

Service Quality of Cloud-Based Applications

Eric Bauer
Randee Adams

 **IEEE**
IEEE PRESS

WILEY

Table of Contents

[IEEE Press](#)

[Title page](#)

[Copyright page](#)

[Figures](#)

[Tables and Equations](#)

[Tables](#)

[Equations](#)

[1: Introduction](#)

[1.1 Approach](#)

[1.2 Target Audience](#)

[1.3 Organization](#)

[Acknowledgments](#)

[I: Context](#)

[2: Application Service Quality](#)

[2.1 Simple Application Model](#)

[2.2 Service Boundaries](#)

[2.3 Key Quality and Performance Indicators](#)

[2.4 Key Application Characteristics](#)

[2.5 Application Service Quality Metrics](#)

[2.6 Technical Service versus Support Service](#)
[2.7 Security Considerations](#)

[3: Cloud Model](#)

[3.1 Roles in Cloud Computing](#)
[3.2 Cloud Service Models](#)
[3.3 Cloud Essential Characteristics](#)
[3.4 Simplified Cloud Architecture](#)
[3.5 Elasticity Measurements](#)
[3.6 Regions and Zones](#)
[3.7 Cloud Awareness](#)

[4: Virtualized Infrastructure Impairments](#)

[4.1 Service Latency, Virtualization, and the Cloud](#)
[4.2 VM Failure](#)
[4.3 Nondelivery of Configured VM Capacity](#)
[4.4 Delivery of Degraded VM Capacity](#)
[4.5 Tail Latency](#)
[4.6 Clock Event Jitter](#)
[4.7 Clock Drift](#)
[4.8 Failed or Slow Allocation and Startup of VM Instance](#)
[4.9 Outlook for Virtualized Infrastructure Impairments](#)

[II: Analysis](#)

[5: Application Redundancy and Cloud Computing](#)

[5.1 Failures, Availability, and Simplex Architectures](#)

[5.2 Improving Software Repair Times via Virtualization](#)

[5.3 Improving Infrastructure Repair Times via Virtualization](#)

[5.4 Redundancy and Recoverability](#)

[5.5 Sequential Redundancy and Concurrent Redundancy](#)

[5.6 Application Service Impact of Virtualization Impairments](#)

[5.7 Data Redundancy](#)

[5.8 Discussion](#)

[6: Load Distribution and Balancing](#)

[6.1 Load Distribution Mechanisms](#)

[6.2 Load Distribution Strategies](#)

[6.3 Proxy Load Balancers](#)

[6.4 Nonproxy Load Distribution](#)

[6.5 Hierarchy of Load Distribution](#)

[6.6 Cloud-Based Load Balancing Challenges](#)

[6.7 The Role of Load Balancing in Support of Redundancy](#)

[6.8 Load Balancing and Availability Zones](#)

[6.9 Workload Service Measurements](#)

[6.10 Operational Considerations](#)

[6.11 Load Balancing and Application Service Quality](#)

[7: Failure Containment](#)

- [7.1 Failure Containment](#)
- [7.2 Points of Failure](#)
- [7.3 Extreme Solution Coresidency](#)
- [7.4 Multitenancy and Solution Containers](#)

[8: Capacity Management](#)

- [8.1 Workload Variations](#)
- [8.2 Traditional Capacity Management](#)
- [8.3 Traditional Overload Control](#)
- [8.4 Capacity Management and Virtualization](#)
- [8.5 Capacity Management in Cloud](#)
- [8.6 Storage Elasticity Considerations](#)
- [8.7 Elasticity and Overload](#)
- [8.8 Operational Considerations](#)
- [8.9 Workload Whipsaw](#)
- [8.10 General Elasticity Risks](#)
- [8.11 Elasticity Failure Scenarios](#)

[9: Release Management](#)

- [9.1 Terminology](#)
- [9.2 Traditional Software Upgrade Strategies](#)
- [9.3 Cloud-Enabled Software Upgrade Strategies](#)
- [9.4 Data Management](#)
- [9.5 Role of Service Orchestration in Software Upgrade](#)
- [9.6 Conclusion](#)

[10: End-to-End Considerations](#)

- [10.1 End-to-End Service Context](#)

- [10.2 Three-Layer End-to-End Service Model](#)
- [10.3 Distributed and Centralized Cloud Data Centers](#)
- [10.4 Multitiered Solution Architectures](#)
- [10.5 Disaster Recovery and Geographic Redundancy](#)

[III: Recommendations](#)

[11: Accountabilities for Service Quality](#)

- [11.1 Traditional Accountability](#)
- [11.2 The Cloud Service Delivery Path](#)
- [11.3 Cloud Accountability](#)
- [11.4 Accountability Case Studies](#)
- [11.5 Service Quality Gap Model](#)
- [11.6 Service Level Agreements](#)

[12: Service Availability Measurement](#)

- [12.1 Parsimonious Service Measurements](#)
- [12.2 Traditional Service Availability Measurement](#)
- [12.3 Evolving Service Availability Measurements](#)
- [12.4 Evolving Hardware Reliability Measurement](#)
- [12.5 Evolving Elasticity Service Availability Measurements](#)
- [12.6 Evolving Release Management Service Availability Measurement](#)
- [12.7 Service Measurement Outlook](#)

