Chapter-4 (part-I)

Cloud platform architecture over virtualized data centers

Cloud Platform Architecture over Virtualized Data Centers

Reference: *Distributed and Cloud Computing From Parallel Processing to the Internet of Things*, Kai Hwang Geoffrey C. Fox, and Jack J. Dongarra, Morgan Kaufmann

Cloud Computing and Service Models

- This section deals with the cloud platform architecture service models, and programming environments.
- Users can access and deploy cloud applications from anywhere in the world at very competitive costs.
- Virtualized cloud platforms are often built on top of large **data centers**.
- Clouds aim to power the next generation of data centers by architecting them as virtual resources over automated hardware, databases, user interfaces, and application environments.

Public, Private, and Hybrid Clouds

- The concept of **cloud computing** has evolved from **cluster**, **grid**, and **utility computing**:
  - Cluster and grid computing leverage the use of many computers in parallel to solve problems of any size.
  - Utility and **Software as a Service** (**SaaS**) provide computing resources as a service with the notion of pay per use.
  - Cloud computing leverages dynamic resources to deliver large numbers of services to end users.
  - Cloud computing is a **high-throughput computing** (**HTC**) paradigm whereby the infrastructure provides the services through a large data center or server farms.
  - The cloud computing model enables users to share access to resources from anywhere at any time through their connected devices.

Public, Private, and Hybrid Clouds

- The cloud will free users to focus on user application development by outsourcing job execution to cloud providers:
  - In this scenario, the computations (programs) are sent to where the data is located, rather than copying the data to millions of desktops.
  - Cloud computing avoids large data movement, resulting in much better network bandwidth utilization.
  - Furthermore, **machine virtualization** has enhanced resource utilization, increased application flexibility, and reduced the total cost of using virtualized data-center resources.

Centralized versus Distributed Computing

- All computations in cloud applications are **distributed** to servers in a **data center**.
  - These are mainly **virtual machines** (**VMs**) in virtual clusters created out of data-center resources.
- **Cloud platforms are systems distributed through virtualization.** As **Figure 4.1** shows, both **public clouds** and **private clouds** are developed in the Internet:
  - Commercial cloud providers such as **Amazon**, **Google**, and **Microsoft** created their platforms to be distributed geographically.
  - This distribution is partially attributed to *fault tolerance*, *response latency reduction*, and even *legal reasons*.

