

Barrie Sosinsky

Cloud Computing

Explore the cloud with this complete guide

Understand all platforms and technologies

Use Google, Amazon, or Microsoft Web services



Bible

Petitioner has added alpha characters to pages (a)- (g). Otherwise, it leaves the original page numbering.

MICROSOFT CORP.
EXHIBIT 1007

The book you need to succeed!

Cloud Computing Bible

Cloud Computing Bible

Barrie Sosinsky



WILEY

Wiley Publishing, Inc.

Cloud Computing Bible

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2011 by Wiley Publishing, Inc., Indianapolis, Indiana

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-90356-8

10 9 8 7 6 5 4 3

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, 201-748-6011, fax 201-748-6008, or online at <http://www.wiley.com/go/permissions>.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Library of Congress Control Number: 2010941512

Trademarks: Wiley, the Wiley logo, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

This book is dedicated to my sister Gina Sosinsky, with love.

About the Author

Barrie Sosinsky has written about computers and technology for more than 25 years beginning with writing about personal computers for the Boston Computer Society in the early 1980s. He has published books on operating systems, applications, databases, desktop publishing, and networking for publishers such as Que, Sybex, Ventana, IDG, Wiley, and others and seen the industry change and reinvent itself several times. His last book was Wiley's *Networking Bible*.

At heart Barrie is a PC enthusiast. He loves building computers, finding and learning about new applications that allow him to do new things, and keeping up with the latest advances in the field of computer technology, which he believes is just in its infancy. Having lived long enough to see the Boston Red Sox win not one but two World Series, he remains committed to living long enough to see grandchildren and to see someone clone a woolly mammoth. To this list (replacing the Red Sox) he adds the new milestone of holding a universal translator in his hands; a device he believes will appear within this decade.

Barrie lives in Medfield Massachusetts about 25 miles southwest of Boston with his six cats Stormy, Shadow, Smokey, Scamper, Slate, and Spat as writing companions; Scout the wonder dog; his son Joseph; his daughter Allie; and his wife Carol; surrounded by pine trees, marauding deer, wild turkeys, and the occasional fox and coyote.

You can reach Barrie at bsosinsky@mindspring.com, where he welcomes your comments and suggestions.

Credits

Senior Acquisitions Editor

Stephanie McComb

Project Editor

Martin V. Minner

Technical Editor

Benjamin M. Schupak

Copy Editor

Gwenette Gaddis

Editorial Director

Robyn Siesky

Editorial Manager

Rosemarie Graham

Business Manager

Amy Knies

Senior Marketing Manager

Sandy Smith

Vice President and Executive Group**Publisher**

Richard Swadley

Vice President and Executive Publisher

Barry Pruett

Project Coordinator

Patrick Redmond

Graphics and Production Specialists

Nikki Gately

Andrea Hornberger

Quality Control Technician

Lindsay Littrell

Proofreading and Indexing

Evelyn Wellborn

Sherry Massey

Preface

Cloud Computing Bible is Wiley's general introduction to an important topic in large book format. A Bible is a book that is meant to be read by knowledgeable readers who are not subject matter experts in a topic but want to have an in-depth introduction to the various individual subjects contained within. It is assumed that the reader of this book may be a generalist, a developer, a system architect, a programmer, or perhaps something else, and therefore the content in a Bible must contain information for each member of this book's audience.

Cloud computing is a vast topic that encompasses many different subjects. To adequately describe what cloud computing offers, we must discuss infrastructure, service-oriented architectures, social networking, unique protocols, open and standard Application Programming Interfaces (APIs), and dozens of other topics. Even a large book can address many of these topics in only an introductory manner. However, this book tries to give you at least the basic information you need on all the related topics, as well as pointers to additional information sources.

In the last several years, many books have been published on cloud computing. Each book has attempted to present some element of the topic for a particular audience. In this book, I do not make the assumption that you are a particular type of reader, nor do I assume that you are approaching the topic with a fresh view. This Bible was written to serve as the introductory course in the topic at a university level, but it is not a textbook. You can pick up and read this book at any particular chapter because the material doesn't build upon itself.

Many topics in this book are unique to this book and are based on published information that is both current and timely. In researching this book, I attempted to bring into the discussion all the new trends, experiments, and products that have made cloud computing such a dynamic area.

Acknowledgments

I want to acknowledge the editorial team at Wiley for giving me the chance to do this book and to work with them again. I also want to thank my literary agent, Matt Wagner, for his assistance; without his vision, this book would not have been created. Finally, I want to thank my family for allowing me the time I spent away from them writing this book.

Contents at a Glance

Preface	xi
Acknowledgments	xiii

Part I: Examining the Value Proposition 1

Chapter 1: Defining Cloud Computing	3
Chapter 2: Assessing the Value Proposition	23
Chapter 3: Understanding Cloud Architecture	45
Chapter 4: Understanding Services and Applications by Type	65

Part II: Using Platforms 91

Chapter 5: Understanding Abstraction and Virtualization	93
Chapter 6: Capacity Planning	113
Chapter 7: Exploring Platform as a Service	133
Chapter 8: Using Google Web Services	151
Chapter 9: Using Amazon Web Services	179
Chapter 10: Using Microsoft Cloud Services	205

Part III: Exploring Cloud Infrastructures. 229

Chapter 11: Managing the Cloud	231
Chapter 12: Understanding Cloud Security	249

Part IV: Understanding Services and Applications 269

Chapter 13: Understanding Service Oriented Architecture	271
Chapter 14: Moving Applications to the Cloud	297
Chapter 15: Working with Cloud-Based Storage	311
Chapter 16: Working with Productivity Software	339
Chapter 17: Using Webmail Services	361
Chapter 18: Communicating with the Cloud	383
Chapter 19: Using Media and Streaming	403

Part V: Using the Mobile Cloud 423

Chapter 20: Working with Mobile Devices	425
Chapter 21: Working with Mobile Web Services	445

Contents

About the Author	viii
Preface	xi
Acknowledgments	xiii

Part I: Examining the Value Proposition	1
--	----------

Chapter 1: Defining Cloud Computing	3
--	----------

Defining Cloud Computing.....	4
Cloud Types	5
The NIST model.....	5
The Cloud Cube Model.....	6
Deployment models	7
Service models	9
Examining the Characteristics of Cloud Computing.....	13
Paradigm shift	13
Benefits of cloud computing.....	16
Disadvantages of cloud computing.....	18
Assessing the Role of Open Standards.....	19
Summary	22

Chapter 2: Assessing the Value Proposition	23
---	-----------

Measuring the Cloud's Value.....	24
Early adopters and new applications	26
The laws of cloudonomics.....	27
Cloud computing obstacles	29
Behavioral factors relating to cloud adoption.....	31
Measuring cloud computing costs	33
Avoiding Capital Expenditures.....	35
Right-sizing.....	36
Computing the Total Cost of Ownership	37
Specifying Service Level Agreements.....	39
Defining Licensing Models.....	41
Summary	42

Chapter 3: Understanding Cloud Architecture 45

Exploring the Cloud Computing Stack.....	46
Composability.....	46
Infrastructure.....	48
Platforms.....	49
Virtual Appliances.....	51
Communication Protocols.....	55
Applications.....	58
Connecting to the Cloud.....	58
The Jolicloud Netbook OS.....	60
Chromium OS: The Browser as an Operating System.....	61
Summary.....	64

Chapter 4: Understanding Services and Applications by Type 65

Defining Infrastructure as a Service (IaaS).....	66
IaaS workloads.....	67
Pods, aggregation, and silos.....	69
Defining Platform as a Service (PaaS).....	70
Defining Software as a Service (SaaS).....	71
SaaS characteristics.....	72
Open SaaS and SOA.....	73
Salesforce.com and CRM SaaS.....	75
Defining Identity as a Service (IDaaS).....	76
What is an identity?.....	76
Networked identity service classes.....	78
Identity system codes of conduct.....	79
IDaaS interoperability.....	79
User authentication.....	81
Authorization markup languages.....	84
Defining Compliance as a Service (CaaS).....	87
Summary.....	88

Part II: Using Platforms 91

Chapter 5: Understanding Abstraction and Virtualization 93

Using Virtualization Technologies.....	94
Load Balancing and Virtualization.....	95
Advanced load balancing.....	96
The Google cloud.....	97
Understanding Hypervisors.....	100
Virtual machine types.....	100
VMware vSphere.....	103

Understanding Machine Imaging	107
Porting Applications	108
The Simple Cloud API	109
AppZero Virtual Application Appliance	109
Summary	111
Chapter 6: Capacity Planning	113
Capacity Planning	114
Defining Baseline and Metrics	115
Baseline measurements	115
System metrics	118
Load testing	121
Resource ceilings	122
Server and instance types	126
Network Capacity	128
Scaling	131
Summary	132
Chapter 7: Exploring Platform as a Service	133
Defining Services	134
Salesforce.com versus Force.com: SaaS versus PaaS	135
Application development	137
Using PaaS Application Frameworks	138
Drupal	140
Eccentex AppBase 3.0	141
LongJump	143
Squarespace	145
WaveMaker	146
Wolf Frameworks	147
Summary	150
Chapter 8: Using Google Web Services	151
Exploring Google Applications	152
Surveying the Google Application Portfolio	153
Indexed search	154
The dark Web	155
Aggregation and disintermediation	156
Productivity applications and services	156
Enterprise offerings	159
AdWords	161
Google Analytics	162
Google Translate	163
Exploring the Google Toolkit	165
The Google APIs	166
Working with the Google App Engine	173
Summary	178

Chapter 9: Using Amazon Web Services.	179
Understanding Amazon Web Services.....	180
Amazon Web Service Components and Services	181
Working with the Elastic Compute Cloud (EC2)	185
Amazon Machine Images.....	186
Pricing models	187
System images and software	188
Creating an account and instance on EC2	190
Working with Amazon Storage Systems.....	196
Amazon Simple Storage System (S3)	196
Amazon Elastic Block Store (EBS)	197
CloudFront	199
Understanding Amazon Database Services	200
Amazon SimpleDB	200
Amazon Relational Database Service (RDS)	201
Choosing a database for AWS.....	202
Summary	202
Chapter 10: Using Microsoft Cloud Services	205
Exploring Microsoft Cloud Services	206
Defining the Windows Azure Platform.....	209
The software plus services approach.....	210
The Azure Platform	210
The Windows Azure service	211
Windows Azure AppFabric	214
Azure Content Delivery Network	216
SQL Azure	217
Windows Azure pricing.....	217
Windows Live services	219
Using Windows Live	221
Windows Live Essentials	224
Windows Live Home.....	226
Windows Live for Mobile	227
Summary	228
 Part III: Exploring Cloud Infrastructures	 229
 Chapter 11: Managing the Cloud	 231
Administrating the Clouds	232
Management responsibilities	234
Lifecycle management	237
Cloud Management Products	237

Emerging Cloud Management Standards.....	242
DMTF cloud management standards.....	243
Cloud Commons and SMI.....	244
Summary	248
Chapter 12: Understanding Cloud Security	249
Securing the Cloud	250
The security boundary	251
Security service boundary	253
Security mapping	255
Securing Data.....	256
Brokered cloud storage access	256
Storage location and tenancy.....	259
Encryption	260
Auditing and compliance	261
Establishing Identity and Presence	263
Identity protocol standards	264
Windows Azure identity standards.....	266
Presence	267
Summary	268
Part IV: Understanding Services and Applications	269
Chapter 13: Understanding Service Oriented Architecture	271
Introducing Service Oriented Architecture	272
Event-driven SOA or SOA 2.0	277
The Enterprise Service Bus	278
Service catalogs	281
Defining SOA Communications	282
Business Process Execution Language.....	286
Business process modeling	287
Managing and Monitoring SOA.....	292
SOA management tools	292
SOA security	293
The Open Cloud Consortium.....	294
Relating SOA and Cloud Computing.....	295
Summary	296
Chapter 14: Moving Applications to the Cloud.	297
Applications in the Clouds.....	298
Functionality mapping	299
Application attributes	300
Cloud service attributes.....	302

Contents

System abstraction	303
Cloud bursting	304
Applications and Cloud APIs	308
Summary	309
Chapter 15: Working with Cloud-Based Storage	311
Measuring the Digital Universe	312
Cloud storage in the Digital Universe	314
Cloud storage definition	315
Provisioning Cloud Storage	316
Unmanaged cloud storage	316
Managed cloud storage	320
Creating cloud storage systems	321
Virtual storage containers	322
Exploring Cloud Backup Solutions	325
Backup types	325
Cloud backup features	327
Cloud attached backup	331
Cloud Storage Interoperability	332
Cloud Data Management Interface (CDMI)	333
Open Cloud Computing Interface (OCCI)	335
Summary	337
Chapter 16: Working with Productivity Software	339
Using Productivity Applications	340
Characteristics of productivity software	341
Online Office Systems	342
Acrobat.com	343
Glide Digital	345
Google Docs	347
Microsoft Office Web Apps	351
ThinkFree Office	354
Zoho Office Suite	355
Summary	360
Chapter 17: Using Webmail Services	361
Exploring the Cloud Mail Services	362
Google Gmail	366
Mail2Web	369
Windows Live Hotmail	370
Yahoo! Mail	373
Working with Syndication Services	375
The RSS and Atom Protocols	376
Newsreaders	377
News Aggregators	378
Summary	382

Chapter 18: Communicating with the Cloud 383

Exploring Instant Messaging	384
Instant messaging clients	384
Instant messaging interoperability	387
Micro-blogs or Short Message Services	388
Exploring Collaboration Technologies	391
Using Social Networks	392
Features	393
List of social networking sites	394
Privacy and security	398
Interaction and interoperability	399
Summary	401

Chapter 19: Using Media and Streaming 403

Understanding the Streaming Process	404
Protocols in Use	405
The cloud computing advantages	405
Audio Streaming	407
Working with VoIP Applications	409
Skype	409
Google Voice and Google Talk	411
Video Streaming	412
Television in the cloud	413
Streaming video formats	416
YouTube	417
YouTube technologies	418
YouTube and the law	420
Summary	421

Part V: Using the Mobile Cloud 423

Chapter 20: Working with Mobile Devices 425

Defining the Mobile Market	426
Connecting to the cloud	427
Adopting mobile cloud applications	428
Feature phones and the cloud	429
Using Smartphones with the Cloud	430
Android	431
Apple iPhone	433
The App Store	434
MobileMe	435
iPhone apps hosting services	436

Contents

Research In Motion BlackBerry	438
Symbian	440
Windows Mobile	441
Summary	443
Chapter 21: Working with Mobile Web Services	445
Understanding Service Types	446
Mobile interoperability	446
Performing Service Discovery	450
Context-aware services	451
MEMS	453
Location awareness	454
Push services	455
The BlackBerry Push Service	457
The Lemonade Profile	459
Using SMS	459
Defining WAP and Other Protocols	461
Performing Synchronization	464
Summary	466
Index	467

Introduction

In the five months that I have been researching and writing *Cloud Computing Bible*, it has become clear to me that most people recognize that cloud computing is a big deal, even if they are not really clear why that is so. Every day newspaper and magazine articles and radio and TV stories report on cloud computing. The phrase “in the cloud” has entered into our colloquial language. You may have heard that the United States government has initiated a “cloud initiative,” or that nearly 75 percent of the developers at Microsoft are currently working on “cloud-related” products, or that a phone or service stores its data in the cloud. The cloud is therefore this amorphous entity that is supposed to represent the future of modern computing.

In reality, the cloud is something that you have been using for a long time now; it is the Internet, along with all the associated standards and protocols that provide a set of Web services to you. When you draw the Internet as a cloud, you are representing one of the essential characteristics of cloud computing: abstraction. In the cloud, resources are pooled and partitioned as needed, and communications are standards-based.

The Internet was begun as a network of networks, with an architecture that was redundant and could survive massive disruption. What the original system architects of the Internet could not have anticipated is that the size of resources attached to it would become massively scalable, which is the second characteristic of cloud computing.

Google’s infrastructure, for example, which is described in this book in Chapter 9, spans 30 data-centers around the world with over a million computers; infrastructure that Google now leases out to developers upon which applications may be staged. So the third and equally as important characteristic of cloud computing is that the cloud is a “utility” and that services are provided using a pay-as-you-go model.

A computing utility has been a dream of computer scientists and industry luminaries for several decades. With a utility model of computing, an application can start small and grow to be enormous overnight. This democratization of computing means that any application has the potential to scale, and that even the smallest seed planted in the cloud may be a giant.

Cloud computing will affect your life in the following ways in the next ten years:

- Applications in the cloud will replace applications that are local to your devices.
- Information will become cheaper, more ubiquitous, and easier to find because the cloud makes it cheaper to scale applications and connections to always-on networks such as wireless carriers that make the information always available.

Introduction

- The cloud will enable new social services by connecting users via social networks that are constructed using multiple cloud services.
- New applications will be easier to create and will be based on standard modular parts.
- It will lessen the role that proprietary operating systems have in our daily computing.
- You will be connected through the cloud wherever you are and at all times.

Frankly, it is hard to predict what new capabilities the cloud may enable. The cloud has a trajectory that is hard to plot and a scope that reaches into so many aspects of our daily life that innovation can occur across a broad range.

Many technologically savvy people have told me they don't understand what the fuss about cloud computing is; in fact, they believe there is nothing new about cloud computing, at least from a technological standpoint. Indeed, they have a point. The technologies that enable cloud computing—system and resource virtualization, thin clients (browsers, for example), virtual private networks and tunneling, and others—are all technologies that existed before anyone ever began to talk about cloud computing. That is all true. Cloud computing is a revolutionary way of architecting and implementing services based on evolutionary changes. *Cloud Computing Bible* attempts to explain how this all came about.

How to Read This Book

Cloud Computing Bible is made up of 21 chapters in five parts. To read this book and get the most out of it, you should know about basic computer operations and theory. You should be able to turn a computer on and know what operating system is running, how processing and input/output is used, and be able to connect with a browser to different Web sites. You should understand the basic user interface elements used by many browsers, such as Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, or Google Chrome.

These are basic skills without which it would be hard to effectively maximize the value contained in this book. If you don't have these skills, Wiley publishes a number of introductory computer books that will give them to you.

It doesn't matter which type of computer operating system you use because most of cloud computing is operating-system-neutral. Indeed, as time goes by, it may not matter whether you use a computer at all. Mobile devices such as smartphones and tablets are on their way to displacing computers in many venues. If you have some familiarity with smartphones, that would be helpful in understanding the last part of this book on mobile-based cloud applications, but it isn't a necessity.

Part I of the book, called "Examining the Value Proposition," defines what cloud computing is and why you should be interested in it. This vocabulary, along with description of cloud architectures

and types, will allow you to discuss cloud computing in a standard way and serves to give you a framework over which you can place all the different service types that make cloud computing such a rich area.

Part II, called “Using Platforms,” looks at the fundamental features that make a cloud computing application unique. You get a background in the concepts of abstraction and virtualization, along with methods for examining how applications are scaled. This part contains several chapters of vendor-specific services that are illustrative of different cloud computing models. In several chapters, I discuss vendors that are thought leaders in different fields of cloud computing. For infrastructure, I’ve chosen to highlight Amazon Web Services, and for platforms and services, you learn about the efforts of Google and Microsoft in cloud computing.

Part III, “Exploring Cloud Infrastructures,” contains two chapters about managing the cloud and working with the cloud securely. The cloud builds on standard distributed networking technologies, applied over systems with large resources, often over federated systems and services.

In Part IV, “Understanding Services and Applications,” the first two chapters describe Service Oriented Architecture and transactions—both of which are important principles in building cloud applications so they are efficient and interoperable—and moving applications to the cloud. The remaining chapters in Part IV describe different types of applications in common use in the cloud today. Those applications are the most highly developed ones in the cloud and have the largest number of users and services. The examples chosen are online backup and storage, Webmail, online productivity applications, messaging, and online media, particularly using streaming technologies.

The book rounds out with two chapters on “Using the Mobile Cloud,” Part V. These chapters describe the rise of the smartphone and its predecessor, the feature phone. These phones are supported by a host of Web services. Since 2008, more traffic has been flowing over wireless networks than wired networks, so it would be hard to underestimate how much impact mobile devices have on the cloud. For vast portions of the world, the cell phone is the only computer most people will know. Mobile Web services use different protocols and technologies and can take into account location and other user profile information that can use the cloud to create a rich user experience.

Please dive into whatever chapter interests you. I hope you enjoy reading about cloud computing as much as I enjoyed writing about it.

Icons

The icons in this book offer you a chance to learn a little more about a topic, refer to a discussion elsewhere in the book, address a problem, or get a little more help. This book offers the following icons:

Introduction

Caution

A Caution icon alerts you to a potential problem that you should be aware of. ■

Note

A Note icon points to a clarification or expansion of the topic being discussed. ■

Tip

Tips are shortcuts you can use to get something done more effectively. ■

Cross-Ref

A Cross-Ref icon provides a reference to related discussions that take place elsewhere in the book. ■

Because this isn't a how-to book, you will find fewer Cautions and Tips in this book than you might find in other Wiley Bibles. However, there are plenty of Notes and Cross-Refs to help guide you in these chapters.

Contacting Us

If, after reviewing this publication, you feel some important information was overlooked or you have any questions concerning cloud computing, you can contact us and let us know your views, opinions, complaints, or suggestions for the next revision.

You can reach the author, Barrie Sosinsky, at the following e-mail address: bsosinsky@mindspring.com.

Understanding Cloud Architecture

Cloud computing is a natural extension of many of the design principles, protocols, plumbing, and systems that have been developed over the past 20 years. However, cloud computing describes some new capabilities that are architected into an application stack and are responsible for the programmability, scalability, and virtualization of resources. One property that differentiates cloud computing is referred to as composability, which is the ability to build applications from component parts.