13: Application Service Quality Requirements

13.1 Service Availability Requirements

13.2 Service Latency Requirements

13.3 Service Reliability Requirements

13.4 Service Accessibility Requirements

13.5 Service Retainability Requirements

13.6 Service Throughput Requirements

13.7 Timestamp Accuracy Requirements

13.8 Elasticity Requirements

13.9 Release Management Requirements

13.10 Disaster Recovery Requirements

14: Virtualized Infrastructure Measurement and Management

14.1 Business Context for Infrastructure Service Quality Measurements

14.2 Cloud Consumer Measurement Options

14.3 Impairment Measurement Strategies

14.4 Managing Virtualized Infrastructure Impairments

15: Analysis of Cloud-Based Applications

15.1 Reliability Block Diagrams and Side-by-Side Analysis

15.2 IaaS Impairment Effects Analysis

15.3 PaaS Failure Effects Analysis

15.4 Workload Distribution Analysis

15.5 Anti-Affinity Analysis

[15.6 Elasticity Analysis](#)

[15.7 Release Management Impact Effects Analysis](#)

[15.8 Recovery Point Objective Analysis](#)

[15.9 Recovery Time Objective Analysis](#)

[16: Testing Considerations](#)

[16.1 Context for Testing](#)

[16.2 Test Strategy](#)

[16.3 Simulating Infrastructure Impairments](#)

[16.4 Test Planning](#)

[17: Connecting the Dots](#)

[17.1 The Application Service Quality Challenge](#)

[17.2 Redundancy and Robustness](#)

[17.3 Design for Scalability](#)

[17.4 Design for Extensibility](#)

[17.5 Design for Failure](#)

[17.6 Planning Considerations](#)

[17.7 Evolving Traditional Applications](#)

[17.8 Concluding Remarks](#)

[Abbreviations](#)

[References](#)

[About the Authors](#)

[Index](#)

IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

IEEE Press Editorial Board 2013

John Anderson, *Editor in Chief*

Linda Shafer	Saeid Nahavandi	George Zobrist
George W. Arnold	Om P. Malik	Tariq Samad
Ekram Hossain	Mary Lanzerotti	Dmitry Goldgof

Kenneth Moore, *Director of IEEE Book and Information Services (BIS)*

Technical Reviewers

Kim W. Tracy, *Northeastern Illinois University*
Rocky Heckman, *CISSP, Architect Advisor, Microsoft*

SERVICE QUALITY OF CLOUD-BASED APPLICATIONS

Eric Bauer
Randee Adams



IEEE PRESS

WILEY

Copyright © 2014 by The Institute of Electrical and Electronics Engineers, Inc.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
All rights reserved

Published simultaneously in Canada

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 Section 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, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. 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: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Bauer, Eric.

Service quality of cloud-based applications / Eric Bauer, Randee Adams.

pages cm

ISBN 978-1-118-76329-2 (cloth)

1. Cloud computing. 2. Application software--Reliability. 3. Quality of service (Computer networks) I. Adams, Randee. II. Title.

QA76.585.B3944 2013

004.67'82--dc23

2013026569

Figures

- [Figure 1.1.](#) Sample Cloud-Based Application.
- [Figure 2.0.](#) Organization of Part I: Context.
- [Figure 2.1.](#) Simple Cloud-Based Application.
- [Figure 2.2.](#) Simple Virtual Machine Service Model.
- [Figure 2.3.](#) Application Service Boundaries.
- [Figure 2.4.](#) KQIs and KPIs.
- [Figure 2.5.](#) Application Consumer and Resource Facing Service Indicators.
- [Figure 2.6.](#) Application Robustness.
- [Figure 2.7.](#) Sample Application Robustness Scenario.
- [Figure 2.8.](#) Interactivity Timeline.
- [Figure 2.9.](#) Service Latency.
- [Figure 2.10.](#) Small Sample Service Latency Distribution.
- [Figure 2.11.](#) Sample Typical Latency Variation by Workload Density.
- [Figure 2.12.](#) Sample Tail Latency Variation by Workload Density.
- [Figure 2.13.](#) Understanding Complimentary Cumulative Distribution Plots.
- [Figure 2.14.](#) Service Latency Optimization Options.
- [Figure 3.1.](#) Cloud Roles for Simple Application.
- [Figure 3.2.](#) Elastic Growth Strategies.
- [Figure 3.3.](#) Simple Model of Cloud Infrastructure.
- [Figure 3.4.](#) Abstract Virtual Machine Server.
- [Figure 3.5.](#) Provisioning Interval (T_{Grow}).
- [Figure 3.6.](#) Release Interval T_{Shrink} .
- [Figure 3.7.](#) VM Scale In and Scale Out.
- [Figure 3.8.](#) Horizontal Elasticity.
- [Figure 3.9.](#) Scale Up and Scale Down of a VM Instance.
- [Figure 3.10.](#) Idealized (Linear) Capacity Agility.
- [Figure 3.11.](#) Slew Rate of Square Wave Amplification.
- [Figure 3.12.](#) Elastic Growth Slew Rate and Linearity.
- [Figure 3.13.](#) Regions and Availability Zones.
- [Figure 4.1.](#) Virtualized Infrastructure Impairments Experienced by Cloud-Based Applications.
- [Figure 4.2.](#) Transaction Latency for Riak Benchmark.

