic are all available through web interfaces using a security model just like the web. The SnapLogic web architecture also provides the ultimate flexibility in functionality and deployment. Data transformations are not restricted to a fixed source or target like traditional ETL engines. The ability to read or write a web interface comes naturally to SnapLogic, allowing the creation of on-demand data services using the same logic as fixed transformations. For deployment, the web architecture means one can choose to run SnapLogic on-premise or hosted in the cloud.

### 3.8.6 The Pervasive DataCloud [9]

Platform (figure 3.6) is unique multi-tenant platform. It provides dynamic "compute capacity in the sky" for deploying on-demand integration and other

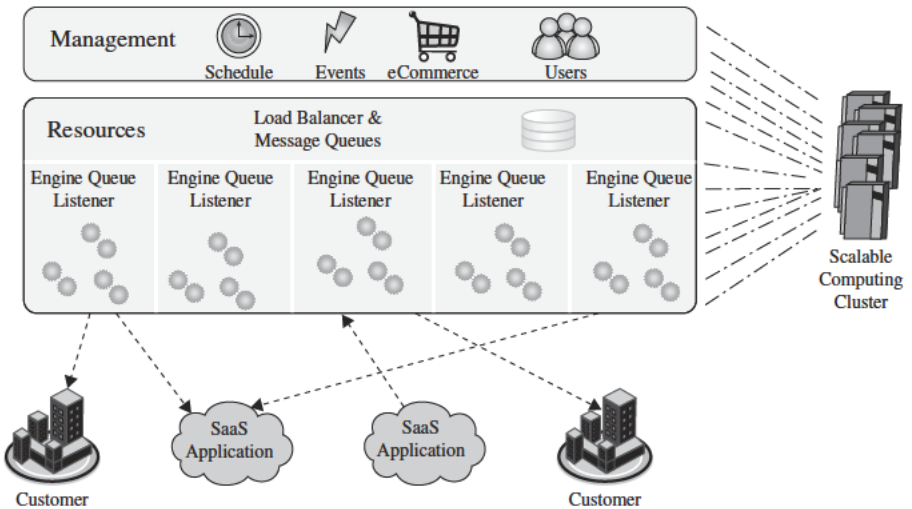



**FIGURE 3.6.** Pervasive Integrator Connects Different Resources.

data-centric applications. Pervasive DataCloud is the first multi-tenant platform for delivering the following.

1. Integration as a Service (IaaS) for both hosted and on-premises applications and data sources
2. Packaged turnkey integration
3. Integration that supports every integration scenario
4. Connectivity to hundreds of different applications and data sources

Pervasive DataCloud hosts Pervasive and its partners' data-centric applications. **Pervasive** uses Pervasive DataCloud as a platform for deploying on-demand integration via

- The **Pervasive DataSynch** family of packaged integrations. These are highly affordable, subscription-based, and packaged integration solutions. They bring a rapid, seamless, turnkey approach to cloud-based integration for popular applications such as Salesforce, QuickBooks and Microsoft Dynamics
- **Pervasive Data Integrator.** This runs on the Cloud or on-premises and is a design-once and deploy anywhere solution to support every integration scenario
  - Data migration, consolidation and conversion
  - ETL / Data warehouse
  - B2B / EDI integration

- Application integration (EAI)
- SaaS /Cloud integration
- SOA / ESB / Web Services
- Data Quality/Governance
- Hubs

Pervasive DataCloud provides multi-tenant, multi-application and multi-customer deployment. Pervasive DataCloud is a platform to deploy applications that are

- **Scalable**—Its multi-tenant architecture can support multiple users and applications for delivery of diverse data-centric solutions such as data integration. The applications themselves scale to handle fluctuating data volumes.
- **Flexible**—Pervasive DataCloud supports SaaS-to-SaaS, SaaS-to-on premise or on-premise to on-premise integration.
- **Easy to Access and Configure**—Customers can access, configure and run Pervasive DataCloud-based integration solutions via a browser.
- **Robust**—Provides automatic delivery of updates as well as monitoring activity by account, application or user, allowing effortless result tracking.
- **Secure—**Uses the best technologies in the market coupled with the best data centers and hosting services to ensure that the service remains secure and available.
- **Affordable**—The platform enables delivery of packaged solutions in a SaaS-friendly pay-as-you-go model.

### 3.8.7 Bluewolf [10]

Has announced its expanded "Integration-as-a-Service" solution, the first to offer ongoing support of integration projects guaranteeing successful integration between diverse SaaS solutions, such as salesforce.com, BigMachines, eAutomate, OpenAir and back office systems (e.g. Oracle, SAP, Great Plains, SQL Service and MySQL). Called the Integrator, the solution is the only one to include proactive monitoring and consulting services to ensure integration success. With remote monitoring of integration jobs via a dashboard included as part of the Integrator solution, Bluewolf proactively alerts its customers of any issues with integration and helps to solves them quickly. For administrative ease, the Bluewolf Integrator is designed with user-friendly administration rules that enable the administrator to manage the flow of data between front and back office systems with little or no IT support. With a Wizard-based approach, the Integrator prompts are presented in simple and non-technical terms. The Bluewolf Integrator integrates with Salesforce, BigMachines,

Oracle, SAP, Microsoft SQL server, MySQL, and supports flat files, such as CSV, XHTML and many more.

### 3.8.8 Online MQ

Online MQ is an Internet-based queuing system. It is a complete and secure online messaging solution for sending and receiving messages over any network. It is a cloud messaging queuing service. In the integration space, messaging middleware as a service is the emerging trend. Here are some of the advantages for using Online MQ.

- **Ease of Use**. It is an easy way for programs that may each be running on different platforms, in different systems and different networks, to communicate with each other without having to write any low-level communication code.
- **No Maintenance**. No need to install any queuing software/server and no need to be concerned with MQ server uptime, upgrades and maintenance.
- **Load Balancing and High Availability**. Load balancing can be achieved on a busy system by arranging for more than one program instance to service a queue. The performance and availability features are being met through clustering. That is, if one system fails, then the second system can take care of users' requests without any delay.
- **Easy Integration**. Online MQ can be used as a web-service (SOAP) and as a REST service. It is fully JMS-compatible and can hence integrate easily with any Java EE application servers. Online MQ is not limited to any specific platform, programming language or communication protocol.

### 3.8.9 CloudMQ [15]

This leverages the power of Amazon Cloud to provide enterprise-grade message queuing capabilities on demand. Messaging allows us to reliably break up a single process into several parts which can then be executed asynchronously. They can be executed within different threads, or even on different machines. The parts communicate by exchanging messages. The messaging framework guarantees that messages get delivered to the right recipient and wake up the appropriate thread when a message arrives. CloudMQ is the easiest way to start exploring integration of messaging into applications since no installation or configuration is necessary.

### 3.8.10 Linxter

Linxter [14] is a cloud messaging framework for connecting all kinds of applications, devices, and systems. Linxter is a behind-the-scenes, message-oriented and cloud-based middleware technology and smoothly automates the complex tasks that developers face when creating communication-based

products and services. With everything becoming Internet-enabled (iPods, clothing, toasters ... anything), Linxter's solution securely, easily, and dynamically connects all these things. Anything that is connected to the Internet can connect to each other through the Linxter's dynamic communication channels. These channels move data between any number of endpoints and the data can be reconfigured on the fly, simplifying the creation of communication-based products and services.

Online MQ, CloudMQ and Linxter are all accomplishing message-based application and service integration. As these suites are being hosted in clouds, messaging is being provided as a service to hundreds of distributed and enterprise applications using the much-maligned multi-tenancy property. "Messaging middleware as a service (MMaaS)" is the grand derivative of the SaaS paradigm. Thus integration as a service (IaaS) is being accomplished through this messaging service. As seen above, there are data mapping tools come handy in linking up different applications and databases that are separated by syntactic, structural, schematic and semantic deviations. Templates are another powerful mechanism being given serious thought these days to minimize the integration complexity. Scores of adaptors for automating the connectivity and subsequently the integration needs are taking off the ground successfully. The integration conundrum has acquired such a big proportion as the SaaS solutions were designed, developed, and deployed without visualizing the need for integration with the resources at the local and corporate servers.

## 3.9 SaaS INTEGRATION SERVICES

We have seen the state-of-the-art cloud-based data integration platforms for real-time data sharing among enterprise information systems and cloud applications. Another fast-emerging option is to link enterprise and cloud systems via messaging. This has forced vendors and service organizations to take message oriented middleware (MoM) to the all-powerful cloud infrastructures. Going forward, there are coordinated and calculated efforts for taking the standards-compatible enterprise service bus (ESB) to clouds in order to guarantee message enrichment, mediation, content and context-based message routing. Thus both loosely or lightly coupled and decoupled cloud services and applications will become a reality soon with the maturity and durability of message-centric and cloud-based service bus suites. We can still visualise the deployment of complex event processing (CEP) engines in clouds in order to capture and capitalise streams of events from diverse sources in different formats and forms in order to infer the existing and emerging situation precisely and concisely. Further on, all kinds of risks, threats, vulnerabilities, opportunities, trends, tips, associations, patterns, and other tactical as well as strategic insights and actionable insights can be deduced to act upon confidently and at real time.

In a highly interoperable environment, seamless and spontaneous composition and collaboration would happen in order to create sophisticated services dynamically. Context-aware applications covering all kinds of constituents and participants (self, surroundings and situation-aware devices, sensors, robots, instruments, media players, utensils, consumer electronics, information appliances, etc.), in a particular environment (home, hotel, hospital, office, station, stadium etc.), enterprise systems, integration middleware, cloud services and knowledge engines can be built and sustained. There are fresh endeavours in order to achieve service composition in cloud ecosystem. Existing frameworks such as service component architecture (SCA) are being revitalised for making it fit for cloud environments. Composite applications, services, data, views and processes will be become cloud-centric and hosted in order to support spatially separated and heterogeneous systems.

### 3.9.1 Informatica On-Demand [11]

Informatica offers a set of innovative on-demand data integration solutions called Informatica On-Demand Services. This is a cluster of easy-to-use SaaS offerings, which facilitate integrating data in SaaS applications, seamlessly and securely across the Internet with data in on-premise applications. The Informatica on-demand service is a subscription-based integration service that provides all the relevant features and functions, using an on-demand or an as-a-service delivery model. This means the integration service is remotely hosted, and thus provides the benefit of not having to purchase or host software. There are a few key benefits to leveraging this maturing technology.

- Rapid development and deployment with zero maintenance of the integration technology.
- Automatically upgraded and continuously enhanced by vendor.
- Proven SaaS integration solutions, such as integration with Salesforce .com, meaning that the connections and the metadata understanding are provided.
- Proven data transfer and translation technology, meaning that core integration services such as connectivity and semantic mediation are built into the technology.

Informatica On-Demand has taken the unique approach of moving its industry leading PowerCenter Data Integration Platform to the hosted model and then configuring it to be a true multi-tenant solution. That means that when developing new features or enhancements, they are immediately made available to all of their customers transparently. That means, no complex software upgrades required and no additional fee is demanded. Fixing, patching, versioning, etc are taken care of by the providers at no cost for the subscribers. Still the service and operation level agreements are being fully met. And the multi-tenant architecture means that bandwidth and scalability are

shared resources so meeting different capacity demands becomes smoother and simpler.

### 3.9.2 Microsoft Internet Service Bus (ISB) [13]

Azure is an upcoming cloud operating system from Microsoft. This makes development, depositing and delivering Web and Windows application on cloud centers easier and cost-effective. Developers' productivity shoots up, customers' preferences are being provided, the enterprise goal of "more with less" gets achieved, etc. Azure is being projected as the comprehensive yet compact cloud framework that comprises a wider variety of enabling tools for a slew of tasks and a growing service portfolio. The primary components are explained below.

***Microsoft .NET Services.*** is a set of Microsoft-built and hosted cloud infrastructure services for building Internet-enabled applications and the ISB acts as the cloud middleware providing diverse applications with a common infrastructure to name, discover, expose, secure and orchestrate web services. The following are the three broad areas.

