Cloud Computing

Automating the Virtualized Data Center

Venkata Josyula
Malcolm Orr
Greg Page

ciscopress.com

Cloud Computing: Automating the Virtualized Data Center

Venkata Josyula
Malcolm Orr
Greg Page

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

Cloud Computing: Automating the Virtualized Data Center

Venkata Josyula
Malcolm Orr
Greg Page

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing December 2011

Library of Congress Cataloging-in-Publication Number is on file.

ISBN-13: 978-1-58720-434-0

ISBN-10: 1-58720-434-7

Warning and Disclaimer

This book is designed to provide information about cloud computing. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc., shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the authors and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact: U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact: International Sales international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Mary Beth Ray

Managing Editor: Sandra Schroeder

Project Editor: Mandie Frank

Editorial Assistant: Vanessa Evans

Cover Designer: Sandra Schroeder

Book Designer: Gary Schroeder

Business Operation Manager, Cisco Press: Anand Sundaram

Manager Global Certification: Erik Ullanderson

Senior Development Editor: Christopher Cleveland

Copy Editor: John Edwards

Technical Editors: Krishna Arji, Eric Charlesworth

Proofreader: Sheri Cain

Indexer: Erika Millen

Composition: Mark Shirar

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

About the Authors

Venkata (Josh) Josyula, Ph.D., CCIE No. 13518, is a distinguished services engineer (DSE) and lead solutions architect in Cisco Services Technology Group (CSTG). He has more than 25 years of diverse experience in network management for telecommunications and IP in a variety of positions, including systems engineering, technical marketing, consulting, customer management, and deployment.

Josh has been with Cisco for 11 years and, prior to that, worked at Bell Laboratories as a distinguished engineer. Josh has written and/or contributed to key ITU-T network management documents and served as advisory director for the TMF board. Josh has published more than 60 technical papers, reports, articles, and books and is frequently called upon by Cisco customers and internal Cisco engineers around the world for advice and presentations and to perform OSS assessment on OSS/BSS architecture and products.

Malcolm Orr (B.S.) is an enterprise architect within the Cisco Services Division. Malcolm focuses on advising telecommunication companies and large enterprise clients on how to architect, build, and operate NGN and cloud platforms. Malcolm has more than 18 years in the IT industry, of which the past 5 years he has spent at Cisco involved in architecting and delivering complex solutions to various clients. He currently is the lead architect for a number of Tier 1 public cloud projects within Cisco. Prior to joining Cisco, Malcolm was a principal consultant at AMDOCS, working on the BT 21CN transformation, and he was one of the founders and the technical director of Harbrook Consultants, a consulting firm specializing in network and system management.

Greg Page (B.A. (Hons.)) is a solutions architect for Cisco Systems within the presales Data Center architecture team. Greg has been working in the IT industry for 16 years (the last 11 with Cisco Systems) in a variety of technical consulting roles specializing in data center architecture and technology in addition to service provider security (CISSP #77673).

About the Technical Reviewers

Krishna Arji is a senior manager at Cisco. In this role, he is responsible for the development of technology that enables delivery of Cisco Services. Krishna has held various positions in the Services Technology Group at Cisco. Most recently, he played a key role in evaluating and developing technologies required for the delivery of cloud planning, design, and implementation services. Under his leadership, his team has developed several tools to perform routing, switching, data center, security, and WLAN assessments of customers' infrastructure. His areas of expertise include networking, software design and development, and data center technologies such as virtualization. Krishna holds a bachelor's degree in electronics and communications engineering, and he has a master's degree in enterprise software technologies. He has a patent pending with USPTO for Automated Assessments of Storage Area Networks (Serial No. 13/115,141).

Eric S. Charlesworth is a Technical Solutions Architect in the WW Data Center/Virtualization & Cloud architecture organization at Cisco Systems. Eric has more than 20 years of experience in the Data Center/Networking field and is currently focused on Cloud Computing and Data Center management. Formerly, he worked in various technical leadership positions at companies such as BellSouth and IBM. Eric is also a member of the review board for the Cloud Credential Council (www.cloudcredential.org) and helped to develop and approve the material in the program, as well as for the Cloud Challenge (www.cloudchallenge.com). As a technical editor, Eric has provided technical edits/reviews for major publishing companies, including Pearson Education and Van Haren Publishing.

Dedications

Venkata (Josh) Josyula Thanks to my family, colleagues, and my management for all the support.

Malcolm Orr To G for all the support, to mum and dad, finally something to make up for my 11+.

Greg Page To SGAL, my family and friends. Thanks for all your support and love over the years.

Acknowledgments

Venkata (Josh) Josyula I want to thank my family for the support at home and also like to thank my manager Sunil Kripalani for the encouragement. In addition, I'd like to thank the reviewers Krishna Arji and Eric Charlesworth. Also, I'd like to thank Charles Conte (now at Juniper), Jason Davis, Gopal Renganathan, Manish Jain, Paul Lam, and many other project members who were part of the DC/V project. Also special thanks to Chris, Mary Beth, and Mandie, from Cisco Press.

Malcolm Orr I would like to thanks James Urquart for his advice around cloud maturity, Aaron Kodra for his support in getting this done, and all my colleagues for putting up with me.

Greg Page I would like to thank my Cisco colleagues for their support, in particular my co-authors Malcolm and Josh, as well as John Evans, Thomas Reid, Eric Charlesworth, Uwe Lambrette, Wouter Belmans; and related to my early years at Cisco, Mark Grayson. Finally, thanks to Wendy Mars for giving me the opportunity and freedom to focus on the then emerging topic of 'Cloud'/IaaS.'

Contents at a Glance

Contents

Introduction

Cloud computing is a paradigm shift in the IT industry similar to the displacement of local electric generators with the electric grid, providing utility computing, and it is changing the nature of competition within the computer industry. There are over a hundred companies that claim they can provide cloud services. However, in most cases, they discuss server provisioning or data center automation.

Many leading IT vendors, such as Amazon, Google, Microsoft, IBM, HP, and Cisco, to name a few, believe that cloud computing is the next logical step in controlling IT resources, as well as a primary means to lower total cost of ownership. More than just an industry buzzword, cloud computing promises to revolutionize the way IT resources are deployed, configured, and managed for years to come. Service providers stand to realize tremendous value from moving toward this "everything as a service" delivery model. By expanding and using their infrastructure as a service, instead of dealing with a number of disparate and incompatible silos or the common single-tenant hosting and colocation model, service providers can offer high value to their customers.

This book provides a practical approach for building an architecture for providing virtualized/cloud services and Infrastructure as a Service (IaaS) specifically. Based on our experiences of working with many industry-leading management software vendors and system integrators, we have provided the most comprehensive knowledge that details how to manage the cloud architecture and provide cloud services. This book details management steps with practical example use cases and best practices to build a cloud that can be used by cloud consumers and providers.

Objectives of This Book

Cloud Computing: Automating the Virtualized Data Center provides exhaustive information on how to build and implement solution architectures for managing the cloud from start to finish. For novice users, this book provides information on clouds and a solution architecture approach for managing the cloud. For experienced, hands-on operations folks, this book provides information on how to set up and provision the Infrastructure as a Service (IaaS). For product specialists, this book covers what service providers look for in their products and discuss how their systems need to interact with other systems to provide an integrated solution that meets end-user needs.

This book evolved as we started working in the lab with major management software vendors to provision an end-to-end cloud infrastructure that consisted of compute, network, and storage resources. During the process, we found that most of the independent software vendors (ISV) could not meet the challenges of provisioning an end-to-end cloud infrastructure. This led us to work with the various Cisco software vendor partners to develop end-to-end integrated solutions for cloud management using Cisco and partner products. The solutions and the best practices in this book provide end-to-end architecture solutions and can be replicated and used in any lab and/or production network for the scenarios described in this book.

How This Book Is Organized

The book is divided into four parts:

Part I: Introduction to Managing Virtualization and Cloud Computing Environments

- **Chapter 1, "Cloud Computing Concepts":** This chapter illustrates the virtualization and cloud concepts. Virtualization and cloud computing are dovetailed, and vendors and solution providers are increasingly using virtualization to build private clouds. This chapter will discuss public, private, and hybrid clouds, as well as the benefits of on-site computing to cloud computing. This chapter will also provide information on types of services that can be provided on top of clouds, such as Infrastructure as a Service (IaaS), Software as a Service (SaaS), barriers to cloud adoption, and cloud benefits and return on investment (ROI).

- **Chapter 2, "Cloud Design Patterns and Use Cases":** This chapter illustrates typical application design patterns and use cases found in most enterprises today and discusses how these can be transitioned into the cloud.

- **Chapter 3, "Data Center Architecture and Technologies":** This chapter provides an overview of the architectural principles and the infrastructure designs needed to support a new generation of "real-time" managed IT service use cases. This chapter focuses on the building blocks, technologies, and con-

cepts that help simplify the design and operation of the data center.

- **Chapter 4, "IT Services":** This chapter describes the classification of IT services from both a business-centric and a technology-centric perspective. In addition, this chapter looks at the underpinning economics of IaaS and the contextual aspects of making a "workload" placement in the cloud, that is, risk versus cost.

- **Chapter 5, "The Cisco Cloud Strategy":** This chapter discusses Cisco Systems' corporate strategy, focusing on the technological, system, and service developments related to the cloud. This chapter also briefly covers the technology evolution toward the cloud to understand how we got to where we are today as an IT industry.

Part II: Managing Cloud Services

- **Chapter 6, "Cloud Management Reference Architecture":** This chapter discusses various industry standards and describes how they can be used to build a reference architecture. This chapter discusses ITIL, TMF, and ITU-TMN standards, and uses these standards to build a cloud reference architecture for process models, cloud frameworks, and management models. It gives recommendations on integration models between various management layers.

- **Chapter 7, "Service Fulfillment":** This chapter describes the details of cloud service fulfillment, also referred to as cloud service provisioning. Service fulfillment is responsible for delivering products and services to the customer. This includes order handling, service configuration and activation, and resource provisioning. Chapter 6 provided two reference architectures from a management perspective. This chapter builds on Chapter 6 and provides details on cloud service fulfillment and an end-to-end logical functional architecture for managing clouds. The end-to-end logical functional architecture is built based on the Tele-Management Forum (TMF) eTOM (enhanced Telecom Operations Map) and Information Technology Infrastructure Library (ITIL) V3 life cycle.

- **Chapter 8, "Service Assurance":** This chapter describes how infrastructure can be automated and how services can be provisioned from the time a customer orders a service to the time the service is provisioned. These services need to be monitored to provide high-quality services to the customers. This chapter discusses proactive and reactive maintenance activities, service monitoring (SLA/QoS), resource status and performance monitoring, and troubleshooting. This includes continuous resource status and performance monitoring to proactively detect possible failures, and the collection of performance data and analysis to identify and resolve potential or real problems.

- **Chapter 9, "Billing and Chargeback":** The ultimate goal of cloud computing is to provide a set of resources on demand when required and to provide an accurate usage of data. The choice to bill/charge or simply show this data to the consumer depends on many factors, all of which are discussed in this chapter. This chapter introduces cloud billing/charging terminology; billing considera-

tions for IaaS, PaaS, and SaaS; process flow from Order-to-Cash (OTC); and the billing/charging architecture for cloud services.

Part III: Managing Cloud Resources

- **Chapter 10, "Technical Building Blocks of IaaS":** This chapter describes how to design and build an IaaS service starting with the basic building blocks and evolving into a full-service catalogue. This chapter also discusses how service data is persisted in the cloud management systems and provides some thoughts on where cloud solutions will challenge traditional CMDB implementations.

- **Chapter 11, "Automating and Orchestration Resources":** Building on Chapter 10, this chapter explores how the service catalogue offers can be realized in the cloud infrastructure and describes best practices around provisioning, activating, and managing cloud services throughout their lifetime.

