A Generic Cloud Architecture Design

- An Internet cloud is envisioned as a public cluster of servers provisioned on demand to perform collective web services or distributed applications using data-center resources. In this section,
- We will discuss cloud design objectives and then present a basic cloud architecture design.

Cloud Platform Design Goals

- Scalability, virtualization, efficiency, and reliability are four major design goals of a cloud computing platform.
- Clouds support Web 2.0 applications. Cloud management receives the user request, finds the correct resources, and then calls the provisioning services which invoke theresources in the cloud.

Enabling Technologies for Clouds

Table 4.3 Cloud-Enabling Technologies in Hardware, Software, and Networking

Technology	Requirements and Benefits
Fast platform deployment	Fast, efficient, and flexible deployment of cloud resources to provide dynamic computing environment to users
Virtual clusters on demand	Virtualized cluster of VMs provisioned to satisfy user demand and virtual cluster reconfigured as workload changes
Multitenant techniques	SaaS for distributing software to a large number of users for their simultaneous use and resource sharing if so desired
Massive data processing	Internet search and web services which often require massive data processing, especially to support personalized services
Web-scale communication	Support for e-commerce, distance education, telemedicine, social networking, digital government, and digital entertainment applications
Distributed storage	Large-scale storage of personal records and public archive information which demands distributed storage over the clouds
Licensing and billing services	License management and billing services which greatly benefit all types of cloud services in utility computing