A platform is a cloud computing service that is both hardware and software. Platforms are used to create more complex software. Virtual appliances are an important example of a platform, and they are becoming a very important standard cloud computing deployment object.

Cloud computing requires some standard protocols with which different layers of hardware, software, and clients can communicate with one another. Many of these protocols are standard Internet protocols. Cloud computing relies on a set of protocols needed to manage interprocess communications that have been developed over the years. The most commonly used set of protocols uses XML as the messaging format, the Simple Object Access Protocol (SOAP) protocol as the object model, and a set of discovery and description protocols based on the Web Services Description Language (WSDL) to manage transactions.

Some completely new clients are under development that are specifically meant to connect to the cloud. These clients have as their focus cloud applications and services, and are often hardened and more securely connected. Two examples presented are Jolicloud and Google Chrome OS. They represent a new client model that is likely to have considerable impact.

IN THIS CHAPTER

Using the cloud computing stack to describe different models

Understanding how platforms and virtual appliances are used

Learning how cloud communications work

Discovering the new world of the cloud client

Exploring the Cloud Computing Stack

Cloud computing builds on the architecture developed for staging large distributed network applications on the Internet over the last 20 years. To these standard networking protocols, cloud computing adds the advances in system virtualization that became available over the last decade. The cloud creates a system where resources can be pooled and partitioned as needed. Cloud architecture can couple software running on virtualized hardware in multiple locations to provide an on-demand service to user-facing hardware and software. It is this unique combination of abstraction and metered service that separates the architectural requirements of cloud computing systems from the general description given for an n-tiered Internet application.

Many descriptions of cloud computing describe it in terms of two architectural layers:

- A client as a front end
- The “cloud” as a backend

This is a very simplistic description because each of these two components is composed of several component layers, complementary functionalities, and a mixture of standard and proprietary protocols. Cloud computing may be differentiated from older models by describing an encapsulated information technology service that is often controlled through an Application Programming Interface (API), thus modifying the services that are delivered over the network.

A cloud can be created within an organization’s own infrastructure or outsourced to another data-center. While resources in a cloud can be real physical resources, more often they are virtualized resources because virtualized resources are easier to modify and optimize. A compute cloud requires virtualized storage to support the staging and storage of data. From a user’s perspective, it is important that the resources appear to be infinitely scalable, that the service be measurable, and that the pricing be metered.

Composability

Applications built in the cloud often have the property of being built from a collection of components, a feature referred to as composability. A composable system uses components to assemble services that can be tailored for a specific purpose using standard parts. A composable component must be:

- **Modular:** It is a self-contained and independent unit that is cooperative, reusable, and replaceable.
- **Stateless:** A transaction is executed without regard to other transactions or requests.

It isn’t an absolute requirement that transactions be stateless, some cloud computing applications provide managed states through brokers, transaction monitors, and service buses. In rarer cases, full transactional systems are deployed in the clouds, but these systems are harder to architect in a distributed architecture.

Chapter 3: Understanding Cloud Architecture

Although cloud computing doesn't require that hardware and software be composable, it is a highly desirable characteristic from a developer or user's standpoint, because it makes system design easier to implement and solutions more portable and interoperable.

There is a tendency for cloud computing systems to become less composable for users as the services incorporate more of the cloud computing stack. From the standpoint of an IaaS (Infrastructure as a Service) vendor such as Amazon Web Services, GoGrid, or Rackspace, it makes no sense to offer non-standard machine instances to customers, because those customers are almost certainly deploying applications built on standard operating systems such as Linux, Windows, Solaris, or some other well-known operating system.

In the next step up the cloud computing stack, PaaS (Platform as a Service) vendors such as Windows Azure or Google AppEngine may narrow the definition of standard parts to standard parts that work with their own platforms, but at least from the standpoint of the individual platform service provider, the intent is to be modular for their own developers.

When you move to the highest degree of integration in cloud computing, which is SaaS (Software as a Service), the notion of composability for users may completely disappear. An SaaS vendor such as Quicken.com or Salesforce.com is delivering an application as a service to a customer, and there's no particular benefit from the standpoint of the service provider that the customer be able to compose its own custom applications. A service provider reselling an SaaS may have the option to offer one module or another, to customize the information contained in the module for a client, to sell the service under their own brand, or to perform some other limited kind of customization, but modifications are generally severely limited.

This idea that composability diminishes going up the cloud computing stack is from the user's point of view. If you are a PaaS or SaaS service provider and your task is to create the platform or service presented to the developer, reseller, or user, the notion of working with a composable system is still a very powerful one. A PaaS or SaaS service provider gets the same benefits from a composable system that a user does—these things, among others:

- Easier to assemble systems
- Cheaper system development
- More reliable operation
- A larger pool of qualified developers
- A logical design methodology

You encounter the trend toward designing composable systems in cloud computing in the widespread adoption of what has come to be called the Service Oriented Architecture (SOA). The essence of a service oriented design is that services are constructed from a set of modules using standard communications and service interfaces. An example of a set of widely used standards describes the services themselves in terms of the Web Services Description Language (WSDL), data exchange between services using some form of XML, and the communications between the services using the SOAP protocol. There are, of course, alternative sets of standards.

Cross-Ref

SOA is described in detail in Chapter 13. ■

What isn't specified is the nature of the module itself; it can be written in any programming language the developer wants. From the standpoint of the system, the module is a black box, and only the interface is well specified. This independence of the internal workings of the module or component means it can be swapped out for a different model, relocated, or replaced at will, provided that the interface specification remains unchanged. That is a powerful benefit to any system or application provider as their products evolve.

Infrastructure

Most large Infrastructure as a Service (IaaS) providers rely on virtual machine technology to deliver servers that can run applications. Virtual servers described in terms of a machine image or instance have characteristics that often can be described in terms of real servers delivering a certain number of microprocessor (CPU) cycles, memory access, and network bandwidth to customers. Virtual machines are containers that are assigned specific resources. The software that runs in the virtual machines is what defines the utility of the cloud computing system.

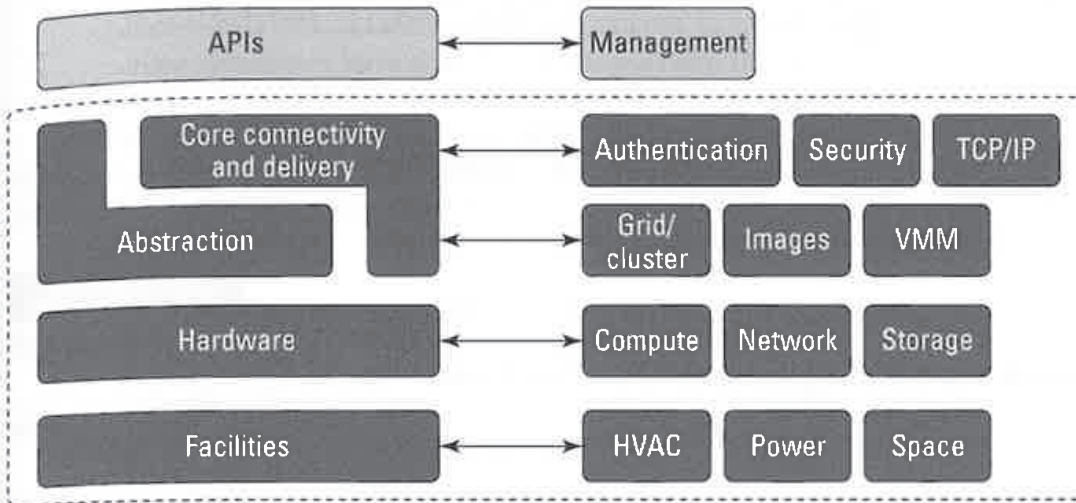
Figure 3.1 shows the portion of the cloud computing stack that is defined as the “server.” In the diagram, the API is shown shaded in gray because it is an optional component that isn't always delivered with the server. The VMM component is the Virtual Machine Monitor, also called a hypervisor. This is the low-level software that allows different operating systems to run in their own memory space and manages I/O for the virtual machines.

The notion of a virtual server presents to an application developer a new way of thinking about and programming applications. For example, when a programmer is creating software that requires several different tasks to be performed in parallel, he might write an application that creates additional threads of execution that must be managed by the application. When a developer creates an application that uses a cloud service, the developer can attach to the appropriate service(s) and allow the application itself to scale the program execution. Thus, an application such as a three-dimensional rendering that might take a long time for a single server to accomplish can be scaled in the cloud to many servers at once for a short period of time, accomplishing the task at a similar or lower price but at a much faster rate.

In future applications, developers will need to balance the architectural needs of their programs so their applications create new threads when it is appropriate or create new virtual machines. Applications will also need to be mindful of how they use cloud resources, when it is appropriate to scale execution to the cloud, how to monitor the instances they are running, and when not to expand their application's usage of the cloud. This will require a new way of thinking about application development, and the ability to scale correctly is something that will have to be architected into applications from the ground up.

FIGURE 3.1

This architectural diagram illustrates the portion of the cloud computing stack that is designated as the server.



Platforms

A platform in the cloud is a software layer that is used to create higher levels of service. As you learned in Chapter 1, many different Platform as a Service (PaaS) providers offer services meant to provide developers with different capabilities. In Chapter 7, PaaS is explored more thoroughly, but for now it is useful to cite three of the major examples that are provided in this book:

- Salesforce.com's Force.com Platform
- Windows Azure Platform
- Google Apps and the Google AppEngine

These three services offer all the hosted hardware and software needed to build and deploy Web applications or services that are custom built by the developer within the context and range of capabilities that the platform allows. Platforms represent nearly the full cloud software stack, missing only the presentation layer that represents the user interface. This is the same portion of the cloud computing stack that is a virtual appliance and is shown in Figure 3.2. What separates a platform from a virtual appliance is that the software that is installed is constructed from components and services and controlled through the API that the platform provider publishes.

It makes sense for operating system vendors to move their development environments into the cloud with the same technologies that have been successfully used to create Web applications. Thus, you might find a platform based on a Sun xVM hypervisor virtual machine that includes a NetBeans Integrated Development Environment (IDE) and that supports the Sun GlassFish

Part I: Examining the Value Proposition

Web stack programmable using Perl or Ruby. For Windows, Microsoft would be similarly interested in providing a platform that allowed Windows developers to run on a Hyper-V VM, use the ASP.NET application framework, support one of its enterprise applications such as SQL Server, and be programmable within Visual Studio—which is essentially what the Azure Platform does. This approach allows someone to develop a program in the cloud that can be used by others.

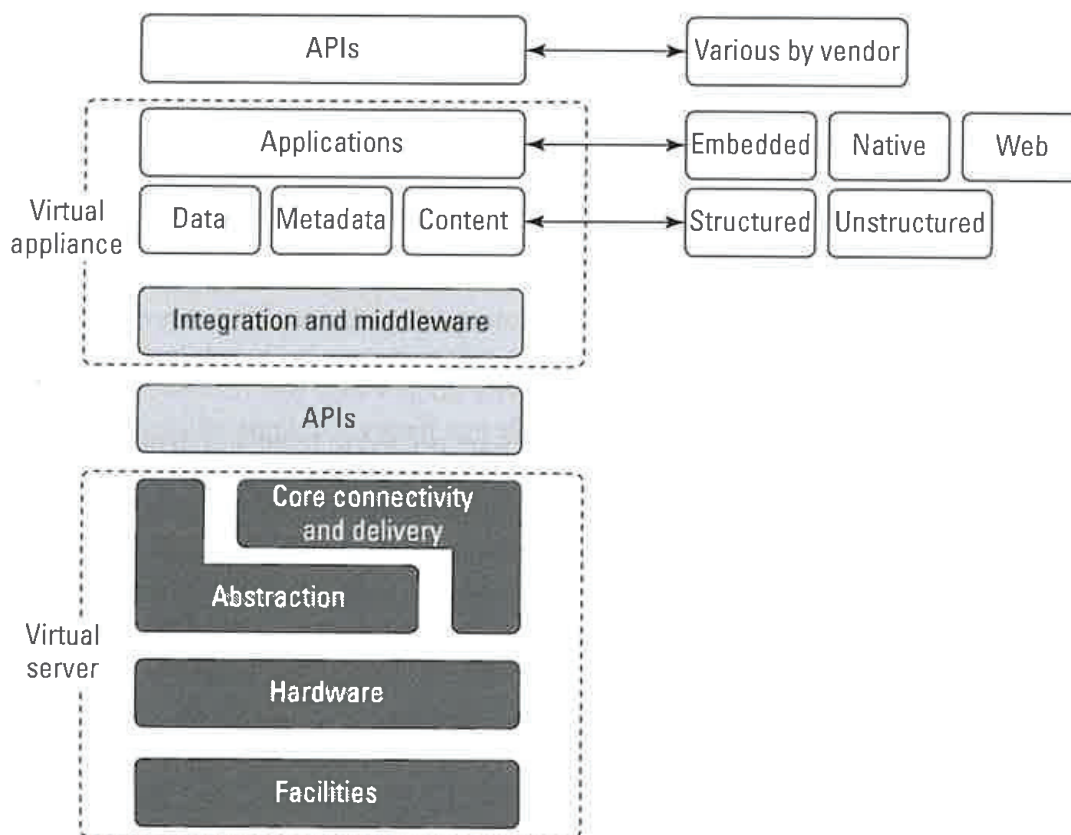
Platforms often come replete with tools and utilities to aid in application design and deployment. Depending upon the vendor, you may find developer tools for team collaboration, testing tools, instrumentation for measuring program performance and attributes, versioning, database and Web service integration, and storage tools. Most platforms begin by establishing a developer community to support the work done in the environment.

Note

To see the entire cloud computing stack, refer to Figure 1.5 in Chapter 1. ■

FIGURE 3.2

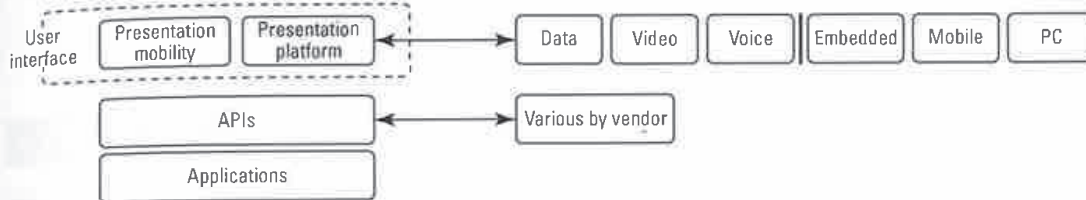
A virtual appliance is software that installs as middleware onto a virtual machine.



Just as a virtual appliance may expose itself to users through an API, so too an application built in the cloud using a platform service would encapsulate the service through its own API. Users would then interact with the platform, consuming services through that API, leaving the platform to manage and scale the service appropriately. Many platforms offer user interface development tools based on HTML, JavaScript, or some other technology. As the Web becomes more media-oriented, many developers have chosen to work with rich Internet environments such as Adobe Flash, Flex, or Air, or alternatives such as Windows Silverlight. A user interface abstracts away the platform API, making those services managed through the UI. Figure 3.3 shows the top portion of the cloud computing stack, which includes the API and the presentation functionality.

FIGURE 3.3

The top of the cloud computing interface includes the user interface and the API for the application layer.



The Application Programming Interface is one of the key differentiators separating cloud computing from the older models of Internet applications, because it is the means for instantiating resources needed to support applications. An API can control data flow, communications, and other important aspects of the cloud application. Unfortunately, each cloud vendor has their own cloud API, none of them are standard, and the best you can hope for is that eventually the major cloud vendor's APIs will interoperate and exchange data. For now, the use of proprietary APIs results in vendor lock-in, which is why you are advised to choose systems that implement APIs based on open standards.

Virtual Appliances

Applications such as a Web server or database server that can run on a virtual machine image are referred to as virtual appliances. The name *virtual appliance* is a little misleading because it conjures up the image of a machine that serves a narrow purpose. Virtual appliances are software installed on virtual servers—application modules that are meant to run a particular machine instance or image type. A virtual appliance is a platform instance. Therefore, virtual appliances occupy the middle of the cloud computing stack (refer to Figure 3.2).

A virtual appliance is a common deployment object in the cloud, and it is one area where there is considerable activity and innovation. One of the major advantages of a virtual appliance is that you can use the appliances as the basis for assembling more complex services, the appliance being one of your standardized components. Virtual appliances remove the need for application configuration and maintenance from your list of system management chores.

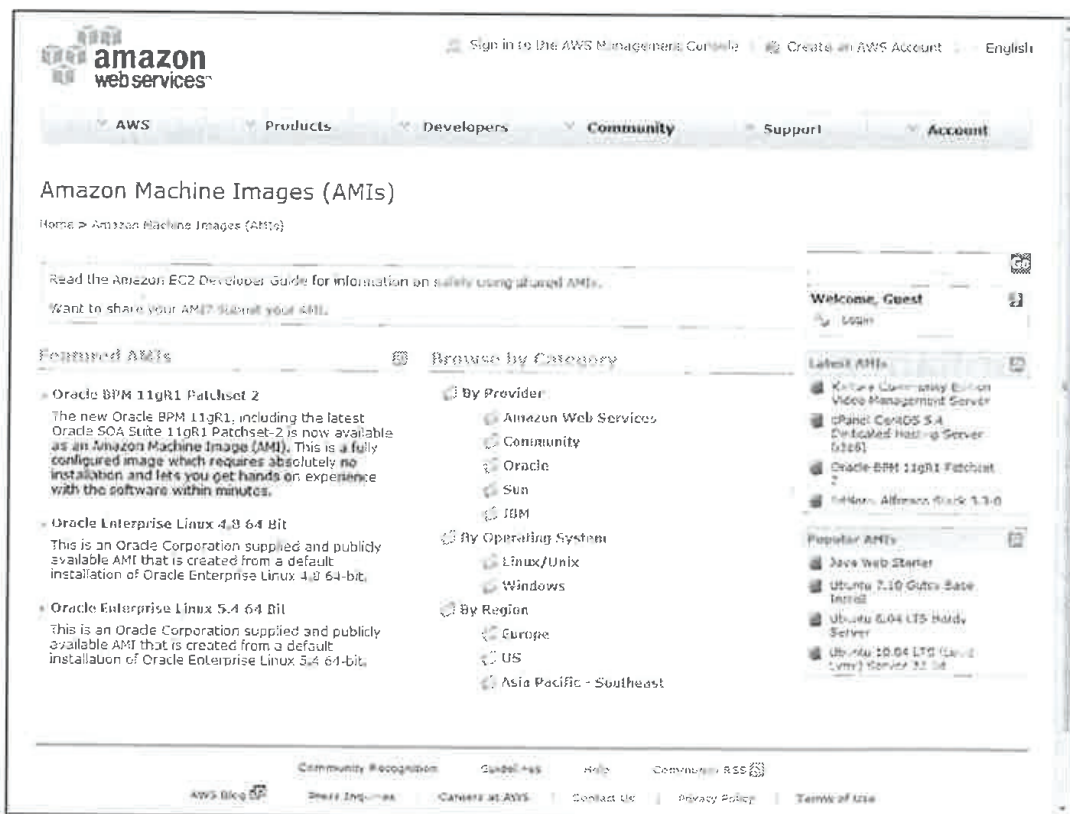
Part I: Examining the Value Proposition

You run across virtual appliances in IaaS systems such as Amazon's Elastic Compute Cloud (EC2), which is discussed in detail in Chapter 9. Amazon Machine Images are virtual appliances that have been packaged to run on the grid of Xen nodes that comprise the Amazon Web Service's EC2 system. Shown in Figure 3.4, the AMI library (<http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=171>) includes a variety of operating systems both proprietary and open source, a set of enterprise applications such as Oracle BPM, SQL Server, and even complete application stacks such as LAMP (Linux, Apache, MySQL, and PHP). Amazon has negotiated licenses from these vendors that are part of your per-use pricing when you run these applications on their servers.

Virtual appliances are far easier to install and run than an application that you must set up yourself. However, virtual appliances are also much larger than the application themselves would be because they are usually bundled with the operating system on which they are meant to run. An application that is 50 or 100MB might require a virtual appliance that is 500MB to 1GB in size. Usually, when a virtual appliance is created, the operating system is stripped of all excess functionality that isn't required by the appliance, because the appliance is meant to be used as is.

FIGURE 3.4

Amazon Machine Images are a collection of virtual appliances that you can install on their Xen hypervisor servers.