***.NET Service Bus.*** The .NET Service Bus (figure 3.7) provides a hosted, secure, and broadly accessible infrastructure for pervasive communication,

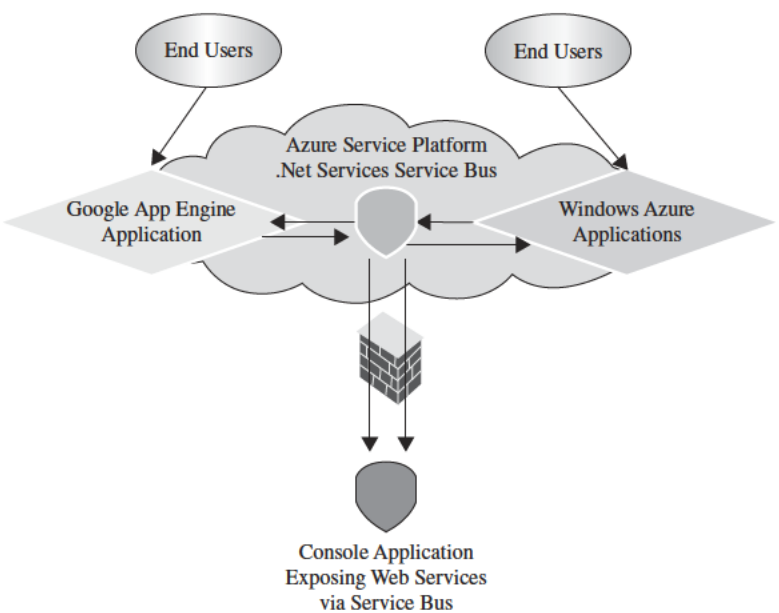



**FIGURE 3.7.** .NET Service Bus.

large-scale event distribution, naming, and service publishing. Services can be exposed through the Service Bus Relay, providing connectivity options for service endpoints that would otherwise be difficult or impossible to reach. Endpoints can be located behind network address translation (NAT) boundaries or bound to frequently changing, dynamically assigned IP addresses, or both.

***.NET Access Control Service.*** The .NET Access Control Service is a hosted, secure, standards-based infrastructure for multiparty, federated authentication, rules-driven, and claims-based authorization. The Access Control Service's capabilities range from simple, one-step, user name/password-based authentication and authorization with Web-style HTTP requests to sophisticated WS-Federation scenarios that employ two or more collaborating WS-Trust Security Token Services. The Access Control Service allows applications to rely on .NET Services solution credentials for simple scenarios or on on-premise enterprise accounts managed in Microsoft Active Directory and federated with the Access Control Service via next-generation Microsoft Active Directory Federation Services.

***.NET Workflow Service.*** The .NET Workflow Service provide a hosted environment for service orchestration based on the familiar Windows Workflow Foundation (WWF) development experience. The Workflow services will provide a set of specialized activities for rules-based control flow, service invocation, as well as message processing and correlation that can be executed on demand, on schedule, and at scale inside the.NET Services environment.

The most important part of the Azure is actually the service bus represented as a WCF architecture. The key capabilities of the Service Bus are

- A **federated namespace** model that provides a shared, hierarchical namespace into which services can be mapped. This allows providing any endpoint with a stable, Internet-accessible URI, regardless of the location.
- A **service registry** service that provides an opt-in model for publishing service endpoints into a lightweight, hierarchical, and RSS-based discovery mechanism.
- A lightweight and scalable **publish/subscribe event bus**.
- A **relay** and **connectivity** service with advanced NAT traversal and pull-mode message delivery capabilities acting as a "perimeter network (also known as DMZ, demilitarized zone, and screened subnet) in the sky" for services that would otherwise be unreachable due to NAT/Firewall restrictions or frequently changing dynamic IP addresses, or that do not allow any incoming connections due to other technical limitations.

***Relay Services.*** Often when we connect a service, it is located behind the firewall and behind the load balancer. Its address is dynamic and can be

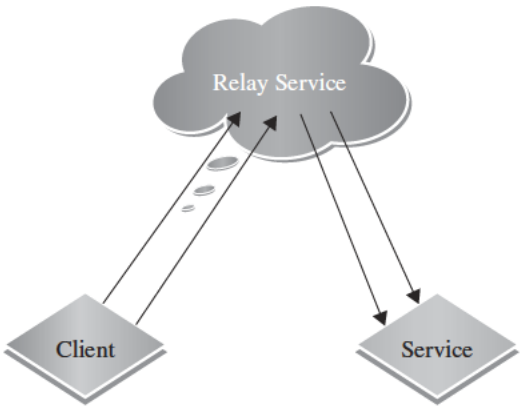



**FIGURE 3.8.** The .NET Relay Service.

resolved only on local network. When we are having the service call-backs to the client, the connectivity challenges lead to scalability, availability and security issues. The solution to Internet connectivity challenges is instead of connecting client directly to the service we can use a relay service as pictorially represented in the relay service figure 3.8.

The Relay service is a service residing in the cloud whose job is to assist the connectivity and relaying the calls to the service. Relay Service solution require both the client and the service intranets to allow connections to the cloud.

## 3.10 BUSINESSES-TO-BUSINESS INTEGRATION (B2Bi) SERVICES

B2Bi has been a mainstream activity for connecting geographically distributed businesses for purposeful and beneficial cooperation. Products vendors have come out with competent B2B hubs and suites for enabling smooth data sharing in standards-compliant manner among the participating enterprises. Now with the surging popularity of clouds, there are serious and sincere efforts to posit these products in clouds in order to deliver B2Bi as a service with very lest investment and maintenance costs. The cloud ideas and ideals lay the strong and stimulating foundation for transitioning from the capital expenditure to operational expenditure and for sustaining the transformed.

There are several proven integration methods in the B2Bi space and they can be captured and capitalized for achieving quicker success and better return and value in the evolving IaaS landscape. B2Bi systems are good candidate for IaaS as they are traditionally employed to automate business processes between manufacturers and their external trading partners such as retail, warehouse, transport, and inventory systems. This means that they provide application-to-application (A2A) connectivity along with functionality that is crucial to

linking internal and external software: i.e. secure data exchange across the corporate firewall. Unlike pure EAI solutions designed only for internal data sharing, B2Bi platforms have the ability to encrypt files for safe passage across the public network, manage large data volumes, transfer batch files, convert disparate file formats and guarantee data accuracy, integrity, confidentiality, and delivery. Just as these abilities ensure smooth communication between manufacturers and their external suppliers or customers, they also enable reliable interchange between hosted and installed applications.

The IaaS model also leverages the adapter libraries developed by B2Bi vendors to provide rapid integration with various business systems. Because the B2Bi partners have the expertise and experience ad can supply pre-built connectors for major ERP, CRM, SCM and other packaged business applications as well as legacy systems from AS400 to MVS and mainframe. The use of a hub-and-spoke centralised architecture further simplifies implementation and provides a good control and grip on the system management and finally this avoids placing an excessive processing burden on the customer side. The hub is installed at the SaaS provider's cloud center to do the heavy lifting such as reformatting files. A spoke unit, typically consisting of a small downloadable Java client, is then deployed at each user site to handle basic tasks such as data transfer. This also eliminates the need for an expensive server-based solution, data mapping and other tasks at the customer location. As the Internet is the principal communication infrastructure, enterprises can leverage the IaaS to sync up with their partners across the continents towards smart and systematic collaboration.

***Cloud- based Enterprise Mashup Integration Services for B2B Scenarios [17].*** There is a vast need for infrequent, situational and ad-hoc B2B applications desired by the mass of business end-users. Enterprise mashup and lightweight composition approaches and tools are promising methods to unleash the huge and untapped potential of empowering end-users to develop or assemble aligned and aware composite services in order to overcome the "long-tail" dilemma. Currently available solutions to support B2B collaborations focus on the automation of long-term business relationships and still lack to provide their users intuitive ways to modify or to extend them according to their ad-hoc or situational needs. Conventional proceeding in the development of such applications directs to an immense use of time and work due to long development cycles and a lack of required business knowledge.

Especially in the area of applications to support B2B collaborations, current offerings are characterized by a high richness but low reach, like B2B hubs that focus on many features enabling electronic collaboration, but lack availability for especially small organizations or even individuals. The other extreme solutions with a low reach but high richness such as web sites, portals and emails, lack standardization and formularization which makes them inappropriate for automated or special enterprises' needs. New development approaches are hence needed to overcome theses hurdles and hitches to involve

non-technical business users into the development process in order to address this long tail syndrome, to realize cost-effectiveness and efficiency gains, and to overcome the traditional constrictions between IT department and business units.

Enterprise Mashups, a kind of new-generation Web-based applications, seem to adequately fulfill the individual and heterogeneous requirements of end-users and foster End User Development (EUD). To shorten the traditional and time-consuming development process, these new breed of applications are developed by non-professional programmers, often in a non-formal, iterative, and collaborative way by assembling existing building blocks.

SOA has been presented as a potent solution to organization's integration dilemmas. ESBs are used to integrate different services within a SOA-driven company. However, most ESBs are not designated for cross-organizational collaboration, and thus problems arise when articulating and aiming such an extended collaboration. SOA simplifies and streamlines the integration of new and third-party services but still it can be done by skilled and experienced developers. End-users usually are not able to realize the wanted integration scenarios. This leads, beneath high costs for integration projects, to the unwanted inflexibility, because integration projects last longer, although market competition demands a timely response to uprising requirements proactively.

Another challenge in B2B integration is the ownership of and responsibility for processes. In many inter-organizational settings, business processes are only sparsely structured and formalized, rather loosely coupled and/or based on ad-hoc cooperation. Inter-organizational collaborations tend to involve more and more participants and the growing number of participants also draws a huge amount of differing requirements. Also, the participants may act according to different roles, controls and priorities. Historically, the focus for collaboration was participation within teams which were managed according to one set of rules.

Now, in supporting supplier and partner co-innovation and customer co-creation, the focus is shifting to collaboration which has to embrace the participants, who are influenced yet restricted by multiple domains of control and disparate processes and practices. This represents the game-changing shift from static B2B approaches to new and dynamic B2B integration, which can adaptively act and react to any unexpected disruptions, can allow a rapid configuration and customization and can manage and moderate the rising complexity by the use of end-to-end business processes.

Both Electronic data interchange translators (EDI) and Managed file transfer (MFT) have a longer history, while B2B gateways only have emerged during the last decade. However, most of the available solutions aim at supporting medium to larger companies, resulting from their high costs and long implementation cycles and times, which make them unaffordable and unattractive to smaller organizations. Consequently, these offerings are not suitable for short-term collaborations, which need to be set up in an ad hoc manner.

***Enterprise Mashup Platforms and Tools.*** Mashups are the adept combination of different and distributed resources including content, data or application functionality. Resources represent the core building blocks for mashups. Resources can be accessed through APIs, which encapsulate the resources and describe the interface through which they are made available. Widgets or gadgets primarily put a face on the underlying resources by providing a graphical representation for them and piping the data received from the resources. Piping can include operators like aggregation, merging or filtering. Mashup platform is a Web based tool that allows the creation of Mashups by piping resources into Gadgets and wiring Gadgets together.

Enterprise Mashups, which are enterprise-scale, aware and ready, are extremely advantages in B2B integration scenes. Mashups can resolve many of the disadvantages of B2B hubs such as low reach due to hard-wired connections. Mashups enable EUD and lightweight connections of systems. Mashups can help adding richness to existing lightweight solutions such as Websites or Portals by adding a certain level of formalization and standardization. Mashups facilitate the ease of mixing and transforming various sources of information internally and from business partners. Complexity in B2B operations is often linked with heterogeneous systems and platforms. The tedious integration process and requirements of various support and maintenance for the software is a major hindrance to today's dynamic B2B integration, especially for the small and medium enterprises.

The Mashup integration services are being implemented as a prototype in the FAST project. The layers of the prototype are illustrated in figure 3.9 illustrating the architecture, which describes how these services work together. The authors of this framework have given an outlook on the technical realization of the services using cloud infrastructures and services.

Prototype architecture shows the services and their relations to each other. The core services are shown within the box in the middle. The external services shown under the box are attached via APIs to allow the usage of third-party offerings to realize their functionality. Users access the services through a Mashup platform of their choice. The Mashup platforms are connected via APIs to the Mashup integration services.

To use the services, users have to identify themselves against the user-access control service. This service is connected to a user management service, which controls the users and their settings. The user management service is connected via an API to allow the usage of external services, e.g. a corporate user database. All data coming from the users go through a translation engine to unify the data objects and protocols, so that different Mashup platforms can be integrated. The translation engine has an interface which allows connecting other external translation engines to add support for additional protocol and data standards. The translated data is forwarded to the routing engine, which is the core of the Mashup integration services. The routing engine takes care of processing the inputs received from the Mashup platforms and forwarding them to the right recipient. The routing is based on rules, which can be configured through an API.