- [Figure 4.3.](#) VM Failure Impairment Example.
- [Figure 4.4.](#) Simplified Nondelivery of VM Capacity Model.
- [Figure 4.5.](#) Characterizing Virtual Machine Nondelivery.
- [Figure 4.6.](#) Nondelivery Impairment Example.
- [Figure 4.7.](#) Simple Virtual Machine Degraded Delivery Model.
- [Figure 4.8.](#) Degraded Resource Capacity Model.
- [Figure 4.9.](#) Degraded Delivery Impairment Example.
- [Figure 4.10.](#) CCDF for Riak Read Benchmark for Three Different Hosting Configurations.
- [Figure 4.11.](#) Tail Latency Impairment Example.
- [Figure 4.12.](#) Sample CCDF for Virtualized Clock Event Jitter.
- [Figure 4.13.](#) Clock Event Jitter Impairment Example.
- [Figure 4.14.](#) Clock Drift Impairment Example.
- [Figure 5.1.](#) Simplex Distributed System.
- [Figure 5.2.](#) Simplex Service Availability.
- [Figure 5.3.](#) Sensitivity of Service Availability to MTRS (Log Scale).
- [Figure 5.4.](#) Traditional versus Virtualized Software Repair Times.
- [Figure 5.5.](#) Traditional Hardware Repair versus Virtualized Infrastructure Restoration Times.
- [Figure 5.6.](#) Simplified VM Repair Logic.
- [Figure 5.7.](#) Sample Automated Virtual Machine Repair-as-a-Service Logic.
- [Figure 5.8.](#) Simple Redundancy Model.
- [Figure 5.9.](#) Simplified High Availability Strategy.
- [Figure 5.10.](#) Failure in a Traditional (Sequential) Redundant Architecture.
- [Figure 5.11.](#) Sequential Redundancy Model.
- [Figure 5.12.](#) Sequential Redundant Architecture Timeline with No Failures.
- [Figure 5.13.](#) Sample Redundant Architecture Timeline with Implicit Failure.
- [Figure 5.14.](#) Sample Redundant Architecture Timeline with Explicit Failure.
- [Figure 5.15.](#) Recovery Times for Traditional Redundancy Architectures.
- [Figure 5.16.](#) Concurrent Redundancy Processing Model.
- [Figure 5.17.](#) Client Controlled Redundant Compute Strategy.
- [Figure 5.18.](#) Client Controlled Redundant Operations.
- [Figure 5.19.](#) Concurrent Redundancy Timeline with Fast but Erroneous Return.
- [Figure 5.20.](#) Hybrid Concurrent with Slow Response.
- [Figure 5.21.](#) Application Service Impact for Very Brief Nondelivery Events.

- [Figure 5.22.](#) Application Service Impact for Brief Nondelivery Events.
- [Figure 5.23.](#) Nondelivery Impact to Redundant Compute Architectures.
- [Figure 5.24.](#) Nondelivery Impact to Hybrid Concurrent Architectures.
- [Figure 6.1.](#) Proxy Load Balancer.
- [Figure 6.2.](#) Proxy Load Balancing.
- [Figure 6.3.](#) Load Balancing between Regions and Availability Zones.
- [Figure 7.1.](#) Reliability Block Diagram of Simplex Sample System (with SPOF).
- [Figure 7.2.](#) Reliability Block Diagram of Redundant Sample System (without SPOF).
- [Figure 7.3.](#) No SPOF Distribution of Component Instances across Virtual Servers.
- [Figure 7.4.](#) Example of No Single Point of Failure with Distributed Component Instances.
- [Figure 7.5.](#) Example of Single Point of Failure with Poorly Distributed Component Instances.
- [Figure 7.6.](#) Simplified VM Server Control.
- [Figure 8.1.](#) Sample Daily Workload Variation (Logarithmic Scale).
- [Figure 8.2.](#) Traditional Maintenance Window.
- [Figure 8.3.](#) Traditional Congestion Control.
- [Figure 8.4.](#) Simplified Elastic Growth of Cloud-Based Applications.
- [Figure 8.5.](#) Simplified Elastic Degrowth of Cloud-Based Applications.
- [Figure 8.6.](#) Sample of Erratic Workload Variation (Linear Scale).
- [Figure 8.7.](#) Typical Elasticity Orchestration Process.
- [Figure 8.8.](#) Example of Workload Whipsaw.
- [Figure 8.9.](#) Elastic Growth Failure Scenarios.
- [Figure 9.1.](#) Traditional Offline Software Upgrade.
- [Figure 9.2.](#) Traditional Online Software Upgrade.
- [Figure 9.3.](#) Type I, “Block Party” Upgrade Strategy.
- [Figure 9.4.](#) Application Elastic Growth and Type I, “Block Party” Upgrade.
- [Figure 9.5.](#) Type II, “One Driver per Bus” Upgrade Strategy.
- [Figure 10.1.](#) Simple End-to-End Application Service Context.
- [Figure 10.2.](#) Service Boundaries in End-to-End Application Service Context.
- [Figure 10.3.](#) Measurement Points 0–4 for Simple End-to-End Context.
- [Figure 10.4.](#) End-to-End Measurement Points for Simple Replicated Solution Context.
- [Figure 10.5.](#) Service Probes across User Service Delivery Path.