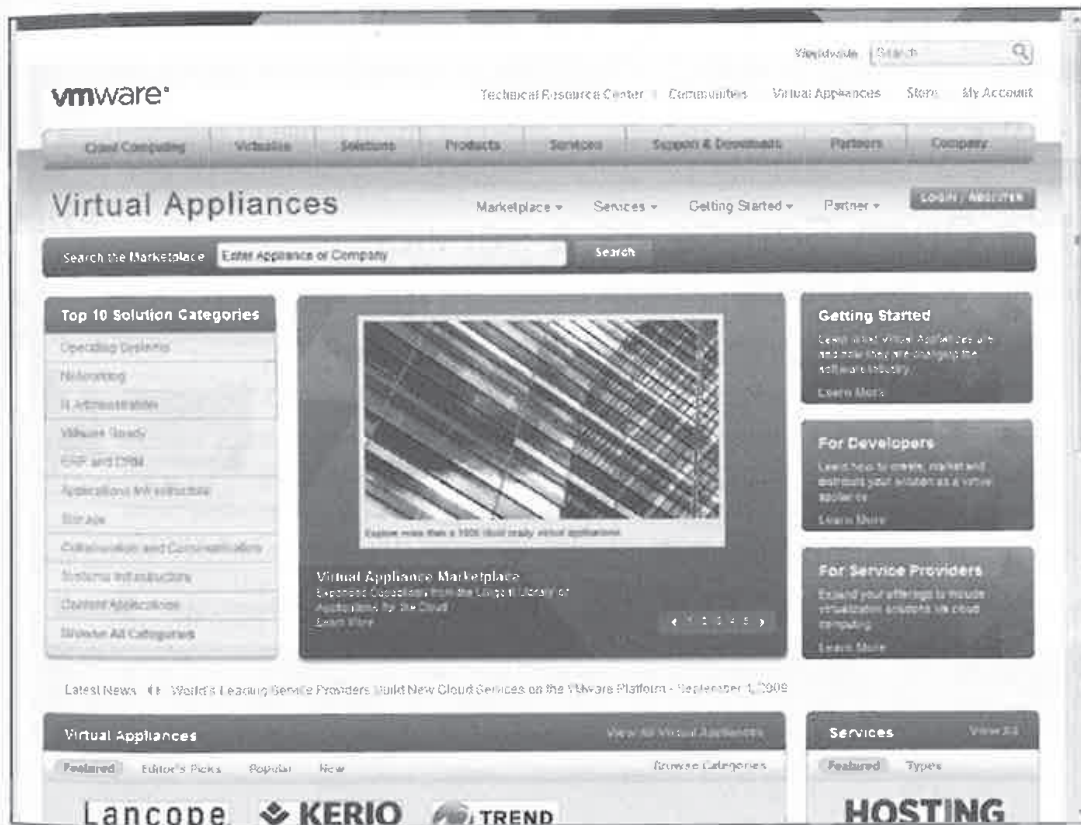
Virtual appliances have begun to affect the PC industry in much the same way that application stores have affected the cell phone industry. You can find various Web sites that either sell or distribute ready-to-use virtual appliances in various forms. Perhaps the best developed of these marketplaces is VMware's Virtual Appliances site (<http://www.vmware.com/appliances/>) shown in Figure 3.5. These appliances are certified by VMware to be ready to use in the enterprise.

Among the other places you can find virtual appliances are at the Web sites of the various operating system vendors, such as Ubuntu, Xen (<http://www.xen.org/>), and others, including these:

- **Bagvapp** (<http://bagside.com/bagvapp/>) offers virtual appliances, including ones based on Windows, all of which run on VMware Player.
- **HelpdeskLive** (<http://helpdesklive.info/download/VirtualBox%20VDI%20free%20images.html>) offers various Linux distributions upon which you can build a virtual machine.
- **Jcinacio** (<http://www.jcinacio.com/>) has Ubuntu appliances.

FIGURE 3.5

VMware's Virtual Appliance marketplace (<http://www.vmware.com/appliances/>) sells virtual appliances that run on VMware's hypervisor in cloud computing applications.

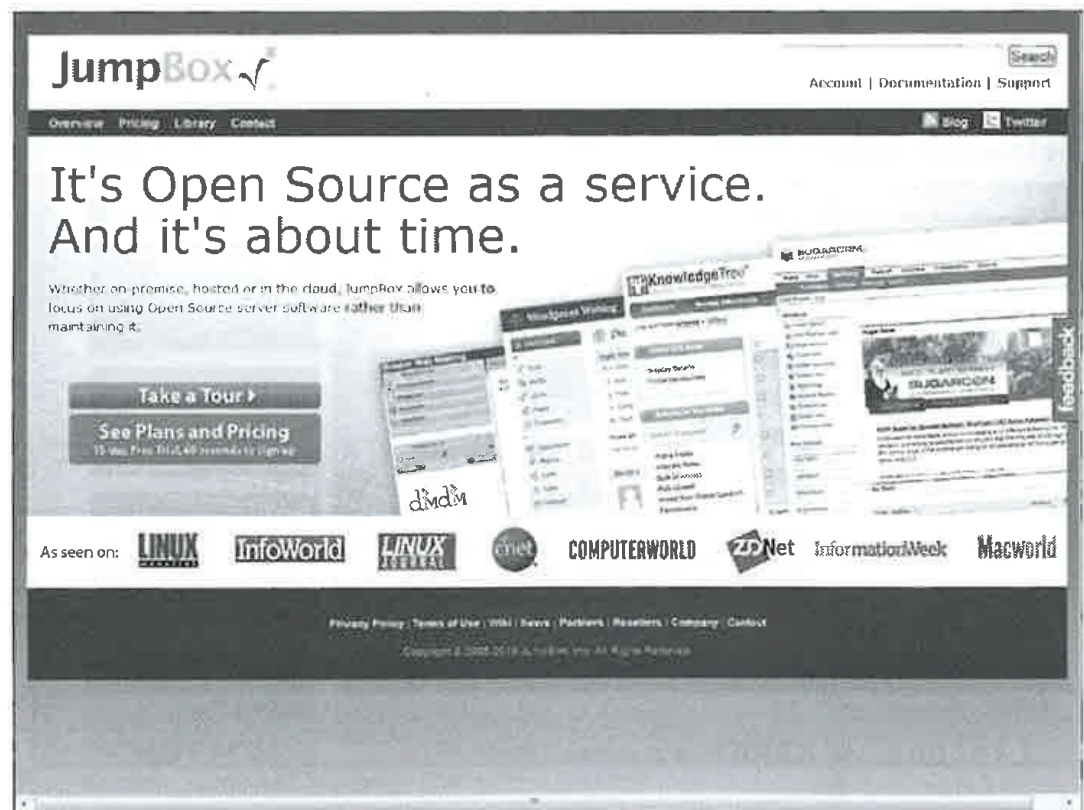


Part I: Examining the Value Proposition

- **Jumpbox** (<http://www.jumpbox.com>) offers open source virtual appliances installed by them as a managed service. Jumpbox offers virtual appliances for many applications including Bugzilla, DokuWiki, Drupal, Joomla!, Nagios, OpenVPN, PostgreSQL, Redmine, WordPress, and many others. Figure 3.6 shows the Jumpbox home page.
- **QEMU** (<http://www.qemu.org/>) is a CPU emulator and virtual machine monitor.
- **Parallels** (<http://ptn.parallels.com/ptn>) hosts a variety of appliances that includes Linux distros, server software, and other products.
- **ThoughtPolice** (<http://www.thoughtpolice.co.uk/vmware/>) offers appliances based on a variety of Linux distributions.
- **VirtualBox** (<http://www.virtualbox.org/>) is a virtual machine technology now owned by Oracle that can run various operating systems and serves as a host for a variety of virtual appliances.
- **Vmachines** (<http://www.vmachines.net/>) is a site with desktop, server, and security-related operating systems that run on VMware.

FIGURE 3.6

Jumpbox (<http://www.jumpbox.com/>) is an open-source virtual appliance installation and management service.



Converting a virtual appliance from one platform to another isn't an easy proposition. Efforts are underway to create file format standards for these types of objects that make this task easier. The best known of these file formats is the Open Virtualization Format (OVF), the work of the Distributed Management Task Force (DMTF) group. Nearly all major virtualization platform vendors support OVF, notably VMware, Microsoft, Oracle, and Citrix.

Cross-Ref

Chapter 5 describes virtual appliances in more detail and gives some examples of vendors offering these types of applications for cloud computing infrastructure development. ■

Communication Protocols

Cloud computing arises from services available over the Internet communicating using the standard Internet protocol suite underpinned by the HTTP and HTTPS transfer protocols. The other protocols and standards that expose compute and data resources in the cloud either format data or communications in packets that are sent over these two transport protocols.

In order to engage in interprocess communication (IPC) processes, many client/server protocols have been applied to distributed networking over the years. Various forms of RPC (Remote Procedure Call) implementations (including DCOM, Java RMI, and CORBA) attempt to solve the problem of engaging services and managing transactions over what is essentially a stateless network. The first of the truly Web-centric RPC technologies was XML-RPC, which uses platform-independent XML data to encode program calls that are transported over HTTP, the networking transport to which nearly everyone is connected.

Note

You can find a full description of the common Internet protocol standards in *Networking Bible* by Barrie Sosinsky, Wiley, 2009. These protocols form the basis for much of the discussion in any good networking textbook. ■

As Internet computing became more firmly entrenched over the last decade, several efforts began to better define methods for describing and discovering services and resources. The most widely used message-passing standard at the moment is the Simple Object Access Protocol (SOAP), which essentially replaces XML-RPC. SOAP uses XML for its messages and uses RPC and HTTP for message passing. SOAP forms the basis for most of the Web services stacks in use today. If you examine the XML file used in a SOAP transaction, you find that it contains a message and the instructions on how to use the message. The message has a set of rules that are translated into application instances and datatypes, and it defines the methods that must be used to initiate procedure calls and then return a response.

Several standards have emerged to allow the discovery and description of Web-based resources. The most commonly used model for discovery and description used with SOAP messaging is the Web Services Description Language (WSDL), a World Wide Web Consortium (<http://www.w3.org/2002/ws/desc/>) Internet standard. WSDL lets a Web service advertise itself in terms of a collection of endpoints or ports associated with a specific network address (URL) that can be

Part I: Examining the Value Proposition

addressed using XML messages to provide a service. In WSDL, a service is a container that performs a set of functions that are exposed to Web protocols. Taken together, the protocol and port are a binding to which messages are passed and operations are performed. A bound service is one that responds to any valid HTTP request sent to it. The important thing to remember about WSDL is that it defines a Web service's public interface.

Cross-Ref

Chapter 13, which describes Service Oriented Architecture, continues the discussion of these various components in building an application in the cloud from a component-based architecture. ■

Using WSDL and SOAP, a number of extensions were created that allow various Web services to describe additional sets of properties and methods that they could provide. These extensions fall under the name WS-*, or the “WS-star” specifications. A number of WS-* extensions are in common use, with the following being the most widely used:

- WS-Addressing
- WS-Discovery
- WS-Eventing
- WS-Federation
- WS-MakeConnection
- WS-Messaging
- WS-MetadataExchange
- WS-Notification
- WS-Policy
- WS-ResourceFramework
- WS-Security
- WS-Transfer
- WS-Trust

These different specifications provide a standard means of adding metadata to a SOAP message by modifying the message header while maintaining the message body structure. In this way, a standard method for metadata exchange is piggybacked onto the WSDL XML message. Each of these different WS-* specifications is in a different state of development.

You use these various WS-* services in your daily work. For example, the Web Services Dynamic Discovery specification (WS-Discovery) is a specification for multicast discovery on a LAN (Local Area Network) that is extended to Web services, most often as SOAP over UDP (User Datagram Protocol). When you open the Network Neighborhood in Windows and use the People Near Me feature, WS-Discovery goes into action and shows you discoverable resources.

These WS-* services carried over XML messages using the SOAP protocol access remote server applications in ways that are becoming increasingly complex. Whereas earlier methods for client/server provided a means through a gateway like CGI to access media content on servers, the current data communications burden servers with accepting and processing very complex requests or engaging their clients in sophisticated negotiations that seek to minimize the amount of processing that must be done and the information that must be exchanged as the response. None of this type of rich media servicing was ever envisaged in the construction of the Internet, and all of it is essentially a kludge.

Over the years, a variety of platform-specific RPC specifications, such as DCOM (Distributed Common Object Model) and CORBA (Common Object Response Broker Architecture), were developed to allow software components that ran on different computers to interoperate with one another. As SOAP and WS-* were developing, those protocols began to build into their specifications server application features from these other technologies in a more platform-independent protocol. What was really needed was a method for standardizing resources on the Web, which is where the idea of REST comes in.

REST stands for Representational State Transfer, and it owes its original description to the work of Roy Fielding, who was also a co-developer of the HTTP protocol. REST assigns a global identifier to a resource so there is a uniform method for accessing information sources. That identifier is a URI expressed in HTTP form. Given a resource then at a known address, various network clients in the form of what are called *user agents* can then communicate with that resource using HTTP commands (requests) to exchange information in the form of documents or files. Typical data transfers might use XML, text, an image file, a JSON document, or some other standard or agreed upon format to perform the data exchange. A transaction following the rules of REST is therefore considered to be RESTful, and this is the basis for cloud computing transactions to be initiated, processed, and completed in most modern implementations.

While REST is heavily used, it is not the only data interchange standard that is used by cloud services. Another example of a data exchange standard is the Atom or Atom Publishing Protocol (APP). Atom is a syndication format that allows for HTTP protocols to create and update information. Microsoft's ADO.NET Data Services Framework is another system for transferring data using a RESTful transaction and standard HTTP commands.

Cloud services span the gamut of computer applications: audio and video streaming, instant messaging, and so forth. Each of these areas uses protocols developed for network use and adapted for use by Web services. The impact of cloud computing on network communication is to encourage the use of open-network protocols in place of proprietary protocols.

For example, in the area of instant messaging, which is a major cloud service, the SIMPLE protocol (which stands for the Session Initiated Protocol for Instant Messaging and Presence Leveraging Extensions) is in widespread use today. SIMPLE is based on the IETF Session Initiation Protocol (SIP) standard and is an open standard. Although most early IM (Instant Messaging) services used proprietary standards, many IM services now support SIMPLE because without this support, it would be hard for the different services to interoperate. Similarly, in the area of VoIP, you find that the open XMPP or the Extensible Messaging and Presence Protocol is used.

Applications

Although the cloud computing stack encompasses many details that describe how clouds are constructed, it is not a perfect vehicle for expressing all the considerations that one must account for in any deployment. An important omission arises from the nature of distributed Web applications and the design of Internet protocols as a stateless service. The Internet was designed to treat each request made to a server as an independent transaction. Therefore, the standard HTTP commands are all atomic in nature: GET to read data, PUT to writedata, and so on.

While stateless servers are easier to architect and stateless transactions are more resilient and can survive outages, much of the useful work that computer systems need to accomplish are stateful. Here's the classic example. When you go to a reservation system to purchase something, you query inventory, reserve the item, and then pay for it. In a multiuser system, if you don't have a stateful system, you cannot know whether the item you reserved has already been taken by another user before you can enter your payment for the item. Should you decide you don't want the item at some later time, it is much easier to restore the item to inventory and return payments or make other adjustments if you can roll back all the steps as a transactional unit.

Much of the really hard development efforts that have gone into making the Web useful in commerce have centered around creating mechanisms to change a set of stateless transactions into stateful ones. The development of transaction servers, message queuing servers, and other middleware is meant to bridge this problem. Cloud computing is no exception to this problem, and to an extent it amplifies the problem by not only making transactions stateless but also virtualizing resources so transactions are always occurring in physically different locations. In cloud computing, a variety of constructs are brought to bear to solve these issues, but these are the two most important concepts:

- The notion of orchestration—that process flow can be choreographed as a service
- The use of what is referred to as a service bus that controls cloud components

In time, other methods for establishing transactional integrity may be developed that are better suited to cloud computing, but these are the standard methods that are now part of the Service Oriented Architecture. Chapter 13 takes up this topic in more detail, but it is something you should certainly keep in mind as you read through the intervening chapters.

Connecting to the Cloud

Clients can connect to a cloud service in a number of different ways. These are the two most common means:

- A Web browser
- A proprietary application

These applications can be running on a server, a PC, a mobile device, or a cell phone. What these devices have in common with either of these application types is that they are exchanging data over an inherently insecure and transient medium. There are three basic methods for securely connecting over a connection:

- Use a secure protocol to transfer data such as SSL (HTTPS), FTPS, or IPsec, or connect using a secure shell such as SSH to connect a client to the cloud.
- Create a virtual connection using a virtual private network (VPN), or with a remote data transfer protocol such as Microsoft RDP or Citrix ICA, where the data is protected by a tunneling mechanism.
- Encrypt the data so that even if the data is intercepted or sniffed, the data will not be meaningful.

The best client connections use two or more of these techniques to communicate with the cloud. In current browser technology, clients rely on the Web service to make available secure connections, but in the future, it is likely that cloud clients will be hardened so the client itself enforces a secure connection.

If you've ever logged into a hotel connection and browsed the network, you may find that often you can access systems on the network that haven't been protected with a firewall; an improperly configured firewall connection to the cloud is even worse. That has led people to drag portable routers with them, which provide a personal hardware firewall; many of these devices have VPN built directly into them.

Other solutions include using VPN software; here are three recommended solutions:

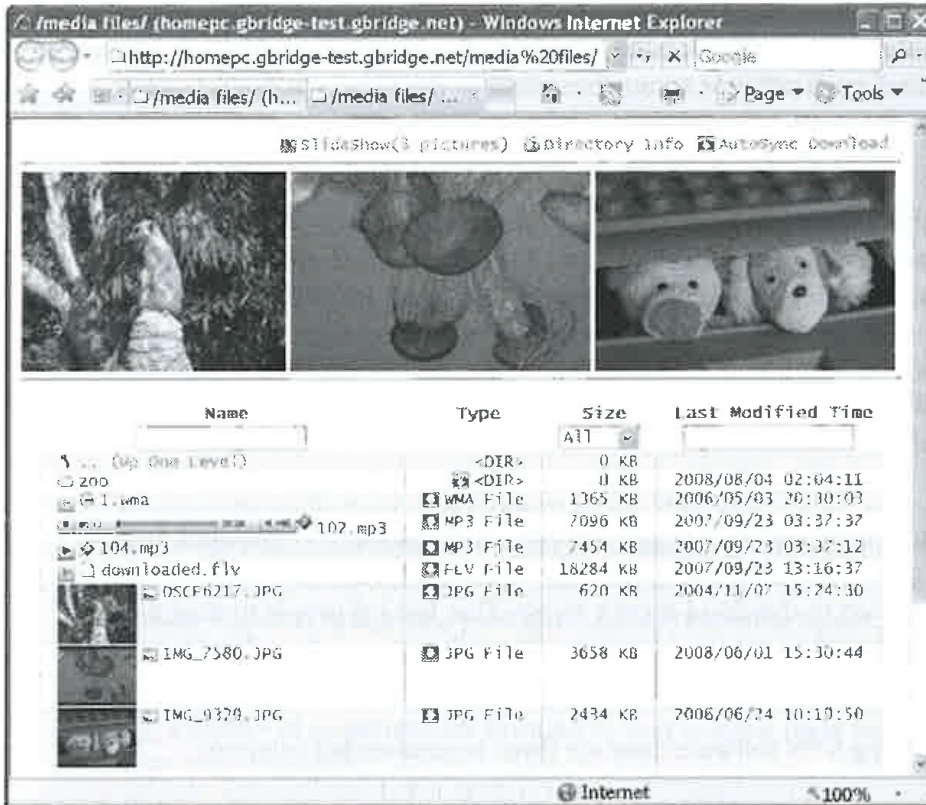
- Hotspot VPN (<http://www.hotspotvpn.com/>)
- AnchorFree Hotspot Shield (<http://hotspotshield.com/>)
- Gbridge (<http://www.gbridge.com/>), a third-party VPN based on Google's GoogleTalk infrastructure

Gbridge is an interesting solution that illustrates the use of VPN over a cloud connection. To use this product, you need to log into the GoogleTalk (or Gtalk) network and connect to another computer using your Google account. Gbridge allows additional people to join a connection when invited and supports collaborative features such as desktop sharing using the Virtual Network Computing (VNC) software, chat, live folder browsing, folder synchronization, and automated backup. Gbridge also works with applications deployed using Google Apps, allowing you to securely connect to these applications using a VPN. Figure 3.7 shows browsing a folder over a VPN connection using Gbridge's SecureShares feature.

Part I: Examining the Value Proposition

FIGURE 3.7

Gbridge provides a means for securely connecting one computer to another using Gtalk. Shown here is the SecureShares folder-browsing feature.



The Jolicloud Netbook OS

The popularity of ultralight netbooks and mobile phones has greatly expanded the potential audience of dedicated cloud computing devices, but until recently these devices ran standard operating systems such as Windows, Linux, and Macintosh on the PC and Android, IOS, and Windows Mobile (among others) on cell phones. The primary differentiation between these devices is whether or not they are capable of running video and animation (particularly Adobe Flash).

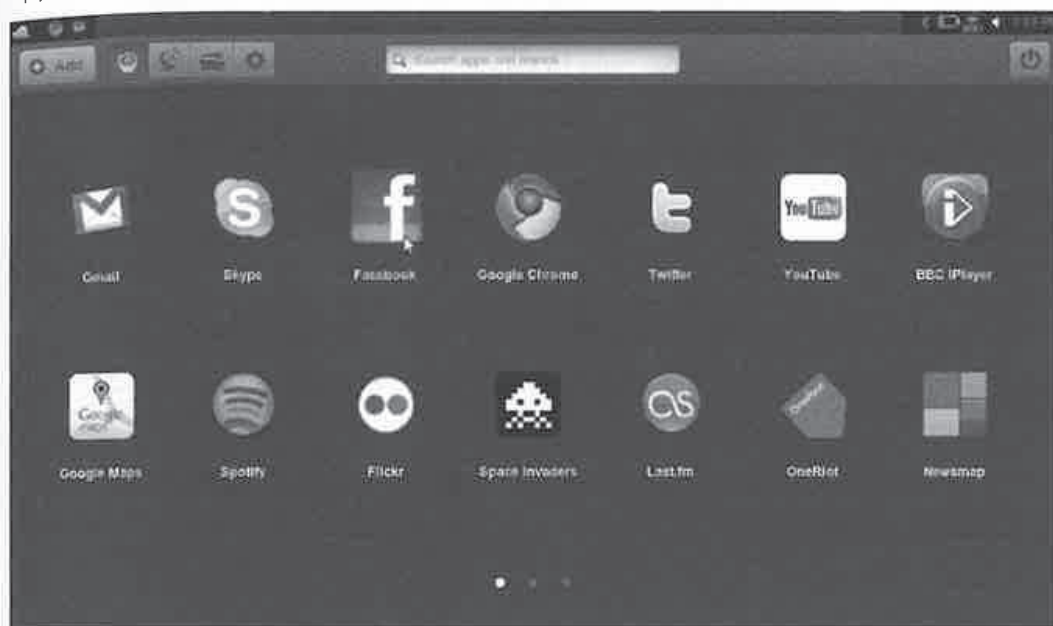
None of these portable devices has been optimized to connect securely to the cloud by narrowing the operating system functions to harden these devices. The French firm Jolicloud (<http://www.jolicloud.com/>) has recently released a lightweight version of Linux designed specifically to run connected to the cloud as a dedicated cloud client. Jolicloud 1.0 ("The Anywhere OS") can be loaded onto a netbook as the only operating system, or it can be set up as a dual boot system that shares files with a Windows partition.