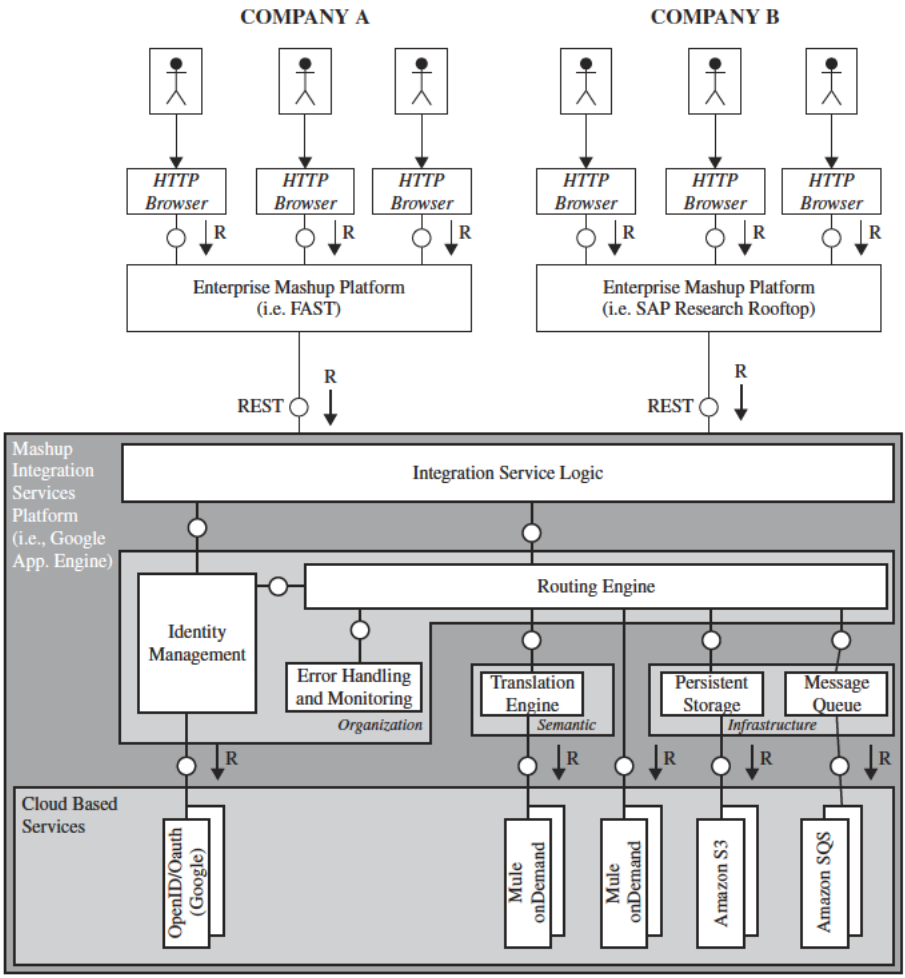



**FIGURE 3.9.** Cloud based Enterprise Mashup Integration Platform Architecture.

To simplify this, a Gadget could be provided for the end-user. The routing engine is also connected to a message queue via an API. Thus, different message queue engines are attachable. The message queue is responsible for storing and forwarding the messages controlled by the routing engine. Beneath the message queue, a persistent storage, also connected via an API to allow exchangeability, is available to store large data. The error handling and monitoring service allows tracking the message-flow to detect errors and to collect statistical data. The Mashup integration service is hosted as a cloud-based service. Also, there are cloud-based services available which provide the functionality required by the integration service. In this way, the Mashup integration service can reuse and leverage the existing cloud services to speed up the implementation.

***Message Queue.*** The message queue could be realized by using Amazon's Simple Queue Service (SQS). SQS is a web-service which provides a queue for messages and stores them until they can be processed. The Mashup integration services, especially the routing engine, can put messages into the queue and recall them when they are needed.

***Persistent Storage.*** Amazon Simple Storage Service5 (S3) is also a web-service. The routing engine can use this service to store large files.

***Translation Engine.*** This is primarily focused on translating between different protocols which the Mashup platforms it connects can understand, e.g. REST or SOAP web services. However, if the need of translation of the objects transferred arises, this could be attached to the translation engine. A company requiring such a service could on the one hand develop such a service and connect it to the Mashup integration services. Another possibility for this would be to connect existing translation services, e.g., the services by Mule on Demand, which is also a cloud-based offering.

***Interaction between the Services.*** The diagram describes the process of a message being delivered and handled by the Mashup Integration Services Platform. The precondition for this process is that a user already established a route to a recipient. After having received a message from an Enterprise Mashup tool via an API, the Integration Services first check the access rights of the sender of the message against an external service. An incoming message is processed only if sender of the message is authorized, that is, he has the right to deliver the message to the recipient and to use the Mashup integration services. If he is not authorized, the processing stops, and an error message gets logged. The error log message is written into a log file, which could reside on Amazon's Simple Storage Service (S3). If the message has been accepted, it is put in the message queue in Amazon's SQS service. If required, the message is being translated into another format, which can also be done by an external, cloud-based service. After that, the services can begin trying delivering the message to a recipient. Evaluating the recipients of the message is based on the rules stored in the routing engine which have been configured by a user before. Finally, the successful delivery of the message can be logged, or an error if one occurred.

## 3.11 A FRAMEWORK OF SENSOR—CLOUD INTEGRATION [3]

In the past few years, wireless sensor networks (WSNs) have been gaining significant attention because of their potentials of enabling of novel and attractive solutions in areas such as industrial automation, environmental monitoring, transportation business, health-care etc. If we add this collection of sensor-derived data to various Web-based social networks or virtual communities, blogs etc., there will be fabulous transitions among and around us.

With the faster adoption of micro and nano technologies, everyday things are destined to become digitally empowered and smart in their operations and offerings. Thus the goal is to link smart materials, appliances, devices, federated messaging middleware, enterprise information systems and packages, ubiquitous services, handhelds, and sensors with one another smartly to build and sustain cool, charismatic and catalytic situation-aware applications. Clouds have emerged as the centralized, compact and capable infrastructure to deliver people-centric and context-aware services to users with all the qualities inherently. This long-term target demands that there has to be a cool connectivity and purposeful interactions between clouds and all these pervasive and minuscule systems. In this section, we explain about a robust and resilient a framework to enable this exploration by integrating sensor networks to clouds. But there are many challenges to enable this framework. The authors of this framework have proposed a pub-sub based model, which simplifies the integration of sensor networks with cloud based community-centric applications. Also there is a need for internetworking cloud providers in case of violation of service level agreement with users.

A virtual community consisting of team of researchers have come together to solve a complex problem and they need data storage, compute capability, security; and they need it all provided now. For example, this team is working on an outbreak of a new virus strain moving through a population. This requires more than a Wiki or other social organization tool. They deploy bio-sensors on patient body to monitor patient condition continuously and to use this data for large and multi-scale simulations to track the spread of infection as well as the virus mutation and possible cures. This may require computational resources and a platform for sharing data and results that are not immediately available to the team.

Traditional HPC approach like Sensor-Grid model can be used in this case, but setting up the infrastructure to deploy it so that it can scale out quickly is not easy in this environment. However, the cloud paradigm is an excellent move. But current cloud providers unfortunately did not address the issue of integrating sensor network with cloud applications and thus have no infrastructure to support this scenario. The virtual organization (VO) needs a place that can be rapidly deployed with social networking and collaboration tools, other specialized applications and tools that can compose sensor data and disseminate them to the VO users based on their subscriptions.

Here, the researchers need to register their interests to get various patients' state (blood pressure, temperature, pulse rate etc.) from bio-sensors for large-scale parallel analysis and to share this information with each other to find useful solution for the problem. So the sensor data needs to be aggregated, processed and disseminated based on subscriptions. On the other hand, as

sensor data require huge computational power and storage, one cloud provider may not handle this requirement. This insists and induces for a dynamic collaboration with other cloud providers. The framework addresses the above issues and provides competent solutions.

To integrate sensor networks to cloud, the authors have proposed a content-based pub-sub model. A pub/sub system encapsulates sensor data into events and provides the services of event publications and subscriptions for asynchronous data exchange among the system entities. MQTT-S is an open topic-based pub-sub protocol that hides the topology of the sensor network and allows data to be delivered based on interests rather than individual device addresses. It allows a transparent data exchange between WSNs and traditional networks and even between different WSNs.

In this framework, like MQTT-S, all of the system complexities reside on the broker's side but it differs from MQTT-S in that it uses content-based pub-sub broker rather than topic-based which is suitable for the application scenarios considered. When an event is published, it is transmitted from a publisher to one or more subscribers without the publisher having to address the message to any specific subscriber. Matching is done by the pub-sub broker outside of the WSN environment. In content-based pub-sub system, sensor data has to be augmented with meta-data to identify the different data fields. For example, a meta-data of a sensor value (also event) can be body temperature, blood pressure etc.

To deliver published sensor data or events to subscribers, an efficient and scalable event matching algorithm is required by the pub-sub broker. This event matching algorithm targets a range predicate case suitable to the application scenarios and it is also efficient and scalable when the number of predicates increases sharply. The framework is shown in figure 3.10. In this framework, sensor data are coming through gateways to a pub/sub broker. Pub/sub broker is required in the system to deliver information to the consumers of SaaS applications as the entire network is very dynamic. On the WSN side, sensor or actuator (SA) devices may change their network addresses at any time. Wireless links are quite likely to fail. Furthermore, SA nodes could also fail at any time and rather than being repaired, it is expected that they will be replaced by new ones. Besides, different SaaS applications can be hosted and run on any machines anywhere on the cloud. In such situations, the conventional approach of using network address as communication means between the SA devices and the applications may be very problematic because of their dynamic and temporal nature.

Moreover, several SaaS applications may have an interest in the same sensor data but for different purposes. In this case, the SA nodes would need to manage and maintain communication means with multiple applications in parallel. This might exceed the limited capabilities of the simple and low-cost SA devices. So pub-sub broker is needed and it is located in the cloud side because of its higher performance in terms of bandwidth and capabilities. It has four components describes as follows:

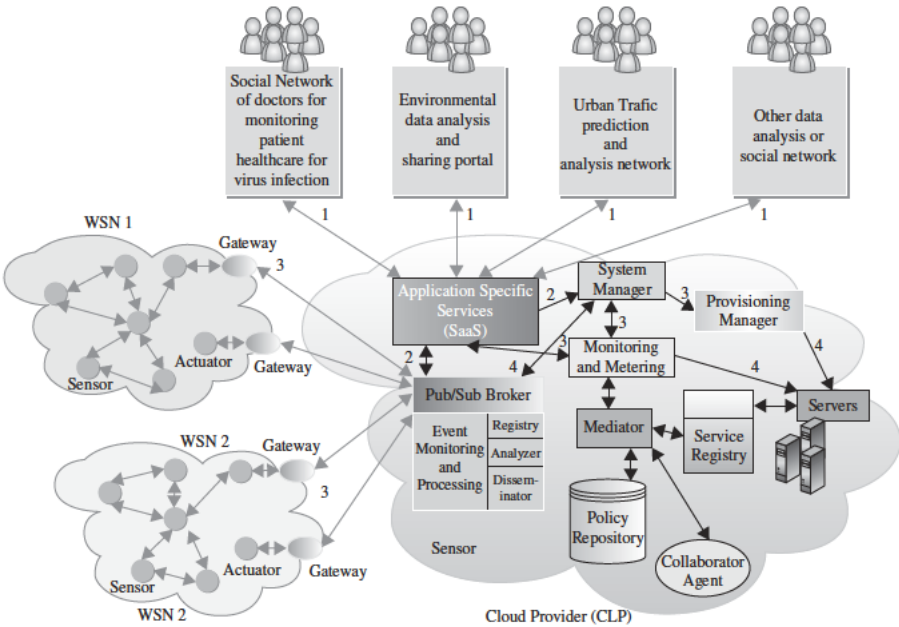



**FIGURE 3.10.** The Framework Architecture of Sensor Cloud Integration.

***Stream monitoring and processing component (SMPC).*** The sensor stream comes in many different forms. In some cases, it is raw data that must be captured, filtered and analyzed on the fly and in other cases, it is stored or cached. The style of computation required depends on the nature of the streams. So the SMPC component running on the cloud monitors the event streams and invokes correct analysis method. Depending on the data rates and the amount of processing that is required, SMP manages parallel execution framework on cloud.

***Registry component (RC).*** Different SaaS applications register to pub-sub broker for various sensor data required by the community user. For each application, registry component stores user subscriptions of that application and sensor data types (temperature, light, pressure etc.) the application is interested in. Also it sends all user subscriptions along with application id to the disseminator component for event delivery.

***Analyzer component (AC).*** When sensor data or events come to the pub-sub broker, analyzer component determines which applications they are belongs to and whether they need periodic or emergency deliver. The events are then passed to the disseminator component to deliver to appropriate users through SasS applications.