- [Figure 10.6.](#) Three Layer Factorization of Sample End to End Solution.
- [Figure 10.7.](#) Estimating Service Impairments across the Three-Layer Model.
- [Figure 10.8.](#) Decomposing a Service Impairment.
- [Figure 10.9.](#) Centralized Cloud Data Center Scenario.
- [Figure 10.10.](#) Distributed Cloud Data Center Scenario.
- [Figure 10.11.](#) Sample Multitier Solution Architecture.
- [Figure 10.12.](#) Disaster Recovery Time and Point Objectives.
- [Figure 10.13.](#) Service Impairment Model of Georedundancy.
- [Figure 11.1.](#) Traditional Three-Way Accountability Split: Suppliers, Customers, External.
- [Figure 11.2.](#) Example Cloud Service Delivery Chain.
- [Figure 11.3.](#) Service Boundaries across Cloud Delivery Chain.
- [Figure 11.4.](#) Functional Responsibilities for Applications Deployed on IaaS.
- [Figure 11.5.](#) Sample Application.
- [Figure 11.6.](#) Service Outage Accountability of Sample Application.
- [Figure 11.7.](#) Application Elasticity Configuration.
- [Figure 11.8.](#) Service Gap Model.
- [Figure 11.9.](#) Service Quality Zone of Tolerance.
- [Figure 11.10.](#) Application's Resource Facing Service Boundary.
- [Figure 11.11.](#) Application's Customer Facing Service Boundary.
- [Figure 12.1.](#) Traditional Service Operation Timeline.
- [Figure 12.2.](#) Sample Application Deployment on Cloud.
- [Figure 12.3.](#) "Network Element" Boundary for Sample Application.
- [Figure 12.4.](#) Logical Measurement Point for Application's Service Availability.
- [Figure 12.5.](#) Reliability Block Diagram of Sample Application (Traditional Deployment).
- [Figure 12.6.](#) Evolving Sample Application to Cloud.
- [Figure 12.7.](#) Reliability Block Diagram of Sample Application on Cloud.
- [Figure 12.8.](#) Side-by-Side Reliability Block Diagrams.
- [Figure 12.9.](#) Accountability of Sample Cloud Based Application.
- [Figure 12.10.](#) Connectivity-as-a-Service as a Nanoscale VPN.
- [Figure 12.11.](#) Sample Application with Database-as-a-Service.
- [Figure 12.12.](#) Accountability of Sample Application with Database-as-a-Service.
- [Figure 12.13.](#) Sample Application with Outboard RAID Storage Array.
- [Figure 12.14.](#) Sample Application with Storage-as-a-Service.
- [Figure 12.15.](#) Accountability of Sample Application with Storage-as-a-Service.

- [Figure 12.16.](#) Virtual Machine Failure Lifecycle.
- [Figure 12.17.](#) Elastic Capacity Growth Timeline.
- [Figure 12.18.](#) Outage Normalization for Type I “Block Party” Release Management.
- [Figure 12.19.](#) Outage Normalization for Type II “One Driver per Bus” Release Management.
- [Figure 13.1.](#) Maximum Acceptable Service Disruption.
- [Figure 14.1.](#) Infrastructure impairments and application impairments.
- [Figure 14.2.](#) Loopback and Service Latency.
- [Figure 14.3.](#) Simplified Measurement Architecture.
- [Figure 15.1.](#) Sample Side-by-Side Reliability Block Diagrams.
- [Figure 15.2.](#) Worst-Case Recovery Point Scenario.
- [Figure 15.3.](#) Best-Case Recovery Point Scenario.
- [Figure 16.1.](#) Measuring Service Disruption Latency.
- [Figure 16.2.](#) Service Disruption Latency for Implicit Failure.
- [Figure 16.3.](#) Sample Endurance Test Case for Cloud-Based Application.
- [Figure 17.1.](#) Virtualized Infrastructure Impairments Experienced by Cloud-Based Applications.
- [Figure 17.2.](#) Application Robustness Challenge.
- [Figure 17.3.](#) Sequential (Traditional) Redundancy.
- [Figure 17.4.](#) Concurrent Redundancy.
- [Figure 17.5.](#) Hybrid Concurrent with Slow Response.
- [Figure 17.6.](#) Type I, “Block Party” Upgrade Strategy.
- [Figure 17.7.](#) Sample Phased Evolution of a Traditional Application.