- **Chapter 12, "Cloud Capacity Management":** Optimizing any infrastructure is challenging, let alone when you factor in the sporadic, real-time demand that the cloud generates. This chapter outlines some of the key capacity challenges, describes the process around developing a capacity model, and discusses deploying tools to support this model.

- **Chapter 13, "Providing the Right Cloud User Experience":** The cloud fundamentally changes the way IT is consumed and delivered, and the key to being a successful cloud provider is the user experience. This chapter defines the typical roles that will interact with the cloud, their requirements, and some typical integration patterns that should be considered to achieve a consistent user experience.

- **Chapter 14, "Adopting Cloud from a Maturity Perspective":** Building and deploying a cloud will, in most cases, touch on organizational, process, and technology areas. Assessing where you as a potential cloud consumer or provider are in these three areas is a critical first step. This chapter provides a simple, extensible framework for assessing cloud maturity.

Part IV: Appendixes

- **Appendix A, "Case Study: Cloud Providers - Hybrid Cloud":** This case study brings together the concepts outlined in the book with an illustrative example showing the choices an IT and a telecommunications company make when considering the cloud from the consumer and provider perspectives. Although it is a fictional example, the case study is drawn from real-world experiences.

- **Appendix B, "Terms and Acronyms":** This appendix lists common acronyms, their expansions, and definitions for the cloud terminology used throughout this book.

Chapter 3

Data Center Architecture and Technologies

In this chapter, you will learn the following:

- How to articulate what is meant by "architecture" in relation to IT design
- Describe the data center architectural building blocks
- Describe the evolution of the data center network design with the impact of virtualization
- Describe the cloud Infrastructure as a Service (IaaS) solution and its functional components
- Describe the network services that are necessary to deliver an "end-to-end" service-level agreement

This chapter provides an overview of the architectural principles and infrastructure designs needed to support a new generation of real-time-managed IT service use cases in the data center. There are many process frameworks and technologies available to architects to deliver a service platform that is both flexible and scalable. From an operational perspective, maintaining visibility and control of the data center that meets the business's governance, risk, and compliance needs is a must. This chapter will discuss the building blocks, technologies, and concepts that help simplify the design and operation, yet deliver real IT value to the business, namely, business continuity and business change.

Architecture

Architecture is a borrowed term that is often overused in technology forums. The Oxford English Dictionary defines architecture as "the art or practice of designing and constructing buildings" and further, "the conceptual structure and logical organization of a computer or computer-based system."

In general, outside the world of civil engineering, the term *architecture* is a poorly understood concept. Although we can understand the concrete concept of a building and the

process of building construction, many of us have trouble understanding the more abstract concepts of a computer or a network and, similarly, the process of constructing an IT system like a service platform. Just like buildings, there are many different kinds of service platforms that draw upon and exhibit different architectural principles.

As an example of early architectural prinicples, requirements and/or guidelines (also known as *artifacts*), Figure 3-1 depicts the the famous drawing of Leonardo Da Vinci's "Vitruvian Man." We are told that the drawing is based on the ideas of a Roman Architect Marcus Vitruvius Pollio that a "perfect building" should be based on the fact (the mainly Christian religious idea) that man is created in the image of God and thus provides the blueprint of "proportional perfection" (that is, the relationship between the length of one body part to another is a constant fixed ratio). It was believed that these ratios can serve as a set of architectural principles when it comes to building design; thus, a "perfect building" can be acheived. Obviously, our ideas on architecture and design are much more secular and science-based today. That said, the Vitruvian Man provides a good a example of the relationship of architecture to design and its implimentation.

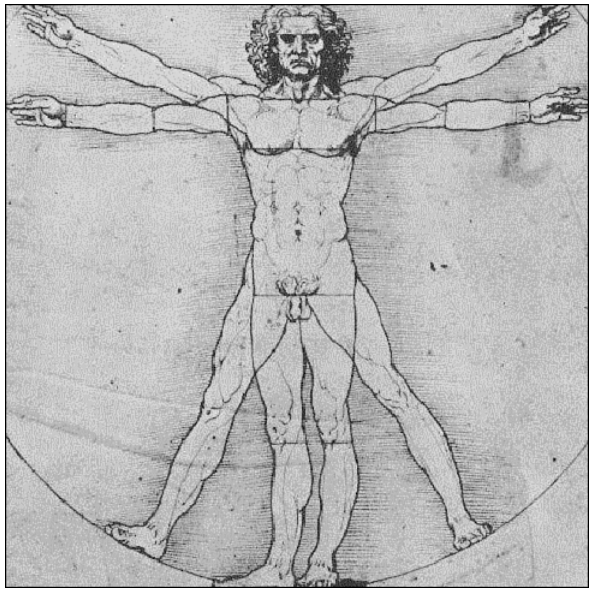

Figure 3-1 *Leonardo da Vinci's Vitruvian Man (Named After the Ancient Roman Architect Vitruvius)*

Even though architecture involves some well-defined activities, our first attempt at a definition uses the words *art* along with *science*. Unfortunately, for practical purposes, this definition is much too vague. But, one thing the definition does indirectly tell us is that architecture is simply part of the process of building things. For example, when building a new services platform, it is being built for a purpose and, when complete, is expected to have certain required principles.

The purpose of a "service delivery platform" is usually described to an architect by means of requirements documents that provide the goals and usage information for the platform that is to be built. Architects are typically individuals who have extensive experience in building IT systems that meet specific business requirements and translating those business requirements into IT engineering requirements. It is then up to subject matter experts (for example, server virtualization, networking, or storage engineers) to interpret the high-level architectural requirements into a low-level design and ultimately implement (build) a system ready for use. Figure 3-2 shows the many-to-one relationship among architecture, design, and implementations. Note that clear and well-understood communication among all stakeholders is essential throughout the project delivery phases to ensure success.

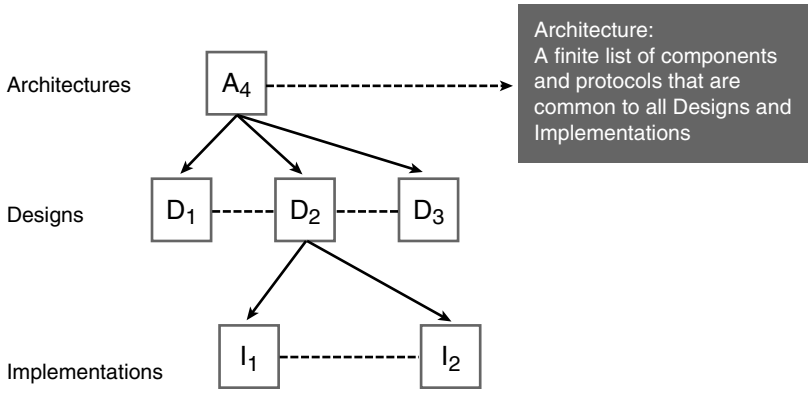

Figure 3-2 *Architecture Shapes the Design and Implementation of a System and/or Service*

Therefore, architecture is primarily used to communicate future system behavior to stakeholders and specify the building blocks for satisfying business requirements (this data is normally referred to as *artifacts*). A stakeholder is usually a person who pays for the effort and/or uses the end result. For example, a stakeholder could be the owner or a tenant of a future service platform, or a business owner or user of an anticipated network. Architecture blueprints are frequently used to communicate attributes of the system to the stakeholders before the system is actually built. In fact, the communication of multiple attributes usually requires multiple architecture documentation or blueprints. Unfortunately, architecture diagrams (usually multiple drawings) are often used incorrectly as design diagrams or vice versa.

With regard to cloud services, architecture must extend beyond on-premises (private cloud) deployments to support hybrid cloud models (hosted cloud, public cloud, community cloud, virtual private cloud, and so on). Architecture must also take into consideration Web 2.0 technologies (consumer social media services) and data access ubiquity (mobility).

Architectural principles that are required for a services platform today would most likely include but not be limited to efficiency, scalability, reliability, interoperability, flexibility, robustness, and modularity. How these principles are designed and implemented into a solution changes all the time as technology evolves.

With regard to implementing and managing architecture, process frameworks and methodologies are now heavily utilized to ensure quality and timely delivery by capitalizing of perceived industry best practices. Chapter 6, "Cloud Management Reference Architecture," covers frameworks in detail.

At this point, it is worth taking a few moments to discuss what exactly "IT value" is from a business perspective. Measuring value from IT investments has traditionally been an inexact science. The consequence is that many IT projects fail to fulfill their anticipated goals. Thus, many CIOs/CTOs today do not have much confidence in the accuracy of total cost of ownership (TCO), or more so, return on investment (ROI) modeling related to potential IT investments. A number of academic research projects with industry partnership have been conducted to look at better ways to approach this challenge.

One example would be the IT Capability Maturity Framework (IT-CMF), developed by the Innovation Value Institute (http://ivi.nuim.ie/ITCMF/index.shtml) along with Intel. Essentially, the IT-CMF provides a "capabilities maturity curve" (five levels of maturity) with a number of associated strategies aimed at delivering increasing IT value, thus ultimately supporting the business to maintain or grow sustainable differentiation in the marketplace.

The concept of *capability maturity* stems from the Software Engineering Institute (SEI), which originally developed what is known as the *Capability Maturity Model Integration (CMMI)*. In addition to the aforementioned IT-CMF, organizations can use the CMMI to map where they stand with respect to the best-in-class offering in relation to defined IT processes within Control Objectives for Information and Related Technology (COBIT) or how-to best practice guides like ITIL (Information Technology Infrastructure Library provides best practice for IT service management). Chapter 4, "IT Services," covers COBIT in detail.

Architectural Building Blocks of a Data Center

Data center design is at an evolutionary crossroads. Massive data growth, challenging economic conditions, and the physical limitations of power, heat, and space are exerting substantial pressure on the enterprise. Finding architectures that can take cost, complexity, and associated risk out of the data center while improving service levels has become a major objective for most enterprises. Consider the challenges facing enterprise IT organizations today.

Data center IT staff is typically asked to address the following data center challenges:

- Improve asset utilization to reduce or defer capital expenses.
- Reduce capital expenses through better management of peak workloads.

- Make data and resources available in real time to provide flexibility and alignment with current and future business agility needs.
- Reduce power and cooling consumption to cut operational costs and align with "green" business practices.
- Reduce deployment/churn time for new/existing services, saving operational costs and gaining competitive advantage in the market.
- Enable/increase innovation through new consumption models and the adoption of new abstraction layers in the architecture.
- Improve availability of services to avoid or reduce the business impact of unplanned outages or failures of service components.
- Maintain information assurance through consistent and robust security posture and processes.

From this set of challenges, you can derive a set of architectural principles that a new services platform would need to exhibit (as outlined in Table 3-1) to address the aforementioned challenges. Those architectural principles can in turn be matched to a set of underpinning technological requirements.

Table 3-1 *Technology to Support Architectural Principles*

Architectural Principles	Technological Requirements
Efficiency	Virtualization of infrastructure with appropriate management tools. Infrastructure homogeneity is driving asset utilization up.
Scalability	Platform scalability can be achieved through explicit protocol choice (for example, TRILL) and hardware selection and also through implicit system design and implementation.
Reliability	Disaster recovery (BCP) planning, testing, and operational tools (for example, VMware's Site Recovery Manager, SNAP, or Clone backup capabilities).
Interoperability	Web-based (XML) APIs, for example, WSDL (W3C) using SOAP or the conceptually simpler RESTful protocol with standards compliance semantics, for example, RFC 4741 NETCONF or TMForum's Multi-Technology Operations Systems Interface (MTOSI) with message binding to "concrete" endpoint protocols.
Flexibility	Software abstraction to enable policy-based management of the underlying infrastructure. Use of "meta models" (frames, rules, and constraints of how to build infrastructure). Encourage independence rather than interdependence among functional components of the platform.