Jolicloud concentrates on building a social platform with automatic software updates and installs. The application launcher is built in HTML 5 and comes preinstalled with Gmail, Skype, Twitter,

Firefox, and other applications. Any HTML 5 browser can be used to work with the Jolicloud interface. Jolicloud maintains a library or App Directory of over 700 applications as part of an app store. When you click to select an application, the company both installs and updates the application going forward, just as the iPhone manages applications on that device. Figure 3.8 shows the Jolicloud interface.

FIGURE 3.8

The Jolicloud cloud client operating system is a social networking platform for netbooks with a dedicated application store.



When you install Jolicloud on multiple devices, the system automatically synchronizes your applications so you are working with the same content on all your devices. You can manage your devices from any cloud-connected device. Your files are also unified in a single location, and the operating system provides access to shared storage cloud services such as box.net, Dropbox, and drop.io, among other services.

Chromium OS: The Browser as an Operating System

The Google Chrome OS is a Linux open-source operating system designed to be a robust cloud client. Unlike many other Linux distributions, Google's Chrome is not a software installation, but is shipped installed on validated hardware from Google-approved OEMs (Original Equipment Manufacturers), just as the Android operating system is shipped on a variety of phones. The intent is to have a tightly coupled hardware offering that supports features in the Chrome OS and that would be highly efficient. Early designs have shown Chrome running on tablet designs that would position it as an Apple iPad competitor running on netbook-type devices in the \$300-400 range.

Part I: Examining the Value Proposition

Note

An OEM or original equipment manufacturer builds systems from components and sells them under a brand name. ■

The expectation is that the first versions of Chrome systems will appear in late 2010, perhaps in both consumer and enterprise offerings. There is also an open-source version of this cloud client called Chromium (<http://www.chromium.org/chromium-os>), which shares the same code base. The Chromium architecture is built as a three-tier system with a hardware layer, the browser and window manager, and a set of system software and utilities.

The Chrome OS has been described as a hardened operating system because it incorporates a sandbox architecture for running applications and also performs automatic updates. Also included in the system is a version of remote desktop connection software that creates an encrypted connection like Microsoft's RDP, Citrix's ICA, or a VNC client. The Chrome OS hardware specification includes a Trusted Platform Module, which provides for a "trusted bootpath" along with a hardware switch that can be used to boot the system into a developer model. In that mode, some of the security features are turned off, allowing the user to reset the system.

Demonstrations of early prototypes of Chrome and Chromium OS systems have shown that they are capable of nearly instantaneous startup. Chromium Linux kernel has adopted the Upstart (<http://upstart.ubuntu.com/>) event-based replacement for the `init` daemon, which is used to launch services concurrently, restore stalled jobs, and perform delayed system startup. The fast boot time is possible because the device is devoid of most of the devices in modern PCs. Chromium has also adopted a set of security routines in firmware that run during startup and store the information necessary to perform verified system restoration.

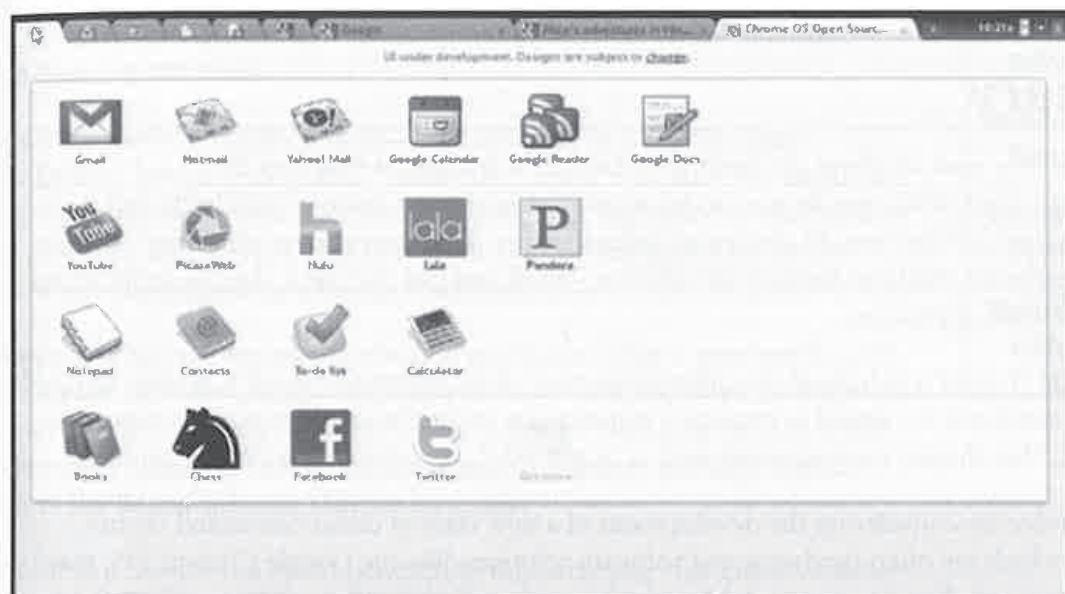
Essentially, the Chrome OS looks like the Chrome browser. Chrome is interesting because Google has essentially stripped down the operating system to run one specific application that connects to the Internet. The user interface is similar to the Chrome Web browser and includes a media player that plays MP3, views JPEGs, and plays media content both online and offline. Adobe Flash is integrated directly into the Chrome OS, just as it is in the Chrome browser. When you launch Chrome, you see links to the major Google cloud applications such as Gmail, Google Apps, and YouTube, along with other major sites such as Facebook, Hulu, Pandora, Twitter, and others. Figure 3.9 shows an early demonstration of the Chrome OS, its multi-tab interface, and its application launcher utility. From this same demonstration is shown the Google Reader application with a page from *Alice in Wonderland*, displayed in Figure 3.10.

Google will include on Chromium the Google Cloud Print service that allows an application to print to any connected printer without accessing a printer driver. This system frees Chromium from having to develop hardware- and OS-specific print subsystems. Instead, a proxy is installed in Chrome that registers a printer with the service, and this proxy manages print jobs for the user.

The Chrome OS devices that appear, as well as the competitors such as the Apple iPad and a host of other similar devices from other system vendors, signal a sea change in the manner in which users access the cloud, and they represent the cloud's impact on the manner in which many users perform their daily work. It's anyone's guess how impactful these introductions will be, but it is clear that they are not simply another competitor to Windows and Macintosh desktop-oriented systems. They represent the move into a cloud-based future where applications run and data is stored remotely. Their success is likely to be contingent upon how fast consumers and businesses become comfortable with the idea of outsourcing these functions. In time, the transition is probably inevitable because the economies of scale and efficiency that cloud computing offers is too compelling to ignore.

FIGURE 3.9

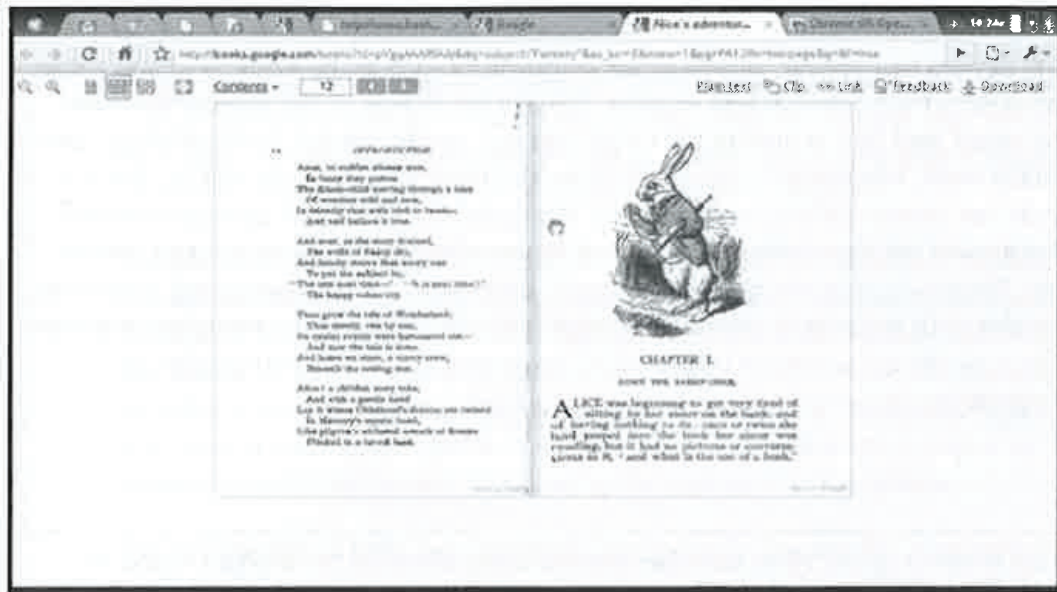
The Chrome OS operating system's application launcher from an early demo of the product found at <http://www.youtube.com/watch?v=ANMrzw7JFzA&feature=channel>



Part I: Examining the Value Proposition

FIGURE 3.10

From the same source as Figure 3.9, this figure illustrates how the Google Reader appears within the Chrome OS as a tab.



Summary

In this chapter, you learned about the various architectural layers that make up the cloud computing stack. Cloud computing may be seen to be an extension of older Internet standards, but it includes some new architecture. In particular, infrastructure components were described, as were platform components. Platform services are software-based and can include a class of applications referred to as virtual appliances.

The focus of the chapter was how all the different parts of cloud computing work together. Many of the important standards are aimed at managing transactions in the cloud and making components work together. This chapter took an initial look at SOAP, XML, and the various WSDL services.

The chapter ended by considering the development of a new class of cloud-connected clients. These clients, which are often hardware and software solutions like the Google Chrome OS, may have major impact in changing the model by which people connect to the Internet and get their work done. They have many advantages, but we will have to wait to see how readily consumers and businesses are willing to adopt this new computing paradigm.

In the next chapter, each of the different Internet service models is more closely described.

Managing the Cloud

Cloud computing deployments must be monitored and managed in order to be optimized for best performance. To the problems associated with analyzing distributed network applications, the cloud adds the complexity of virtual infrastructure. This is one of the most active areas of product development in the entire cloud computing industry, and this chapter introduces you to the different products in this nascent area.

Cloud management software provides capabilities for managing faults, configuration, accounting, performance, and security; this is referred to as FCAPS. Many products address one or more of these areas, and through network frameworks, you can access all five areas. Framework products are being repositioned to work with cloud systems.

Your management responsibilities depend on the particular service model for your cloud deployment. Cloud management includes not only managing resources in the cloud, but managing resources on-premises. The management of resources in the cloud requires new technology, but management of resources on-premises allows vendors to use well-established network management technologies.

The lifecycle of a cloud application includes six defined parts, and each must be managed. In this chapter, the tasks associated with each stage are described.

Efforts are underway to develop cloud management interoperability standards. One effort you learn about in this chapter is the DMTF's (Distributed Management Task Force) Open Cloud Standards Incubator. The goal of these efforts is to develop management tools that work with any cloud type. Another group called the Cloud Commons is developing a technology called the Service Measurement Index (SMI). SMI aims to deploy methods for measuring various aspects of cloud performance in a standard way.

IN THIS CHAPTER

Learning about network management software

Viewing the essential monitoring features

Using lifecycle management techniques

Discovering emerging network management interoperability standards

Administrating the Clouds

The explosive growth in cloud computing services has led many vendors to rename their products and reposition them to get in on the gold rush in the clouds. What was once a network management product is now a cloud management product. Nevertheless, this is one area of technology that is very actively funded, comes replete with interesting startups, has been the focus of several recent strategic acquisitions, and has resulted in some interesting product alliances. Let's join the party and see what all the fuss is about.

These fundamental features are offered by traditional network management systems:

- Administration of resources
- Configuring resources
- Enforcing security
- Monitoring operations
- Optimizing performance
- Policy management
- Performing maintenance
- Provisioning of resources

Network management systems are often described in terms of the acronym FCAPS, which stands for these features:

- Fault
- Configuration
- Accounting
- Performance
- Security

Most network management packages have one or more of these characteristics; no single package provides all five elements of FCAPS.

To get the complete set of all five of these management areas from a single vendor, you would need to adopt a network management framework. These large network management frameworks were industry leaders several years back: BMC PATROL, CA Unicenter, IBM Tivoli, HP OpenView, and Microsoft System Center. Network framework products have been sliced and diced in many different ways over the years, and they are rebranded from time to time. Today, for example, BMC PATROL is now part of BMC ProactiveNet Performance Management (<http://www.bmc.com/products/product-listing/ProactiveNet-Performance-Management.html>), HP OpenView has been split (https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-10^36657_4000_100) into a set of HP Manager products.

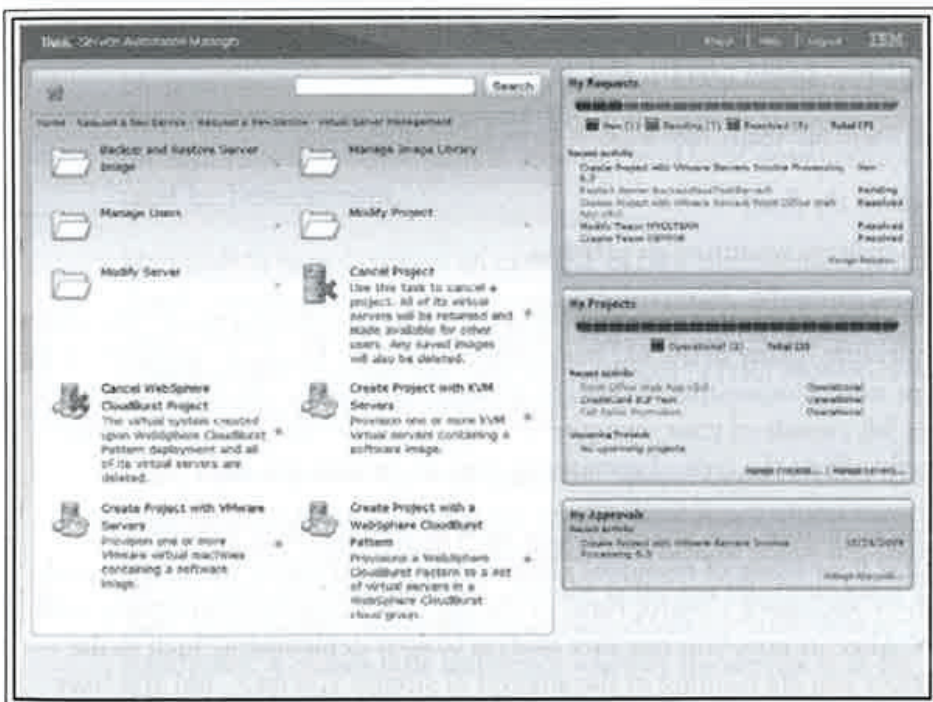
The impact that cloud computing is having on network frameworks is profound. These five vendors have (or soon will have) products for cloud management. Computer Associates, for example, has completely repositioned its network management portfolio as an IT Management Software as a Service. Find the cloud products for these five large cloud vendors at the following URLs:

- BMC Cloud Computing (<http://www.bmc.com/solutions/esm-initiative/cloud-computing.html>)
- Computer Associates Cloud Solutions (<http://www.ca.com/us/cloud-computing.aspx>)
- HP Cloud Computing (<http://h20338.www2.hp.com/enterprise/w1/en/technologies/cloud-computing-overview.html>)
- IBM Cloud Computing (<http://www.ibm.com/ibm/cloud/>)
- Microsoft Cloud Services (<http://www.microsoft.com/cloud/>)

Figure 11.1 shows IBM Tivoli Service Automation Manager, a framework tool for managing cloud infrastructure.

FIGURE 11.1

Tivoli Service Automation Manager lets you create and stage cloud-based servers.



Management responsibilities

What separates a network management package from a cloud computing management package is the “cloudly” characteristics that cloud management service must have:

- Billing is on a pay-as-you-go basis.
- The management service is extremely scalable.
- The management service is ubiquitous.
- Communication between the cloud and other systems uses cloud networking standards.

To monitor an entire cloud computing deployment stack, you monitor six different categories:

1. End-user services such as HTTP, TCP, POP3/SMTP, and others
2. Browser performance on the client
3. Application monitoring in the cloud, such as Apache, MySQL, and so on
4. Cloud infrastructure monitoring of services such as Amazon Web Services, GoGrid, Rackspace, and others
5. Machine instance monitoring where the service measures processor utilization, memory usage, disk consumption, queue lengths, and other important parameters
6. Network monitoring and discovery using standard protocols like the Simple Network Management Protocol (SNMP), Configuration Management Database (CMDB), Windows Management Instrumentation (WMI), and the like

It's important to note that there are really two aspects to cloud management:

Managing resources *in the cloud*

Using the cloud to manage resources *on-premises*

When you move to a cloud computing architecture from a traditional networked model like client/server or a three-tier architecture, many of the old management tasks for processes going on in the cloud become irrelevant or nearly impossible to manage because the tools to effectively manage resources of various kinds fall outside of your own purview. In the cloud, the particular service model you are using directly affects the type of monitoring you are responsible for.

Consider the case of an Infrastructure as a Service vendor such as Amazon Web Services or Rackspace. You can monitor your usage of resources either through their native monitoring tools like Amazon CloudWatch or Rackspace Control Panel or through the numerous third-party tools that work with these sites' APIs. In IaaS, you can alter aspects of your deployment, such as the number of machine instances you are running or the amount of storage you have, but you have very limited control over many important aspects of the operation. For example, your network bandwidth is locked into the type of instance you deploy. Even if you can provision more bandwidth, you likely have no control over how network traffic flows into and out of the system, whether there is packet prioritization, how routing is done, and other important characteristics.

The situation—as you move first to Platform as a Service (PaaS) like Windows Azure or Google App Engine and then onto Software as a Service (SaaS) for which Salesforce.com is a prime example—becomes even more restrictive. When you deploy an application on Google's PaaS App Engine cloud service, the Administration Console provides you with the following monitoring capabilities:

- Create a new application, and set it up in your domain.
- Invite other people to be part of developing your application.
- View data and error logs.
- Analyze your network traffic.
- Browse the application datastore, and manage its indexes.
- View the application's scheduled tasks.
- Test the application, and swap out versions.

However, you have almost no operational control. Essentially, Google App Engine lets you deploy the application and monitor it, and that's about it. All the management of devices, networks, and other aspects of the platform are managed by Google. You have even less control when you are selling software in the cloud, as you would with Salesforce.com.

Figure 11.2 graphically summarizes the management responsibilities by service model type.

The second aspect of cloud management is the role that cloud-based services can play in managing on-premises resources. From the standpoint of the client, a cloud service provider is no different than any other networked service. The full range of network management capabilities may be brought to bear to solve mobile, desktop, and local server issues, and the same sets of tools can be used for measurement.

Microsoft System Center is an example of how management products are being adapted for the cloud. System Center provides tools for managing Windows servers and desktops. The management services include an Operations Manager, the Windows Service Update Service (WSUS), a Configuration Manager for asset management, a Data Protection Manager, and a Virtual Machine Manager, among other components.

One of these service sets was called the System Center Online Desktop Manager (SCODM). Microsoft has taken SCODM and repositioned it as a cloud-based service for managing updates, monitoring PCs for license compliance and health, enforcing security policies, and using Forefront protect systems from malware, and the company has branded it as Windows Intune (<http://www.microsoft.com/windows/windowsintune/default.aspx>). From the client's standpoint, it makes little difference whether the service is in the cloud or on a set of servers in a datacenter. The benefit of a cloud management service accrues to the organization responsible for managing the desktops or mobile devices. Figure 11.3 shows an Overview screen from the beta version of Windows Intune. The product is due to be released in the first or second quarter of 2011.