Tables and Equations

Tables

[TABLE 2.1.](#) Mean Opinion Scores [P.800]

[TABLE 13.1.](#) Service Availability and Downtime Ratings

Equations

[Equation 2.1.](#) Availability Formula

[Equation 5.1.](#) Simplex Availability

[Equation 5.2.](#) Traditional Availability

[Equation 10.1.](#) Estimating General End-to-End Service Impairments

[Equation 10.2.](#) Estimating End-to-End Service Downtime

[Equation 10.3.](#) Estimating End-to-End Service Availability

[Equation 10.4.](#) Estimating End-to-End Typical Service Latency

[Equation 10.5.](#) Estimating End-to-End Service Defect Rate

[Equation 10.6.](#) Estimating End-to-End Service Accessibility

[Equation 10.7.](#) Estimating End to End Service Retainability (as DPM)

[Equation 13.1.](#) DPM via Operations Attempted and Operations Successful

[Equation 13.2.](#) DPM via Operations Attempted and Operations Failed

[Equation 13.3.](#) DPM via Operations Successful and Operations Failed

[Equation 14.1.](#) Computing VM FITs

[Equation 14.2.](#) Converting FITs to MTBF

1

Introduction

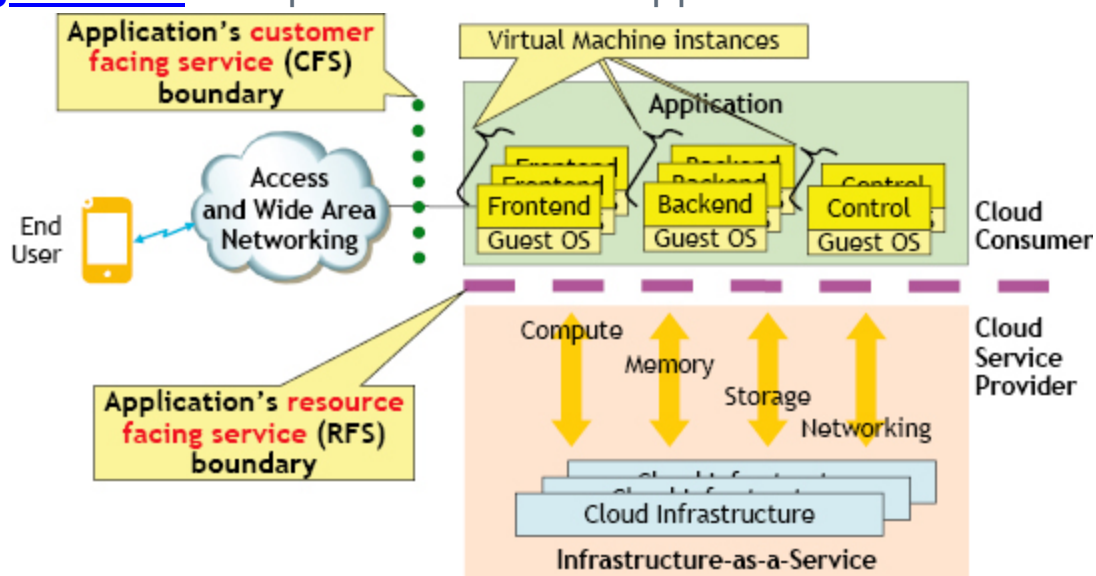
Customers expect that applications and services deployed on cloud computing infrastructure will deliver comparable service quality, reliability, availability, and latency as when deployed on traditional, native hardware configurations. Cloud computing infrastructure introduces a new family of service impairment risks based on the virtualized compute, memory, storage, and networking resources that an Infrastructure-as-a-Service (IaaS) provider delivers to hosted application instances. As a result, application developers and cloud consumers must mitigate these impairments to assure that application service delivered to end users is not unacceptably impacted. This book methodically analyzes the impacts of cloud infrastructure impairments on application service delivered to end users, as well as the opportunities for improvement afforded by cloud. The book also recommends architectures, policies, and other techniques to maximize the likelihood of delivering comparable or better service to end users when applications are deployed to cloud.

1.1 Approach

Cloud-based application software executes within a set of virtual machine instances, and each individual virtual machine instance relies on virtualized compute, memory, storage, and networking service delivered by the underlying cloud infrastructure. As shown in [Figure 1.1](#), the application

presents **customer facing service** toward end users across the dotted service boundary, and consumes virtualized resources offered by the Infrastructure-as-a-Service provider across the dashed **resource facing service** boundary. The application's service quality experienced by the end users is primarily a function of the application's architecture and software quality, as well as the service quality of the virtualized infrastructure offered by the IaaS across the resource facing service boundary, and the access and wide area networking that connects the end user to the application instance. This book considers both the new impairments and opportunities of virtualized resources offered to applications deployed on cloud and how user service quality experienced by end users can be maximized. By ignoring service impairments of the end user's device, and access and wide area network, one can narrowly consider how application service quality differs when a particular application is hosted on cloud infrastructure compared with when it is natively deployed on traditional hardware.

Figure 1.1. Sample Cloud-Based Application.



The key technical difference for application software between native deployment and cloud deployment is that

native deployments offer the application's (guest) operating system direct access to the physical compute, memory, storage, and network resources, while cloud deployment inserts a layer of hypervisor or virtual machine management software between the guest operating system and the physical hardware. This layer of hypervisor or virtual machine management software enables sophisticated resource sharing, technical features, and operational policies. However, the hypervisor or virtual machine management layer does not deliver perfect hardware emulation to the guest operating system and application software, and these imperfections can adversely impact application service delivered to end users. While [Figure 1.1](#) illustrates application deployment to a single data center, real world applications are often deployed to multiple data centers to improve user service quality by shortening transport latency to end users, to support business continuity and disaster recovery, and for other business reasons. Application service quality for deployment across multiple data centers is also considered in this book.

This book considers how application architectures, configurations, validation, and operational policies should evolve so that the acceptable application service quality can be delivered to end users even when application software is deployed on cloud infrastructure. This book approaches application service quality from the end users perspective while considering standards and recommendations from NIST, TM Forum, QuEST Forum, ODCA, ISO, ITIL, and so on.