Table 3-1 *Technology to Support Architectural Principles*

Architectural Principles	Technological Requirements
Modularity	Commonality of the underlying building blocks that can support scale-out and scale-up heterogeneous workload requirements with common integration points (web-based APIs). That is, integrated compute stacks or infrastructure packages (for example, a Vblock or a FlexPod). Programmatic workflows versus script-based workflows (discussed later in this chapter) along with the aforementioned software abstraction help deliver modularity of software tools.
Security	The appropriate countermeasures (tools, systems, processes, and protocols) relative to risk assessment derived from the threat model. Technology countermeasures are systems based, security in depth. Bespoke implementations/design patterns required to meet varied hosted tenant visibility and control requirements necessitated by regulatory compliance.
Robustness	System design and implementation—tools, methods, processes, and people that assist to mitigate collateral damage of a failure or failures internal to the administratively controlled system or even to external service dependencies to ensure service continuity.

Industry Direction and Operational and Technical Phasing

New technologies, such as multicore CPU, multisocket motherboards, inexpensive memory, and Peripheral Component Interconnect (PCI) bus technology, represent an evolution in the computing environment. These advancements, in addition to abstraction technologies (for example, virtual machine monitors [VMM], also known as hypervisor software), provide access to greater performance and resource utilization at a time of exponential growth of digital data and globalization through the Internet. Multithreaded applications designed to use these resources are both bandwidth intensive and require higher performance and efficiency from the underlying infrastructure.

Over the last few years, there have been iterative developments to the virtual infrastructure. Basic hypervisor technology with relatively simple virtual switches embedded in the hypervisor/VMM kernel have given way to far more sophisticated third-party distributed virtual switches (DVS) (for example, the Cisco Nexus 1000V) that bring together the operational domains of virtual server and the network, delivering consistent and integrated policy deployments. Other use cases, such as live migration of a VM, require orchestration of (physical and virtual) server, network, storage, and other dependencies to enable uninterrupted service continuity. Placement of capability and function needs to be carefully considered. Not every capability and function will have an optimal substantiation as a virtual entity; some might require physical substantiation because of performance or compliance reasons. So going forward, we see a hybrid model taking shape, with each capability and function being assessed for optimal placement with the architecture and design.

Although data center performance requirements are growing, IT managers are seeking ways to limit physical expansion by increasing the utilization of current resources. Server consolidation by means of server virtualization has become an appealing option. The use of multiple virtual machines takes full advantage of a physical server's computing potential and enables a rapid response to shifting data center demands. This rapid increase in computing power, coupled with the increased use of VM environments, is increasing the demand for higher bandwidth and at the same time creating additional challenges for the supporting networks.

Power consumption and efficiency continue to be some of the top concerns facing data center operators and designers. Data center facilities are designed with a specific power budget, in kilowatts per rack (or watts per square foot). Per-rack power consumption and cooling capacity have steadily increased over the past several years. Growth in the number of servers and advancement in electronic components continue to consume power at an exponentially increasing rate. Per-rack power requirements constrain the number of racks a data center can support, resulting in data centers that are out of capacity even though there is plenty of unused space.

Several metrics exist today that can help determine how efficient a data center operation is. These metrics apply differently to different types of systems, for example, facilities, network, server, and storage systems. For example, Cisco IT uses a measure of power per work unit performed instead of a measure of power per port because the latter approach does not account for certain use cases—the availability, power capacity, and density profile of mail, file, and print services will be very different from those of mission-critical web and security services. Furthermore, Cisco IT recognizes that just a measure of the network is not indicative of the entire data center operation. This is one of several reasons why Cisco has joined The Green Grid (www.thegreengrid.org), which focuses on developing data center–wide metrics for power efficiency. The power usage effectiveness (PUE) and data center efficiency (DCE) metrics detailed in the document "The Green Grid Metrics: Describing Data Center Power Efficiency" are ways to start addressing this challenge. Typically, the largest consumer of power and the most inefficient system in the data center is the Computer Room Air Conditioning (CRAC). At the time of this writing, state-of-the-art data centers have PUE values in the region of 1.2/1.1, whereas typical values would be in the range of 1.8–2.5. (For further reading on data center facilities, check out the book *Build the Best Data Center Facility for Your Business*, by Douglas Alger from Cisco Press.)

Cabling also represents a significant portion of a typical data center budget. Cable sprawl can limit data center deployments by obstructing airflows and requiring complex cooling system solutions. IT departments around the world are looking for innovative solutions that will enable them to keep up with this rapid growth with increased efficiency and low cost. We will discuss Unified Fabric (enabled by virtualization of network I/O) later in this chapter.

Current Barriers to Cloud/Utility Computing/ITaaS

It's clear that a lack of trust in current cloud offerings is the main barrier to broader adoption of cloud computing. Without trust, the economics and increased flexibility of cloud computing make little difference. For example, from a workload placement perspective, how does a customer make a cost-versus-risk (Governance, Risk, Compliance [GRC]) assessment without transparency of the information being provided? Transparency requires well-defined notations of service definition, audit, and accountancy. Multiple industry surveys attest to this. For example, as shown in Figure 3-3, Colt Technology Services' CIO Cloud Survey 2011 shows that most CIOs consider security as a barrier to cloud service adoption, and this is ahead of standing up the service (integration issues)! So how should we respond to these concerns?

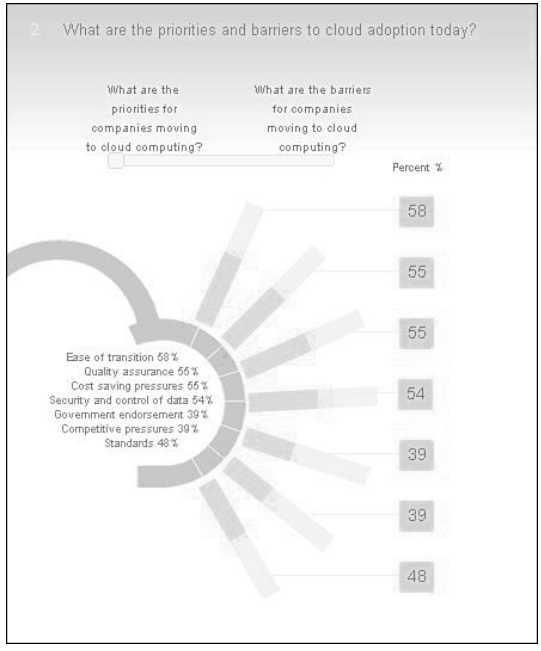

Figure 3-3 *CTS' CIO Cloud Survey 2011 (www.colt.net/cio-research)*

Trust in the cloud, Cisco believes, centers on five core concepts. These challenges keep business leaders and IT professionals alike up at night, and Cisco is working to address them with our partners:

- **Security:** Are there sufficient information assurance (IA) processes and tools to enforce confidentiality, integrity, and availability of the corporate data assets? Fears around multitenancy, the ability to monitor and record effectively, and the transparency of security events are foremost in customers' minds.

- **Control:** Can IT maintain direct control to decide how and where data and software are deployed, used, and destroyed in a multitenant and virtual, morphing infrastructure?
- **Service-level management:** Is it reliable? That is, can the appropriate Resource Usage Records (RUR) be obtained and measured appropriately for accurate billing? What if there's an outage? Can each application get the necessary resources and priority needed to run predictably in the cloud (capacity planning and business continuance planning)?
- **Compliance:** Will my cloud environment conform with mandated regulatory, legal, and general industry requirements (for example, PCI DSS, HIPAA, and Sarbanes-Oxley)?
- **Interoperability:** Will there be a vendor lock-in given the proprietary nature of today's public clouds? The Internet today has proven popular to enterprise businesses in part because of the ability to reduce risk through "multihoming" network connectivity to multiple Internet service providers that have diverse and distinct physical infrastructures.

For cloud solutions to be truly secure and trusted, Cisco believes they need an underlying network that can be relied upon to support cloud workloads.

To solve some of these fundamental challenges in the data center, many organizations are undertaking a journey. Figure 3-4 represents the general direction in which the IT industry is heading. The figure maps the operational phases (Consolidation, Virtualization, Automation, and so on) to enabling technology phases (Unified Fabric, Unified Computing, and so on).

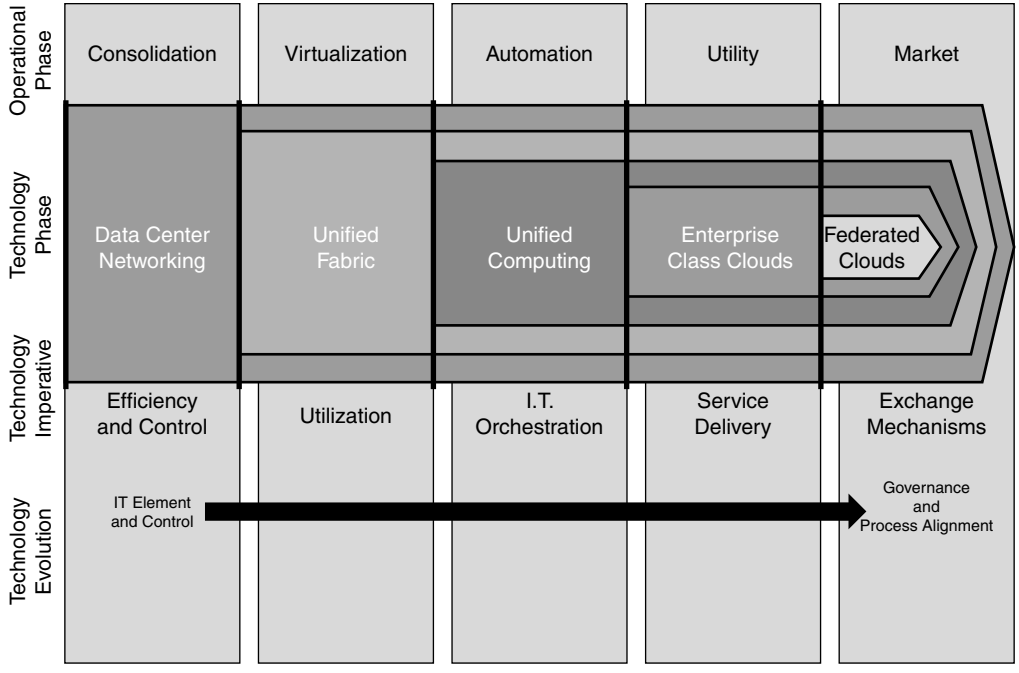

Figure 3-4 *Operational and Technological Evolution Stages of IT*

Organizations that are moving toward the adoption and utilization of cloud services tend to follow these technological phases:

1. Adoption of a broad IP WAN that is highly available (either through an ISP or self-built over dark fiber) enables centralization and consolidation of IT services. Application-aware services are layered on top of the WAN to intelligently manage application performance.
2. Executing on a virtualization strategy for server, storage, networking, and networking services (session load balancing, security apps, and so on) enables greater flexibility in the substantiation of services in regard to physical location, thereby enabling the ability to arrange such service to optimize infrastructure utilization.
3. Service automation enables greater operational efficiencies related to change control, ultimately paving the way to an economically viable on-demand service consumption model. In other words, building the "service factory."
4. Utility computing model includes the ability meter, chargeback, and bill customer on a pay-as-you-use (PAYU) basis. Showback is also a popular service: the ability to show current, real-time service and quota usage/consumption including future trending. This allows customers to understand and control their IT consumption. Showback is a fundamental requirement of service transparency.
5. Market creation through a common framework incorporating governance with a service ontology that facilitates the act of arbitrating between different service offerings and service providers.

Phase 1: The Adoption of a Broad IP WAN That Is Highly Available

This connectivity between remote locations allows IT services that were previously distributed (both from a geographic and organizational sense) to now be centralized, providing better operational control over those IT assets.

The constraint of this phase is that many applications were written to operate over a LAN and not a WAN environment. Rather than rewriting applications, the optimal economic path forward is to utilize application-aware, network-deployed services to enable a consistent Quality of Experience (QoE) to the end consumer of the service. These services tend to fall under the banner of Application Performance Management (APM) (www.cisco.com/go/apm). APM includes capabilities such as visibility into application response times, analysis of which applications and branch offices use how much bandwidth, and the ability to prioritize mission-critical applications, such as those from Oracle and SAP, as well as collaboration applications such as Microsoft SharePoint and Citrix.