***Disseminator component (DC).*** For each SaaS application, it disseminates sensor data or events to subscribed users using the event matching algorithm. It can utilize cloud's parallel execution framework for fast event delivery. The pub-sub components workflow in the framework is as follows:

Users register their information and subscriptions to various SaaS applications which then transfer all this information to pub/sub broker registry. When sensor data reaches to the system from gateways, event/stream monitoring and processing component (SMPC) in the pub/sub broker determines whether it needs processing or just store for periodic send or for immediate delivery. If sensor data needs periodic/ emergency delivery, the analyzer determines which SaaS applications the events belong to and then passes the events to the disseminator along with application ids. The disseminator, using the event matching algorithm, finds appropriate subscribers for each application and delivers the events for use.

Besides the pub-sub broker, the authors have proposed to include three other components: mediator, policy repository (PR) and collaborator agent (CA) along with system manager, provisioning manager, monitoring and metering and service registry in the sensor-cloud framework to enable VO based dynamic collaboration of primary cloud providers with other cloud providers in case of SLA violations for burst resource demand. These three components collectively act as a "gateway" for a given CLP in creation of a new VO. They are described as follows:

***Mediator.*** The (resource) mediator is a policy-driven entity within a VO to ensure that the participating entities are able to adapt to changing circumstances and are able to achieve their objectives in a dynamic and uncertain environment. Once a VO is established, the mediator controls which resources to be used of the collaborating CLPs, how this decision is taken, and which policies are being used. When performing automated collaboration, the mediator will also direct any decision making during negotiations, policy management, and scheduling. A mediator holds the initial policies for VO creation and works in conjunction with its local Collaborating Agent (CA) to discover external resources and to negotiate with other CLPs.

***Policy Repository (PR).*** The PR virtualizes all of the policies within the VO. It includes the mediator policies, VO creation policies along with any policies for resources delegated to the VO as a result of a collaborating arrangement. These policies form a set of rules to administer, manage, and control access to VO resources. They provide a way to manage the components in the face of complex technologies.

***Collaborating Agent (CA).*** The CA is a policy-driven resource discovery module for VO creation and is used as a conduit by the mediator to exchange policy and resource information with other CLPs. It is used by a primary CLP to discover the collaborating CLPs' (external) resources, as well as to let them

know about the local policies and service requirements prior to commencement of the actual negotiation by the mediator.

On concluding, to deliver published sensor data or events to appropriate users of cloud applications, an efficient and scalable event-matching algorithm called Statistical Group Index Matching (SGIM) is proposed and leveraged. The authors also have evaluated its performance and compared with existing algorithms in a cloud based ubiquitous health-care application scenario. The authors in the research paper have clearly described this algorithm that in sync with the framework enables sensor-cloud connectivity to utilize the ever-expanding sensor data for various next generation community-centric sensing applications on the cloud. It can be seen that the computational tools needed to launch this exploration is more appropriately built from the data center "cloud" computing model than the traditional HPC approaches or Grid approaches. The authors have embedded a content-based pub-sub model to enable this framework.

## 3.12 SaaS INTEGRATION APPLIANCES

Appliances are a good fit for high-performance requirements. Clouds too have gone in the same path and today there are cloud appliances (also termed as "cloud in a box"). In this section, we are to see an integration appliance.

***Cast Iron Systems [12].*** This is quite different from the above-mentioned schemes. Appliances with relevant software etched inside are being established as a high-performance and hardware-centric solution for several IT needs. Very frequently we read and hear about a variety of integration appliances considering the complexities of connectivity, transformation, routing, mediation and governance for streamlining and simplifying business integration. Even the total cloud infrastructure comprising the prefabricated software modules is being produced as an appliance (cloud in a box). This facilitates building private clouds quicker and easier. Further on, appliance solution is being taken to clouds in order to provide the appliance functionality and feature as a service. "Appliance as a service" is a major trend sweeping the cloud service provider (CSP) industry.

Cast Iron Systems (**www.ibm.com**) provides pre-configured solutions for each of today's leading enterprise and On-Demand applications. These solutions, built using the Cast Iron product offerings offer out-of-the-box connectivity to specific applications, and template integration processes (TIPs) for the most common integration scenarios. For example, the Cast Iron solution for salesforce.com comes with built-in AppExchange connectivity, and TIPs for customer master, product master and contact data integration. Cast Iron solutions enable customers to rapidly complete application-specific integrations using a "configuration, not coding" approach. By using a pre-configured template, rather than starting from scratch with complex software tools and

writing lots of code, enterprises complete business-critical projects in days rather than months. Large and midsize companies in a variety of industries use Cast Iron solutions to solve their most common integration needs. From the image below, it is clear Cast Iron systems have readymade.

## 3.13 CONCLUSION

SaaS in sync with cloud computing has brought in strategic shifts for businesses as well as IT industries. Increasingly SaaS applications are being hosted in cloud infrastructures and the pervasive Internet is the primary communication infrastructure. These combinations of game-changing concepts and infrastructures have really come as a boon and blessing as the world is going through the economic slump and instability. The goal of "more with less" is being accomplished with the maturity of these freshly plucked and published ideas. Applications are studiously being moved to clouds, which are exposed as services, which are delivered via the Internet to user agents or humans and accessed through the ubiquitous web browsers. The unprecedented adoption is to instigate and instil a number of innovations as it has already created a lot of buzz on newer business, pricing, delivery and accessibility models. Ubiquity and utility will become common connotations. Value-added business transformation, augmentation, optimization along with on-demand IT will be the ultimate output. In the midst of all the enthusiasm and optimism, there are some restricting factors that need to be precisely factored out and resolved comprehensively in order to create an extended ecosystem for intelligent collaboration. Integration is one such issue and hence a number of approaches are being articulated by professionals. Product vendors, consulting and service organizations are eagerly coming out with integration platforms, patterns, processes, and best practices. There are generic as well as specific (niche) solutions. Pure SaaS middleware as well as standalone middleware solutions are being studied and prescribed based on "as-is" situation and to-be" aspiration. As the business and technical cases of cloud middleware suites are steadily evolving and enlarging, the realization of internet service bus (the internet-scale ESB) is being touted as the next big thing for the exotic cloud space. In this chapter, we have elaborated and expounded the need for a creative and futuristic ISB that streamlines and simplifies the integration among clouds (public, private, and hybrid).

## REFERENCES

1. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report No. UCB/EECS 2009 28, University of California at Berkley, USA, Feb. 10, 2009.

2. R. Buyya, C. S. Yeo, and S. Venugopal, *Market Oriented Cloud Computing Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*, Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, Sept. 25–27, 2008, Dalian, China.
3. Arista, "Cloud Networking: Design Patterns for 'Cloud Centric' Application Environments", January 2009.
4. http://www.jitterbit.com
5. http://www.dell.com
6. http://www.bungeeconnect.com/
7. http://www.opsource.net/
8. http://www.snaplogic.com
9. http://www.pervasiveintegration.com/
10. http://www.bluewolf.com
11. http://www.informaticaondemand.com
12. http://www.castiron.com/
13. http://www.microsoft.com/azure/servicebus.mspx
14. http://linxter.com/
15. http://www.cloudmq.com/
16. Mohammad Mehedi Hassan et al., "A framework of sensor cloud integration opportunities and challenges", Proceedings of the Conference On Ubiquitous Information Management and Communication, Korea, 2009.
17. Robert G. Siebeck et al., "Cloudbased Enterprise Mashup Integration Services for B2B Scenarios", MEM2009 workshop, Spain, 2009.

# CHAPTER 4

# THE ENTERPRISE CLOUD COMPUTING PARADIGM

TARIQ ELLAHI, BENOIT HUDZIA, HUI LI, MAIK A. LINDNER, and PHILIP ROBINSON

## 4.1 INTRODUCTION

Cloud computing is still in its early stages and constantly undergoing changes as new vendors, offers, services appear in the cloud market. This evolution of the cloud computing model is driven by cloud providers bringing new services to the ecosystem or revamped and efficient exiting services primarily triggered by the ever changing requirements by the consumers. However, cloud computing is predominantly adopted by start-ups or SMEs so far, and wide-scale enterprise adoption of cloud computing model is still in its infancy. Enterprises are still carefully contemplating the various usage models where cloud computing can be employed to support their business operations. Enterprises will place stringent requirements on cloud providers to pave the way for more widespread adoption of cloud computing, leading to what is known as the enterprise cloud paradigm computing. Enterprise cloud computing is the alignment of a cloud computing model with an organization's business objectives (profit, return on investment, reduction of operations costs) and processes. This chapter explores this paradigm with respect to its motivations, objectives, strategies and methods.

Section 4.2 describes a selection of deployment models and strategies for enterprise cloud computing, while Section 4.3 discusses the issues of moving [traditional] enterprise applications to the cloud. Section 4.4 describes the technical and market evolution for enterprise cloud computing, describing some potential opportunities for multiple stakeholders in the provision of enterprise cloud computing.

*Cloud Computing: Principles and Paradigms*, Edited by Rajkumar Buyya, James Broberg and Andrzej Goscinski 

## 4.2 BACKGROUND

According to NIST [1], cloud computing is composed of five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. The ways in which these characteristics are manifested in an enterprise context vary according to the deployment model employed.

### 4.2.1 Relevant Deployment Models for Enterprise Cloud Computing

There are some general cloud deployment models that are accepted by the majority of cloud stakeholders today, as suggested by the references [1] and [2] and discussed in the following:

- ***Public clouds*** are provided by a designated service provider for general public under a utility based pay-per-use consumption model. The cloud resources are hosted generally on the service provider's premises. Popular examples of public clouds are Amazon's AWS (EC2, S3 etc.), Rackspace Cloud Suite, and Microsoft's Azure Service Platform.
- ***Private clouds*** are built, operated, and managed by an organization for its internal use only to support its business operations exclusively. Public, private, and government organizations worldwide are adopting this model to exploit the cloud benefits like flexibility, cost reduction, agility and so on.
  - ***Virtual private clouds*** are a derivative of the private cloud deployment model but are further characterized by an isolated and secure segment of resources, created as an overlay on top of public cloud infrastructure using advanced network virtualization capabilities. Some of the public cloud vendors that offer this capability include Amazon Virtual Private Cloud [3], OpSource Cloud [4], and Skytap Virtual Lab [5].
- ***Community clouds*** are shared by several organizations and support a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). They may be managed by the organizations or a third party and may exist on premise or off premise [1]. One example of this is OpenCirrus [6] formed by HP, Intel, Yahoo, and others.
- ***Managed clouds*** arise when the physical infrastructure is owned by and/or physically located in the organization's data centers with an extension of management and security control plane controlled by the managed service provider [2]. This deployment model isn't widely agreed upon, however, some vendors like ENKI [7] and NaviSite's NaviCloud offers claim to be managed cloud offerings.
- ***Hybrid clouds*** are a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application

portability (e.g., cloud bursting for load-balancing between clouds) [1]. Recently some cloud vendors have started offering solutions which can be used to enable these hybrid cloud deployment models. Some examples of these offerings include Amazon Virtual Private Cloud [3], Skytap Virtual Lab [5], and CohesiveFT VPN-Cubed [8]. These solutions work by creating IPSec VPN tunneling capabilities to connect the public cloud infrastructure to the on-premise cloud resources.

The selection of a deployment model depends on the opportunities to increase earnings and reduce costs i.e. capital expenses (CAPEX) and operating expenses (OPEX). Such opportunities can also have an element of timeliness associated with it, in that decisions that lead to losses today could be done with a vision of increased earnings and cost reductions in a foreseeable future.

### 4.2.2 Adoption and Consumption Strategies

The selection of strategies for enterprise cloud computing is critical for IT capability as well as for the earnings and costs the organization experiences, motivating efforts toward convergence of business strategies and IT. Some critical questions toward this convergence in the enterprise cloud paradigm are as follows:

- Will an enterprise cloud strategy increase overall business value?
- Are the effort and risks associated with transitioning to an enterprise cloud strategy worth it?
- Which areas of business and IT capability should be considered for the enterprise cloud?
- Which cloud offerings are relevant for the purposes of an organization?
- How can the process of transitioning to an enterprise cloud strategy be piloted and systematically executed?

These questions are addressed from two strategic perspectives: (1) adoption and (2) consumption. Figure 4.1 illustrates a framework for enterprise cloud adoption strategies, where an organization makes a decision to adopt a cloud computing model based on fundamental drivers for cloud computing—scalability, availability, cost and convenience. The notion of a Cloud Data Center (CDC) is used, where the CDC could be an external, internal or federated provider of infrastructure, platform or software services.

An optimal adoption decision cannot be established for all cases because the types of resources (infrastructure, storage, software) obtained from a CDC depend on the size of the organisation understanding of IT impact on business, predictability of workloads, flexibility of existing IT landscape and available budget/resources for testing and piloting. The strategic decisions using these four basic drivers are described in following, stating objectives, conditions and actions.