Part III: Exploring Cloud Infrastructures

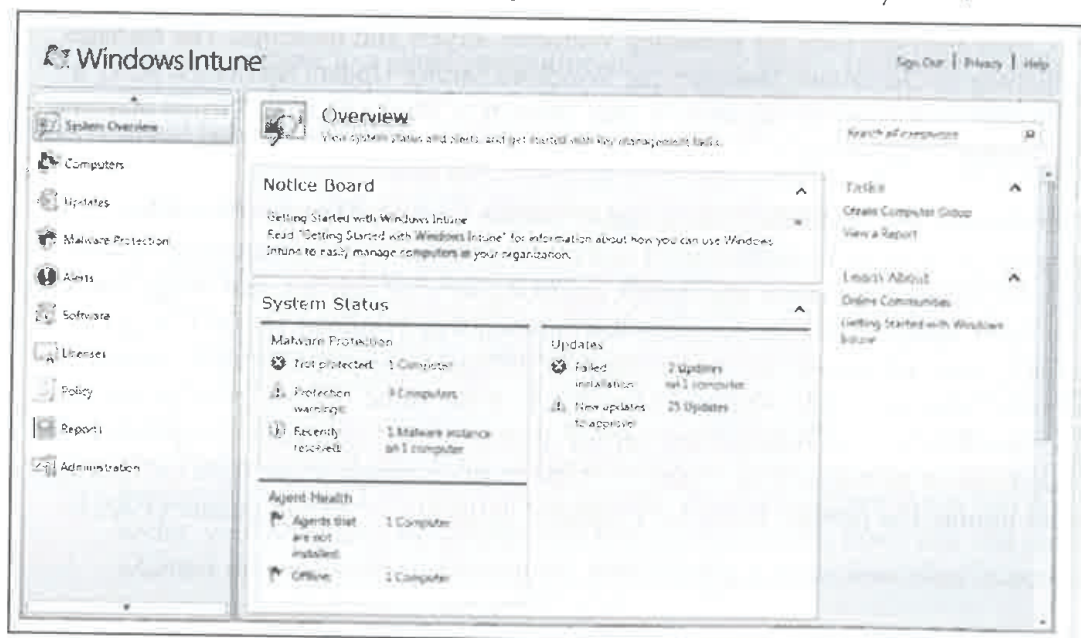
FIGURE 11.2

Management responsibilities by service model type

	Hosted	Managed services	Cloud (IaaS)	Cloud (PaaS)	SaaS
Example(s)	Hosted infrastructure	Network VoIP	Amazon AWS, Rackspace Cloud server	Google App Engine, Microsoft Azure	Salesforce.com
IT primary responsibilities					
Provider primary responsibilities		<i>Varies by business agreement</i>			
Shared responsibilities					
 Business service/user satisfaction  Application  Database  Server  Operating system  Network					

FIGURE 11.3

Intune is Microsoft's cloud-based management service for Windows systems.



Lifecycle management

Cloud services have a defined lifecycle, just like any other system deployment. A management program has to touch on each of the six different stages in that lifecycle:

1. The definition of the service as a template for creating instances
Tasks performed in Phase 1 include the creation, updating, and deletion of service templates.
2. Client interactions with the service, usually through an SLA (Service Level Agreement) contract
This phase manages client relationships and creates and manages service contracts.
3. The deployment of an instance to the cloud and the runtime management of instances
Tasks performed in Phase 3 include the creation, updating, and deletion of service offerings.
4. The definition of the attributes of the service while in operation and performance of modifications of its properties
The chief task during this management phase is to perform service optimization and customization.
5. Management of the operation of instances and routine maintenance
During Phase 5, you must monitor resources, track and respond to events, and perform reporting and billing functions.
6. Retirement of the service
End of life tasks include data protection and system migration, archiving, and service contract termination.

Cloud Management Products

Cloud management software and services is a very young industry, and as such, it has a very large number of companies, some with new products and others with older products competing in this area. Table 11.1 shows some of the current players in this market, along with the products they either are offering or are promising in the very near future. When considering products in cloud management, you should be aware that—as in all new areas of technology—there is considerable churn as companies grow, get acquired, or fail along the way. It is entirely possible that if you return to this list a year or two after this book is published, half of these products or services will no longer exist as listed; you should keep this in mind.

TABLE 11.1

Cloud and Web Monitoring Solutions

Product	URL	Description
AbiCloud	http://www.abiquo.com/	Virtual machine conversion and management
Amazon CloudWatch	http://aws.amazon.com/cloudwatch/	AWS dashboard
BMC Cloud Computing Initiative	http://www.bmc.com/solutions/esm-initiative/cloud-computing.html	Cloud planning, lifecycle management, optimization, and guidance
CA Cloud Connected Management Suite	http://www.ca.com/us/cloud-solutions.aspx	CA Cloud Insight, CA Cloud Compose, CA Cloud Optimize, and CA Cloud Orchestrate are described below
Cacti	http://www.cacti.net/	Network performance graphing solution
CloudKick	https://www.cloudkick.com/	Cloud server monitoring
Dell Scalent	http://www.scalent.com/index.php	Virtualization provisioning system that will be rolled into Dell's Advanced Infrastructure Manager (AIM)
Elastra	http://www.elastra.com/	Federated hybrid cloud management software
Ganglia	http://ganglia.info/	Distributed network monitoring software
Gomez	http://www.gomez.com/	Web site monitoring and analytics
HP Cloud Computing	http://h20338.www2.hp.com/enterprise/w1/en/technologies/cloud-computing-overview.html	A variety of management products and services, both released and under development
Hyperic	http://www.hyperic.com/	Performance management for virtualized Java Apps with VMware integration
IBM Service Management and Cloud Computing	http://www-01.ibm.com/software/tivoli/solutions/cloudcomputing/	Various IBM Tivoli managers and monitors
Internetseer	http://www.internetseer.com/home/index.xtp	Web site monitoring service
Intune	http://www.microsoft.com/windows/windowsintune/default.aspx	Cloud-based Windows system management
Keynote	http://www.keynote.com/	Web, mobile, streaming, and customer test and measurement products

Product	URL	Description
ManageEngine OpManager	http://www.manageengine.com/network-performance-management.html	Network and server monitoring, server desk, event and security management
ManageIQ	http://www.manageiq.com/	Enterprise Virtualization Management Suite (EVM) that provides monitoring, provisioning, and cloud integration services
Managed Methods JaxView	http://managedmethods.com/	SOA management tool
Monit	http://mmonit.com/monit/	Unix system monitoring and management
Montis	http://portal.monitis.com/index.php/home	Cloud-based monitoring service
Morph	http://mor.ph/	Infrastructure management, provisioning, deployment, and monitoring tools
Nagios	http://www.nagios.org/	Network monitoring system
NetIQ	http://www.netiq.com/	Network management, monitoring, deployment, and security software
New Relic RPM	http://www.newrelic.com/	Java and Ruby application monitor and troubleshooting
Nimsoft	http://www.ca.com/us/products/detail/CA-Nimsoft-Monitoring-Solution.aspx	Cloud monitoring software
OpenQRM	http://www.openqrm.com/	Data center management platform
Pareto Networks	http://www.paretonetworks.com/	Cloud provisioning and deployment
Pingdom	http://www.pingdom.com/	Web site and server uptime and performing monitoring
RightScale	http://www.rightscale.com/	Automated virtual server scaling
ScienceLogic	http://www.sciencelogic.com/	Datacenter and cloud management solutions and appliances
Scout	http://scoutapp.com/	Hosted server management service
ServiceUptime	http://www.serviceuptime.com/	Web site monitoring service
Site24X7	http://site24x7.com/	Web site monitoring service
Solarwinds	http://www.solarwinds.com/	Network monitoring and management software

continued

Part III: Exploring Cloud Infrastructures

TABLE 11.1 (continued)

Product	URL	Description
Tapinsystems	http://www.tapinsystems.com/home	Provisioning and management service
Univa UD	http://univaud.com/index.php	Application and infrastructure management software for hybrid multi-clouds
VMware Hyperic	http://www.springsource.com/	Performance management for VMware deployed Java applications
Webmetrics	http://www.webmetrics.com/	Web performance management, load testing, and application monitor for cloud services
WebSitePulse	http://www.websitepulse.com/	Server, Web site, and application monitoring service
Whatsup Gold	http://www.whatsupgold.com/	Network monitoring and management software
Zenoss	http://www.zenoss.com/	IT operations monitoring
Zeus	http://www.zeus.com/	Web-based application traffic manager

The core management features offered by most cloud management service products include the following:

- Support of different cloud types
- Creation and provisioning of different types of cloud resources, such as machine instances, storage, or staged applications
- Performance reporting including availability and uptime, response time, resource quota usage, and other characteristics
- The creation of dashboards that can be customized for a particular client's needs

Automated deployment on IaaS systems represents one class of cloud management services. One of the more interesting and successful vendors in this area is Rightscale (<http://www.rightscale.com/>) whose software allows clients to stage and manage applications on AWS (Amazon Web Service), Eucalyptus, Rackspace, and the Chef Multicloud framework or a combination of these cloud types. Rightscale creates cloud-ready server templates and provides the automation and orchestration necessary to deploy them. Eucalyptus and Rackspace both use Amazon EC2 and S3 services, although Eucalyptus is open source and portable. RightScale server templates and the Rightscript technology are highly configurable and can be run under batch control. The RightScale user interface also provides real-time measurements of individual server instances.

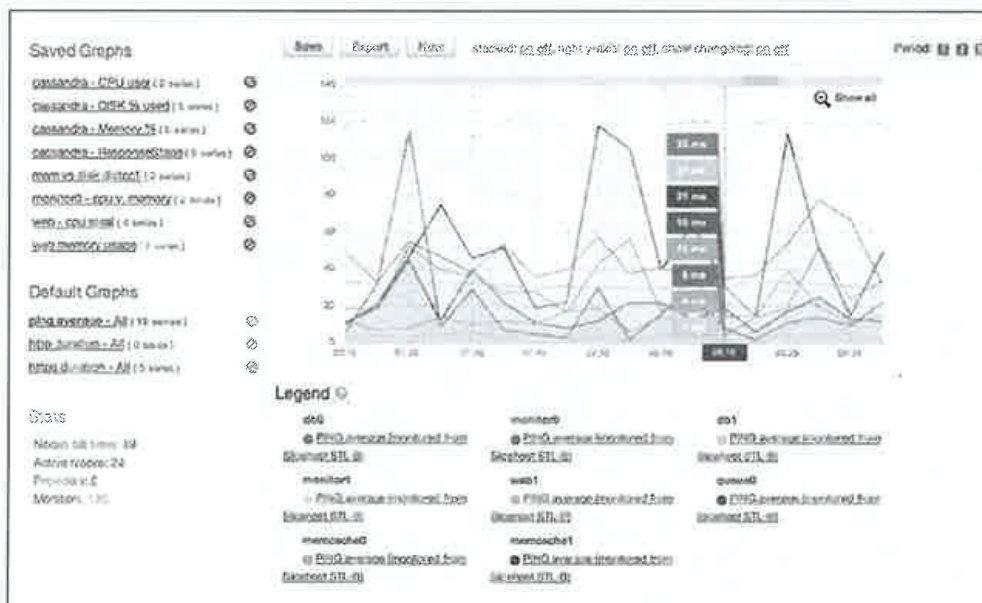
Cloudkick (<https://www.cloudkick.com/>) is another infrastructure monitoring solution that is well regarded. Its service is noted for being agnostic and working with multiple vendor cloud platforms. The Cloudkick user interface is designed for rapid deployment assessment, and its at-a-glance-monitoring Insight module is particularly easy to use. Figure 11.4 shows the Insight

module, and Figure 11.5 shows Cloudkick’s real-time server visualization tool, which is one of the more interesting presentation tools we’ve seen. In Figure 11.5, the circles are servers with their location on the different axes based on observed metrics. Powerful servers are larger circles, and the colors indicate the current state of the server.

Users have commented on Cloudkick's instant launching being difficult, and both Cloudkick and RightScale are known to be easy to use with Linux virtual servers and less so with Windows instances.

FIGURE 11.4

Cloudkick's Insight module (https://www.cloudkick.com/site_media/images/graphs2.png) is powerful and particularly easy to use.



All of the service models support monitoring solutions, most often through interaction with the service API. Tapping into a service API allows management software to perform command actions that a user would normally perform. Some of these APIs are themselves scriptable, while in some cases, scripting is supported in the management software.

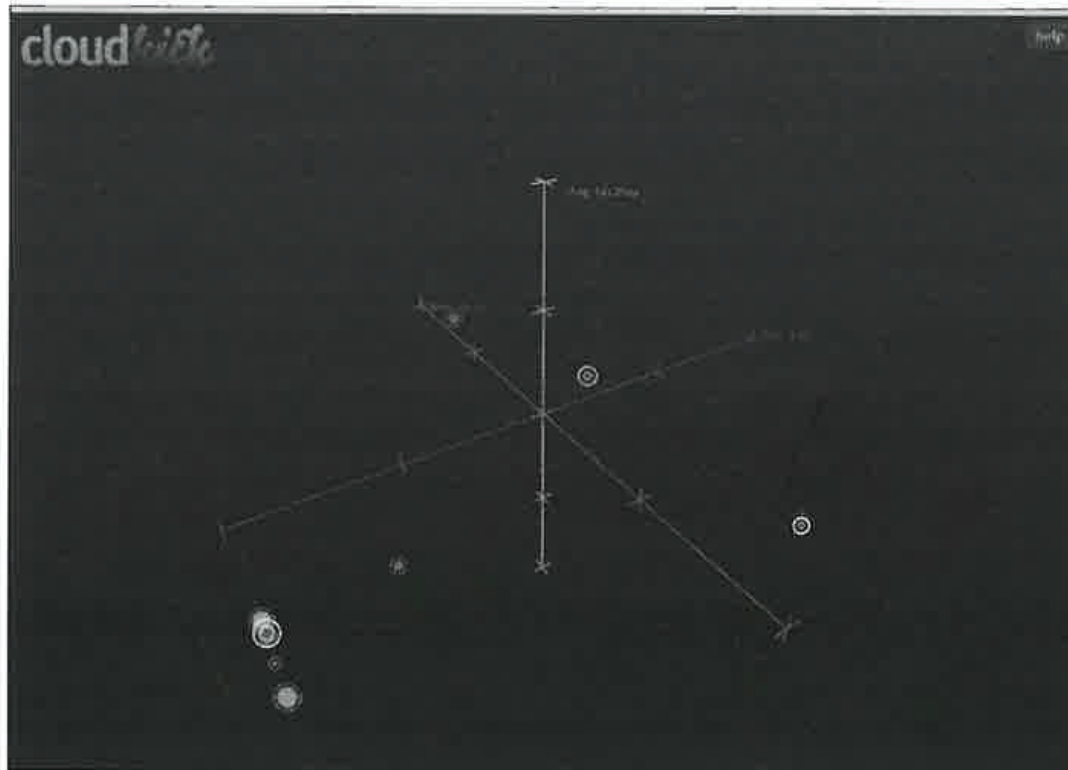
One key differentiator in monitoring and management software is whether the service needs to install an agent or it performs its service without an agent. The monitoring function normally can be performed through direct interaction with a cloud service or client using processes such as an HTTP GET or a network command like PING. For management functions, an agent is helpful in that it can provide needed hooks to manipulate a cloud resource. Agents also, as a general rule, are useful in helping to solve problems associated with firewall NAT traversal.

ManageIQ (<http://www.manageiq.com/>) and Service-now.com offer an integrated cloud stack that combines the ManageIQ Enterprise Virtualization Management Suite with Service-Now.com's ITSM SaaS service. The system has offers management, discovery, CMDB synchronization,

and automated provisioning services. You can integrate these services into your Web applications using an open API that these companies offer.

FIGURE 11.5

The Cloudkick visualization demo (<https://www.cloudkick.com/viz/demo/>) provides a real-time graphical illustration of the state of monitored servers.



Distributed network applications often benefit from the deployment of a management appliance. Because cloud services tend to distribute applications across multiple sites, physical appliances need to be deployed in different locations—something that only cloud service providers can do. However, there has been a tendency to create virtual appliances, and those can be deployed as server instances wherever an application is deployed. Pareto Networks (<http://www.pareto-networks.com/>) has a cloud computing service that can monitor and manage distributed network services using a physical or virtual appliance. The system can be used to control and provision network services. Pareto Networks plans to add an API to this service.

Emerging Cloud Management Standards

As it stands now, different cloud service providers use different technologies for creating and managing cloud resources. As the area matures, cloud providers are going to be under considerable

pressure from large cloud users like the federal government to conform to standards and make their systems interoperable with one another. No entity is likely to want to make a major investment in a service that is a silo or from which data is difficult to stage or to extract. To this end, a number of large industry players such as VMware, IBM, Microsoft, Citrix, and HP have gotten together to create standards that can be used to promote cloud interoperability. In the section that follows, you learn about the work of the DMTF in this area.

Another effort just getting underway has been started by CA (the company formerly known as Computer Associates) in association with Carnegie Mellon called the Cloud Commons. This effort is aimed at creating an industry community and working group, and promoting a set of monitoring standards that were part of CA's cloud technology portfolio but are now open sourced.

DMTF cloud management standards

The Distributed Management Task Force (DMTF; see <http://www.dmtf.org/>) is an industry organization that develops industry system management standards for platform interoperability. Its membership is a “who's who” in computing, and since its founding in 1992, the group has been responsible for several industry standards, most notably the Common Information Model (CIM). The DMTF organizes itself into a set of working groups that are tasked with specifying standards for different areas of technology.

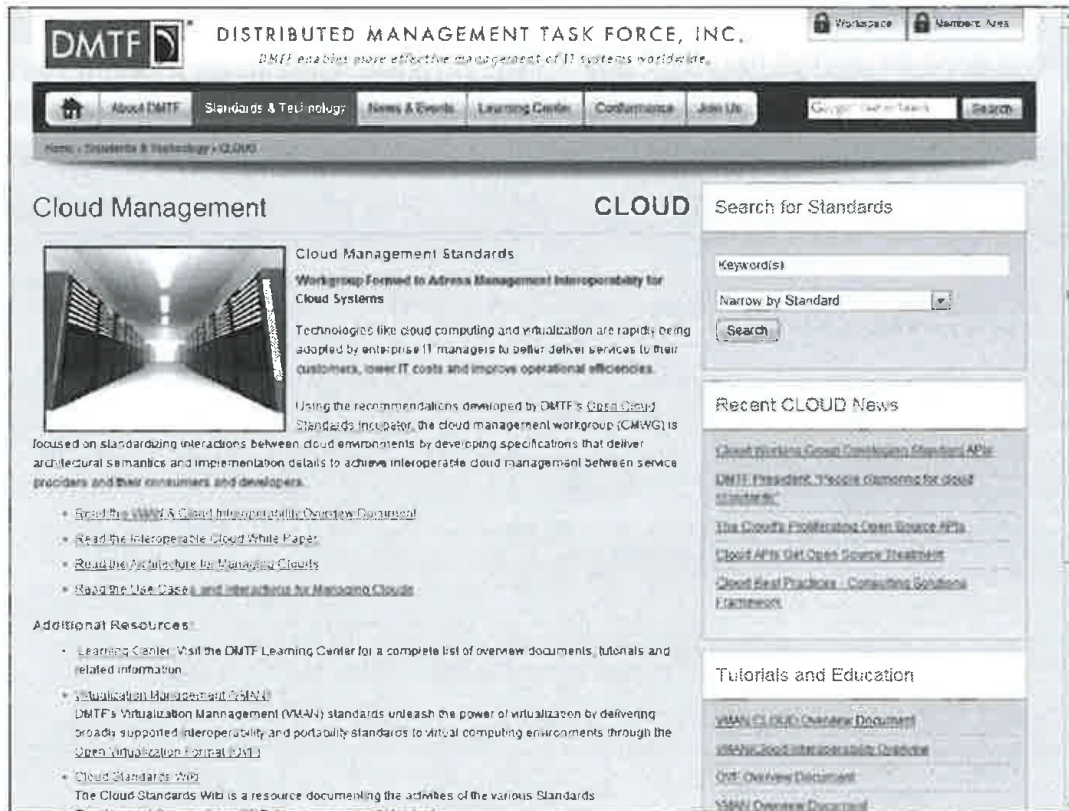
A recent standard called the Virtualization Management Initiative (VMAN) was developed to extend CIM to virtual computer system management. VMAN has resulted in the creation of the Open Virtualization Format (OVF), which describes a standard method for creating, packaging, and provisioning virtual appliances. OVF is essentially a container and a file format that is open and both hypervisor- and processor-architecture-agnostic. Since OVF was announced in 2009, vendors such as VirtualBox, AbiCloud, IBM, Red Hat, and VMWare have announced or introduced products that use OVF.

It was, therefore, a natural extension of the work that DMTF does in virtualization to solve management issues in cloud computing. DMTF has created a working group called the Open Cloud Standards Incubator (OCSI) to help develop interoperability standards for managing interactions between and in public, private, and hybrid cloud systems. The group is focused on describing resource management and security protocols, packaging methods, and network management technologies. The Web site of the Cloud Management group (<http://dmtof.org/standards/cloud>) is shown in Figure 11.6.

DMTF's cloud management efforts are really in their initial stages, but the group has broad industry support. Part of the group's task is to provide industry education, so you can find a number of white papers and technology briefs published on this site. It's an effort that's worth checking back with over time. Although the OCSI's work has not yet been joined by Amazon or Salesforce.com, a set of open standards that extend the use of industry standard protocols—such as the Common Information Model (CIM), the Open Virtualization Format (OVF), and WBEM—to the cloud are going to be hard for vendors to resist.

FIGURE 11.6

DMTF (<http://dmtf.org/standards/cloud>) has a large and important effort underway for developing cloud interoperability management standards.



Cloud Commons and SMI

CA Technologies (<http://www.ca.com>), the company once known as Computer Associates, has taken some of its technologies in measuring distributed network performance metrics and repositioned its products as the following:

- CA Cloud Insight, a cloud metrics measurement service
- CA Cloud Compose, a deployment service
- CA Cloud Optimize, a cloud optimization service
- CA Cloud Orchestrate, a workflow control and policy based automation service

Taken together, these products form the basis for CA's Cloud Connected Management Suite (<http://www.ca.com/us/cloud-solutions.aspx>).