Specific capabilities to deliver APM are as follows:

- **Performance monitoring:** Both in the network (transactions) and in the data center (application processing).

- **Reporting:** For example, application SLA reporting requires service contextualization of monitoring data to understand the data in relation to its expected or requested performance parameters. These parameters are gleaned from who the service owner is and the terms of his service contract.
- **Application visibility and control:** Application control gives service providers dynamic and adaptive tools to monitor and assure application performance.

Phase 2: Executing on a Virtualization Strategy for Server, Storage, Networking, and Networking Services

There are many solutions available on the market to enable server virtualization. Virtualization is the concept of creating a "sandbox" environment, where the computer hardware is abstracted to an operating system. The operating system is presented generic hardware devices that allow the virtualization software to pass messages to the physical hardware such as CPUs, memory, disks, and networking devices. These sandbox environments, also known as virtual machines (VM), include the operating system, the applications, and the configurations of a physical server. VMs are hardware independent, making them very portable so that they can run on any server.

Virtualization technology can also be applicable to many different areas such as networking and storage. LAN switching, for example, has the concept of a virtual LAN (VLAN) and routing with Virtual Routing and Forwarding (VRF) tables; storage-area networks have something similar in terms of virtual storage-area networks (VSAN), vFiler for NFS storage virtualization, and so on.

However, there is a price to pay for all this virtualization: management complexity. As virtual resources become abstracted from physical resources, existing management tools and methodologies start to break down in regard to their control effectiveness, particularly when one starts adding scale into the equation. New management capabilities, both implicit within infrastructure components or explicitly in external management tools, are required to provide the visibility and control service operations teams required to manage the risk to the business.

Unified Fabric based on IEEE Data Center Bridging (DCB) standards (more later) is a form of abstraction, this time by virtualizing Ethernet. However, this technology unifies the way that servers and storage resources are connected, how application delivery and core data center services are provisioned, how servers and data center resources are interconnected to scale, and how server and network virtualization is orchestrated.

To complement the usage of VMs, virtual applications (vApp) have also been brought into the data center architecture to provide policy enforcement within the new virtual infrastructure, again to help manage risk. Virtual machine-aware network services such as VMware's vShield and Virtual Network Services from Cisco allow administrators to provide services that are aware of tenant ownership of VMs and enforce service domain isolation (that is, the DMZ). The Cisco Virtual Network Services solution is also aware of the location of VMs. Ultimately, this technology allows the administrator to tie together service policy to location and ownership of an application residing with a VM container.

The Cisco Nexus 1000V vPath technology allows policy-based traffic steering to "invoke" vApp services (also known as *policy enforcement points [PEP]*), even if they reside on a separate physical ESX host. This is the start of *Intelligent Service Fabrics (ISF)*, where the traditional IP or MAC-based forwarding behavior is "policy hijacked" to substantiate service chain–based forwarding behavior.

Server and network virtualization have been driven primarily by the economic benefits of consolidation and higher utilization of physical server and network assets. vApps and ISF change the economics through efficiency gains of providing network-residing services that can be invoked on demand and dimensioned to need rather than to the design constraints of the traditional traffic steering methods.

Virtualization, or rather the act of abstraction from the underlying physical infrastructure, provides the basis of new types of IT services that potentially can be more dynamic in nature, as illustrated in Figure 3-5.

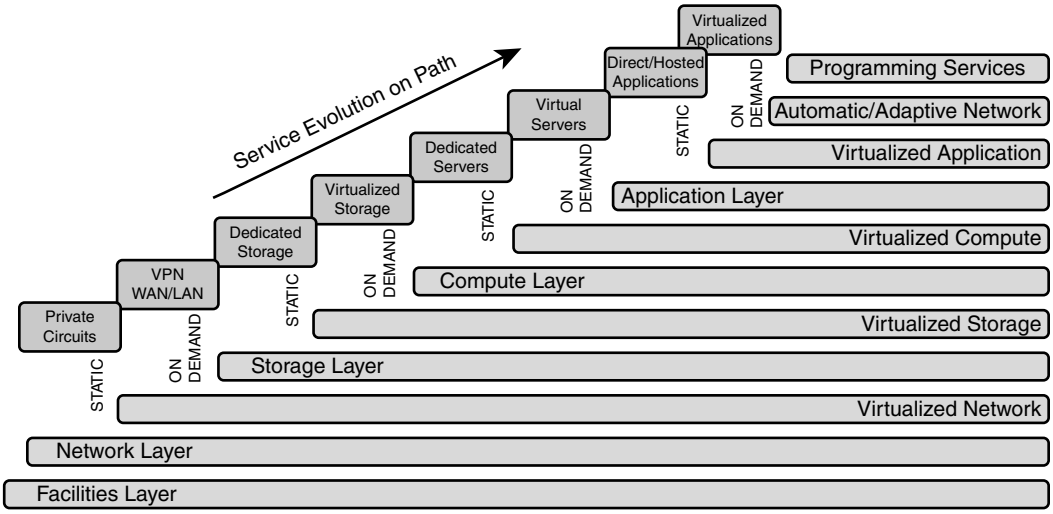

Figure 3-5 *IT Service Enablement Through Abstraction/Virtualization of IT Domains*

Phase 3: Service Automation

Service automation, working hand in hand with a virtualized infrastructure, is a key enabler in delivering dynamic services. From an IaaS perspective, this phase means the policy-driven provisioning of IT services though the use of automated task workflow, whether that involves business tasks (also known as Business Process Operations Management [BPOM]) or IT tasks (also known as IT Orchestration).

Traditionally, this has been too costly to be economically effective because of the reliance on script-based automation tooling. Scripting is linear in nature (makes rollback challenging); more importantly, it tightly couples workflow to process execution logic to

assets. In other words, if an architect wants or needs to change an IT asset (for example, a server type/supplier) or change the workflow or process execution logic within a workflow step/node in response to a business need, a lot of new scripting is required. It's like building a LEGO brick wall with all the bricks glued together. More often than not, a new wall is cheaper and easier to develop than trying to replace or change individual blocks.

Two main developments have now made service automation a more economically viable option:

- Standards-based web APIs and protocols (for example, SOAP and RESTful) have helped reduce integration complexity and costs through the ability to reuse.
- Programmatic-based workflow tools helped to decouple/abstract workflow from process execution logic from assets. Contemporary IT orchestration tools, such as Enterprise Orchestrator from Cisco and BMC's Atrium Orchestrator, allow system designers to make changes to the workflow (including invoking and managing parallel tasks) or to insert new workflow steps or change assets through reusable *adaptors* without having to start from scratch. Using the LEGO wall analogy, individual bricks of the wall can be relatively easily interchanged without having to build a new wall.

Note that a third component is necessary to make programmatic service automation a success, namely, an *intelligent infrastructure* by which the complexity of the low-level device configuration syntax is abstracted from the northbound system's management tools. This means higher-level management tools only need to know the *policy semantics*. In other words, an orchestration system need only ask for a chocolate cake and the element manager, now based on a well-defined (programmatic) object-based data model, will translate that request into the required ingredients and, furthermore, how they those ingredients should be mixed together and in what quantities.

A practical example is the Cisco Unified Compute System (UCS) with its single data model exposed through a single transactional-based rich XML API (other APIs are supported!). This allows policy-driven consumption of the physical compute layer. To do this, UCS provides a layer of abstraction between its XML data model and the underlying hardware through *application gateways* that do the translation of the policy semantics as necessary to execute state change of a hardware component (such as BIOS settings).

Phase 4: Utility Computing Model

This phase involves the ability to monitor, meter, and track resource usage for chargeback billing. The goal is for self-service provisioning (on-demand allocation of compute resources), in essence turning IT into a utility service.

In any IT environment, it is crucial to maintain knowledge of allocation and utilization of resources. Metering and performance analysis of these resources enable cost efficiency, service consistency, and subsequently the capabilities IT needs for trending, capacity management, threshold management (service-level agreements [SLA]), and pay-for-use chargeback.

In many IT environments today, dedicated physical servers and their associated applications, as well as maintenance and licensing costs, can be mapped to the department using them, making the billing relatively straightforward for such resources. In a shared virtual environment, however, the task of calculating the IT operational cost for each consumer in real time is a challenging problem to solve.

Pay for use, where the end customers are charged based on their usage and consumption of a service, has long been used by such businesses as utilities and wireless phone providers. Increasingly, *pay-per-use* has gained acceptance in enterprise computing as IT works in parallel to lower costs across infrastructures, applications, and services.

One of the top concerns of IT leadership teams implementing a utility platform is this: If the promise of pay-per-use is driving service adoption in a cloud, how do the providers of the service track the service usage and bill for it accordingly?

IT providers have typically struggled with billing solution metrics that do not adequately represent all the resources consumed as part of a given service. The primary goal of any chargeback solution requires consistent visibility into the infrastructure to meter resource usage per customer and the cost to serve for a given service. Today, this often requires cobbling together multiple solutions or even developing custom solutions for metering.

This creates not only up-front costs, but longer-term inefficiencies. IT providers quickly become overwhelmed building new functionality into the metering system every time they add a service or infrastructure component.

The dynamic nature of a virtual converged infrastructure and its associated layers of abstraction being a benefit to the IT operation conversely increase the metering complexity. An optimal chargeback solution provides businesses with the true allocation breakdown of costs and services delivered in a converged infrastructure.

The business goals for metering and chargeback typically include the following:

- Reporting on allocation and utilization of resources by business unit or customer
- Developing an accurate cost-to-serve model, where utilization can be applied to each user
- Providing a method for managing IT demand, facilitating capacity planning, forecasting, and budgeting
- Reporting on relevant SLA performance

Chargeback and billing requires three main steps:

Step 1. Data collection

Step 2. Chargeback mediation (correlating and aggregating data collected from the various system components into a billing record of the service owner customer)

Step 3. Billing and reporting (applying the pricing model to collected data) and generating a periodic billing report

Phase 5: Market

In mainstream economics, the concept of a market is any structure that allows buyers and sellers to exchange any type of goods, services, and information. The exchange of goods or services for money (an agreed-upon medium of exchange) is a transaction.

For a marketplace to be built to exchange IT services as an exchangeable commodity, the participants in that market need to agree on common service definitions or have an ontology that aligns not only technology but also business definitions. The alignment of process and governance among the market participants is desirable, particularly when "mashing up" service components from different providers/authors to deliver an end-to-end service.

To be more detailed, a *service* has two aspects:

- **Business:** The business aspect is required for marketplace and a technical aspect for exchange and delivery. The business part needs product definition, relationships (ontology), collateral, pricing, and so on.
- **Technical:** The technical aspect needs fulfillment, assurance, and governance aspects.

In the marketplace, there will be various players/participants who take on a variety and/or combination of roles. There would be exchange providers (also known as *service aggregators* or *cloud service brokers*), service developers, product manufacturers, service providers, service resellers, service integrators, and finally consumers (or even prosumers).

Design Evolution in the Data Center

This section provides an overview of the emerging technologies in the data center, how they are supporting architectural principles outlined previously, how they are influencing design and implementation of infrastructure, and ultimately their value in regard to delivering IT as a service.

First, we will look at Layer 2 physical and logical topology evolution. Figure 3-6 shows the design evolution of an OSI Layer 2 topology in the data center. Moving from left to right, you can see the physical topology changing in the number of active interfaces between the functional layers of the data center. This evolution is necessary to support the current and future service use cases.

Virtualization technologies such as VMware ESX Server and clustering solutions such as Microsoft Cluster Service currently require Layer 2 Ethernet connectivity to function properly. With the increased use of these types of technologies in data centers and now even across data center locations, organizations are shifting from a highly scalable Layer 3 network model to a highly scalable Layer 2 model. This shift is causing changes in the technologies used to manage large Layer 2 network environments, including migration away from Spanning Tree Protocol (STP) as a primary loop management technology toward new technologies, such as vPC and IETF TRILL (Transparent Interconnection of Lots of Links).