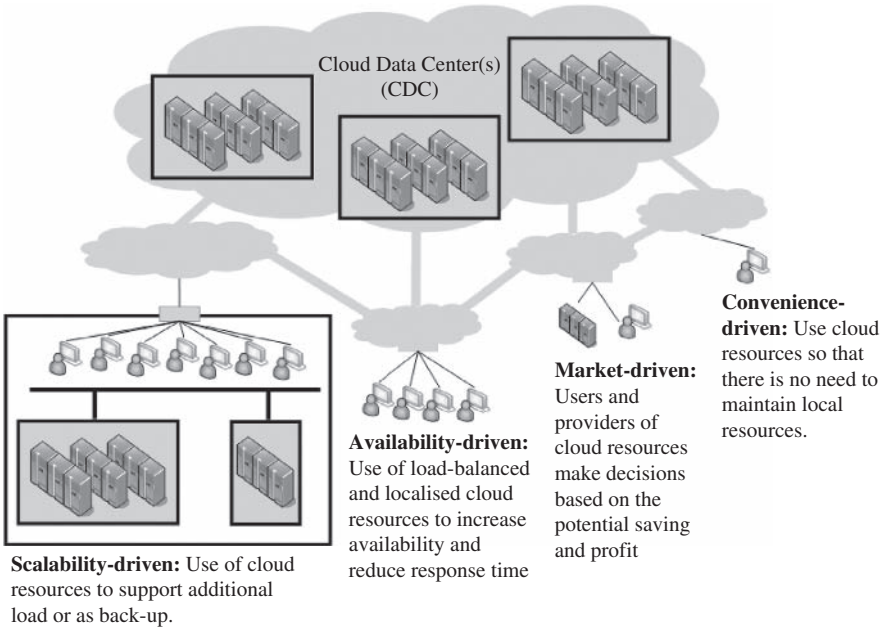
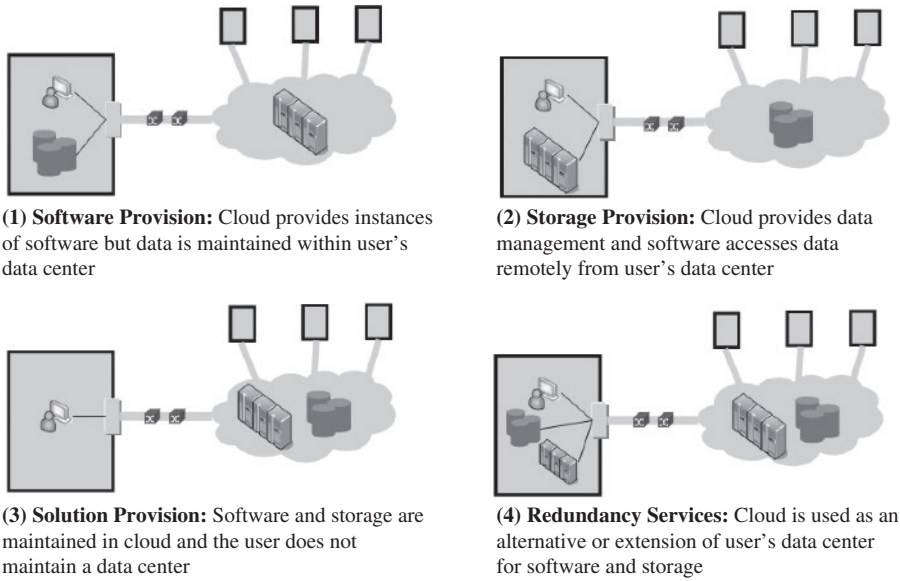



**FIGURE 4.1.** Enterprise cloud adoption strategies using fundamental cloud drivers.

1. **Scalability-Driven Strategy.** The objective is to support increasing workloads of the organization without investment and expenses exceeding returns. The conditions are that the effort, costs (CAPEX and OPEX) and time involved in accessing and installing IT capability on a CDC are less than going through a standard hardware and software procurement and licensing process. Scalability will often make use of the IaaS delivery model because the fundamental need of the organization is to have compute power or storage capacity readily available.
2. **Availability-Driven Strategy.** Availability has close relations to scalability but is more concerned with the assurance that IT capabilities and functions are accessible, usable and acceptable by the standards of users. This is hence the objective of this basic enterprise cloud strategy. The conditions of this strategy are that there exist unpredictable usage peaks and locales, yet the risks (probability and impact) of not being able to satisfy demand outweigh the costs of acquiring the IT capability from a CDC.
3. **Market-Driven Strategy.** This strategy is more attractive and viable for small, agile organizations that do not have (or wish to have) massive investments in their IT infrastructure. The objective here is to identify and acquire the "best deals" for IT capabilities as demand and supply change, enabling ongoing reductions in OPEX and CAPEX. There is however always the need to support customer-driven service management based



**(1) Software Provision:** Cloud provides instances of software but data is maintained within user's data center

**(2) Storage Provision:** Cloud provides data management and software accesses data remotely from user's data center

**(3) Solution Provision:** Software and storage are maintained in cloud and the user does not maintain a data center

**(4) Redundancy Services:** Cloud is used as an alternative or extension of user's data center for software and storage

**FIGURE 4.2.** Enterprise cloud consumption strategies.

on their profiles and requests service requirements [9]. The conditions for this strategy would be the existence of standardized interfaces between and across CDCs, where the means by which customers access their resources on the CDC, deploy software/data and migrate software/data are uniformed. Ongoing efforts in the *Open Cloud Computing Interface (OCCI) Working Group* and the *Open Cloud Consortium (OCC)* are steps toward achieving these standards. Other features such as bidding, negotiation, service discovery and brokering would also be required at communal, regional or global scales.

4. **Convenience-Driven Strategy.** The objective is to reduce the load and need for dedicated system administrators and to make access to IT capabilities by users easier, regardless of their location and connectivity (e.g. over the Internet). The expectation is that the cost of obtaining IT capabilities from a CDC and making them accessible to users is significantly lower than the cost of having a dedicated administrator. However, it should be noted that, according to a recent Gartner study [10], the major reason for discontinuing with cloud-related strategies is the difficulty with integration, ahead of issues with the costs of services.

The consumption strategies make a distinction between data and application logic because there are questions of programming models used, data sensitivity, software licensing and expected response times that need to be considered. Figure 4.2 illustrates a set of enterprise cloud consumption strategies, where an

organization makes decisions about how to best deploy its data and software using its internal resources and those of a selected CDC.

There are four consumptions strategies identified, where the differences in objectives, conditions and actions reflect the decision of an organization to trade-off hosting costs, controllability and resource elasticity of IT resources for software and data. These are discussed in the following.

1. **Software Provision.** This strategy is relevant when the elasticity requirement is high for software and low for data, the controllability concerns are low for software and high for data, and the cost reduction concerns for software are high, while cost reduction is not a priority for data, given the high controllability concerns for data, that is, data are highly sensitive. Implementing this strategy sees an organization requesting either software to be delivered as a service (SaaS) by the CDC or access to some portion of the CDC's compute infrastructure as a service (IaaS), such that it can deploy its application software on the provisioned resources. However, the organization chooses to maintain its data internally and hence needs to provide a means for the software running in the CDC to access data within its domain. This will entail changing some properties at the firewall or maintaining additional, supplementary software for secure access such as VPN, application-level proxy/gateway or wrapper software that could make the data base accessible via a remote messaging or service interface. According to a recent Gartner survey [10], the major hindrance to SaaS adoption is still the pricing and the lack of compelling indicators that the long-term investment in SaaS will be more cost-effective than traditional on-site maintenance of software.
2. **Storage Provision.** This strategy is relevant when the elasticity requirements is high for data and low for software, while the controllability of software is more critical than for data. This can be the case for data intensive applications, where the results from processing in the application are more critical and sensitive than the data itself. Furthermore, the cost reduction for data resources is a high concern, whereas cost for software, given its criticality, is not an issue for the organization within reasonable means. Other advantages of this strategy include the ease of sharing data between organizations, availability, fast provisioning, and management of storage utilization, because storage is a resource that is constantly in demand. Hasan, Yurcik and Myagmar [11] show in their study of storage service providers that reputation as storage vendors and the existence of established business relationships are major success and sustainability factors in this market.
3. **Solution Provision.** This strategy is relevant when the elasticity and cost reduction requirements are high for software and data, but the controllability requirements can be entrusted to the CDC. It is not the case that controllability is an insignificant requirement; it is rather the case that the

organization trusts the CDC sufficiently to manage access and usage control of its software and data. In some cases the organization might have greater trust in the CDC maintaining and securing its applications and data than it does in its own administrative capabilities. In other words, there are perceived gains in controllability for placing the entire IT solution (software and data) in the domain of the CDC. Solution provision also seemed like a more viable strategy than software or storage provision strategies, given the limitations of bandwidth between software and data that persists, especially for query-intensive solutions. Such a strategy is also attractive for testing systems, because these generally will not contain sensitive data (i.e., only test data) and are not the production-time versions of the software.

4. **Redundancy Services.** This strategy can be considered as a hybrid enterprise cloud strategy, where the organization switches between traditional, software, storage or solution management based on changes in its operational conditions and business demands. The trade-offs between controllability and cost reduction will therefore vary based on changes in load experienced by the organization. The strategy is referred to as the "redundancy strategy" because the CDC is used for situations such as disaster recovery, fail-over and load-balancing. Software, storage or solution services can be implemented using redundancy, such that users are redirected for the purpose of maintaining availability of functionality or performance/response times experienced by the user of the service. Business continuity is then the objective of this strategy, given that downtime and degradation of QoS can result in massive losses. There is however a cost for redundancy, because the subscription and access to redundant services needs to be maintained.

Even though an organization may find a strategy that appears to provide it significant benefits, this does not mean that immediate adoption of the strategy is advised or that the returns on investment will be observed immediately. There are still many issues to be considered when moving enterprise applications to the cloud paradigm.

## 4.3 ISSUES FOR ENTERPRISE APPLICATIONS ON THE CLOUD

Enterprise Resource Planning (ERP) is the most comprehensive definition of enterprise application today. The purpose of ERP solutions is to equip enterprises with a tool to optimize their underlying business processes with a seamless, integrated information flow from suppliers through to manufacturing and distribution [12] and the ability to effectively plan and control all resources [13], [14], necessary in the face of growing consumer demands, globalization and competition [15]. For these reasons, ERP solutions have emerged as the core of successful information management and the enterprise backbone of

nearly any organization [16]. Organizations that have successfully implemented the ERP systems are reaping the benefits of having integrating working environment, standardized process and operational benefits to the organization [17]. However, as the market rapidly changes, organizations need new solutions for remaining competitive, such that they will constantly need to improve their business practices and procedures. For this reason the enterprise cloud computing paradigm is becoming attractive as a potential ERP execution environment. Nevertheless, such a transition will *require a balance of strategic and operational steps guided by socio-technical considerations, continuous evaluation, and tracking mechanisms* [18].

One of the first issues is that of infrastructure availability. Al-Mashari [19] and Yasser [20] argued that adequate IT infrastructure, hardware and networking are crucial for an ERP system's success. It is clear that ERP implementation involves a complex transition from legacy information systems and business processes to an integrated IT infrastructure and common business process throughout the organization. Hardware selection is driven by the organization's choice of an ERP software package. The ERP software vendor generally certifies which hardware (and hardware configurations) must be used to run the ERP system. This factor has always been considered critical [17]. The IaaS offerings hence bear promising, but also challenging future scenarios for the implementation of ERP systems.

One of the ongoing discussions concerning future scenarios considers varying infrastructure requirements and constraints given different workloads and development phases. Recent surveys among companies in North America and Europe with enterprise-wide IT systems showed that nearly all kinds of workloads are seen to be suitable to be transferred to IaaS offerings. Interest in use for production applications is nearly as high as for test and development use. One might think that companies will be much more comfortable with test and development workloads at an external service provider than with production workloads, where they must be more cautious. However, respondents in surveys said they were either just as comfortable, or only up to 8% less comfortable, deploying production workloads on "the cloud" as they were deploying test and development workloads. When the responses for all workload types are aggregated together, two-thirds or more of firms are willing to put at least one workload type into an IaaS offering at a service provider [21]. More technical issues for enterprise cloud computing adoption arise when considering the operational characteristics and behaviors of transactional and analytical applications [22], which extend and underlie the capabilities of ERP.