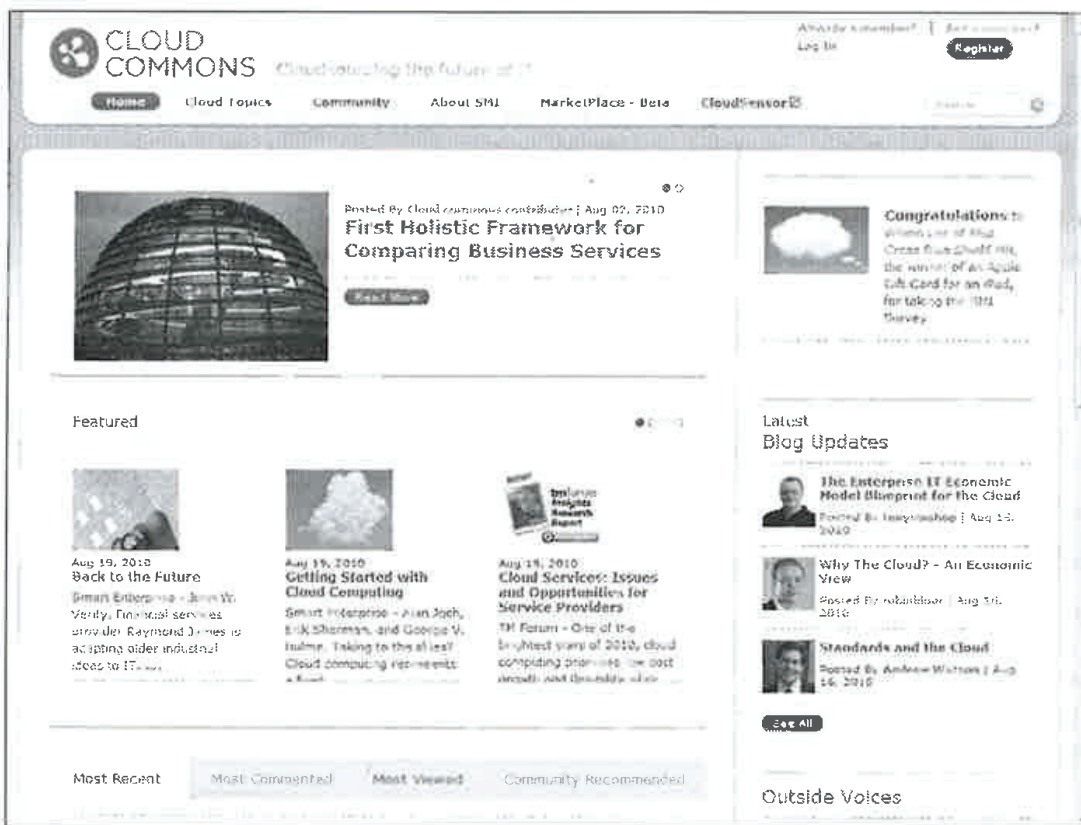
CA has lots of experience in this area through its Unicenter management suite and the products that were spawned from it. The company also has invested in cloud vendors such as 3Tera,

Oblicore, and Cassatt to create their cloud services. CA acquired Nimsoft in March 2010. Nimsoft has a monitoring and management package called Nimsoft United Monitoring that creates a monitoring portal with customizable dashboards. The system can gather information from up to 100 types of data points and can work with both Google and Rackspace cloud deployments. Among the data points that can be monitored are resource usage and UPS status.

At the heart of CA Cloud Insight is a method for measuring different cloud metrics that creates what CA calls a Service Measurement Index or SMI. The SMI measures things like SLA compliance, cost, and other values and rolls them up into a score. To help allow SMI to gain traction in the industry, CA has donated the core technology to the Software Engineering Institute at Carnegie Mellon as part of what is called the SMI Consortium. This same group is responsible for the Capability Maturity Model Integration (CMMI) process optimization technology and other efforts. The second CA initiative is the funding of an industry online community called the Cloud Commons (<http://www.cloudcommons.com/>), the home page of which is shown in Figure 11.7.

FIGURE 11.7

The Cloud Commons (<http://www.cloudcommons.com/web/guest>) is a new online community founded by CA to promote information exchange on cloud services and the SMI standard.



Because the Cloud Commons is brand new, it is hard to tell whether this group will have impact in the cloud community, but it is an interesting effort. The hope is that not only will this site establish CA's performance metrics, but that community users will eventually provide detailed information and ratings on particular services.

To demonstrate the potential of cloud-based metrics, the Cloud Commons has built a dashboard called the CloudSensor that monitors the performance of the major cloud-based services in real time.

This tool measures the performance of the following:

- RackSpace file creation and deletion
- E-mail availability (system uptime) based on Google Gmail, Windows Live Hotmail, and Yahoo! Mail
- Amazon Web Services server creation/destruction times at four AWS sites
- Dashboard Response Times for the consoles of AWS, Amazon, Google App Status, RackSpace Cloud, and Salesforce
- Windows Azure storage benchmarks
- Windows Azure SQL benchmarks

It is meant to demonstrate the value of cloud performance measurements. These metrics are based on real-time data derived from real transactions. Each chart shows the last two hours of activity. Figure 11.8 shows the CloudSensor performance dashboard.

The Service Measurement Index (SMI) is based on a set of measurement technologies forming the SMI Framework that CA donated to the SMI Consortium. It measures cloud-based services in six areas:

- Agility
- Capability
- Cost
- Quality
- Risk
- Security

These form a set of Key Performance Indicators (KPI) that can be used to compare one service to another. Figure 11.9 shows the different characteristics that make up each of the KPIs of the Service Measurement Index.

FIGURE 11.8

The CloudSensor (<http://dashboard.atgcloud.info:5001/cloudsensor/cloud-sensor.html>) dashboard displays real-time cloud service performance metrics.

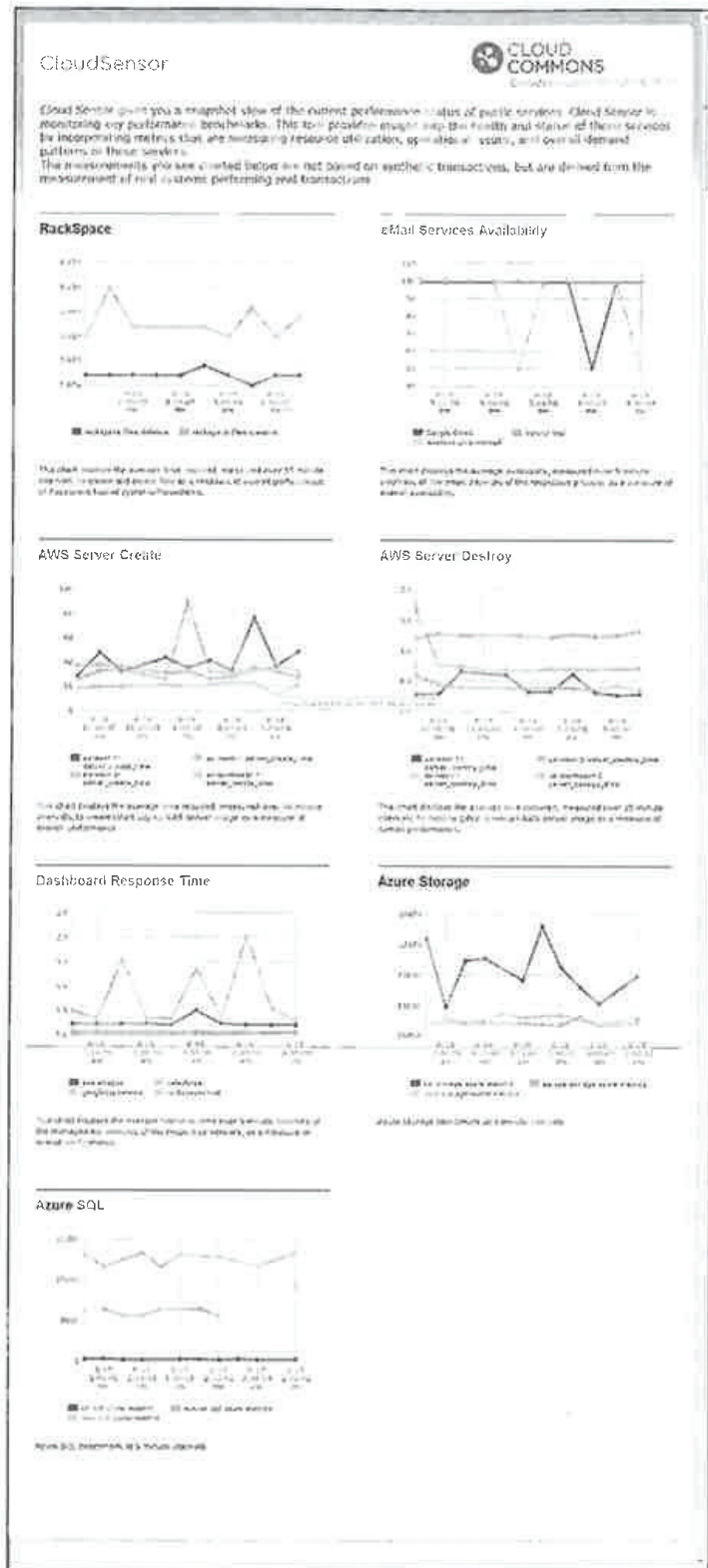
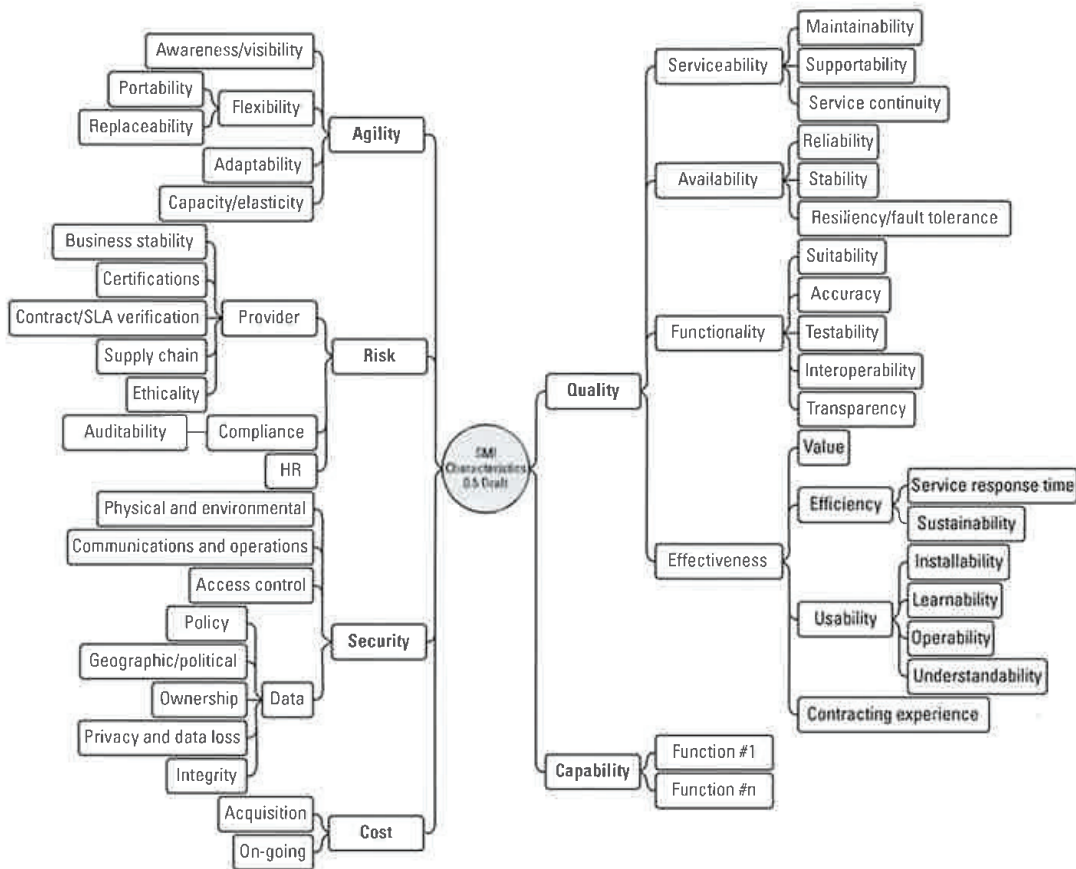


FIGURE 11.9

SMI defined characteristics (Source: “The Details behind the Service Measurement Index” by Keith Allen, 2010)



It's too early to determine whether SMI will gain traction, but the positioning of the technology as an open industry working group makes the project very interesting and worthy of note.

Summary

Cloud management is an important and growing area of technology. In this chapter, you learned about some of the products being offered or developed to address common management problems. Among the management tasks are deployment, monitoring, configuration, optimization, and often security. Nearly all network management software vendors are repositioning their products to work with cloud systems. Some network management is available from within the cloud service providers' platforms. Many of the software systems utilize the service provider's API to manage, monitor, and control resources. The use of virtualization has spawned many new products in this area.

In Chapter 12, you learn about security used in cloud computing. This is an extension of our discussion on network management, because security is another means of controlling access through network policies.

Understanding Cloud Security

Cloud computing has lots of unique properties that make it very valuable. Unfortunately, many of those properties make security a singular concern. Many of the tools and techniques that you would use to protect your data, comply with regulations, and maintain the integrity of your systems are complicated by the fact that you are sharing your systems with others and many times outsourcing their operations as well. Cloud computing service providers are well aware of these concerns and have developed new technologies to address them.

Different types of cloud computing service models provide different levels of security services. You get the least amount of built in security with an Infrastructure as a Service provider, and the most with a Software as a Service provider. This chapter presents the concept of a security boundary separating the client's and vendor's responsibilities.

Adapting your on-premises systems to a cloud model requires that you determine what security mechanisms are required and mapping those to controls that exist in your chosen cloud service provider. When you identify missing security elements in the cloud, you can use that mapping to work to close the gap.

Storing data in the cloud is of particular concern. Data should be transferred and stored in an encrypted format. You can use proxy and brokerage services to separate clients from direct access to shared cloud storage.

Logging, auditing, and regulatory compliance are all features that require planning in cloud computing systems. They are among the services that need to be negotiated in Service Level Agreements.

Also in this chapter, you learn about identity and related protocols from a security standpoint. The concept of presence as it relates to identity is also introduced.

IN THIS CHAPTER

Reviewing cloud security concerns

Understanding how cloud data can be secured

Planning for security in your system

Learning how identity is used to allow secure cloud access

Securing the Cloud

The Internet was designed primarily to be resilient; it was not designed to be secure. Any distributed application has a much greater attack surface than an application that is closely held on a Local Area Network. Cloud computing has all the vulnerabilities associated with Internet applications, and additional vulnerabilities arise from pooled, virtualized, and outsourced resources.

In the report “Assessing the Security Risks of Cloud Computing,” Jay Heiser and Mark Nicolett of the Gartner Group (<http://www.gartner.com/DisplayDocument?id=685308>) highlighted the following areas of cloud computing that they felt were uniquely troublesome:

- Auditing
- Data integrity
- e-Discovery for legal compliance
- Privacy
- Recovery
- Regulatory compliance

Your risks in any cloud deployment are dependent upon the particular cloud service model chosen and the type of cloud on which you deploy your applications. In order to evaluate your risks, you need to perform the following analysis:

1. Determine which resources (data, services, or applications) you are planning to move to the cloud.
2. Determine the sensitivity of the resource to risk.
Risks that need to be evaluated are loss of privacy, unauthorized access by others, loss of data, and interruptions in availability.
3. Determine the risk associated with the particular cloud type for a resource.
Cloud types include public, private (both external and internal), hybrid, and shared community types. With each type, you need to consider where data and functionality will be maintained.
4. Take into account the particular cloud service model that you will be using.
Different models such as IaaS, SaaS, and PaaS require their customers to be responsible for security at different levels of the service stack.
5. If you have selected a particular cloud service provider, you need to evaluate its system to understand how data is transferred, where it is stored, and how to move data both in and out of the cloud.
You may want to consider building a flowchart that shows the overall mechanism of the system you are intending to use or are currently using.

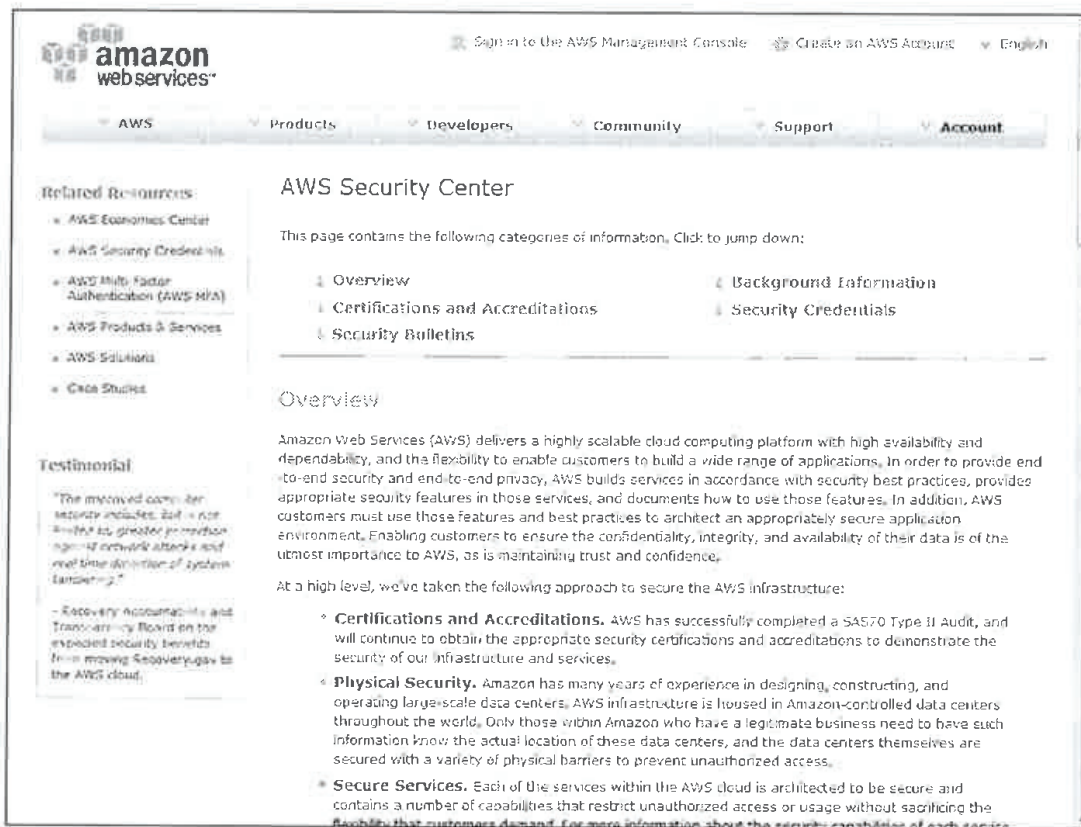
One technique for maintaining security is to have “golden” system image references that you can return to when needed. The ability to take a system image off-line and analyze the image for

vulnerabilities or compromise is invaluable. The compromised image is a primary forensics tool. Many cloud providers offer a snapshot feature that can create a copy of the client's entire environment; this includes not only machine images, but applications and data, network interfaces, firewalls, and switch access. If you feel that a system has been compromised, you can replace that image with a known good version and contain the problem.

Many vendors maintain a security page where they list their various resources, certifications, and credentials. One of the more developed offerings is the AWS Security Center, shown in Figure 12.1, where you can download some backgrounders, white papers, and case studies related to the Amazon Web Service's security controls and mechanisms.

FIGURE 12.1

The AWS Security Center (<http://aws.amazon.com/security/>) is a good place to start learning about how Amazon Web Services protects users of its IaaS service.



The security boundary

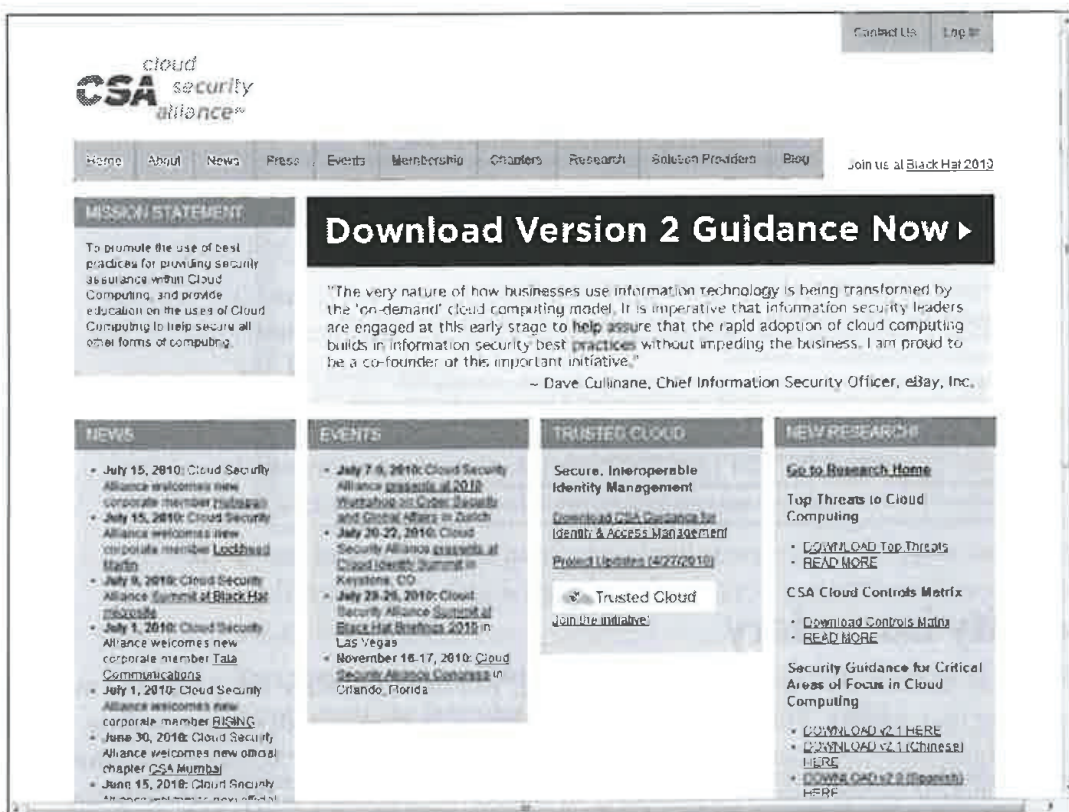
In order to concisely discuss security in cloud computing, you need to define the particular model of cloud computing that applies. This nomenclature provides a framework for understanding what security is already built into the system, who has responsibility for a particular security mechanism, and where the boundary between the responsibility of the service provider is separate from the responsibility of the customer.