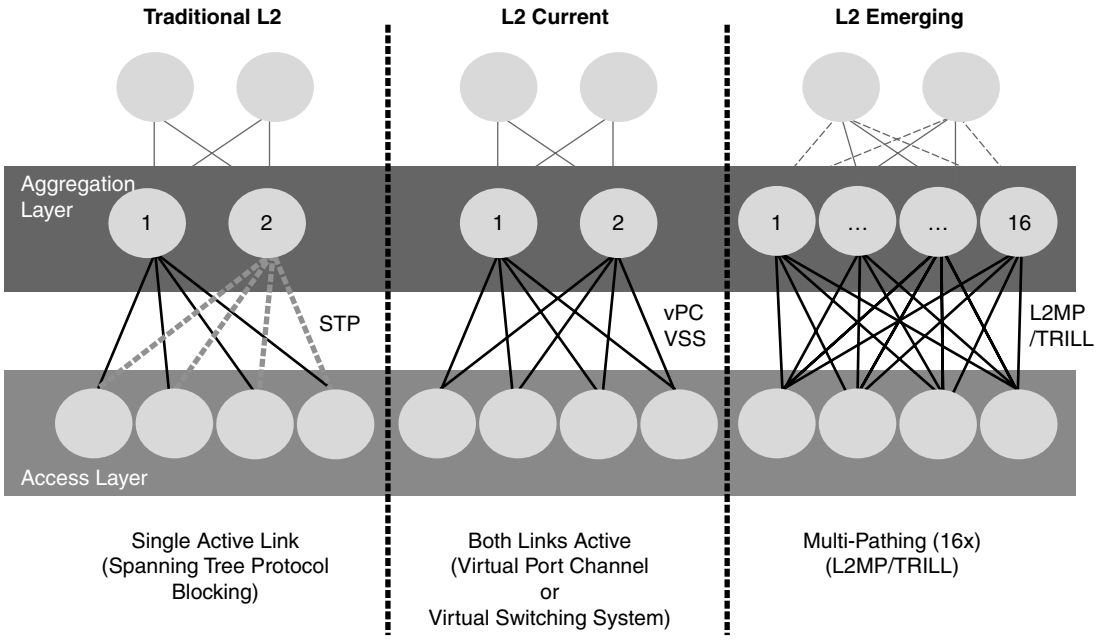

Figure 3-6 *Evolution of OSI Layer 2 in the Data Center*

In early Layer 2 Ethernet network environments, it was necessary to develop protocol and control mechanisms that limited the disastrous effects of a topology loop in the network. STP was the primary solution to this problem, providing a loop detection and loop management capability for Layer 2 Ethernet networks. This protocol has gone through a number of enhancements and extensions, and although it scales to very large network environments, it still has one suboptimal principle: To break loops in a network, only one active path is allowed from one device to another, regardless of how many actual connections might exist in the network. Although STP is a robust and scalable solution to redundancy in a Layer 2 network, the single logical link creates two problems:

- Half (or more) of the available system bandwidth is off limits to data traffic.
- A failure of the active link tends to cause multiple seconds of system-wide data loss while the network reevaluates the new "best" solution for network forwarding in the Layer 2 network.

Although enhancements to STP reduce the overhead of the rediscovery process and allow a Layer 2 network to reconverge much faster, the delay can still be too great for some networks. In addition, no efficient dynamic mechanism exists for using all the available bandwidth in a robust network with STP loop management.

An early enhancement to Layer 2 Ethernet networks was PortChannel technology (now standardized as IEEE 802.3ad PortChannel technology), in which multiple links between

two participating devices can use all the links between the devices to forward traffic by using a load-balancing algorithm that equally balances traffic across the available Inter-Switch Links (ISL) while also managing the loop problem by bundling the links as one logical link. This logical construct keeps the remote device from forwarding broadcast and unicast frames back to the logical link, thereby breaking the loop that actually exists in the network. PortChannel technology has one other primary benefit: It can potentially deal with a link loss in the bundle in less than a second, with little loss of traffic and no effect on the active STP topology.

Introducing Virtual PortChannel (vPC)

The biggest limitation in classic PortChannel communication is that the PortChannel operates only between two devices. In large networks, the support of multiple devices together is often a design requirement to provide some form of hardware failure alternate path. This alternate path is often connected in a way that would cause a loop, limiting the benefits gained with PortChannel technology to a single path. To address this limitation, the Cisco NX-OS Software platform provides a technology called virtual PortChannel (vPC). Although a pair of switches acting as a vPC peer endpoint looks like a single logical entity to PortChannel-attached devices, the two devices that act as the logical PortChannel endpoint are still two separate devices. This environment combines the benefits of hardware redundancy with the benefits of PortChannel loop management. The other main benefit of migration to an all-PortChannel-based loop management mechanism is that link recovery is potentially much faster. STP can recover from a link failure in approximately 6 seconds, while an all-PortChannel-based solution has the potential for failure recovery in less than a second.

Although vPC is not the only technology that provides this solution, other solutions tend to have a number of deficiencies that limit their practical implementation, especially when deployed at the core or distribution layer of a dense high-speed network. All multichassis PortChannel technologies still need a direct link between the two devices acting as the PortChannel endpoints. This link is often much smaller than the aggregate bandwidth of the vPCs connected to the endpoint pair. Cisco technologies such as vPC are specifically designed to limit the use of this ISL specifically to switch management traffic and the occasional traffic flow from a failed network port. Technologies from other vendors are not designed with this goal in mind, and in fact, are dramatically limited in scale especially because they require the use of the ISL for control traffic and approximately half the data throughput of the peer devices. For a small environment, this approach might be adequate, but it will not suffice for an environment in which many terabits of data traffic might be present.

Introducing Layer 2 Multi-Pathing (L2MP)

IETF Transparent Interconnection of Lots of Links (TRILL) is a new Layer 2 topology-based capability. With the Nexus 7000 switch, Cisco already supports a prestandards version of TRILL called *FabricPath*, enabling customers to benefit from this technology

before the ratification of the IETF TRILL standard. (For the Nexus 7000 switch, the migration from Cisco FabricPath to IETF TRILL protocol, a simple software upgrade migration path is planned. In other words, no hardware upgrades are required.) Generically, we will refer to TRILL and FabricPath as "Layer 2 Multi-Pathing (L2MP)."

The operational benefits of L2MP are as follows:

- Enables Layer 2 multipathing in the Layer 2 DC network (up to 16 links). This provides much greater *cross-sectional bandwidth* for both client-to-server (North-to-South) and server-to-server (West-to-East) traffic.
- Provides built-in loop prevention and mitigation with no need to use the STP. This significantly reduces the operational risk associated with the day-to-day management and troubleshooting of a nontopology-based protocol, like STP.
- Provides a single control plane for unknown unicast, unicast, broadcast, and multicast traffic.
- Enhances mobility and virtualization in the FabricPath network with a larger OSI Layer 2 domain. It also helps with simplifying service automation workflow by simply having less service dependencies to configure and manage.

What follows is an amusing poem by Ray Perlner that can be found in the IETF TRILL draft that captures the benefits of building a topology free of STP:

I hope that we shall one day see,

A graph more lovely than a tree.

A graph to boost efficiency,

While still configuration-free.

A network where RBridges can,

Route packets to their target LAN.

The paths they find, to our elation,

Are least cost paths to destination!

With packet hop counts we now see,

The network need not be loop-free!

RBridges work transparently,

Without a common spanning tree.

(Source: Algorhyme V2, by Ray Perlner from IETF draft-perlman-trill-rbridge-protocol)

Network Services and Fabric Evolution in the Data Center

This section looks at the evolution of data center networking from an Ethernet protocol (OSI Layer 2) virtualization perspective. The section then looks at how network services (for example, firewalls, load balancers, and so on) are evolving within the data center.

Figure 3-7 illustrates the two evolution trends happening in the data center.

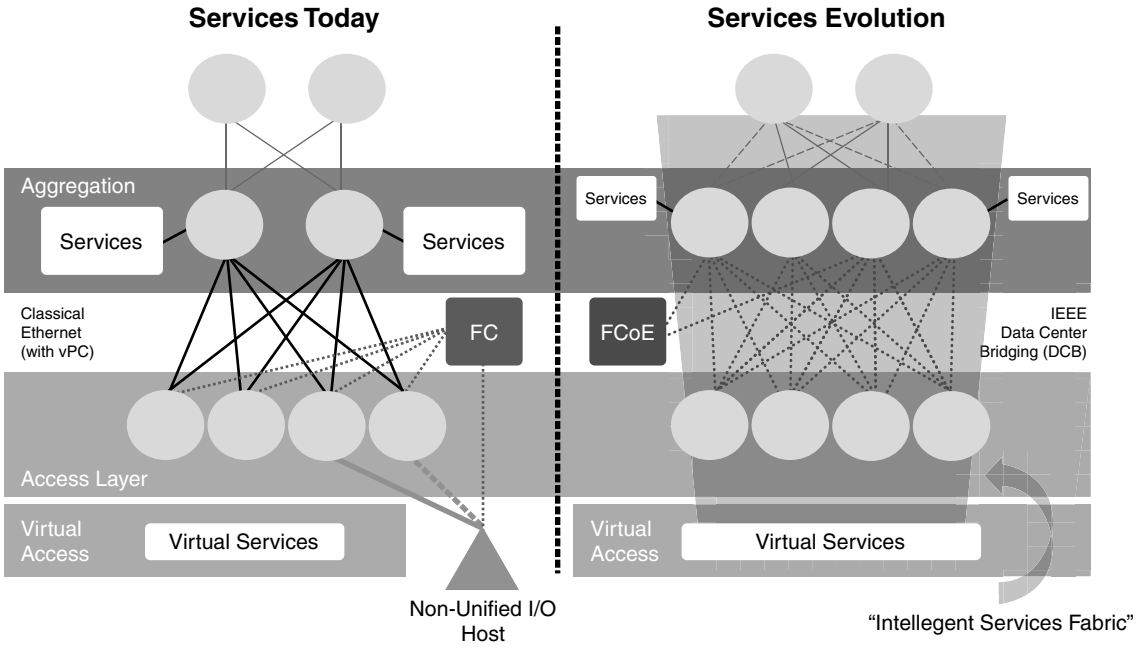

Figure 3-7 *Evolution of I/O Fabric and Service Deployment in the DC*

1. Virtualization of Data Center Network I/O

From a supply-side perspective, the transition to a converged I/O infrastructure fabric is a result of the evolution of network technology to the point where a single fabric has sufficient throughput, low-enough latency, sufficient reliability, and lower-enough cost to be the economically viable solution for the data center network today.

From the demand side, multicore CPUs spawning the development of virtualized compute infrastructures have placed increased demand of I/O bandwidth at the access layer of the data center. In addition to bandwidth, virtual machine mobility also requires the flexibility of service dependencies such as storage. Unified I/O infrastructure fabric enables the abstraction of the overlay service (for example, file [IP] or block-based [FC] storage) that supports the architectural principle of flexibility: "Wire once, any protocol, any time."

Abstraction between the virtual network infrastructure and the physical networking causes its own challenge in regard to maintaining end-to-end control of service traffic from a policy enforcement perspective. Virtual Network Link (VN-Link) is a set of standards-based solutions from Cisco that enables policy-based network abstraction to recouple the virtual and physical network policy domains.

Cisco and other major industry vendors have made standardization proposals in the IEEE to address networking challenges in virtualized environments. The resulting standards tracks are IEEE 802.1Qbg Edge Virtual Bridging and IEEE 802.1Qbh Bridge Port Extension.