A Generic Cloud Architecture

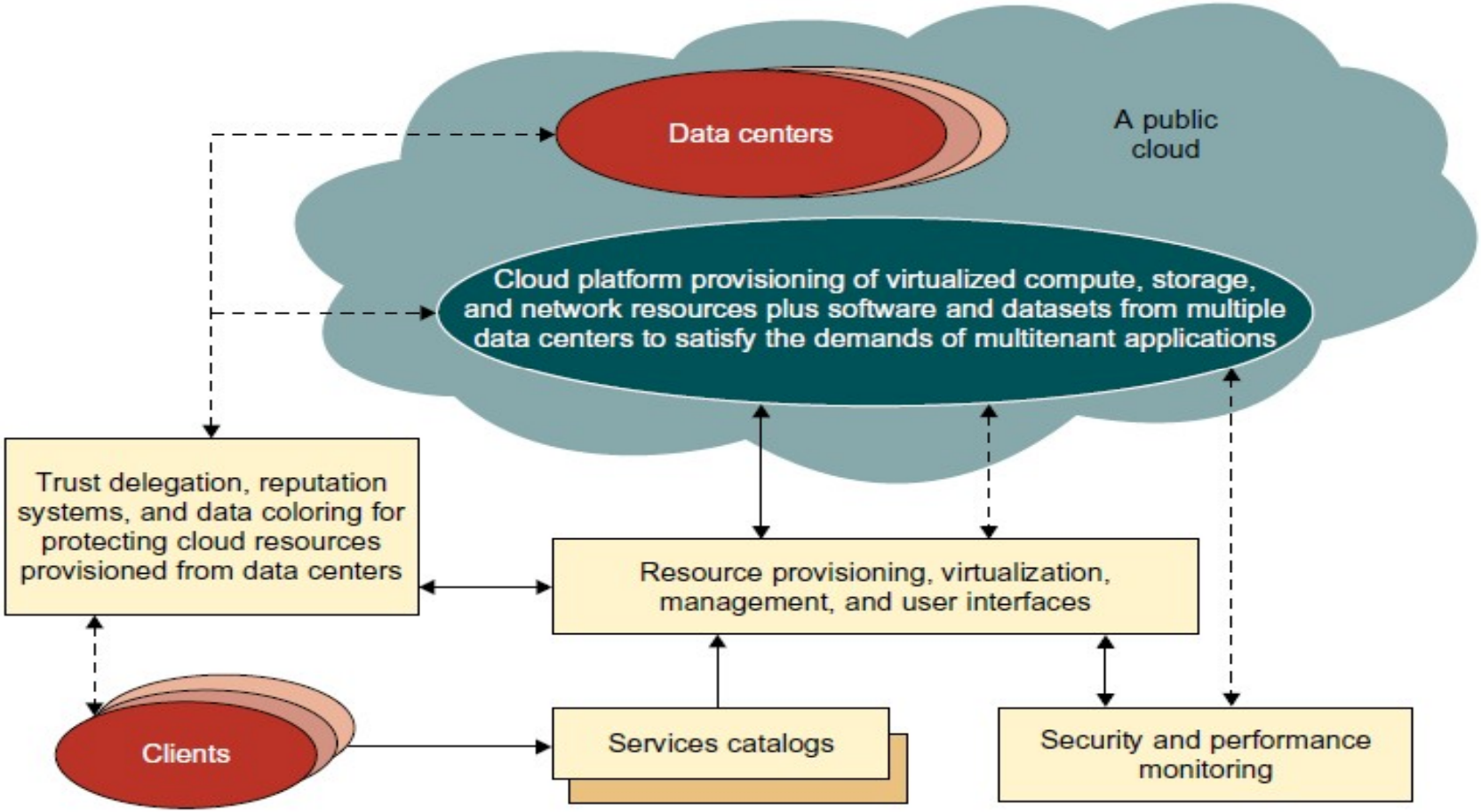

FIGURE 4.14

A security-aware cloud platform built with a virtual cluster of VMs, storage, and networking resources over the data-center servers operated by providers.

(Courtesy of K. Hwang and D. Li, 2010 [36])