All of Chapter 1 was concerned with defining what cloud computing is and defining the lexicon of cloud computing. There are many definitions and acronyms in the area of cloud computing that will probably not survive long. The most commonly used model based on U.S. National Institute of Standards and Technology (NIST; <http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html>) separates deployment models from service models and assigns those models a set of service attributes. Deployment models are cloud types: community, hybrid, private, and public clouds. Service models follow the SPI Model for three forms of service delivery: Software, Platform, and Infrastructure as a Service. In the NIST model, as you may recall, it was not required that a cloud use virtualization to pool resources, nor did that model require that a cloud support multi-tenancy. It is just these factors that make security such a complicated proposition in cloud computing.

Chapter 1 also presented the Cloud Security Alliance (CSA; <http://www.cloudsecurityalliance.org/>) cloud computing stack model, which shows how different functional units in a network stack relate to one another. As you may recall from Chapter 1, this model can be used to separate the different service models from one another. CSA is an industry working group that studies security issues in cloud computing and offers recommendations to its members. The work of the group is open and available, and you can download its guidance from its home page, shown in Figure 12.2.

FIGURE 12.2

The Cloud Security Alliance (CSA) home page at <http://www.cloudsecurityalliance.org/> offers a number of resources to anyone concerned with securing his cloud deployment.



The CSA partitions its guidance into a set of operational domains:

- Governance and enterprise risk management
- Legal and electronic discovery
- Compliance and audit
- Information lifecycle management
- Portability and interoperability
- Traditional security, business continuity, and disaster recovery
- Datacenter operations
- Incidence response, notification, and remediation
- Application security
- Encryption and key management
- Identity and access management
- Virtualization

You can download the group's current work in these areas from the different sections of its Web site.

One key difference between the NIST model and the CSA is that the CSA considers multi-tenancy to be an essential element in cloud computing. Multi-tenancy adds a number of additional security concerns to cloud computing that need to be accounted for. In multi-tenancy, different customers must be isolated, their data segmented, and their service accounted for. To provide these features, the cloud service provider must provide a policy-based environment that is capable of supporting different levels and quality of service, usually using different pricing models. Multi-tenancy expresses itself in different ways in the different cloud deployment models and imposes security concerns in different places.

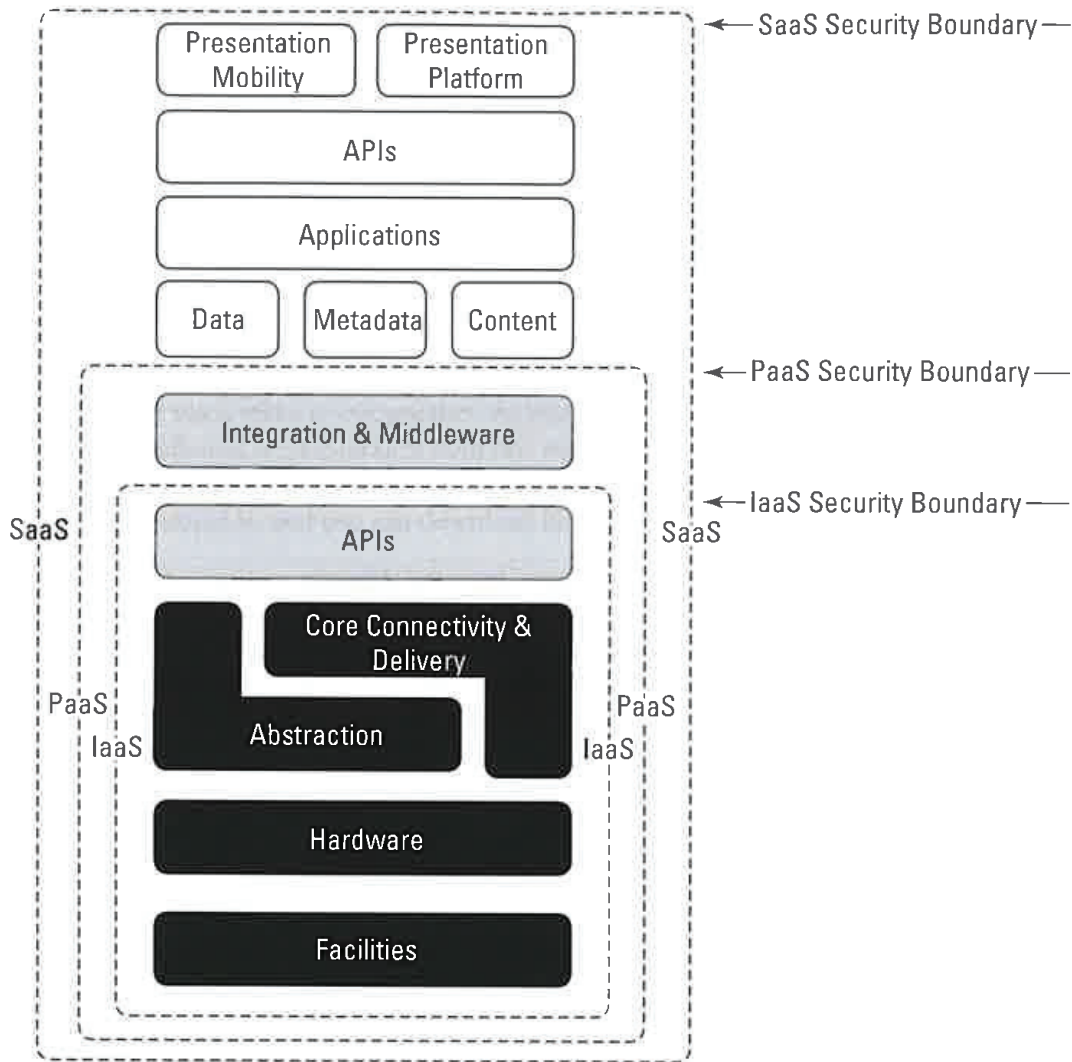
Security service boundary

The CSA functional cloud computing hardware/software stack is the Cloud Reference Model. This model, which was discussed in Chapter 1, is reproduced in Figure 12.3. IaaS is the lowest level service, with PaaS and SaaS the next two services above. As you move upward in the stack, each service model inherits the capabilities of the model beneath it, as well as all the inherent security concerns and risk factors. IaaS supplies the infrastructure; PaaS adds application development frameworks, transactions, and control structures; and SaaS is an operating environment with applications, management, and the user interface. As you ascend the stack, IaaS has the least levels of integrated functionality and the lowest levels of integrated security, and SaaS has the most.

The most important lesson from this discussion of architecture is that each different type of cloud service delivery model creates a security boundary at which the cloud service provider's responsibilities end and the customer's responsibilities begin. Any security mechanism below the security boundary must be built into the system, and any security mechanism above must be maintained by the customer. As you move up the stack, it becomes more important to make sure that the type and level of security is part of your Service Level Agreement.

FIGURE 12.3

The CSA Cloud Reference Model with security boundaries shown



In the SaaS model, the vendor provides security as part of the Service Level Agreement, with the compliance, governance, and liability levels stipulated under the contract for the entire stack. For the PaaS model, the security boundary may be defined for the vendor to include the software framework and middleware layer. In the PaaS model, the customer would be responsible for the security of the application and UI at the top of the stack. The model with the least built-in security is IaaS, where everything that involves software of any kind is the customer's problem. Numerous definitions of services tend to muddy this picture by adding or removing elements of the various functions from any particular offering, thus blurring which party has responsibility for which features, but the overall analysis is still useful.

In thinking about the Cloud Security Reference Model in relationship to security needs, a fundamental distinction may be made between the nature of how services are provided versus where those services are located. A private cloud may be internal or external to an organization, and although a public cloud is most often external, there is no requirement that this mapping be made so. Cloud computing has a tendency to blur the location of the defined security perimeter in such a way that the previous notions of network firewalls and edge defenses often no longer apply.

This makes the location of trust boundaries in cloud computing rather ill defined, dynamic, and subject to change depending upon a number of factors. Establishing trust boundaries and creating a new perimeter defense that is consistent with your cloud computing network is an important consideration. The key to understanding where to place security mechanisms is to understand where physically in the cloud resources are deployed and consumed, what those resources are, who manages the resources, and what mechanisms are used to control them. Those factors help you gauge where systems are located and what areas of compliance you need to build into your system.

Table 12.1 lists some of the different service models and lists the parties responsible for security in the different instances.

TABLE 12.1

Security Responsibilities by Service Model

Model Type	Infrastructure Security Management	Infrastructure Owner	Infrastructure Location	Trust Condition
Hybrid	Both vendor and customer	Both vendor and customer	Both on- and off-premises	Both trusted and untrusted
Private/Community	Customer	Customer	On- or off-premises	Trusted
Private/Community	Customer	Vendor	Off- or on-premises	Trusted
Private/Community	Vendor	Customer	On- or off-premises	Trusted
Private/Community	Vendor	Vendor	Off- or on-premises	Trusted
Public	Vendor	Vendor	Off-premises	Untrusted

Security mapping

The cloud service model you choose determines where in the proposed deployment the variety of security features, compliance auditing, and other requirements must be placed. To determine the particular security mechanisms you need, you must perform a mapping of the particular cloud service model to the particular application you are deploying. These mechanisms must be supported by the various controls that are provided by your service provider, your organization, or a third party. It's unlikely that you will be able to duplicate security routines that are possible on-premises, but this analysis allows you to determine what coverage you need.

A security control model includes the security that you normally use for your applications, data, management, network, and physical hardware. You may also need to account for any compliance standards that are required for your industry. A compliance standard can be any government regulatory framework such as Payment Card Industry Data Security Standards (PCI-DSS), Health Insurance Portability and Accountability Act (HIPPA), Gramm–Leach–Bliley Act (GLBA), or the Sarbanes–Oxley Act (SOX) that requires you operate in a certain way and keep records.

Essentially, you are looking to identify the missing features that would be required for an on-premises deployment and seek to find their replacements in the cloud computing model. As you assign accountability for different aspects of security and contract away the operational responsibility to others, you want to make sure they remain accountable for the security you need.

Securing Data

Securing data sent to, received from, and stored in the cloud is the single largest security concern that most organizations should have with cloud computing. As with any WAN traffic, you must assume that any data can be intercepted and modified. That's why, as a matter of course, traffic to a cloud service provider and stored off-premises is encrypted. This is as true for general data as it is for any passwords or account IDs.

These are the key mechanisms for protecting data mechanisms:

- Access control
- Auditing
- Authentication
- Authorization

Whatever service model you choose should have mechanisms operating in all four areas that meet your security requirements, whether they are operating through the cloud service provider or your own local infrastructure.

Brokered cloud storage access

The problem with the data you store in the cloud is that it can be located anywhere in the cloud service provider's system: in another datacenter, another state or province, and in many cases even in another country. With other types of system architectures, such as client/server, you could count on a firewall to serve as your network's security perimeter; cloud computing has no physical system that serves this purpose. Therefore, to protect your cloud storage assets, you want to find a way to isolate data from direct client access.

One approach to isolating storage in the cloud from direct client access is to create layered access to the data. In one scheme, two services are created: a broker with full access to storage but no access to the client, and a proxy with no access to storage but access to both the client and broker. The location of the proxy and the broker is not important (they can be local or in the cloud); what is important is that these two services are in the direct data path between the client and data stored in the cloud.

Under this system, when a client makes a request for data, here's what happens:

1. The request goes to the external service interface (or endpoint) of the proxy, which has only a partial trust.
2. The proxy, using its internal interface, forwards the request to the broker.
3. The broker requests the data from the cloud storage system.
4. The storage system returns the results to the broker.
5. The broker returns the results to the proxy.
6. The proxy completes the response by sending the data requested to the client.

Figure 12.4 shows this storage “proxy” system graphically.

Note

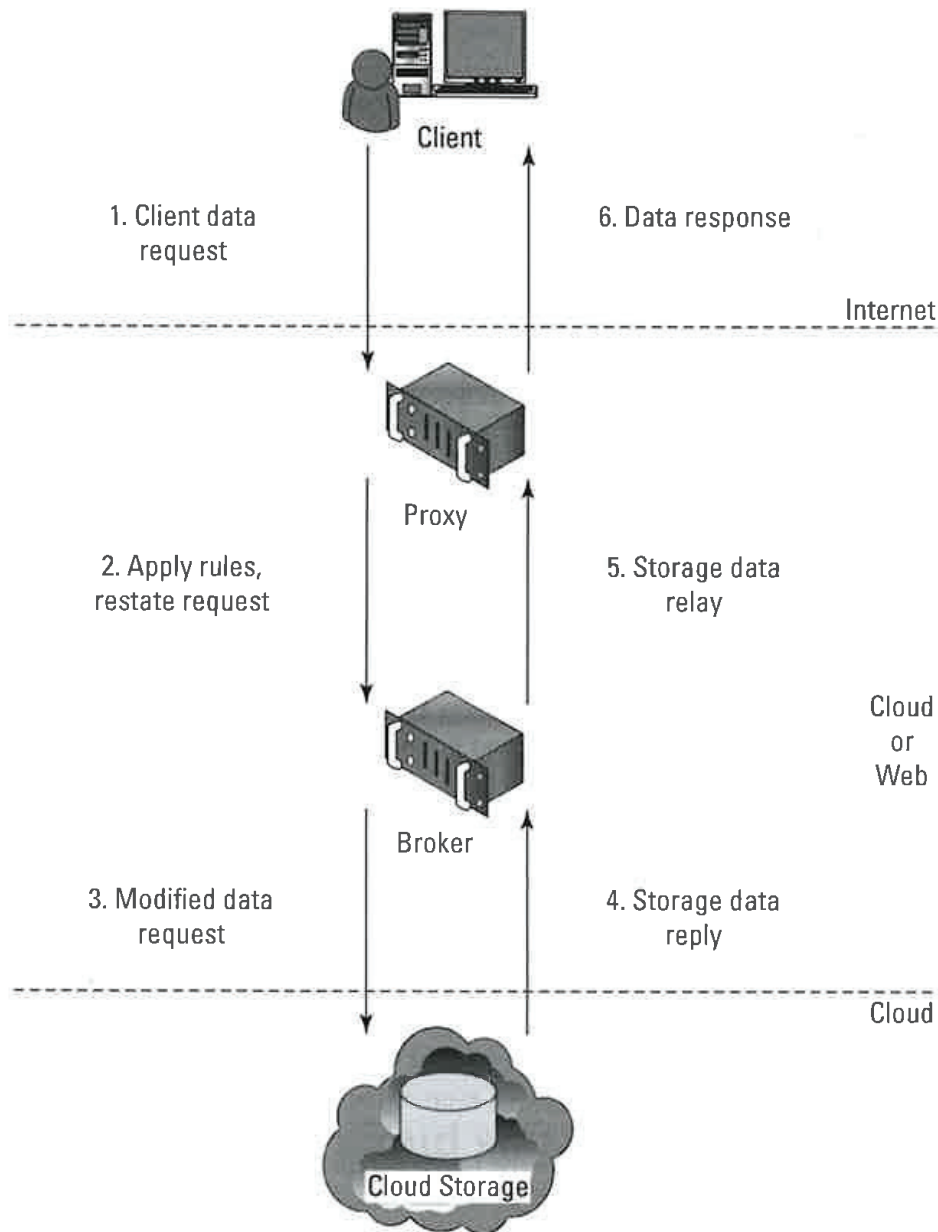
This discussion is based on a white paper called “Security Best Practices For Developing Windows Azure Applications,” by Andrew Marshall, Michael Howard, Grant Bugher, and Brian Harden that you can find at <http://download.microsoft.com/download/7/3/E/73E4EE93-559F-4D0F-A6FC-7FEC5F1542D1/SecurityBestPracticesWindowsAzureApps.docx>. In their presentation, the proxy service is called the Gatekeeper and assigned a Windows Server Web Role, and the broker is called the KeyMaster and assigned a Worker Role. ■

This design relies on the proxy service to impose some rules that allow it to safely request data that is appropriate to that particular client based on the client's identity and relay that request to the broker. The broker does not need full access to the cloud storage, but it may be configured to grant READ and QUERY operations, while not allowing APPEND or DELETE. The proxy has a limited trust role, while the broker can run with higher privileges or even as native code.

The use of multiple encryption keys can further separate the proxy service from the storage account. If you use two separate keys to create two different data zones—one for the untrusted communication between the proxy and broker services, and another a trusted zone between the broker and the cloud storage—you create a situation where there is further separation between the different service roles.

FIGURE 12.4

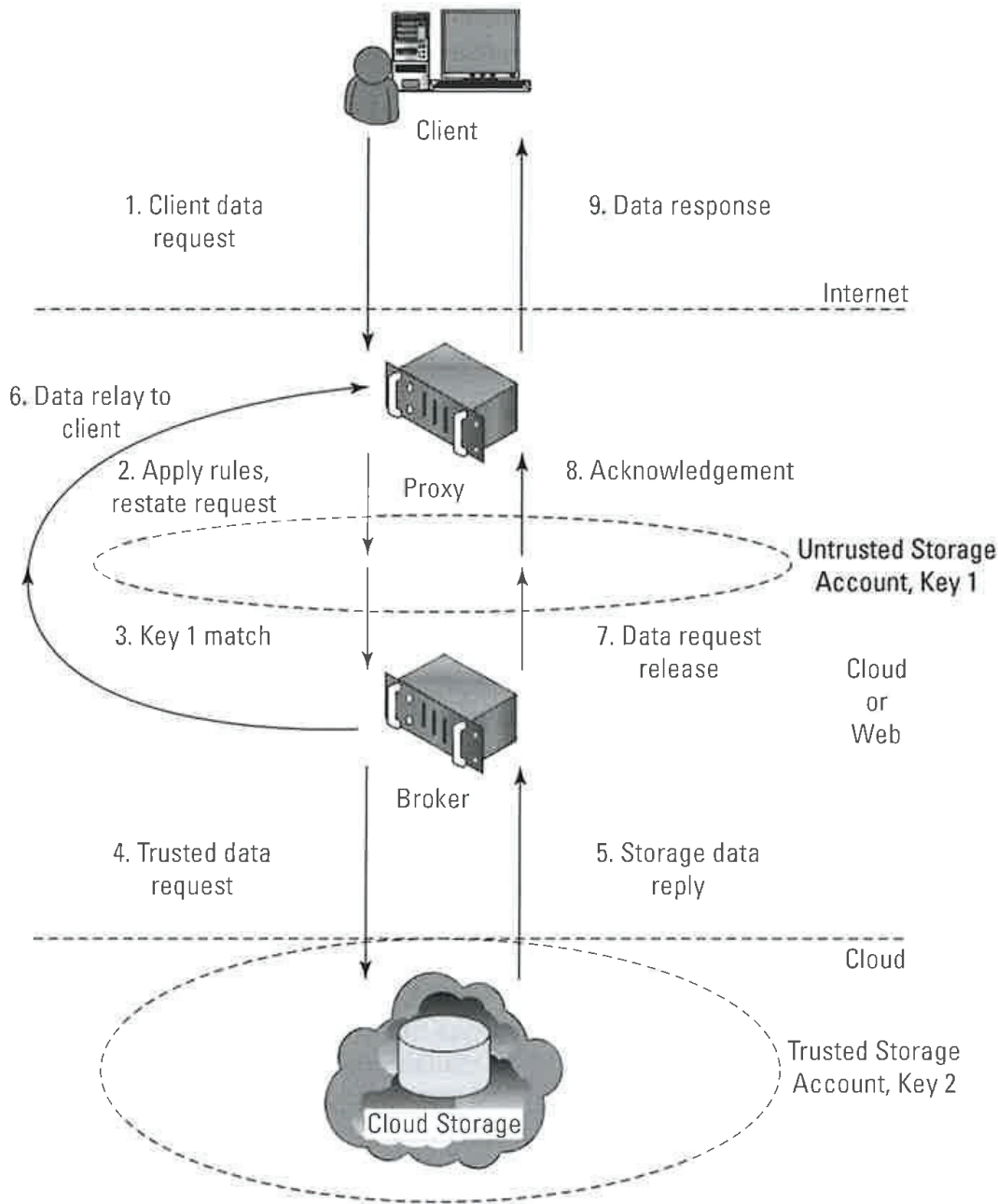
In this design, direct access to cloud storage is eliminated in favor of a proxy/broker service.



Even if the proxy service is compromised, that service does not have access to the trusted key necessary to access the cloud storage account. In the multi-key solution, shown in Figure 12.5, you have not only eliminated all internal service endpoints, but you also have eliminated the need to have the proxy service run at a reduced trust level.

FIGURE 12.5

The creation of storage zones with associated encryption keys can further protect cloud storage from unauthorized access.



Storage location and tenancy

Some cloud service providers negotiate as part of their Service Level Agreements to contractually store and process data in locations that are predetermined by their contract. Not all do. If you can

get the commitment for specific data site storage, then you also should make sure the cloud vendor is under contract to conform to local privacy laws.

Because data stored in the cloud is usually stored from multiple tenants, each vendor has its own unique method for segregating one customer's data from another. It's important to have some understanding of how your specific service provider maintains data segregation.

Another question to ask a cloud storage provider is who is provided privileged access to storage. The more you know about how the vendor hires its IT staff and the security mechanism put into place to protect storage, the better.