The Data Center Bridging (DCB) architecture is based on a collection of open-standard Ethernet extensions developed through the IEEE 802.1 working group to improve and expand Ethernet networking and management capabilities in the data center. It helps ensure delivery over lossless fabrics and I/O convergence onto a unified fabric. Each element of this architecture enhances the DCB implementation and creates a robust Ethernet infrastructure to meet data center requirements now and in the future. Table 3-2 lists the main features and benefits of the DCB architecture.

Table 3-2 *Features and Benefits of Data Center Bridging*

Feature	Benefit
Priority-based Flow Control (PFC) (IEEE 802.1 Qbb)	Provides the capability to manage a bursty, single-traffic source on a multiprotocol link
Enhanced Transmission Selection (ETS) (IEEE 802.1 Qaz)	Enables bandwidth management between traffic types for multiprotocol links
Congestion Notification (IEEE 802.1 Qau)	Addresses the problem of sustained congestion by moving corrective action to the network edge
Data Center Bridging Exchange (DCBX) Protocol	Allows autoexchange of Ethernet parameters between switches and endpoints

IEEE DCB builds on classical Ethernet's strengths, adds several crucial extensions to provide the next-generation infrastructure for data center networks, and delivers unified fabric. We will now describe how each of the main features of the DCB architecture contributes to a robust Ethernet network capable of meeting today's growing application requirements and responding to future data center network needs.

Priority-based Flow Control (PFC) enables link sharing that is critical to I/O consolidation. For link sharing to succeed, large bursts from one traffic type must not affect other traffic types, large queues of traffic from one traffic type must not starve other traffic types' resources, and optimization for one traffic type must not create high latency for small messages of other traffic types. The Ethernet pause mechanism can be used to control the effects of one traffic type on another. PFC is an enhancement to the pause

mechanism. PFC enables pause based on user priorities or classes of service. A physical link divided into eight virtual links, PFC provides the capability to use pause frame on a single virtual link without affecting traffic on the other virtual links (the classical Ethernet pause option stops all traffic on a link). Enabling pause based on user priority allows administrators to create lossless links for traffic requiring no-drop service, such as Fibre Channel over Ethernet (FCoE), while retaining packet-drop congestion management for IP traffic.

Traffic within the same PFC class can be grouped together and yet treated differently within each group. ETS provides prioritized processing based on bandwidth allocation, low latency, or best effort, resulting in per-group traffic class allocation. Extending the virtual link concept, the network interface controller (NIC) provides virtual interface queues, one for each traffic class. Each virtual interface queue is accountable for managing its allotted bandwidth for its traffic group, but has flexibility within the group to dynamically manage the traffic. For example, virtual link 3 (of 8) for the IP class of traffic might have a high-priority designation and a best effort within that same class, with the virtual link 3 class sharing a defined percentage of the overall link with other traffic classes. ETS allows differentiation among traffic of the same priority class, thus creating priority groups.

In addition to IEEE DCB standards, Cisco Nexus data center switches include enhancements such as FCoE multihop capabilities and lossless fabric to enable construction of a Unified Fabric.

At this point to avoid any confusion, note that the term *Converged Enhanced Ethernet (CEE)* was defined by "CEE Authors," an ad hoc group that consisted of over 50 developers from a broad range of networking companies that made prestandard proposals to the IEEE prior to the IEEE 802.1 Working Group completing DCB standards.

FCoE is the next evolution of the Fibre Channel networking and Small Computer System Interface (SCSI) block storage connectivity model. FCoE maps Fibre Channel onto Layer 2 Ethernet, allowing the combination of LAN and SAN traffic onto a link and enabling SAN users to take advantage of the economy of scale, robust vendor community, and road map of Ethernet. The combination of LAN and SAN traffic on a link is called unified fabric. Unified fabric eliminates adapters, cables, and devices, resulting in savings that can extend the life of the data center. FCoE enhances server virtualization initiatives with the availability of standard server I/O, which supports the LAN and all forms of Ethernet-based storage networking, eliminating specialized networks from the data center. FCoE is an industry standard developed by the same standards body that creates and maintains all Fibre Channel standards. FCoE is specified under INCITS as FC-BB-5.

FCoE is evolutionary in that it is compatible with the installed base of Fibre Channel as well as being the next step in capability. FCoE can be implemented in stages nondisruptively on installed SANs. FCoE simply tunnels a full Fibre Channel frame onto Ethernet. With the strategy of frame encapsulation and deencapsulation, frames are moved, without overhead, between FCoE and Fibre Channel ports to allow connection to installed Fibre Channel.

For a comprehensive and detailed review of DCB, TRILL, FCoE and other emerging protocols, refer to the book *I/O Consolidation in the Data Center*, by Silvano Gai and Claudio DeSanti from Cisco Press.

2. Virtualization of Network Services

Application networking services, such as load balancers and WAN accelerators, have become integral building blocks in modern data center designs. These Layer 4–7 services provide service scalability, improve application performance, enhance end-user productivity, help reduce infrastructure costs through optimal resource utilization, and monitor quality of service. They also provide security services (that is, policy enforcement points [PEP] such as firewalls and intrusion protection systems [IPS]) to isolate applications and resources in consolidated data centers and cloud environments that along with other control mechanisms and hardened processes, ensure compliance and reduce risk.

Deploying Layer 4 through 7 services in virtual data centers has, however, been extremely challenging. Traditional service deployments are completely at odds with highly scalable virtual data center designs, with mobile workloads, dynamic networks, and strict SLAs. Security, as aforementioned, is just one required service that is frequently cited as the biggest challenge to enterprises adopting cost-saving virtualization and cloud-computing architectures.

As illustrated in Figure 3-8, Cisco Nexus 7000 Series switches can be segmented into virtual devices based on business need. These segmented virtual switches are referred to as *virtual device contexts (VDC)*. Each configured VDC presents itself as a unique device to connected users within the framework of that physical switch. VDCs therefore deliver true segmentation of network traffic, context-level fault isolation, and management through the creation of independent hardware and software partitions. The VDC runs as a separate logical entity within the switch, maintaining its own unique set of running software processes, having its own configuration, and being managed by a separate administrator.

The possible use cases for VDCs include the following:

- Offer a secure network partition for the traffic of multiple departments, enabling departments to administer and maintain their own configurations independently
- Facilitate the collapsing of multiple tiers within a data center for total cost reduction in both capital and operational expenses, with greater asset utilization
- Test new configuration or connectivity options on isolated VDCs on the production network, which can dramatically improve the time to deploy services

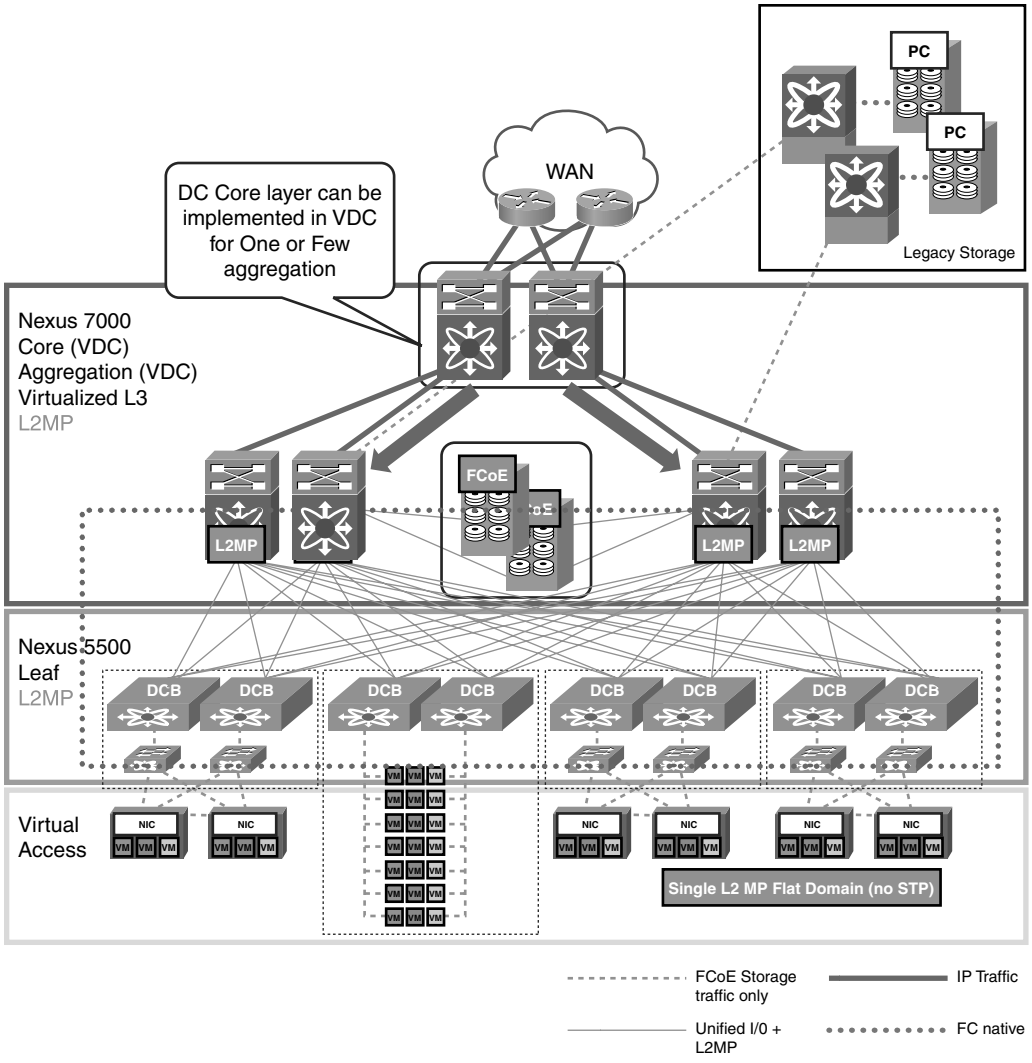

Figure 3-8 *Collapsing of the Vertical Hierarchy with Nexus 7000 Virtual Device Contexts (VDC)*

Multitenancy in the Data Center

Figure 3-9 shows multitenant infrastructure providing end-to-end logical separation between different tenants and shows how a cloud IaaS provider can provide a robust end-to-end multitenant services platform. *Multitenant* in this context is the ability to share a single physical and logical set of infrastructure across many stakeholders and customers. This is nothing revolutionary; the operational model to isolate customers from one another has been well established in wide-area networks (WAN) using technologies such as

Multi-Protocol Label Switching (MPLS). Therefore, multitenancy in the DC is an evolution of a well-established paradigm, albeit with some additional technologies such as VLANs and Virtual Network Tags (VN-Tag) combined with virtualized network services (for example, session load balancers, firewalls, and IPS PEP instances).