### 4.3.1 Considering Transactional and Analytical Capabilities

Transactional type of applications or so-called OLTP (On-line Transaction Processing) applications, refer to a class of systems that manage transaction-oriented applications, typically using relational databases. These applications rely on strong ACID (*atomicity, consistency, isolation, durability*) properties

and are relatively write/update-intensive. Typical OLTP-type ERP components are sales and distributions (SD), banking and financials, customer relationship management (CRM) and supply chain management (SCM). These applications face major technical and non-technical challenges to deploy in cloud environments. For instance, they provide mission-critical functions and enterprises have clear security and privacy concerns. The classical transactional systems typically use a shared-everything architecture, while cloud platforms mostly consist of shared-nothing commodity hardware. ACID properties are also difficult to guarantee given the concurrent cloud-based data management and storage systems. Opportunities arise while the highly complex enterprise applications are decomposed into simpler functional components, which are characterized and engineered accordingly. For example, salesforce.com focuses on CRM-related applications and provides both a hosted software and development platform. Companies such as taleo.com offer on-demand Human Relationship (HR) applications and are gaining momentum in the SaaS market. A suite of core business applications as managed services can also be an attractive option, especially for small and medium companies. Despite the big engineering challenges, leading software providers are offering tailored business suite solutions as hosted services (e.g. SAP Business ByDesign).

Secondly, analytical types of applications or so-called OLAP (Online Analytical Processing) applications, are used to efficiently answer multi-dimensional queries for analysis, reporting, and decision support. Typical OLAP applications are business reporting, marketing, budgeting and forecasting, to name a few, which belong to the larger Business Intelligence (BI) category [23]. These systems tend to be read-most or read-only, and ACID guarantees are typically not required. Because of its data-intensive and data-parallel nature, this type of applications can benefit greatly from the elastic compute and storage available in the cloud. Business Intelligence and analytical applications are relatively better suited to run in a cloud platform with a shared-nothing architecture and commodity hardware. Opportunities arise in the vision of Analytics as a Service, or Agile Analytics [24]. Data sources residing within private or public clouds, can be processed using elastic computing resources on-demand, accessible via APIs, web services, SQL, BI, and data mining tools. Of course security, data integrity, and other issues can not be overlooked, but a cloud way offers a direction with unmatched performance and TCO (total cost of ownership) benefits toward large-scale analytic processing. Leading providers have been offering on-demand BI and analytics services (e.g. BusinessObjects' ondemand.com and Cognos Now!). Startup companies and niche players (e.g. Brist, PivotLink, Oco) provide a range of SaaS BI products from reporting to ETL (Extract, Transform, Load).

One can conclude that analytical applications will benefit more than their transactional counterparts from the opportunities created by cloud computing, especially on compute elasticity and efficiency. The success of separate functional components such as CRM and HR offered as hosted services has been observed, such that predictions of an integrated suite of core enterprise

functionalities emerging as on-demand solutions for small and medium enterprises can be made, given that the transition challenges can be overcome.

## 4.4 TRANSITION CHALLENGES

The very concept of cloud represents a leap from traditional approach for IT to deliver mission critical services. With any leap comes the gap of risk and challenges to overcome. These challenges can be classified in five different categories, which are the five aspects of the enterprise cloud stages: build, develop, migrate, run, and consume (Figure 4.3).

At the moment, the private and hybrid models (Section 4.2) appear as most relevant for comprehensive ERP transitioning and will hence be considered in this discussion of challenges. The first immediate challenge facing organizations, embarking on this transition, is the understanding of the state of their own IT assets and what is already, can, and cannot be *sublimed* (the process of transitioning from physical to less visible vapor). Based on the information gathered by this audit they need to evaluate what can be salvaged from the existing infrastructure and how high in the cloud stack they should venture. Most companies are likely to stick to IaaS. However, major development shops may envisage delving into the PaaS and SaaS sphere. Shifting the current architecture requires us to scrap a good chunk of it, which should be taken literally. However, we already see a sprawl of small cloud island appearing within corporations. As this *unplanned cloud* spreads throughout the organization, coherency becomes a challenge. The requirement for a company-wide cloud approach should then become the number one priority of the CIO, especially when it comes to having a coherent and cost effective development and migration of services on this architecture.

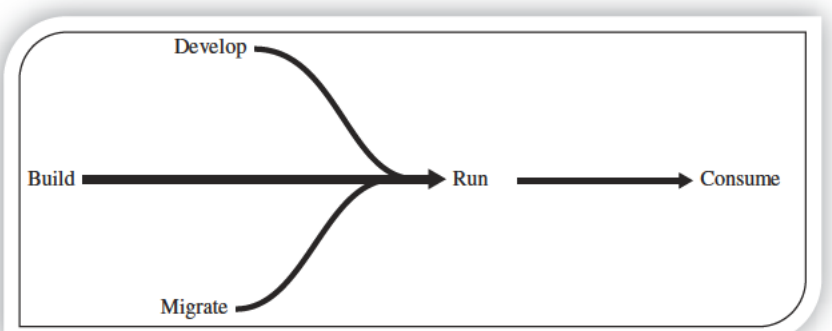



**FIGURE 4.3.** Five stages of the cloud.

A second challenge is migration of existing or "legacy" applications to "the cloud." The expected average lifetime of ERP product is ~15 years, which means that companies will need to face this aspect sooner than later as they try to evolve toward the new IT paradigm. An applications migration is not a straightforward process. It is risky, and doesn't always guarantee a better service delivery. Firstly, the guarantee that the migration process can be agnostic of the underlying, chosen cloud technology must be provided. If such a process can be automated, a company will still face the same amount of planning, negotiation and testing required for risk mitigation as classical software. It is yet to be proven that companies will be able to balance such expense with the cost cutting, scalability and performance promised by the cloud.

Because migrating to the cloud depends on the concept of decoupling of processes, work needs to be organized using a process (or service) centric model, rather than the standard "silo" one commonly used in IT: server, network, storage, database, and so on. Not all applications will be able to handle such migration without a tedious and costly overall reengineering. However, if companies decide to (re-) develop from scratch, they will face a completely different kind of hurdle: governance, reliability, security/trust, data management, and control/predictability [25] [26]. The ownership of enterprise data conjugated with the integration with others applications integration in and from outside the cloud is one of the key challenges. Future enterprise application development frameworks will need to enable the separation of data management from ownership. From this, it can be extrapolated that SOA, as a style, underlies the architecture and, moreover, the operation of the enterprise cloud.

Challenges for cloud operations can be divided into running the enterprise cloud and running applications on the enterprise cloud. In the first case, companies face difficulties in terms of the changing IT operations of their day today operation. It requires upgrading and updating all the IT department's components. One of these has been notoriously hard to upgrade: the human factor; bringing staff up to speed on the requirements of cloud computing with respect to architecture, implementation, and operation has always been a tedious task.

Once the IT organization has either been upgraded to provide cloud or is able to tap into cloud resource, they face the difficulty of maintaining the services in the cloud. The first one will be to maintain interoperability between in-house infrastructure and service and the CDC (Cloud Data Center).

Furthermore, inasmuch as elasticity is touted as the killer features for enterprise applications, most of the enterprise applications do not really face such wild variations in load to date, such that they need to resort to the cloud for on-demand capacity. More fundamentally, most enterprise apps don't support such features (apart from the few ones built from the ground up for clouds). Before leveraging such features, much more basic functionalities are problematic: monitoring, troubleshooting, and comprehensive capacity planning are actually missing in most offers. Without such features it becomes

very hard to gain visibility into the return on investment and the consumption of cloud services.

Today there are two major cloud pricing models: Allocation based and Usage based [27]. The first one is provided by the poster child of cloud computing, namely, Amazon. The principle relies on allocation of resource for a fixed amount of time. The second model does not require any reservation of resource, and the cloud would simply allocate them as a per need basis. When this model combine two typical pricing models: Utility (pay-per-use) and subscription based (fixed per duration charge)—we see the number of variation of offers exploding. Finding the right combination of billing and consumption model for the service is a daunting task. However, the challenge doesn't just stop there. As companies need to evaluate the offers they need to also include the hidden costs such as lost IP, risk, migration, delays and provider overheads. This combination can be compared to trying to choose a new mobile with carrier plan. Not to mention that some providers are proposing to introduce a subscription scheme in order to palliate with their limited resource within their unlimited offer. This is similar to what ISPs would have done with their content rationing strategies. The market dynamics will hence evolve alongside the technology for the enterprise cloud computing paradigm.

## 4.5 ENTERPRISE CLOUD TECHNOLOGY AND MARKET EVOLUTION

This section discusses the potential factors which will influence this evolution of cloud computing and today's enterprise landscapes to the enterprise computing paradigm, featuring the convergence of business and IT and an open, service oriented marketplace.

### 4.5.1 Technology Drivers for Enterprise Cloud Computing Evolution

One of the main factors driving this evolution is the concern by all the stakeholders in the cloud ecosystem of vendor lock-in, which includes the barriers of proprietary interfaces, formats, and protocols employed by the cloud vendors. As an increasing number of organizations and enterprises formulate cloud adoption strategies and execution plans, requirements of open, interoperable standards for cloud management interfaces and protocols, data formats and so on will emerge. This will put pressure on cloud providers to build their offering on open interoperable standards to be considered as a candidate by enterprises. There have been a number initiatives emerging in this space. For example, OGF OCCI [28] for compute clouds, SNIA CDMI [29] for storage and data management, DMTF Virtualization Management (VMAN) [30], and DMTF Cloud Incubator [31], to name a few of these standardization initiatives. Widespread participation in these initiatives is still lacking especially amongst the big cloud vendors like Amazon, Google, and Microsoft, who currently do not actively participate in these efforts. True interoperability across

the board in the near future seems unlikely. However, if achieved, it could lead to facilitation of advanced scenarios and thus drive the mainstream adoption of the enterprise cloud computing paradigm. Another reason standards-based cloud offers are critical for the evolution and spread of this paradigm is the fact that standards drive choice and choice drives the market. From another perspective, in the presence of standards-based cloud offers, third party vendors will be able to develop and offer value added management capabilities in the form of independent cloud management tools. Moreover, vendors with existing IT management tools in the market would be able to extend these tools to manage cloud solutions, hence facilitating organizations to preserve their existing investments in IT management solutions and use them for managing their hybrid cloud deployments.

Part of preserving investments is maintaining the assurance that cloud resources and services powering the business operations perform according to the business requirements. Underperforming resources or service disruptions lead to business and financial loss, reduced business credibility, reputation, and marginalized user productivity. In the face of lack of control over the environment in which the resources and services are operating, enterprise would like sufficient assurances and guarantees to eliminate performance issues, and lack of compliance to security or governance standards (e.g. PCI, HPIAA, SOX, etc.) which can potentially lead to service disruptions, business loss, or damaged reputation. Service level agreements (SLA) can prove to be a useful instrument in facilitating enterprises' trust in cloud-based services. Currently, the cloud solutions come with primitive or non existing SLAs. This is surely bound to change; as the cloud market gets crowded with increasing number of cloud offers, providers have to gain some competitive differentiation to capture larger share of the market. This is particularly true for market segments represented by enterprises and large organizations. Enterprise will be particularly interested to choose the offering with sophisticated SLAs providing assurances for the issues mentioned above.

Another important factor in this regard is lack of insights into the performance and health of the resources and service deployed on the cloud, such that this is another area of technology evolution that will be pushed. Currently, cloud providers don't offer sophisticated monitoring and reporting capabilities which can allow customers to comprehend and analyze the operations of these resources and services. However, recently, solutions have started to emerge to address this issue [32–34]. Nonetheless, this is one of the areas where cloud providers need to improve their offerings. It is believed that the situation will then improve because the enterprise cloud adoption phenomenon will make it imperative for the cloud providers to deliver sophisticated monitoring and reporting capabilities for the customers. This requirement would become ever more critical with the introduction of sophisticated SLAs, because customers would like to get insights into the service and resource behaviors for detecting SLA compliance violations. Moreover, cloud providers would need to expose this information through a standardized programmatic

interface so customers can feed this information into their planning tools. Another important advancement that would emerge is to enable third-party independent vendors to measure the performance and health of resources and services deployed on cloud. This would prove to be a critical capability empowering third-party organizations to act as independent auditors especially with respect to SLA compliance auditing and for mediating the SLA penalty related issues.

Looking into the cloud services stack (IaaS, PaaS, SaaS) [1], the applications space or SaaS has the most growth potential. As forecasted by the analyst IDC [35], applications will account for 38% of $44.2 billion cloud services market by 2103. Enterprises have already started to adopt some SaaS based solutions; however, these are primarily the edge applications like supplier management, talent management, performance management and so on as compared to the core business processes. These SaaS based applications need to be integrated to the backed applications located on-premise. These integration capabilities would drive the mainstream SaaS adoption by enterprises. Moreover, organizations would opt for SaaS applications from multiple service providers to cater for various operational segments of an enterprise. This adds an extra dimension of complexity because the integration mechanisms need to weave SaaS application from various providers and eventually integrate them to the on-premise core business applications seamlessly. Another emerging trend in the cloud application space is the divergence from the traditional RDBMS based data store backend. Cloud computing has given rise to alternative data storage technologies (Amazon Dynamo, Facebook Cassandra, Google BigTable, etc.) based on key-type storage models as compared to the relational model, which has been the mainstream choice for data storage for enterprise applications. Recently launched NoSQL movement is gaining momentum, and enterprise application developers will start adopting these alternative data storage technologies as a data layer for these enterprise applications.