Layered Cloud Architectural Development

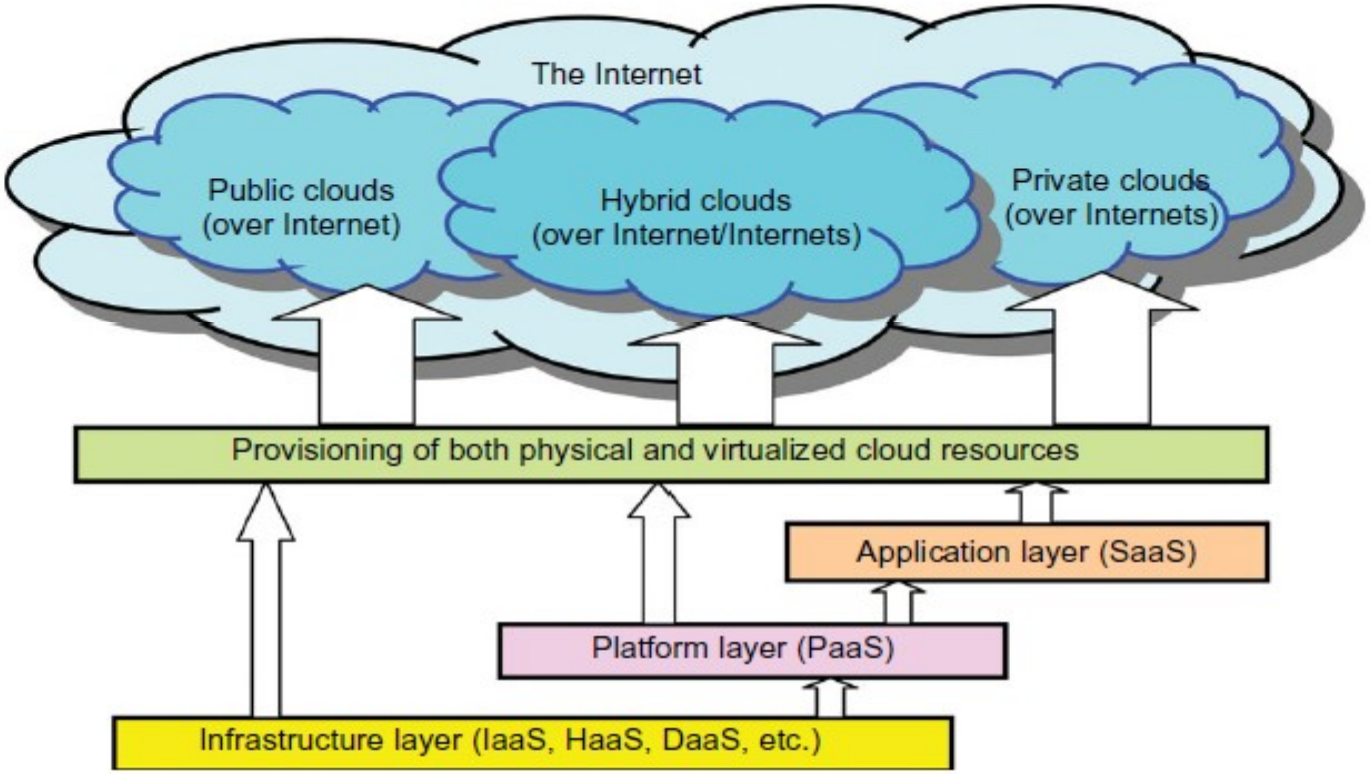

FIGURE 4.15

Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet.