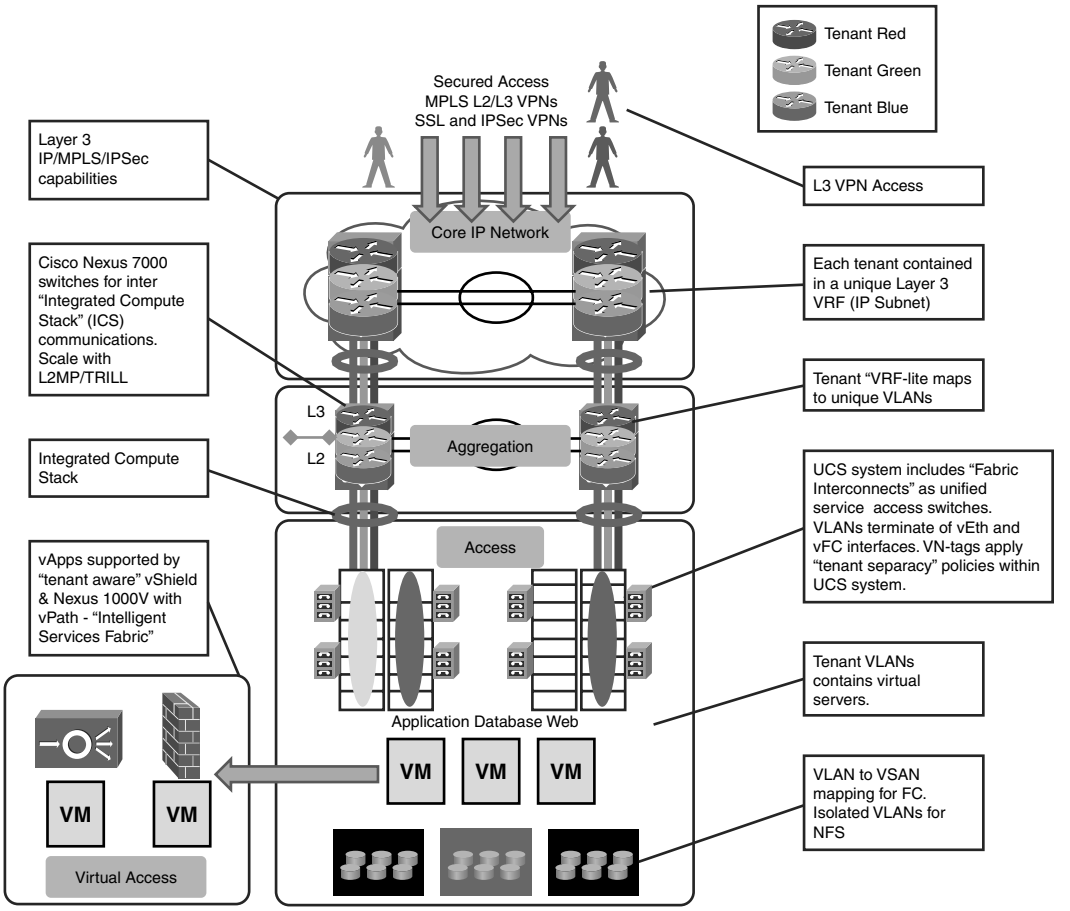

Figure 3-9 *End-to-End "Separacy"—Building the Multitenant Infrastructure*

In addition to multitenancy, architects need to think about how to provide multitier applications and their associated network and service design, including from a security posture perspective a multizone overlay capability. In other words, to build a functional and secure service, one needs to take into account multiple functional demands, as illustrated in Figure 3-10.

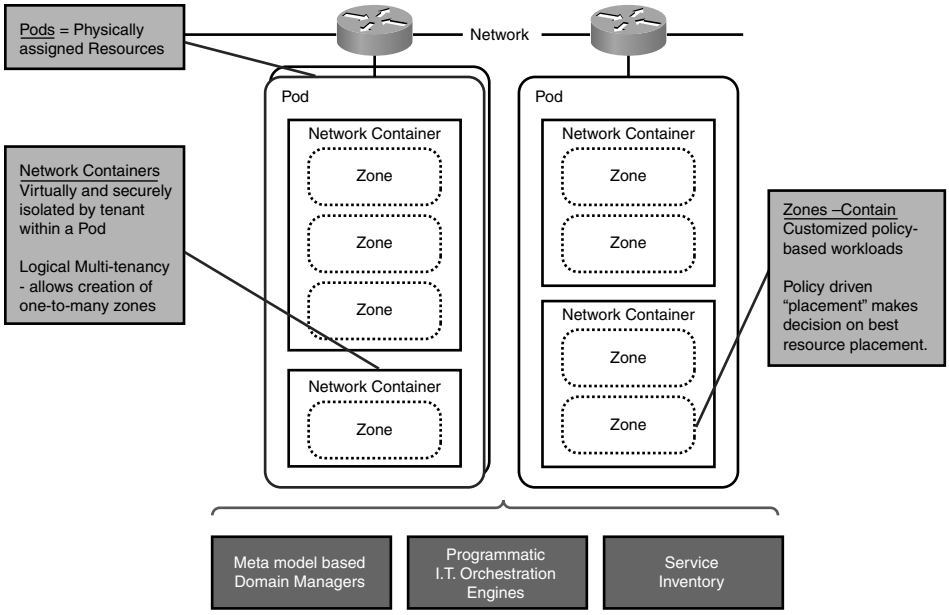

Figure 3-10 *Example of a Hierarchical Architecture Incorporating Multitenancy, Multitier, and Multizoning Attributes for an IaaS Platform (Source: Cisco Systems VMDC 2.0 Solution)*

The challenge is being able to "stitch" together the required service components (each with their own operational-level agreement (OLAs underpin an SLA) to form a service chain that delivers the end-to-end service attributes (legally formalized by a service-level agreement [SLA]) that the end customer desires. This has to be done within the context of the application tier design and security zoning requirements.

Real-time capacity and capability posture reporting of a given infrastructure are only just beginning to be delivered to the market. Traditional ITIL Configuration Management Systems (CMS) have not been designed to run in real-time environments. The consequence is that to deploy a service chain with known quantitative and qualitative attributes, one must take a structured approach to service deployment/service activation. This structured approach requires a predefined infrastructure modeling of the capacity and capability of service elements and their proximity and adjacency to each other. A predefined service chain, known more colloquially as a network container, can therefore be activated on the infrastructure as a known unit of consumption. A service chain is a group of technical topology building blocks, as illustrated in Figure 3-11.

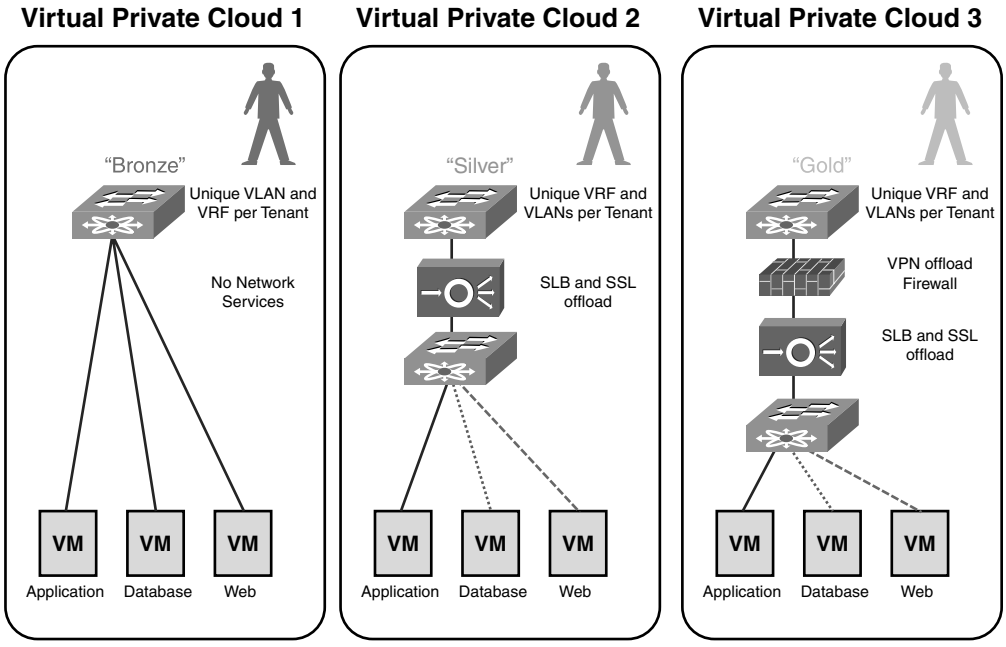

Figure 3-11 *Network Containers for Virtual Private Cloud Deployment*

As real-time IT capacity- and capability-reporting tooling becomes available, ostensibly requiring autodiscovery and reporting capabilities of all infrastructure in a addition to flexible meta models and data (that is, rules on how a component can connect to other components—for example, a firewall instance can connect to a VLAN but not a VRF), providers and customers will be able to take an unstructured approach to service chain deployments. In other words, a customer will be able to create his own blueprint and publish within his own service catalogue to consume or even publish the blueprint into the provider's service portfolio for others to consume, thereby enabling a "prosumer" model (*prosumer* being a portmanteau of producer and consumer).

Service Assurance

As illustrated in Figure 3-12, SLAs have evolved through necessity from those based only on general network performance in Layers 1 through 3 (measuring metrics such as jitter and availability), to SLAs increasingly focused on network performance for specific applications (as managed by technologies such as a WAN optimization controller), to SLAs based on specific application metrics and business process SLAs based on key performance indicators (KPI) such as cycle time or productivity rate. Examples of KPIs are the number of airline passengers who check in per hour or the number of new customer accounts provisioned.

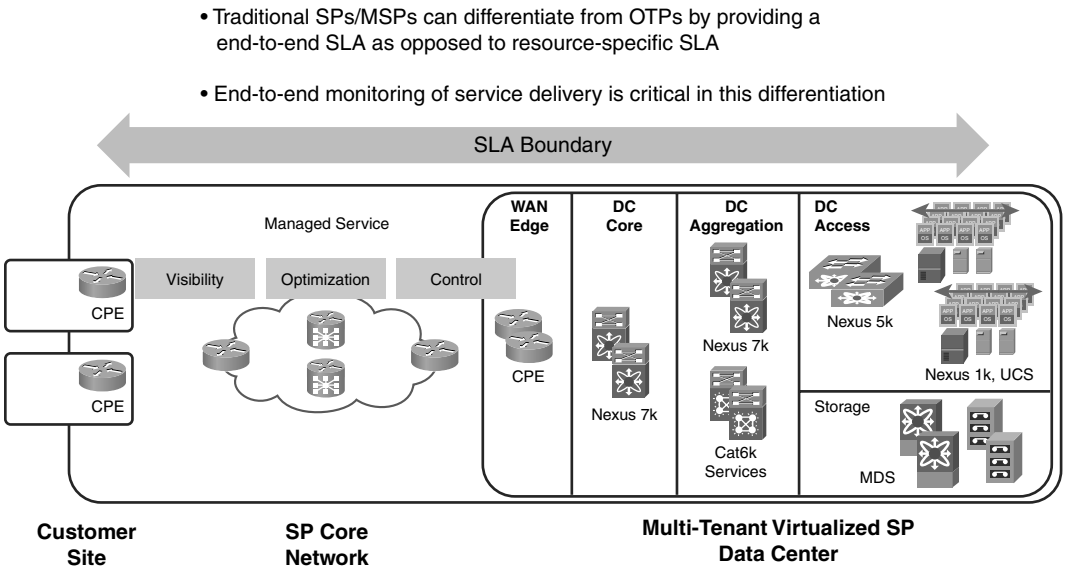

Figure 3-12 *Expanding the SLA Boundary*

Customers expect that their critical business processes (such as payroll and order fulfillment) will always be available and that sufficient resources are provided by the service provider to ensure application performance, even in the event that a server fails or if a data center becomes unavailable. This requires cloud providers to be able to scale up data center resources, ensure the mobility of virtual machines within the data center and across data centers, and provide supplemental computer resources in another data center, if needed.

With their combined data center and Cisco IP NGN assets, service providers can attract relationships with independent software vendors with SaaS offerings, where end customers purchase services from the SaaS provider while the service provider delivers an assured end-to-end application experience.

In addition to SLAs for performance over the WAN and SLAs for application availability, customers expect that their hosted applications will have security protection in an external hosting environment. In many cases, they want the cloud service provider to improve the performance of applications in the data center and over the WAN, minimizing application response times and mitigating the effects of latency and congestion.

With their private IP/MPLS networks, cloud service providers can enhance application performance and availability in the cloud and deliver the visibility, monitoring, and reporting that customers require for assurance. As cloud service providers engineer their solutions, they should consider how they can continue to improve on their service offerings to support not only network and application SLAs but also SLAs for application transactions and business processes.

Service assurance solutions today need to cope with rapidly changing infrastructure configurations as well as understand the status of a service with the backdrop of ever-changing customer ownership of a service. The solution also needs to understand the context of a service that can span traditionally separate IT domains, such as the IP WAN and the Data Center Network (DCN).

Ideally, such a solution should ideally be based on a single platform and code base design that eliminates some of the complexities of understanding a service in a dynamic environment. This makes it easier to understand and support the cloud services platform and also eliminates costly and time-consuming product integration work. However, the single-platform design should not detract from scalability and performance that would be required in a large virtual public cloud environment and obviously with an HA deployment model supported.