The platform services segment of the cloud market is still in its early phases. Currently, PaaS is predominantly used for developing and deploying situational applications to exploit the rapid development cycles especially to cope with the scenarios that are constrained by limited timeframe to bring the solutions to the market. However, most of the development platforms and tools addressing this market segment are delivered by small startups and are proprietary technologies. Since the technologies are still evolving, providers are focusing on innovation aspects and gaining competitive edge over other providers. As these technologies evolve into maturity, the PaaS market will consolidate into a smaller number of service providers. Moreover, big traditional software vendors will also join this market which will potentially trigger this consolidation through acquisitions and mergers. These views are along the lines of the research published by Gartner [36]. Key findings published in this report were that through 2011, development platforms and tools targeting cloud deployment will remain highly proprietary and until then, the focus of these service providers would be on innovation over market viability. Gartner

predicts that from 2011 to 2015 market competition and maturing developer practises will drive consolidation around a small group of industry-dominant cloud technology providers.

The IaaS segment is typically attractive for small companies or startups that don't have enough capital and human resources to afford internal infrastructures. However, enterprises and large organizations are experimenting with external cloud infrastructure providers as well. According to a Forrester report published last year [37], enterprises were experimenting with IaaS in various contexts for examples R&D-type projects for testing new services and applications and low-priority business applications. The report also quotes a multinational telecommunication company running an internal cloud for wikis and intranet sites and was beginning to test mission critical applications. The report also quotes the same enterprise to have achieved 30% cost reduction by adopting the cloud computing model. However, we will see this trend adopted by an increasing number of enterprises opting for IaaS services. A recent Forrester report [21] published in May 2009 supports this claim as according to the survey, 25% enterprises are either experimenting or thinking about adopting external cloud providers various types of enterprise applications and workloads. As more and more vendors enter the IaaS cloud segment, cloud providers will strive to gain competitive advantage by adopting various optimization strategies or value added services to the customers. Open standards based cloud interfaces will gain attraction for increasing the likelihood of being chosen by enterprises. Cloud providers will provide transparency into their operations and environments through sophisticated monitoring and reporting capabilities for the consumer to track and control their costs based on the consumption and usage information.

A recent report published by Gartner [36] presents an interesting perspective on cloud evolution. The report argues that as cloud services proliferate, services would become complex to be handled directly by the consumers. To cope with these scenarios, meta-services or cloud brokerage services will emerge. These brokerages will use several types of brokers and platforms to enhance service delivery and, ultimately service value. According to Gartner, before these scenarios can be enabled, there is a need for brokerage business to use these brokers and platforms. According to Gartner, the following types of cloud service brokerages (CSB) are foreseen:

- Cloud Service Intermediation. An intermediation broker providers a service that directly enhances a given service delivered one or more service consumers, essentially on top of a given service to enhance a specific capability.
- Aggregation. An aggregation brokerage service combines multiple services into one or more new services.
- Cloud Service Arbitrage. These services will provide flexibility and opportunistic choices for the service aggregator.

The above shows that there is potential for various large, medium, and small organizations to become players in the enterprise cloud marketplace. The dynamics of such a marketplace are still to be explored as the enabling technologies and standards continue to mature.

## 4.6 BUSINESS DRIVERS TOWARD A MARKETPLACE FOR ENTERPRISE CLOUD COMPUTING

In order to create an overview of offerings and consuming players on the market, it is important to understand the forces on the market and motivations of each player. Porter [39] offers a framework for the industry analysis and business strategy development. Within this framework the actors, products, and business models are clarified and structured.

The Porter model consists of five influencing factors/views (forces) on the market (Figure 4.4). In the traditional economic model, competition among rival companies drives profits to zero, thus forcing companies to strive for a competitive advantage over their rivals. The intensity of rivalry on the market is traditionally influenced by industry-specific characteristics [40]:

- Rivalry: The amount of companies dealing with cloud and virtualization technology is quite high at the moment; this might be a sign for high

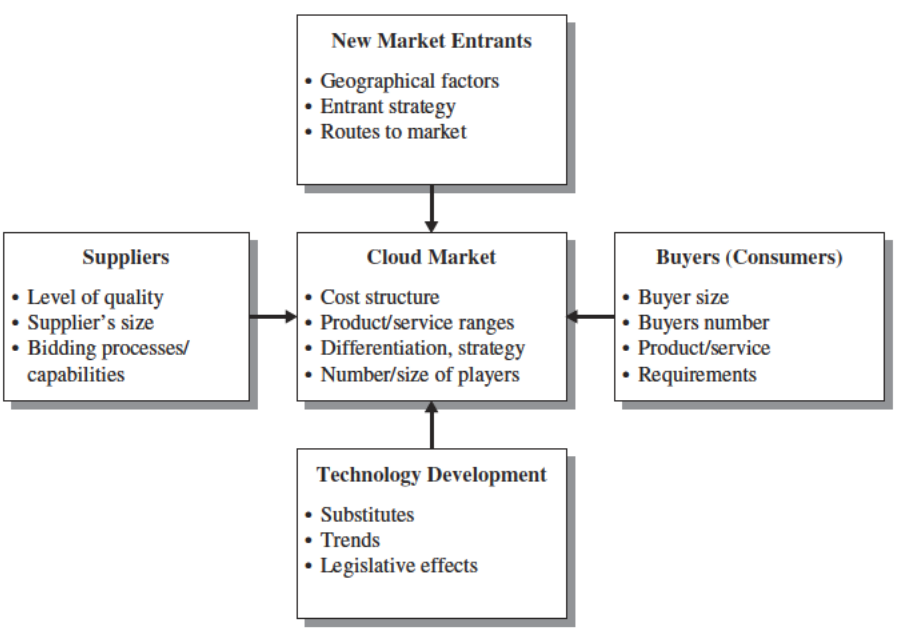



**FIGURE 4.4.** Porter's five forces market model (adjusted for the cloud market) [38].

rivalry. But also the products and offers are quite various, so many niche products tend to become established.

- Obviously, the cloud-virtualization market is presently booming and will keep growing during the next years. Therefore the fight for customers and struggle for market share will begin once the market becomes saturated and companies start offering comparable products.
- The initial costs for huge data centers are enormous. By building up federations of computing and storing utilities, smaller companies can try to make use of this scale effect as well.
- Low switching costs or high exit barriers influence rivalry. When a customer can freely switch from one product to another, there is a greater struggle to capture customers. From the opposite point of view high exit barriers discourage customers to buy into a new technology. The trends towards standardization of formats and architectures try to face this problem and tackle it. Most current cloud providers are only paying attention to standards related to the interaction with the end user. However, standards for clouds interoperability are still to be developed [41].

Monitoring the cloud market and observing current trends will show when the expected shakeout will take place and which companies will have the most accepted and economic offers by then [42]. After this shakeout, the whole buzz and hype around cloud computing is expected to be over and mature solutions will evolve. It is then that concrete business models will emerge. These business models will consider various fields, including e-business, strategy, supply chain management and information systems [43], [44], but will now need to emphasize the value of ICT-driven innovations for organizations and users [45]. Furthermore, static perspectives on business models will not be viable in such an ICT-centric environment, given that organizations often have to review their business model in order to keep in line with fast changing environments like the cloud market for the ICT sector [46], from development to exploitation [45]. With a few exceptions [47–49], most literature has taken a fairly static perspective on business models.

For dynamic business models for ICT, it is important to incorporate general phases of a product development. Thus, phasing models help to understand how innovation and change affect the evolution of the markets, and its consequences for company strategies and business models [50]. As argued by Kijl [51], the three main phases are R&D, implementation/roll-out, and market phase, which include the subphases of market offerings, maturity, and decline. These three main phases, influencing the business model, are used in a framework, visualized in Figure 4.5.

Figure 4.5 also outlines which external drivers are expected to play a dominant role throughout the phases [52]. Technology is the most important driver for the development of new business models in the ICT sector and will

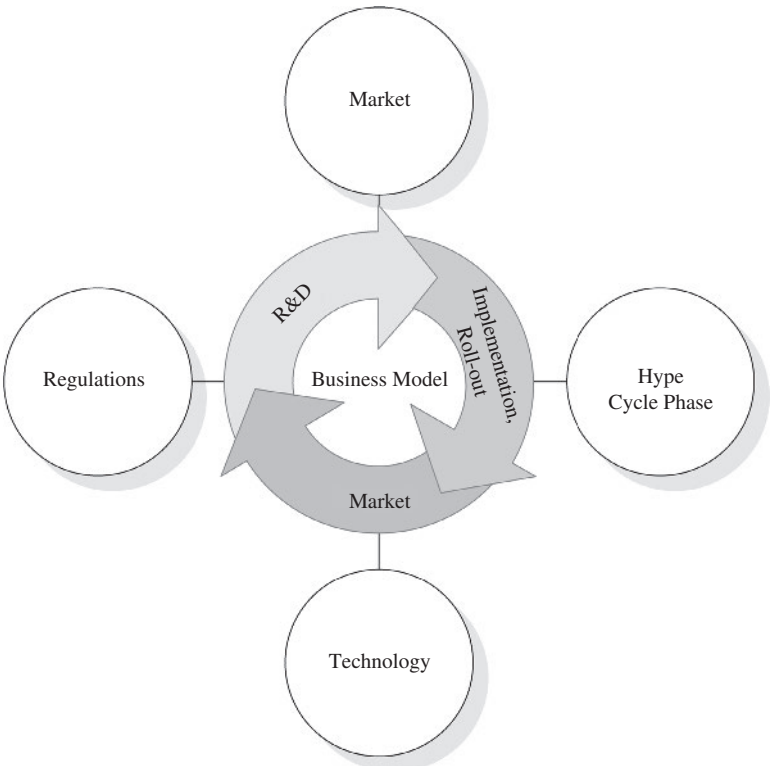



**FIGURE 4.5.** Dynamic business models (based on [49] extend by influence factors identified by [50]).

undoubtedly continue to be a major influencer of the enterprise cloud computing evolution. However, it can be assumed that market developments and regulation can also trigger opportunities for the development of new products and services in this paradigm. Changes in market opportunities or regulation enable new product and/or service definitions as well as underlying business models. There are already various players in the cloud computing market offering various services [53]. However, they still struggle for market share and it is very likely that they will diversify their offers in order to meet all the market requirements. During these efforts, some of them will reach the mainstream and achieve a critical mass for the market while others will pass away or exist as niche offers after the shakeout. It is increasingly necessary to have a comprehensive model of drivers for business model dynamics [40], [45], [54], including knowledge of actors, products and market. This is also motivated by Porter [40], Kijl [51], and Bouwman and MacInnes [52]. How then would such a business model be manifested?

## 4.7 THE CLOUD SUPPLY CHAIN

One indicator of what such a business model would look like is in the complexity of deploying, securing, interconnecting and maintaining enterprise landscapes and solutions such as ERP, as discussed in Section 4.3. The concept of a Cloud Supply Chain (C-SC) and hence Cloud Supply Chain Management (C-SCM) appear to be viable future business models for the enterprise cloud computing paradigm. The idea of C-SCM represents the management of a network of interconnected businesses involved in the end-to-end provision of product and service packages required by customers. The established understanding of a supply chain is two or more parties linked by a flow of goods, information, and funds [55], [56] A specific definition for a C-SC is hence: "two or more parties linked by the provision of cloud services, related information and funds." Figure 4.6 represents a concept for the C-SC, showing the flow of products along different organizations such as hardware suppliers, software component suppliers, data center operators, distributors and the end customer.