Centralized versus Distributed Computing

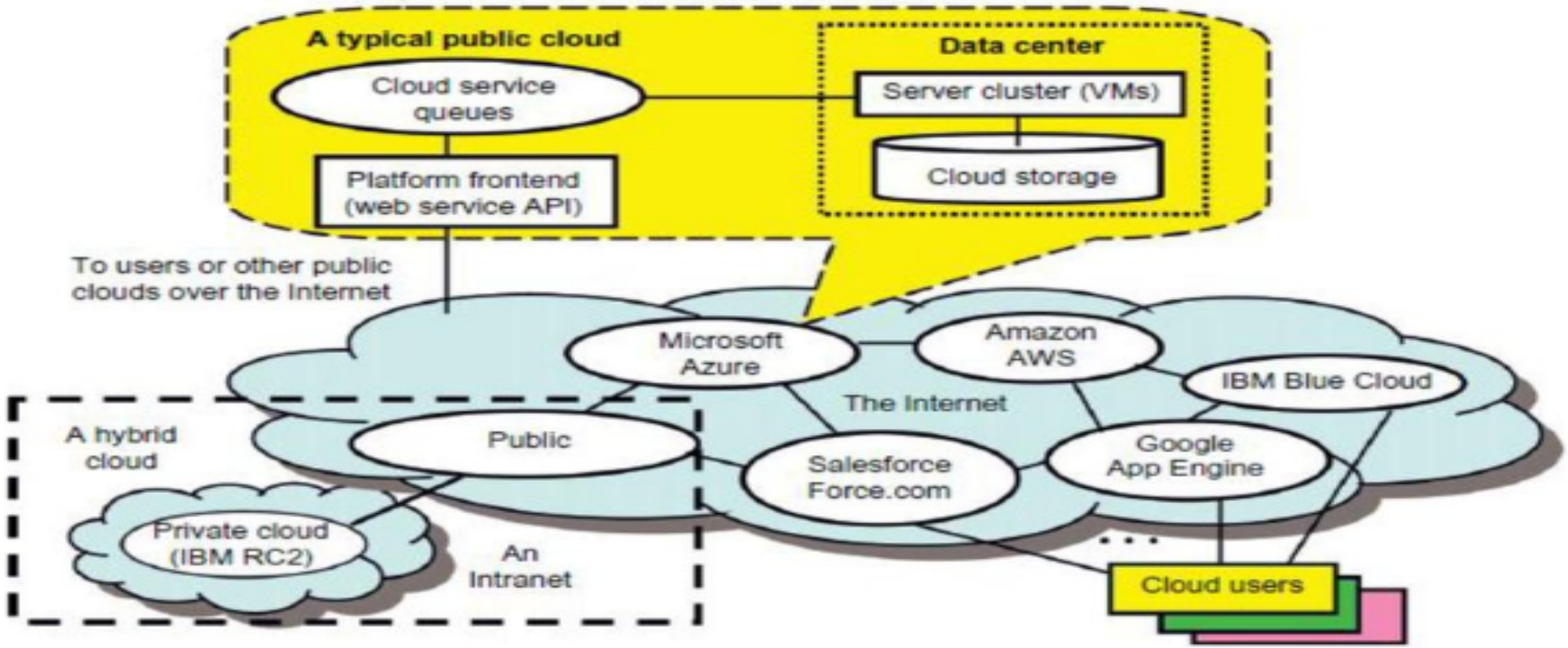


FIGURE 4.1

Public, private, and hybrid clouds illustrated by functional architecture and connectivity of representative clouds available by 2011.

Public Clouds

- A **public cloud** is built over the Internet and can be accessed by any user who has paid for the service.
  - **Public clouds** are owned by service providers and are accessible through a subscription.
    - The callout box in top of **Figure 4.1** shows the architecture of a typical public cloud.
  - Many public clouds are available, including **Google App Engine (GAE)**, **Amazon Web Services** (**AWS**), **Microsoft Azure**, **IBM Blue Cloud**, and Salesforce.com's **Force.com**.
  - The providers of the aforementioned clouds are commercial providers that offer a *publicly accessible remote interface* for creating and managing **VM** instances .

Private Clouds

- A **private cloud** is built within the domain of an **intranet** owned by a single organization.
- Therefore, **it is client owned and managed**, and its access is limited to the owning clients and their partners.
  - Its deployment was not meant to sell capacity over the Internet through publicly accessible interfaces.
- A **private cloud** is supposed to deliver more efficient and convenient cloud services.
- It may impact the cloud standardization, while retaining greater customization and organizational control.

Hybrid Clouds

- A **hybrid cloud** is built with **both public** and **private clouds**, as shown at the lower-left corner of **Figure 4.1**.
- **Private clouds** can also support a **hybrid cloud** model by supplementing *local infrastructure with computing capacity* from an external public cloud.
  - For example, the **Research Compute Cloud** (**RC2**) is a **private cloud**, built by IBM, that interconnects the computing and IT resources at eight IBM Research Centers scattered throughout the United States, Europe, and Asia.
- A **hybrid cloud** provides access to clients, the partner network, and third parties.

Data-Center Networking Structure

- The core of a cloud is the **server cluster** (or **VM cluster**).
- The **gateway** nodes provide the access points of the service from the outside world.
  - These gateway nodes can be also used for security control of the entire cloud platform.
- In physical clusters and traditional grids, users expect static demand of resources:
  - Clouds are designed to handle fluctuating workloads, and thus demand variable resources dynamically.
    - Private clouds will satisfy this demand if properly designed and managed.

Data-Center Networking Structure

- **Data centers** and **supercomputers** also differ in networking requirements, as illustrated in **Figure 4.2**.
- **Supercomputers** use custom-designed high-bandwidth networks such as fat trees or 3D torus networks.
  - **Data center networks** are mostly IP-based commodity networks, such as the 10 Gbps Ethernet network, which is optimized for Internet access.
- **Figure 4.2** shows a **multilayer structure** for accessing the Internet:
  - The server racks are at the bottom **Layer 2**, and they are connected through fast switches (S) as the hardware core.
  - The **data center** is connected to the Internet at Layer 3 with many **access routers** (**ARs**) and **border routers** (**BRs**).