Northbound and southbound integration to third-party tools, with well-defined and documented message format and workflow that allow direct message interaction and web integration APIs, is an absolute basic requirement to build a functional system.

An IaaS assurance deployment requires a real-time and extensible data model that can support the following:

- Normalized object representation of multiple types of devices and domain managers, their components, and configuration
- Flexible enough to represent networking equipment, operating systems, data center environmental equipment, standalone and chassis servers, and domain managers such as vSphere, vCloud Director, and Cisco UCS
- Able to manage multiple overlapping relationships among and between managed resources
- Peer relationships, such as common membership in groups
- Parent-child relationships, such as the relationship between a UCS chassis and blade
- Fixed dependency relationships, such as the relationship between a process and an operating system
- Mobile dependency relationships, such as the relationship between a VM and its current host system
- Cross-silo discovered relationships, such as the relationship between a virtual host and a logical unit number (LUN) that represents network attached logical storage volume
- Linkages between managed objects and management data streams, such as event database and performance metrics
- Security boundaries between sets of managed objects and subsets of users to enable use in multitenant environments
- Developer-extensible to allow common capabilities to be developed for all customers
- Field-extensible to enable services teams and customers to meet uncommon or unique requirements

The ability to define logical relationships among service elements to represent the technical definition of a service is a critical step in providing a service-oriented impact analysis.

Service elements include

- **Physical:** Systems, infrastructure, and network devices
- **Logical:** Aspects of a service that must be measured or evaluated
- **Virtual:** Software components, for example, processes
- **Reference:** Elements represented by other domain managers

In addition, to understand the service components, the service element relationships are both fixed and dynamic and need to be tracked. Fixed relationships identify definitions, such as the fact that this web application belongs to this service. Dynamic relationships are managed by the model, such as identifying as an example which Cisco UCS chassis is hosting an ESX server where a virtual machine supporting this service is currently running.

Service policies evaluate the state of and relationships among elements and provide impact roll-up so that the services affected by a low-level device failure are known. They assist in root cause identification so that from the service a multilevel deep failure in the infrastructure can be seen to provide up, down, and degraded service states. (For example, if a single web server in a load-balanced group is down, the service might be degraded.) Finally, service policies provide event storm filtering, roll-up, and windowing functions.

All this information, service elements, relationships, and service policies provide service visualization that allows operations to quickly determine the current state of a service, service elements, and current dynamic network and infrastructure resources, and in addition allow service definition and tuning. A good example of a service assurance tool that supports these attributes and capabilities can be found at www.zenoss.com.

Evolution of the Services Platform

Organizations tend to adopt a phased strategy when building a utility service platform. Figure 3-13 shows a four-step approach that actually is a simplification of the actual journey to be undertaken by the end customer. How such a goal is realized does heavily depend on the current state of the architecture. For example, are we starting from a greenfield or brownfield deployment? What services are to be offered to whom, at what price and when? All these factors need decided up front during the service creation phase.

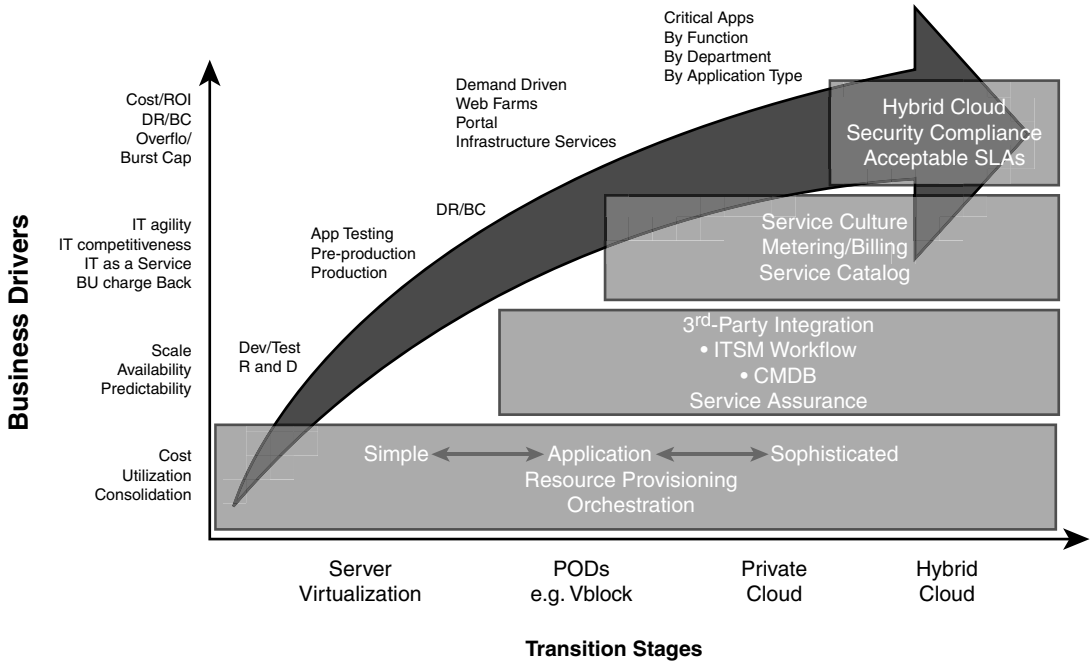

Figure 3-13 *Evolving Customer Needs from Service Platform*

The phasing closely maps to the IT industry evolution we discussed earlier in this chapter:

1. Virtualization of the end-to-end infrastructure
2. Automation of the service life cycle from provisioning to assurance
3. Deployment and integration of the service infrastructure, that is, customer portal, billing, and CRM
4. Deployment of intercloud technologies and protocols to enable migration of workloads and their dependencies

Migration of existing applications onto the new services platform requires extensive research and planning in regard not only to the technical feasibility but also the feasibility in regard to current operational and governance constraints which, with this authors' experience to date, prove to be the most challenging aspects to get right. It is essential that technical and business stakeholders work together to ensure success.

Building a virtualization management strategy at tool set is key to success for the first phase. The benefits gained through virtualization can be lost without an effective virtualization management strategy. Virtual server management requires changes in policies surrounding operations, naming conventions, chargeback, and security. Although many

server and desktop virtualization technologies come with their own sets of management capabilities, businesses should also evaluate third-party tools to plug any gaps in management. These tools should answer questions such as, "How much infrastructure capacity and capability do I have?" or "What are the service dependencies?" in real time.

Virtualization, as discussed earlier, helps to deliver infrastructure multitenant capability. This means the ability to group and manage a set of constrained resources (normally virtual) that can be used exclusively by a customer and is isolated from other customer-assigned resources at both the data and management planes (for example, customer portal and life cycle management tools). A good example of a tool that can achieve this level of abstraction is the Cisco Cloud Portal (CCP) that provides RBAC-based entitlement views and management or, from a service activation approach example, network containers as aforementioned in this chapter.

The second phase is to introduce service automation through (ideally) end-to-end IT orchestration (also known as Run Book Automation) and service assurance tools. This phase is all about speed and quality of IT service delivery at scale, with predictability and availability at lower change management costs. In short, this is providing IT Service Management (ITSM) in a workflow-based structured way using best-practice methodologies.

This second phase is a natural progression of software tool development to manage data center infrastructure, physical and virtual. This development timeline is shown in Figure 3-14. The IT industry is now adopting 'programmatic' software to model underlying infrastructure capability and capacity. Within these software models, technology and business rules can be built within to ensure compliance and standardization of IT infrastructure. We discuss an example of this type programmatic model tooling in Chapter 5 when we discuss the 'Network Hypervisor' product.

The third phase is building and integrating service capabilities. Service-enabling tools include a customer portal and a service catalogue in conjunction with SLA reporting and metering, chargeback, and reporting. (*Service catalogue* is often an overused term that actually consists of multiple capabilities, for example, portfolio management, demand management, request catalogue, and life cycle management). From an operational perspective, integration of IT orchestration software (for example, Cisco Enterprise Orchestrator) along with smart domain/resource management tools completes the end-to-end service enablement of infrastructure. This third phase is about changing the culture of the business to one that is service lead rather than product lead. This requires organizational, process, and governance changes within the business.

Technology to support the fourth phase of this journey is only just starting to appear in the marketplace at the time of this writing. The ability to migrate workloads and service chains over large distances between (cloud) service providers requires an entire range of technological and service-related constraints that are being addressed. Chapter 5, "The Cisco Cloud Strategy," will discuss some of these constraints in detail.

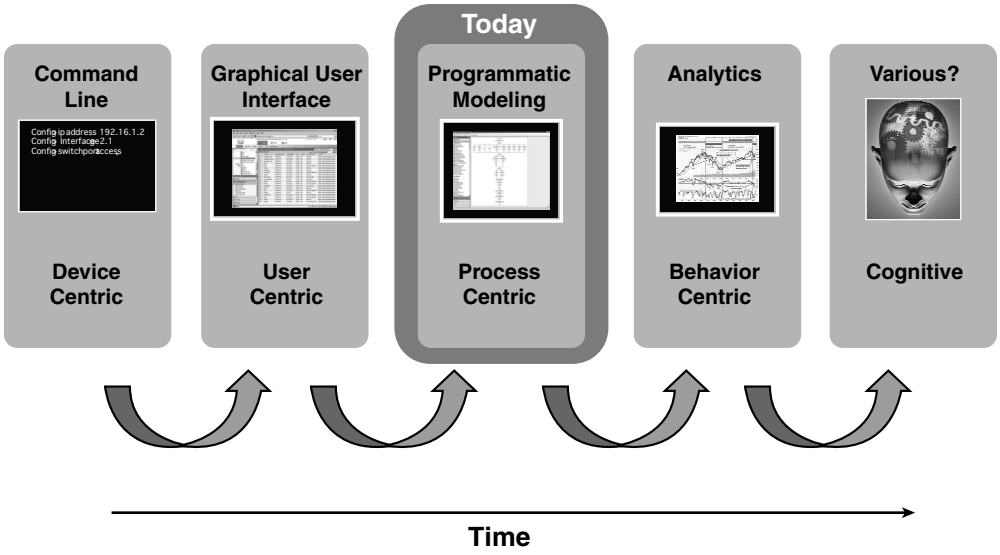

Figure 3-14 *Evolution of Data Center Management Tools*

Summary

Cisco believes that the network platform is a foundational component of a utility service platform as it is critical to providing intelligent connectivity within and beyond the data center. With the right built-in and external tools, the network is ideally placed to provide a secure, trusted, and robust services platform.

The network is the natural home for management and enforcement of policies relating to risk, performance, and cost. Only the network sees all data, connected resources, and user interactions within and between clouds. The network is thus uniquely positioned to monitor and meter usage and performance of distributed services and infrastructure. An analogy for the network in this context would be the human body's autonomic nervous system (ANS) that acts as a system (functioning largely below the level of consciousness) that controls visceral (inner organ) functions. ANS is usually divided into sensory (afferent) and motor (efferent) subsystems that is analogous to visibility and control capabilities we need from a services platform to derive a desired outcome. Indeed, at the time of this writing, there is a lot of academic research into managing complex network systems, might they be biological, social, or traditional IT networking. Management tools for the data center and wider networks have moved from a user-centric focus (for example, GUI design) to today's process-centric programmatic capabilities. In the future, the focus will most likely shift toward behavioral- and then cognitive-based capabilities.

The network also has a pivotal role to play in promoting resilience and reliability. For example, the network, with its unique end-to-end visibility, helps support dynamic orchestration and redirection of workloads through embedded policy-based control capabilities. The network is inherently aware of the physical location of resources and users. Context-aware services can anticipate the needs of users and deploy resources appropriately, balancing end-user experience, risk management, and the cost of service.

Index

Numbers

A

B

C

D

E

F

G

H

I

J-K

L

M

N

O

P

Q

R

S

T

U

V

W

X-Y-Z