1.2 Target Audience

This book provides application architects, developers, and testers with guidance on architecting and engineering applications that meet their customers' and end users' service reliability, availability, quality, and latency

expectations. Product managers, program managers, and project managers will also gain deeper insights into the service quality risks and mitigations that must be addressed to assure that an application deployed onto cloud infrastructure consistently meets or exceeds customers' expectations for user service quality.

1.3 Organization

The work is organized into three parts: context, analysis, and recommendations. **Part I: Context** frames the context of service quality of cloud-based applications via the following:

- *“Application Service Quality”* ([Chapter 2](#)). Defines the application service metrics that will be used throughout this work: service availability, service latency, service reliability, service accessibility, service retainability, service throughput, and timestamp accuracy.
- *“Cloud Model”* ([Chapter 3](#)). Explains how application deployment on cloud infrastructure differs from traditional application deployment from both a technical and an operational point of view, as well as what new opportunities are presented by rapid elasticity and massive resource pools.
- *“Virtualized Infrastructure Impairments”* ([Chapter 4](#)). Explains the infrastructure service impairments that applications running in virtual machines on cloud infrastructure must mitigate to assure acceptable quality of service to end users. The application service impacts of the impairments defined in this chapter will be rigorously considered in [Part II](#): Analysis.

Part II: Analysis methodically considers how application service defined in [Chapter 2](#), “Application Service Quality,” is impacted by the infrastructure impairments enumerated

in [Chapter 4](#), “Virtualized Infrastructure Impairments,” across the following topics:

- “*Application Redundancy and Cloud Computing*” ([Chapter 5](#)). Reviews fundamental redundancy architectures (simplex, sequential redundancy, concurrent redundancy, and hybrid concurrent redundancy) and considers their ability to mitigate application service quality impact when confronted with virtualized infrastructure impairments.
- “*Load Distribution and Balancing*” ([Chapter 6](#)). Methodically analyzes work load distribution and balancing for applications.
- “*Failure Containment*” ([Chapter 7](#)). Considers how virtualization and cloud help shape failure containment strategies for applications.
- “*Capacity Management*” ([Chapter 8](#)). Methodically analyzes application service risks related to rapid elasticity and online capacity growth and degrowth.
- “*Release Management*” ([Chapter 9](#)). Considers how virtualization and cloud can be leveraged to support release management actions.
- “*End-to-End Considerations*” ([Chapter 10](#)). Explains how application service quality impairments accumulate across the end-to-end service delivery path. The chapter also considers service quality implications of deploying applications to smaller cloud data centers that are closer to end users versus deploying to larger, regional cloud data centers that are farther from end users. Disaster recovery and georedundancy are also discussed.

[Part III: Recommendations](#) covers the following:

- “*Accountabilities for Service Quality*” ([Chapter 11](#)). Explains how cloud deployment profoundly changes traditional accountabilities for service quality and offers guidance for framing accountabilities across the cloud service delivery chain. The chapter also uses the service

gap model to review how to connect specification, architecture, implementation, validation, deployment, and monitoring of applications to assure that expectations are met. Service level agreements are also considered.

- *“Service Availability Measurement”* ([Chapter 12](#)). Explains how traditional application service availability measurements can be applied to cloud-based application deployments, thereby enabling efficient side-by-side comparisons of service availability performance.
- *“Application Service Quality Requirements”* ([Chapter 13](#)). Reviews high level service quality requirements for applications deployed to cloud.
- *“Virtualized Infrastructure Measurement and Management”* ([Chapter 14](#)). Reviews strategies for quantitatively measuring virtualized infrastructure impairments on production systems, along with strategies to mitigate the application service quality risks of unacceptable infrastructure performance.
- *“Analysis of Cloud-Based Applications”* ([Chapter 15](#)). Presents a suite of analysis techniques to rigorously assess the service quality risks and mitigations of a target application architecture.
- *“Testing Considerations”* ([Chapter 16](#)). Considers testing of cloud-based applications to assure that service quality expectations are likely to be met consistently despite inevitable virtualized infrastructure impairments.
- *“Connecting the Dots”* ([Chapter 17](#)). Discusses how to apply the recommendations of [Part III](#) to both existing and new applications to mitigate the service quality risks introduced in [Part I](#): Basics and analyzed in [Part II](#): Analysis.

As many readers are likely to study sections based on the technical needs of their business and their professional interest rather than strictly following this work's running

order, cross-references are included throughout the work so readers can, say, dive into detailed [Part II](#) analysis sections, and follow cross-references back into [Part I](#) for basic definitions and follow references forward to [Part III](#) for recommendations. A detailed index is included to help readers quickly locate material.