Data-Center Networking Structure

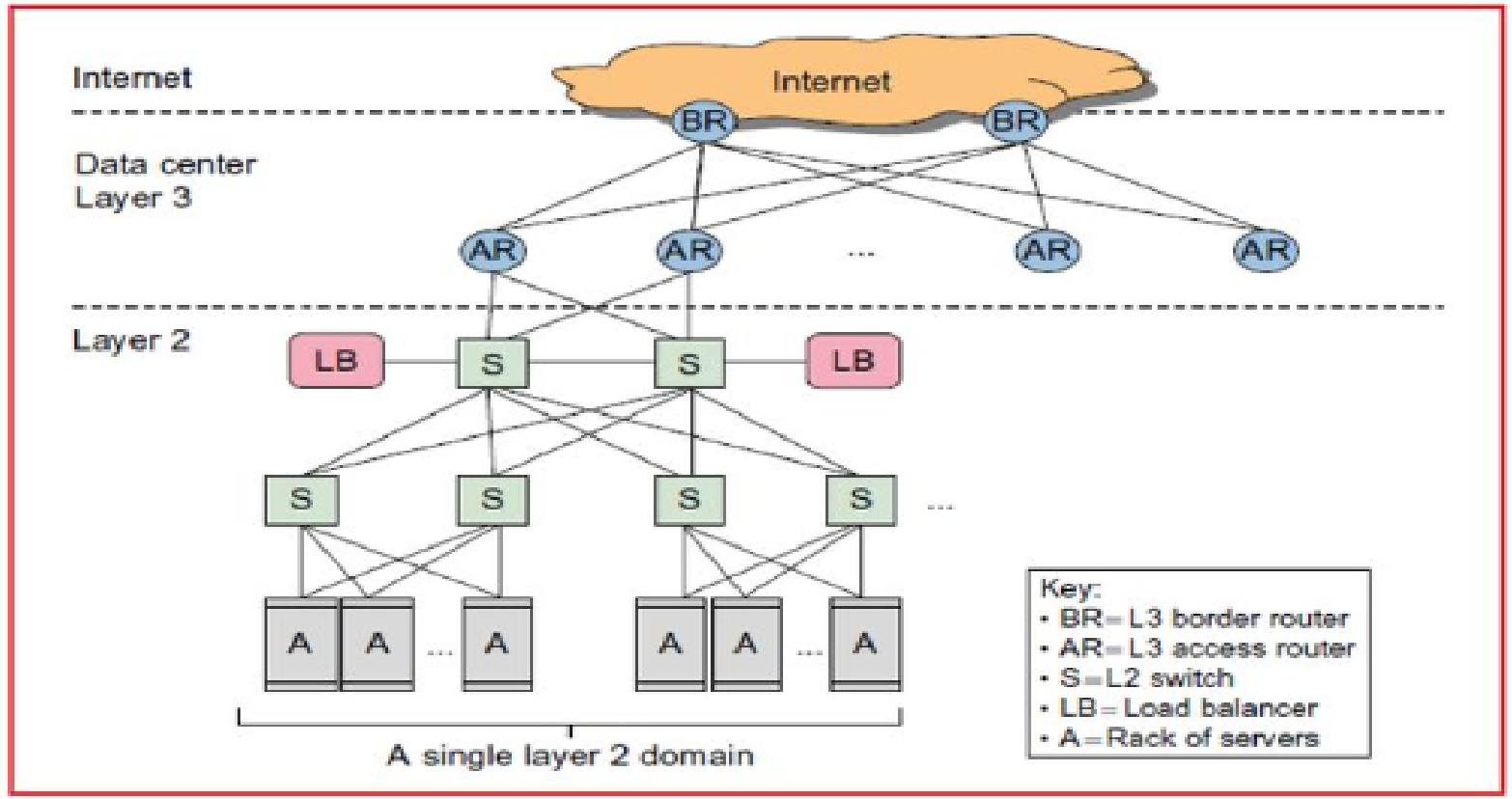


FIGURE 4.2

Standard data-center networking for the cloud to access the Internet.

Cloud Development Trends

- Although most clouds built since 2010 are large public clouds, the authors believe **private clouds** will grow much faster than **public clouds** in the future:
  - **Private clouds** are easier to secure and more trustworthy within a company or organization.
  - Once **private clouds** become mature and better secured, they could be open or converted to public clouds.
  - Therefore, the boundary between **public** and **private** clouds could be blurred in the future.
    - Most likely, most future clouds will be **hybrid** in nature.

Cloud Development Trends

  - For example, an e-mail application can run in the **service-access nodes** and provide the user interface for outside users; the application can get the service from the **internal cloud computing services** (e.g., the e-mail storage service).

- These nodes are called **runtime supporting service nodes**.
  - For example, there might be **distributed locking services** for supporting specific applications.
- Finally, it is possible that there will be some **independent service nodes:**
  - Those nodes would provide independent services for other nodes in the cluster.

Cloud Ecosystem and Enabling Technologies

- **Cloud computing** platforms differ from conventional **computing platforms** in many aspects:
  - The **traditional computing model** involves buying the HW, acquiring the necessary system SW, installing the system, testing the configuration, and executing the application code and management of resources.
  - What is even worse is that this cycle repeats itself in about every 18 months, meaning the machine we bought becomes obsolete every 18 months.