Most cloud service providers store data in an encrypted form. While encryption is important and effective, it does present its own set of problems. When there is a problem with encrypted data, the result is that the data may not be recoverable. It is worth considering what type of encryption the cloud provider uses and to check that the system has been planned and tested by security experts.

Regardless of where your data is located, you should know what impact a disaster or interruption will have on your service and your data. Any cloud provider that doesn't offer the ability to replicate data and application infrastructure across multiple sites cannot recover your information in a timely manner. You should know how disaster recovery affects your data and how long it takes to do a complete restoration.

Encryption

Strong encryption technology is a core technology for protecting data in transit to and from the cloud as well as data stored in the cloud. It is or will be required by law. The goal of encrypted cloud storage is to create a virtual private storage system that maintains confidentiality and data integrity while maintaining the benefits of cloud storage: ubiquitous, reliable, shared data storage. Encryption should separate stored data (data at rest) from data in transit.

Depending upon the particular cloud provider, you can create multiple accounts with different keys as you saw in the example with Windows Azure Platform in the previous section. Microsoft allows up to five security accounts per client, and you can use these different accounts to create different zones. On Amazon Web Service, you can create multiple keys and rotate those keys during different sessions.

Although encryption protects your data from unauthorized access, it does nothing to prevent data loss. Indeed, a common means for losing encrypted data is to lose the keys that provide access to the data. Therefore, you need to approach key management seriously. Keys should have a defined lifecycle. Among the schemes used to protect keys are the creation of secure key stores that have restricted role-based access, automated key stores backup, and recovery techniques. It's a good idea to separate key management from the cloud provider that hosts your data.

One standard for interoperable cloud-based key management is the OASIS Key Management Interoperability Protocol (KMIP; <http://www.oasis-open.org/committees/kmip/>). IEEE 1619.3 (https://siswg.net/index.php?option=com_docman) also covers both storage encryption and key management for shared storage.

Auditing and compliance

Logging is the recording of events into a repository; auditing is the ability to monitor the events to understand performance. Logging and auditing is an important function because it is not only necessary for evaluation performance, but it is also used to investigate security and when illegal activity has been perpetrated. Logs should record system, application, and security events, at the very minimum.

Logging and auditing are unfortunately one of the weaker aspects of early cloud computing service offerings.

Cloud service providers often have proprietary log formats that you need to be aware of. Whatever monitoring and analysis tools you use need to be aware of these logs and able to work with them. Often, providers offer monitoring tools of their own, many in the form of a dashboard with the potential to customize the information you see through either the interface or programmatically using the vendor's API. You want to make full use of those built-in services.

Because cloud services are both multitenant and multisite operations, the logging activity and data for different clients may not only be co-located, they may also be moving across a landscape of different hosts and sites. You can't simply expect that an investigation will be provided with the necessary information at the time of discovery unless it is part of your Service Level Agreement. Even an SLA with the appropriate obligations contained in it may not be enough to guarantee you will get the information you need when the time comes. It is wise to determine whether the cloud service provider has been able to successfully support investigations in the past.

As it stands now, nearly all regulations were written without keeping cloud computing in mind. A regulator or auditor isn't likely to be familiar with the nature of running applications and storing data in the cloud. Even so, laws are written to ensure compliance, and the client is held responsible for compliance under the laws of the governing bodies that apply to the location where the processing or storage takes place.

Therefore, you must understand the following:

- Which regulations apply to your use of a particular cloud computing service
- Which regulations apply to the cloud service provider and where the demarcation line falls for responsibilities
- How your cloud service provider will support your need for information associated with regulation
- How to work with the regulator to provide the information necessary regardless of who had the responsibility to collect the data

Traditional service providers are much more likely to be the subject of security certifications and external audits of their facilities and procedures than cloud service providers. That makes the willingness for a cloud service provider to subject its service to regulatory compliance scrutiny an important factor in your selection of that provider over another. In the case of a cloud service provider who shows reluctance to or limits the scrutiny of its operations, it is probably wise to use the

Part III: Exploring Cloud Infrastructures

service in ways that limit your exposure to risk. For example, although encrypting stored data is always a good policy, you also might want to consider not storing any sensitive information on that provider's system.

As it stands now, clients must guarantee their own regulatory compliance, even when their data is in the care of the service provider. You must ensure that your data is secure and that its integrity has not been compromised. When multiple regulatory entities are involved, as there surely are between site locations and different countries, then that burden to satisfy the laws of those governments is also your responsibility.

For any company with clients in multiple countries, the burden of regulatory compliance is onerous. While organizations such as the EEC (European Economic Community) or Common Market provide some relief for European regulation, countries such as the United States, Japan, China, and others each have their own sets of requirements. This makes regulatory compliance one of the most actively developing and important areas of cloud computing technology.

This situation is likely to change. On March 1, 2010, Massachusetts passed a law that requires companies that provide sensitive personal information on Massachusetts residents to encrypt data transmitted and stored on their systems. Businesses are required to limit the amount of personal data collected, monitor data usage, keep a data inventory, and be able to present a security plan on how they will keep the data safe. The steps require that companies verify that any third-party services they use conform to these requirements and that there be language in all SLAs that enforce these protections. The law takes full effect in March 2012.

Going forward, you want to ensure the following:

- You have contracts reviewed by your legal staff.
- You have a right-to-audit clause in your SLA.
- You review any third parties who are service providers and assess their impact on security and regulatory compliance.
- You understand the scope of the regulations that apply to your cloud computing applications and services.
- You consider what steps you must take to comply with the demands of regulations that apply.
- You consider adjusting your procedures to comply with regulations.
- You collect and maintain the evidence of your compliance with regulations.
- You determine whether your cloud service provider can provide an audit statement that is SAS 70 Type II-compliant.

The ISO/IEC 27001/27002 standard for information security management systems has a roadmap for mission-critical services that you may want to discuss with your cloud service provider. Amazon Web Services supports SAS70 Type II Audits.

Becoming a cloud service provider requires a large investment, but as we all know, even large companies can fail. When a cloud service provider fails, it may close or more likely be acquired by another company. You likely wouldn't use a service provider that you suspected of being in difficulty, but problems develop over years and cloud computing has a certain degree of vendor lock-in to it. That is, when you have created a cloud-based service, it can be difficult or often impossible to move it to another service provider. You should be aware of what happens to your data if the cloud service provider fails. At the very least, you would want to make sure your data could be obtained in a format that could be accessed by on-premise applications.

The various attributes of cloud computing make it difficult to respond to incidents, but that doesn't mean you should consider drawing up security incidence response policies. Although cloud computing creates shared responsibilities, it is often up to the client to initiate the inquiry that gets the ball rolling. You should be prepared to provide clear information to your cloud service provider about what you consider to be an incident or a breach in security and what are simply suspicious events.

Establishing Identity and Presence

Chapter 4 introduced the concept of identities, some of the protocols that support them, and some of the services that can work with them. Identities also are tied to the concept of accounts and can be used for contacts or "ID cards." Identities also are important from a security standpoint because they can be used to authenticate client requests for services in a distributed network system such as the Internet or, in this case, for cloud computing services.

Identity management is a primary mechanism for controlling access to data in the cloud, preventing unauthorized uses, maintaining user roles, and complying with regulations. The sections that follow describe some of the different security aspects of identity and the related concept of "presence." For this conversation, you can consider presence to be the mapping of an authenticated identity to a known location. Presence is important in cloud computing because it adds context that can modify services and service delivery.

Cloud computing requires the following:

- That you establish an identity
- That the identity be authenticated
- That the authentication be portable
- That authentication provide access to cloud resources

When applied to a number of users in a cloud computing system, these requirements describe systems that must provision identities, provide mechanisms that manage credentials and authentication, allow identities to be federated, and support a variety of user profiles and access policies. Automating these processes can be a major management task, just as they are for on-premises operations.

Identity protocol standards

The protocols that provide identity services have been and are under active development, and several form the basis for efforts to create interoperability among services.

OpenID 2.0 (<http://openid.net/>) is the standard associated with creating an identity and having a third-party service authenticate the use of that digital identity. It is the key to creating Single Sign-On (SSO) systems. Some cloud service providers have adopted OpenID as a service, and its use is growing.

In Chapter 4, you learned how OpenID is associated with contact cards such as vCards and InfoCards. In that chapter, I briefly discussed how OpenID provides access to important Web sites and how some Web sites allow you to use your logins based on OpenID from another site to gain access to their site.

OpenID doesn't specify the means for authentication of an identity, and it is up to the particular system how the authentication process is executed. Authentication can be by a Challenge and Response Protocol (CHAP), through a physical smart card, or using a flying finger or evil eye through a biometric measurement. In OpenID, the authentication procedure has the following steps:

1. The end-user uses a program like a browser that is called a user agent to enter an OpenID identifier, which is in the form of a URL or XRI.
An OpenID might take the form of `name.openid.provider.org`.
2. The OpenID is presented to a service that provides access to the resource that is desired.
3. An entity called a relaying party queries the OpenID identity provider to authenticate the veracity of the OpenID credentials.
4. The authentication is sent back to the relaying party from the identity provider and access is either provided or denied.

According to a report by one of OpenID's directors called "OpenID 2009 Year in Review" by Brian Kissel (<http://openid.net/2009/12/16/openid-2009-year-in-review/>), there were over 1 billion OpenID accounts accepted by 9 million sites on the Internet.

The second protocol used to present identity-based claims in cloud computing is a set of authorization markup languages that create files in the form of being XACML and SAML. These protocols were described in Chapter 4 in detail, so I only mention them in passing here. SAML (Security Assertion Markup Language; http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security) is gaining growing acceptance among cloud service providers. It is a standard of OASIS and an XML standard for passing authentication and authorization between an identity provider and the service provider. SAML is a complimentary mechanism to OpenID and is used to create SSO systems.

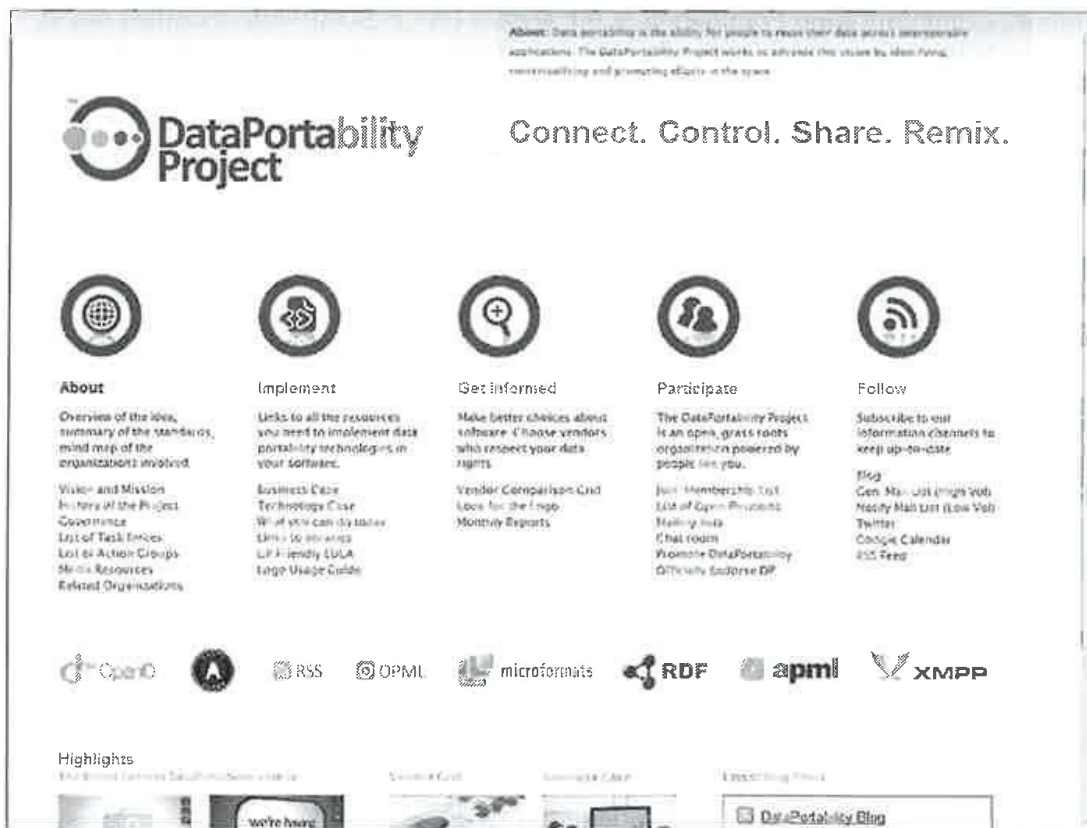
Taken as a unit, OpenID and SAML are being positioned to be the standard authentication mechanism for clients accessing cloud services. It is particularly important for services such as mashups that draw information from two or more data services.

An open standard called OAuth (<http://oauth.net/>) provides a token service that can be used to present validated access to resources. OAuth is similar to OpenID, but provides a different mechanism for shared access. The use of OAuth tokens allows clients to present credentials that contain no account information (userID or password) to a cloud service. The token comes with a defined period after which it can no longer be used. Several important cloud service providers have begun to make OAuth APIs available based on the OAuth 2.0 standard, most notably Facebook's Graph API and the Google Data API.

The DataPortability Project (<http://dataportability.org/>) is an industry working group that promotes data interoperability between applications, and the group's work touches on a number of the emerging standards mentioned in this section. The group's Web site is shown in Figure 12.6.

FIGURE 12.6

The home page of the DataPortability Project, an industry working group that promotes open identity standards



A number of vendors have created server products, such as Identity and Access Managers (IAMs), to support these various standards.

Windows Azure identity standards

The Windows Azure Platform uses a claims-based identity based on open authentication and access protocols and is a good example of a service implementing the standards described in the previous section. These standards may be used without modification on a system that is running in the cloud or on-premises, in keeping with Microsoft's S+S (software plus services) approach to cloud computing.

Windows Azure security draws on the following three services:

- Active Directory Federation Services 2.0
- Windows Azure AppFabric Access Control Service
- Windows Identity Foundation (WIF)

The Windows Identity Foundation offers .NET developers Visual Studio integration of WS-Federation and WS-Trust open standards. ASP.NET Web applications created with WIF integrate the Windows Communication Foundation SOAP service (WCF-SOAP) into a unified object model. This allows WIF to have full access to the features of WS-Security and to work with tokens in the SAML format.

WIF relies on third-party authentication and accepts authentication requests from these services in the form of a set of claims. Claims are independent of where a user account or application is located, thus allowing claims to be used in single sign-on systems (SSO). Claims support both simple resource access and the Role Based Access Control (RBAC) policies that can be enforced by Windows group policies.

Active Directory Federation Services 2.0 (AD FS) is a Security Token Service (STS) that allows users to authenticate their access to applications both locally and in the cloud with a claims-based identity. Anyone who has an account in the local Windows directory can access an application; AD FS creates and retains trust relationships with federated systems. AD FS uses WS-Federation, WS-Trust, and SAML, which allows users to access a system based on IBM, Novell, SAP, and many other vendors.

The final piece of the Windows Azure Platform claims-based identity system is built directly into the AppFabric Access Control (AC) service. You may recall from Chapter 10 that AppFabric is a service bus for Azure components that supports REST Web services. Included in AppFabric are authentication and claims-based authorization access. These can be simple logons or more complex schemes supported by AD FS. AC allows authorization to be located anywhere and allows developers to separate identity from their application.

The claims-based identity in AC is based on the OAuth Web Resource Authorization Protocol (OAuth WRAP), which works with various REST APIs. The OAuth 2.0 protocol seems to be gaining acceptance in the cloud computing industry, because SAML tokens can be accepted by many vendors.

Presence

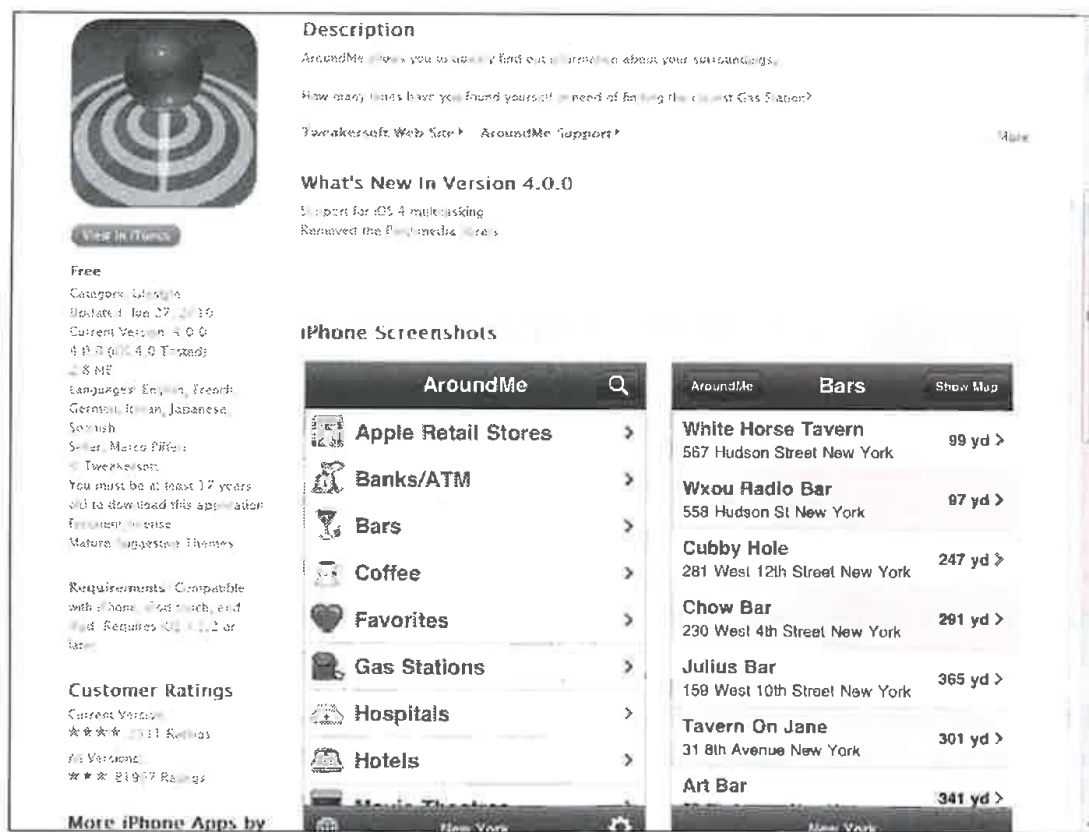
Presence is a fundamental concept in computer science. It is used on networks to indicate the status of available parties and their location. Commands like the WHO command in Linux that list users logged into the network go all the way back to the first network operating systems.

Presence provides not only identity, but status and, as part of status, location. The status is referred to as the presence state, the identity is the presentity, and the service that manages presence is called the presence service. Many presence services rely on agents called watchers, which are small programs that relay a client's ability to connect. Among the cloud computing services that rely on presence information are telephony systems such as VoIP, instant messaging services (IM), and geo-location-based systems such as GPS. Presence is playing an important role in cell phones, particularly smart phones.

When you access an application such as AroundMe on the Apple iPhone, which lists businesses, services, and restaurants in your vicinity, you are using an example of a presence service. Figure 12.7 shows the AroundMe app with some sample results. The presence service is provided by the GPS locator inside the phone, which provides a location through AT&T (the service provider) to the application. Presence is an essential and growing component of cloud-based services, and it adds a tremendous amount of value to the ubiquity that a cloud network offers.

FIGURE 12.7

The AroundMe iPhone app is an example of an application that makes use of a presence service.



As cloud computing becomes more pervasive and vendors attempt to create federated systems, emerging presence services will become more important. Microsoft's Windows Identity Foundation (described in the previous section) created under the Geneva Framework project is one example of an attempt to create a claims-based presence system. WIF allows different systems to interoperate using a variety of authentication methods, including LDAP and Active Directory, OpenID, LiveID, Microsoft CardSpace, and Novell Digital Me.

The Internet Engineering Task Force (IETF) has developed a standard called the Extensible Messaging and Presence Protocol (XMPP) that can be used with a federation system called the Jabber Extensible Communications Platform (Jabber XCP) to provide presence information. Among the services that use Jabber XCP are the Defense Information Systems Agency (DISA), Google Talk, Earthlink, Facebook, the National Weather Service, Twitter, the U.S. Marine Corps, and the U.S. Joint Forces Command (USJFCOM).

Jabber XCP is popular because it is an extensible development platform that is platform-independent and supports several communications protocols, such as the Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) and Instant Messaging and Presence Service (IMPS).

The notion of applying presence services over the standard Service Oriented Architecture (SOA) protocols such as SOAP/REST/HTTP is that in SOA all these protocols support unidirectional data exchange. You request a service/data, and a response is supplied. SOA architectures don't scale well and can't supply high-speed data transfers required by the collaboration services that are based on presence service technologies. SOA also has the problem of services that have trouble penetrating firewalls. It was these barriers that Jabber and XMPP were created to solve, and you will find that protocol incorporated into a number of cloud computing SaaS services. AOL, Apple, Google, IBM, and others are using this technology in some of their applications today.

Summary

In this chapter, you learned about many of the issues concerning security and cloud computing. This is a rapidly changing area of great importance to anyone considering deploying systems or storing data in the cloud. Security may, in fact, be the single most important area of cloud computing that you need to plan for. Depending upon the type of service you use, you may find many security services already built into your system.

An issue you need to consider is how to protect data in transit to and stored in the cloud. Data encryption, access restrictions, and data protection services were described. Multi-tenancy, system virtualization, and other factors make monitoring, regulatory compliance, and incidence response more challenging than on-premises systems are. A key concept for controlling access to cloud resources is identity management.

In Chapter 13, the standards used to create cloud systems are described. These standards create what is called the Service Oriented Architecture. You also learn how cloud systems can be built from many different interoperable modular parts.