Acknowledgments

The authors acknowledge the consistent support of Dan Johnson, Annie Lequesne, Sam Samuel, and Lawrence Cowsar that enabled us to complete this work. Expert technical feedback was provided by Mark Clougherty, Roger Maitland, Rich Sohn, John Haller, Dan Eustace, Geeta Chauhan, Karsten Oberle, Kristof Boeynaems, Tony Imperato, and Chuck Salisbury. Data and practical insights were shared by Karen Woest, Srujal Shah, Pete Fales, and many others. Bob Brownlie offered keen insights into service measurements and accountabilities. Expert review and insight on release management for virtualized applications was provided by Bruce Collier. The work benefited greatly from insightful review feedback from Mark Cameron. Iraj Saniee, Katherine Guo, Indra Widjaja, Davide Cherubini, and Karsten Oberle offered keen and substantial insights. The authors gratefully acknowledge the external reviewers who took time to provide thorough review and thoughtful feedback that materially improved this book: Tim Coote, Steve Woodward, Herbert Ristock, Kim Tracy, and Xuemei Zhang.

The authors welcome feedback on this book; readers may e-mail us at Eric.Bauer@alcatel-lucent.com and Randee.Adams@alcatel-lucent.com.

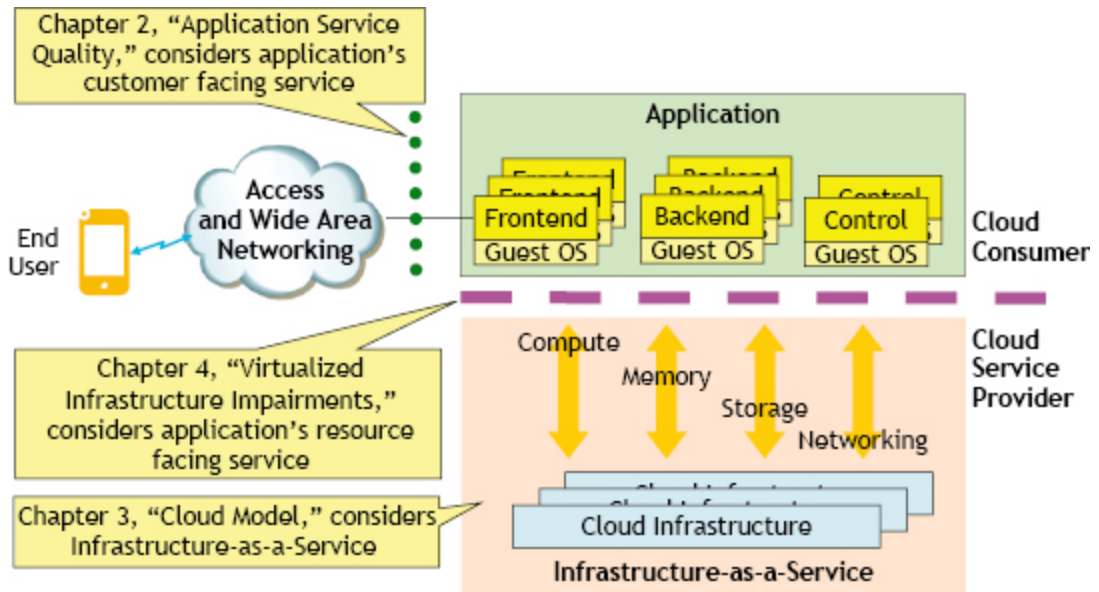
I

Context

[Figure 2.0](#) frames the context of this book: cloud-based applications rely on virtualized compute, memory, storage, and networking resources to provide information services to end users via access and wide area networks. The application's primary quality focus is on the user service delivered across the application's customer facing service boundary (dotted line in [Figure 2.0](#)).

- [Chapter 2](#), *"Application Service Quality,"* focuses on application service delivered across that boundary. The application itself relies on virtualized computer, memory, storage, and networking delivered by the cloud service provider to execute application software.
- [Chapter 3](#), *"Cloud Model,"* frames the context of the cloud service that supports this virtualized infrastructure.
- [Chapter 4](#), *"Virtualized Infrastructure Impairments,"* focuses on the service impairments presented to application components across the application's resource facing service boundary.

[Figure 2.0.](#) Organization of Part I: Context.



Application Service Quality

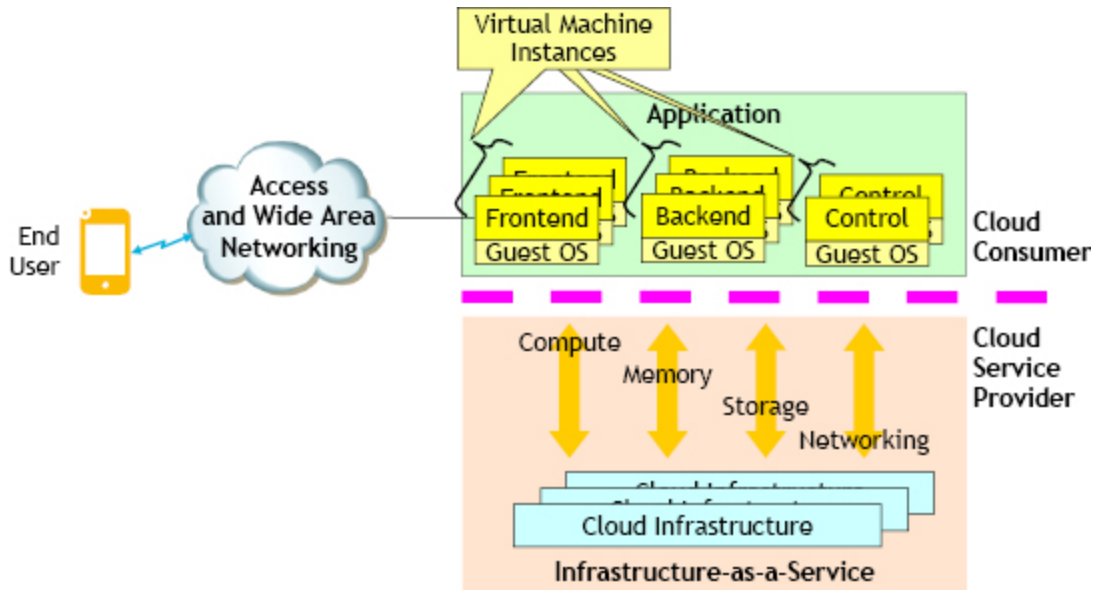
This section considers the service offered by applications to end users and the metrics used to characterize the quality of that service. A handful of common service quality metrics that characterize application service quality are detailed. These user service key quality indicators (KQIs) are considered in depth in [Part II](#): Analysis.