Figure 4.6 also makes a distinction between innovative and functional products in the C-SC. Fisher classifies products primarily on the basis of their demand patterns into two categories: primarily functional or primarily innovative [57]. Due to their stability, functional products favor competition, which leads to low profit margins and, as a consequence of their properties, to low inventory costs, low product variety, low stockout costs, and low obsolescence [58], [57]. Innovative products are characterized by additional (other) reasons for a customer in addition to basic needs that lead to purchase, unpredictable demand (that is high uncertainties, difficult to forecast and variable demand), and short product life cycles (typically 3 months to 1 year). Cloud services

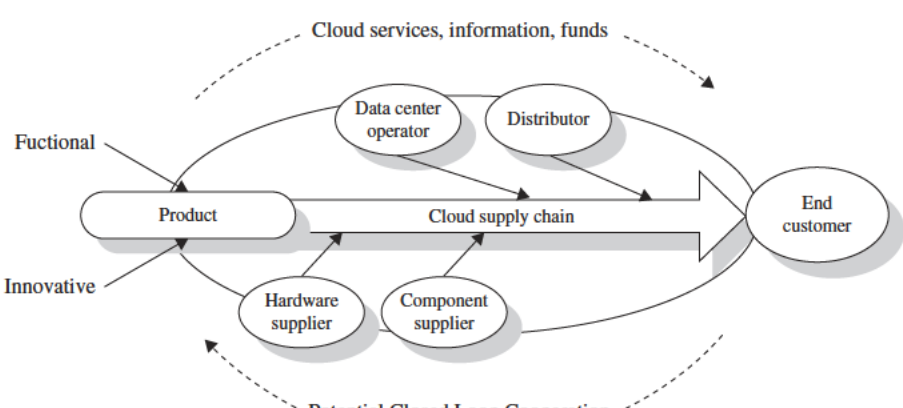



**FIGURE 4.6.** Cloud supply chain (C SC).

should fulfill basic needs of customers and favor competition due to their reproducibility. They however also show characteristics of innovative products as the demand is in general unpredictable (on-demand business model) and have due to adjustments to competitors and changing market requirements very short development circles. Table 4.1 presents a comparison of Traditional

**TABLE 4.1. Comparison of Traditional and Emerging ICT Supply Chains[a]**

| | Traditional Supply Chain Concepts | | Emerging ICT Concepts |
|---|---|---|---|
| | Efficient SC | Responsive SC | Cloud SC |
| Primary goal | Supply demand at the lowest level of cost | Respond quickly to demand (changes) | Supply demand at the lowest level of costs and respond quickly to demand |
| Product design strategy | Maximize performance at the minimum product cost | Create modularity to allow postponement of product differentiation | Create modularity to allow individual setting while maximizing the performance of services |
| Pricing strategy | Lower margins because price is a prime customer driver | Higher margins, because price is not a prime customer driver | Lower margins, as high competition and comparable products |
| Manufacturing strategy | Lower costs through high utilization | Maintain capacity flexibility to meet unexpected demand | High utilization while flexible reaction on demand |
| Inventory strategy | Minimize inventory to lower cost | Maintain buffer inventory to meet unexpected demand | Optimize of buffer for unpredicted demand, and best utilization |
| Lead time strategy | Reduce but not at the expense of costs | Aggressively reduce even if the costs are significant | Strong service level agreements (SLA) for ad hoc provision |
| Supplier strategy | Select based on cost and quality | Select based on speed, flexibility, and quantity | Select on complex optimum of speed, cost, and flexibility |
| Transportation strategy | Greater reliance on low cost modes | Greater reliance on responsive modes | Implement highly responsive and low cost modes |

[a] Based on references 54 and 57.

Supply Chain concepts such as the efficient SC and responsive SC and a new concept for emerging ICT as the cloud computing area with cloud services as traded products.

This mixed characterization is furthermore reflected when it comes to the classification of efficient vs. responsive Supply Chains. Whereas functional products would preferable go into efficient Supply Chains, the main aim of responsive Supply Chains fits the categorization of innovative product. Cachon and Fisher [58] show that within the supply chain the sharing of information (e.g. accounting and billing) is not the only contributor to SC cost, but it is the management and restructuring of services, information, and funds for an optimization of the chain that are expensive [60].

## 4.8 SUMMARY

In this chapter, the enterprise cloud computing paradigm has been discussed, with respect to opportunities, challenges and strategies for cloud adoption and consumption. With reference to Gartner's hype cycle [61], enterprise cloud computing and related technologies is already in the first phase called "inflated expectation," but it is likely to move quite quickly into the "trough of disillusionment" [62]. At the moment the main adopters of cloud computing are small companies and startups, where the issue of legacy of IT investments is not present. Large enterprises continue to wrestle with the arguments for adopting such a model, given the perceived risks and effort incurred. From an analysis of existing offerings, the current models do not fully meet the criteria of enterprise IT as yet. Progress continues at an accelerated pace, boosted by the rich and vibrant ecosystem being developed by start-up and now major IT vendors. It can hence be foreseen that the enterprise cloud computing paradigm could see a rise within the next 10 years. Evidence is found in the increasing development of enterprise applications tailored for this environment and the reductions in cost for development, testing and operation. However, the cloud model will not predate the classical way of consuming software services to extinction; they will just evolve and adapt. It will have far reaching consequences for years to come within the software, IT services vendors and even IT hardware, as it reshapes the IT landscape.

## ACKNOWLEDGMENTS

This chapter combines insights that have been drawn from various EU and Invest-NI funded projects in SAP Research Belfast. These include SLA@SOI (FP7-216556), RESERVOIR (FP7-215605), XtreemOS (FP6-IST-033576), and Virtex (Invest-NI).

## REFERENCES


1. NIST, "Working Definition of Cloud Computing," 2009.
2. Cloud Security Alliance, "Guidance for Critical Areas of Focus in Cloud Computing," 2009.
3. Amazon, "Amazon Virtual Private Cloud."
4. OpSource, "OpSource Cloud."
5. Skytap, "Virtual Lab."
6. OpenCirrus, "OpenCirrus Cloud Computing Research Testbed."
7. ENKI, "Outsourced IT Operations."
8. CohesiveFT, "VPN Cubed."
9. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I., "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst. 25, 6 (Jun. 2009)*, 2009, pp. 599–616.
10. Ben Pring and Twiggy Lo, "Dataquest Insight: SaaS Adoption Trends in the U.S. and UK," 2009.
11. Hasan, R., Yurcik, W., and Myagmar, S., "The evolution of storage service providers: techniques and challenges to outsourcing storage," *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability (Fairfax, VA, USA, November 11 11, 2005). StorageSS '05. ACM*, 2005.
12. McDermott, T, "MES and ERP: Creating Synergy with Industry Specific Solu tions," *APICS The Performance Advantage, vol. 9, no. 11, November 1999*, 1999, pp. 40–3.
13. Miller, GJ, "Lean and ERP: Can they Co exist?," *PROACTION Management Consultants www.proaction.net/HTML_papers/LeanERPCompat.html.*
14. Sandoe, K., Corbitt, G., and Boykin, R., *Enterprise Integration*, New York: 2001.
15. Ferguson, B, "Implementing Supply Chain Management," *Production and Inven tory Management Journal, vol. 41, no. 2*, 2000, pp. 64–7.
16. Nash, K.S, "Companies don't learn from previous IT snafus," *ComputerWorld 32–3*, 2000.
17. T.R. Bhatti, "CRITICAL SUCCESS FACTORS FOR THE IMPLEMENTA TION OF ENTERPRISE RESOURCE PLANNING (ERP): EMPIRICAL VALIDATION," *The Second International Conference on Innovation in Informa tion Technology (IIT'05)*, p. 2005.
18. Al Mudimigh A., Zairi M., and Al Mashiri M., "ERP software implementation: an integrative framework," *European Journal of Information Systems, 10*, 2001, pp. 216–226.
19. Al Mashari, M., "Enterprise resource planning (ERP) systems: a research agenda," *Industrial Management & Data Systems, Vol. 102, No. 3*, 2002, pp. 165–170.
20. Yasser Jarrar, "ERP Implementation and Critical Success Factors, The Role and Impact of Business Process Management," *Proceedings of The 2000 IEE Interna tional Conference on Management of Innovation and Technology, Singapore*, 2000, pp. 167–178.
21. Frank E. Gillet, "Conventional Wisdom is Wrong About Cloud IaaS," 2009.

22. D. Abadi, "Data Management in the Cloud: Limitations and Opportunities," *IEEE Data Engineering Bulletin, 32(1)*, 2009.
23. E. Thomsen, *OLAP Solutions: Building Multi dimensional Information Systems*, 2002.
24. Oliver Ratzesberger, "Analytics as a Service."
25. Daniel J. and Abadi, H., "Data Management in the Cloud: Limitations and Opportunities," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2009.
26. Jay Heiser and Mark Nicolett, "Assessing the Security Risks of Cloud Comput ing," *Gartner Report*, Jun. 2008.
27. Weiss, A., "Computing in the clouds," *netWorker 11, 4 (Dec. 2007)*, 2007.
28. Open Cloud Computing Interface (OCCI) Working Group, "OCCI."
29. SNIA, "Cloud Data Management Initiative (CDMI)."
30. DMTF, "Virtualization Management Initiative (VMAN)."
31. DMTF, "Open Cloud Standards Incubator."
32. Amazon, "CloudWatch."
33. Hyperic, "CloudStatus."
34. Nimsoft, "Unified Monitoring."
35. IDC, "IT Cloud Services Forecast 2009–2013."
36. Mark Driver, "Cloud Application Infrastructure Technologies Need Seven More Years to Mature," 2008.
37. James Staten, "Is Cloud Ready for Enterprises?," 2008.
38. Daryl C. Plummer and L. Frank Kenney, "Three Types of Cloud Brokerages Will Enhance Cloud Services," 2009.
39. Michael E. Porter, *Competitive Strategy: Techniques for Analyzing Industries and Competitors*, 1980.
40. M.E. Porter, "Competitive Strategy: Techniques for Analyzing Industries and Competitors," *Ed. The Free Press*, 1980.
41. EGEE, "Enabling Grids for E sciencE An EGEE Comparative study: Grids and Clouds evolution or revolution?," 2008.
42. D. Reeves, *Data center strategies: Vmware: Welcome to the game*, 2008.
43. J. Hedman and T. Kalling, "The business model concept: theoretical under pinnings and empirical illustrations," *European Journal of Information Sciences*, 2003, pp. 49–59.
44. S.M. Shafer, H.J. Smith, and J.C. Linder, "The power ofbusiness models. Business Horizons," *European Journal of Information Sciences*, vol. 48, 2005, pp. 199–207.
45. M. de Reuver, H. Bouwman, and I. MacInnes, "What Drives Business Model Dynamics? A Case Survey," *Management of eBusiness, 2007. WCMeB 2007. Eighth World Congress on the*, Jul. 2007, pp. 2–2.
46. A. Afuah and C. Tucci, "Internet Business Models and Strategies," *Boston McGraw Hill*, 2003.
47. P. Andries, K. Debackere, and B. Van Looy, "Effective business model adaptation strategies for new technology based ventures," *PREBEM Conference on Business Economics*, vol. 9, 2006.

48. I. MacInnes, "Dynamic business model framework for emerging technologies," *International Journal of Services Technology and Management*, vol. 6, 2005, pp. 3–19.
49. V.L. Vaccaro and D.Y. Cohn, "The Evolution of Business Models and Marketing Strategies in the Music Industry," *JMM The International Journal on Media Management*, vol. 6, 2004, pp. 46–58.
50. A. Afuah and C.L. Tucci, "Internet Business Models and Strategies," *Mcgraw Hill*, 2001.
51. B. Kijl, "Developing a dynamic business model framework for emerging mobile services," *ITS 16th European Regional Conference*, 2005.
52. H. Bouwman and I. MacInnes, "Dynamic Business Model Framework for Value Webs," *39th Annual Hawaii International Conference on System Sciences*, 2006.
53. M. Crandell, "Defogging cloud computing: A taxonomy refresh the net," *Gigaom*, Sep. 2008.
54. L.M. Vaquero, L. Rodero Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," *Strategic Management Journal*, vol. 22, 2009.
55. Tsay, A., Agrawal, N., and Nahmias, S., "Modeling supply chain contracts: a review," *Tayur, S., Ganeshan, R., and Magazine, M., editors, Quantitative Models for Supply Chain Management, Kluwer's International Series in Operations Research & Management Science, chapter 10, pages 299–336. Kluwer Academic Publishers, Boston, MA, USA. F.S.Hillier, series editor*, 1998.
56. Paulitsch, M., "Dynamic Coordination of Supply Chains," 2003.
57. Fisher, M., "What is the right supply chain for your product?," *Harvard Business Review, pages 105–116*, 1997.
58. Lee, H., "Aligning supply chain strategies with product uncertainties," *California Management Review, 44(3):105–119*, 2002.
59. Chopra, S. and Meindl, P., "Supply Chain Management: Strategy, Planning, and Operation," *Prentice Hall, Inc., Upper Saddle River, New Jersey, USA, 1st edition.*, 2001.
60. Cachon, G. and Fisher, M., "Supply chain inventory management and the value of shared information," *Management Science, 46(8):1032–1048*, 2000.
61. Gartner, "Hype Cycle for Emerging Technologies," 2008.
62. McKinsey&Company, "Clearing the air on cloud computing," Mar. 2009.