Market-Oriented Cloud Architecture

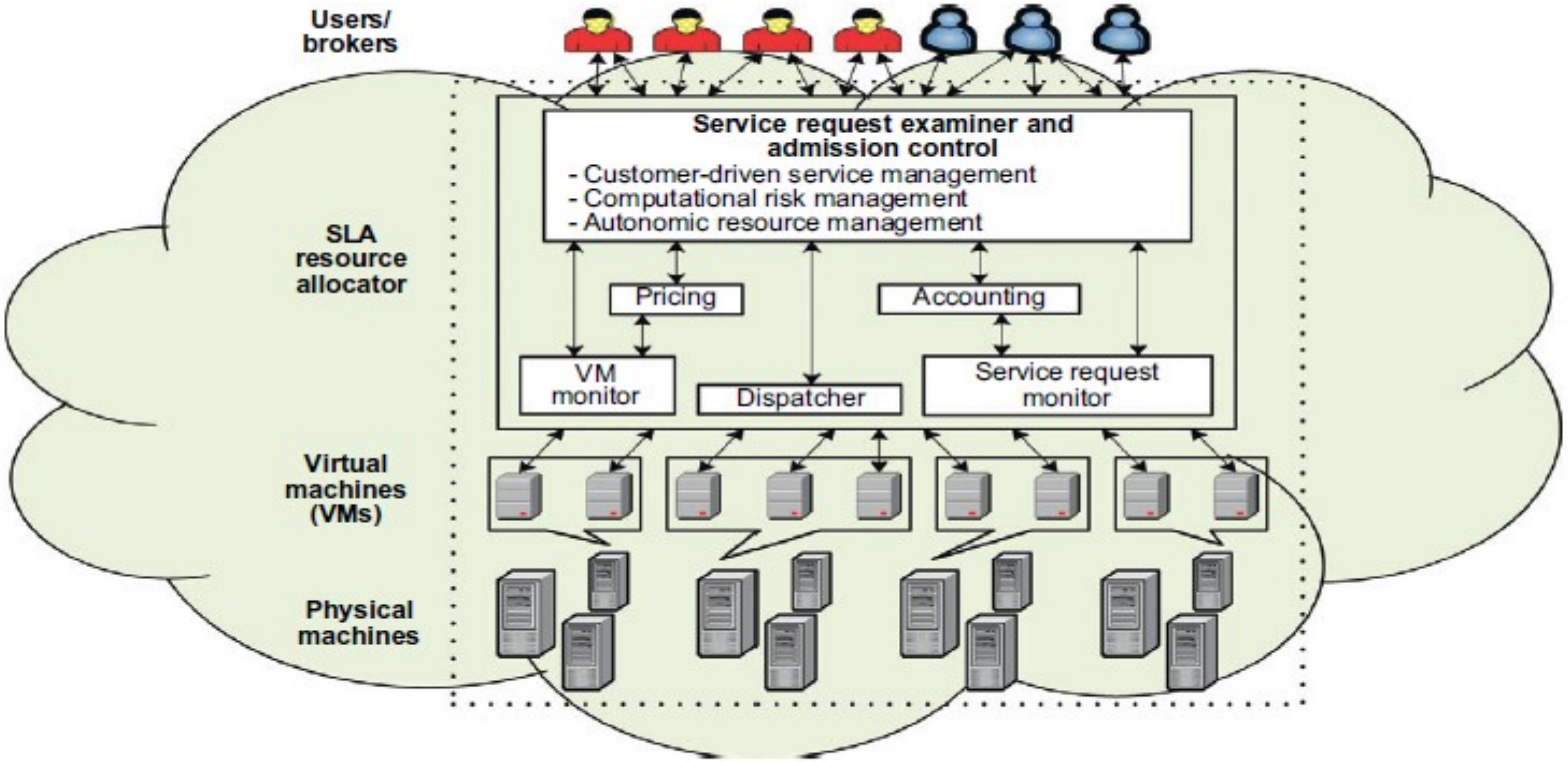

FIGURE 4.16

Market-oriented cloud architecture to expand/shrink leasing of resources with variation in QoS/demand from users.

(*Courtesy of Raj Buyya, et al. [11]*)

- market-oriented resource management is necessary to regulate the supply and demand of cloud resources to achieve market equilibrium between supply and demand.

Quality of Service Factors

- The data center comprises multiple computing servers that provide resources to meet service demands.
- In the case of a cloud as a commercial offering to enable crucial business operations of companies, there are critical QoS parameters to consider in a service request, such as time, cost, reliability,and trust/security.

Virtualization Support and Disaster Recovery

- Hardware Virtualization
- Virtualization Support in Public Clouds
- Storage Virtualization for Green Data Centers
- Virtualization for IaaS
- VM Cloning for Disaster Recovery

Table 4.4 Virtualized Resources in Compute, Storage, and Network Clouds [4]

Provider	AWS	Microsoft Azure	GAE
Compute cloud with virtual cluster of servers	x86 instruction set, Xen VMs, resource elasticity allows scalability through virtual cluster, or a third party such as RightScale must provide the cluster	Common language runtime VMs provisioned by declarative descriptions	Predefined application framework handlers written in Python, automatic scaling up and down, server failover inconsistent with the web applications
Storage cloud with virtual storage	Models for block store (EBS) and augmented key/blob store (SimpleDB), automatic scaling varies from EBS to fully automatic (SimpleDB, S3)	SQL Data Services (restricted view of SQL Server), Azure storage service	MegaStore/BigTable
Network cloud services	Declarative IP-level topology; placement details hidden, security groups restricting communication, availability zones isolate network failure, elastic IP applied	Automatic with user's declarative descriptions or roles of app. components	Fixed topology to accommodate three-tier web app. structure, scaling up and down is automatic and programmer-invisible

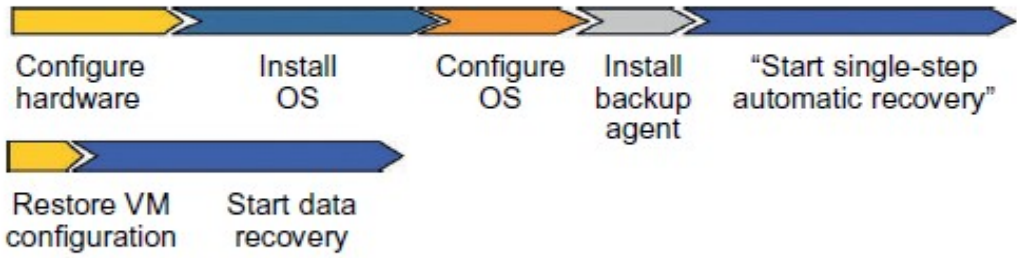

FIGURE 4.18

Recovery overhead of a conventional disaster recovery scheme, compared with that required to recover from live migration of VMs.

Architectural Design Challenges

- **Challenge 1—Service Availability and Data Lock-in Problem**
- **Challenge 2—Data Privacy and Security Concerns**
- **Challenge 3—Unpredictable Performance and Bottlenecks**
- **Challenge4—Distributed Storage and Widespread Software Bugs**
- **Challenge 5—Cloud Scalability, Interoperability, and Standardization**
- **Challenge 6—Software Licensing and Reputation Sharing**