2.1 Simple Application Model

[Figure 2.1](#) illustrates a simple cloud-based application with a pool of frontend components distributing work across a pool of backend components. The suite of frontend and backend components is managed by a pair of control components that provide management visibility and control for the entire application instance. Each of the application's components, along with their supporting guest operating systems, execute in distinct virtual machine instances served by the cloud service provider. The Distributed Management Task Force (DMTF) defines ***virtual machine*** as:

the complete environment that supports the execution of guest software. A virtual machine is a full encapsulation of the virtual hardware, virtual disks, and the metadata associated with it. Virtual machines allow multiplexing of the underlying physical machine through a software layer called a hypervisor. [DSP0243]

[Figure 2.1.](#) Simple Cloud-Based Application.



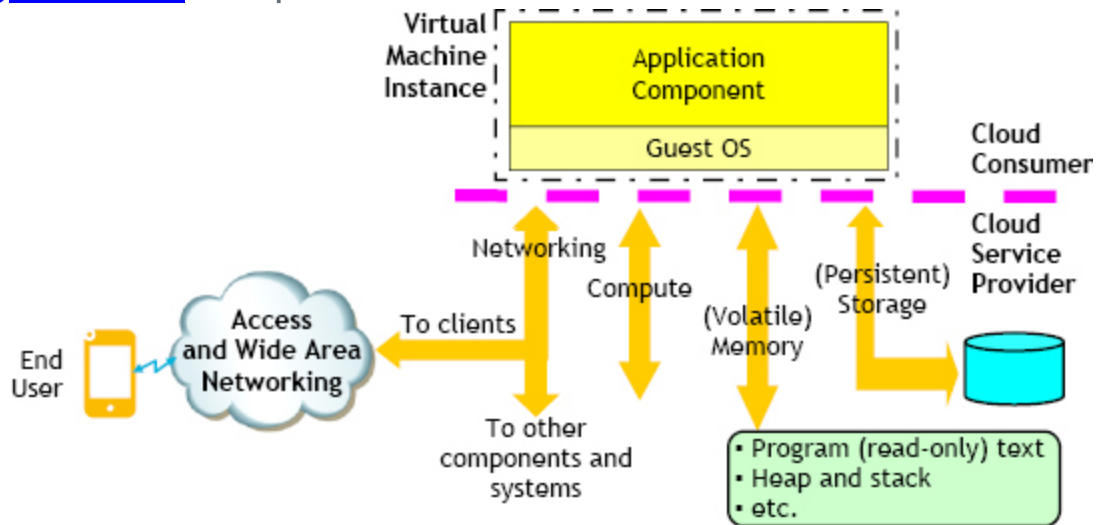
For simplicity, this simple model ignores systems that directly support the application, such as security appliances that protect the application from external attack, domain name servers, and so on.

[Figure 2.2](#) shows a single application component deployed in a virtual machine on cloud infrastructure. The application software and its underlying operating system—referred to as a *guest OS*—run within a virtual machine instance that emulates a dedicated physical server. The cloud service provider's infrastructure delivers the following resource services to the application's guest OS instance:

- *Networking*. Application software is networked to other application components, application clients, and other systems.
- *Compute*. Application programs ultimately execute on a physical processor.
- *(Volatile) Memory*. Applications execute programs out of memory, using heap memory, stack storage, shared memory, and main memory to maintain dynamic data, such as application state
- *(Persistent) Storage*. Applications maintain program executables, configuration, and application data on

persistent storage in files and file systems.

Figure 2.2. Simple Virtual Machine Service Model.



2.2 Service Boundaries

It is useful to define boundaries that demark applications and service offerings to better understand the dependencies, interactions, roles, and responsibilities of each element in overall user service delivery. This work will focus on the two high-level application service boundaries shown in [Figure 2.3](#):

- Application's **customer facing service** (CFS) boundary (dotted line in [Figure 2.3](#)), which demarks the edge of the application instance that faces users. User service reliability, such as call completion rate, and service latency, such as call setup, are well-known service quality measurements of telecommunications customer facing service.
- Application's **resource facing service** (RFS) boundary (dashed line in [Figure 2.3](#)), which demarks the boundary between the application's guest OS instances executing in virtual machine instances and the virtual compute, memory, storage, and networking provided by the cloud