Cloud Design Objectives

- The following list highlights **six design objectives** for cloud computing:
  - **Shifting computing from desktops to data centers** over the internet.
  - **Service provisioning and cloud economics** by providers supply cloud services signing **SLA**s with consumers and end users.
  - **Scalability in performance** by improving software and infrastructure services help to scale in performance.
  - **Data privacy protection** by providing cloud as a trusted services.
  - **High quality of cloud services** by improving the QoS of cloud to make clouds interoperable among multiple providers.
  - **New standards and interfaces** by providing universally accepted APIs and access protocols for high portability and flexibility of virtualized applications.

Cost Model

Overall, cloud computing will reduce computing costs significantly for both small users and large enterprises. Computing economics does show a big gap between traditional IT users and cloud users.

The savings in acquiring expensive computers up front releases a lot of burden for startup companies.

The fact that cloud users only pay for operational expenses and do not have to invest in permanent equipment is especially attractive to massive numbers of small users.

This is a major driving force for cloud computing to become appealing to most enterprises and heavy computer users.

In fact, any IT users whose capital expenses are under more pressure than their operational expenses should consider sending their overflow work to utility computing or cloud service providers.

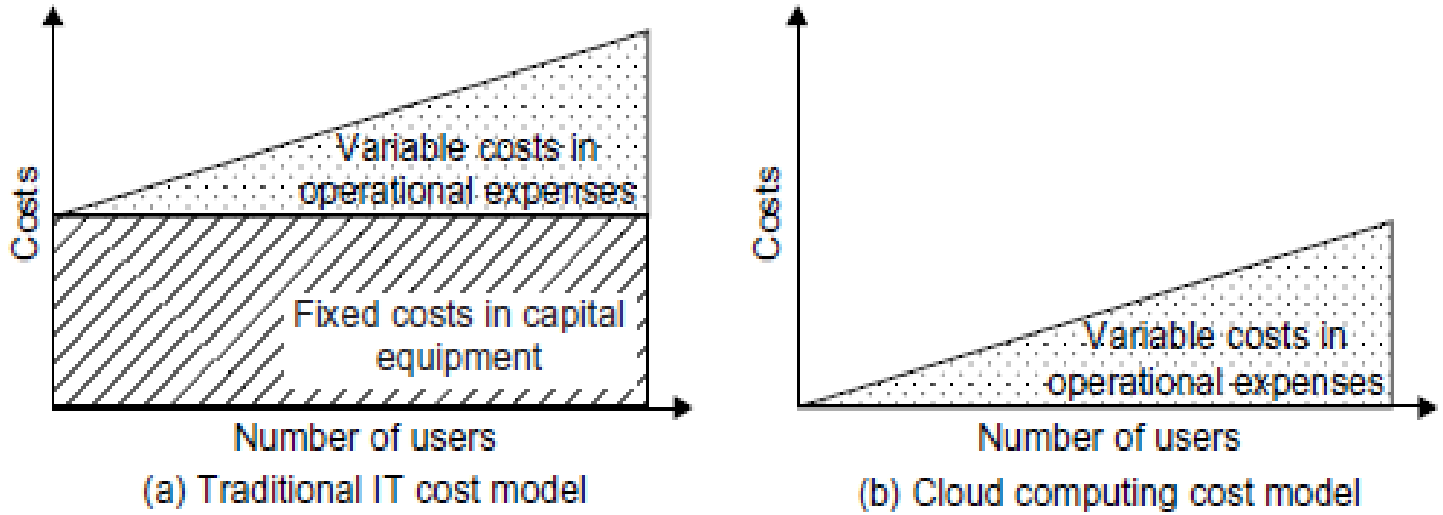


FIGURE 4.3

Computing economics between traditional IT users and cloud users.

Cloud Ecosystems

- With the emergence of various Internet clouds, an **ecosystem** of providers, users, and technologies has been appeared.
- This **ecosystem** has evolved around **public clouds**:
  - Strong interest is growing in **open source cloud computing** tools that let organizations build their own **infrastructure-as-a-service (IaaS)** clouds using their internal infrastructures.
- **Private** and **hybrid** clouds are **not exclusive**, since public clouds are involved in both cloud types:
  - A **private/hybrid** cloud allows remote access to its resources over the Internet using remote web service interfaces such as that used in **Amazon EC2**.
  - An **ecosystem** was suggested by Sotomayor (**Figure 4.4**) for building **private clouds**.

Cloud Ecosystems

- The **Figure 4.4** suggested **four levels of ecosystem** development in a **private cloud**:
  - **At the user end**, consumers demand a flexible platform.
  - **At the cloud management level**, the cloud manager provides virtualized resources over an IaaS platform.
  - **At the virtual infrastructure (VI) management level**, the manager allocates VMs over multiple server clusters.
  - **At the VM management level**, the VM managers handle VMs installed on individual host machines.
- An **ecosystem of cloud** tools attempts to span both cloud management and **VI** management.

Cloud Ecosystems

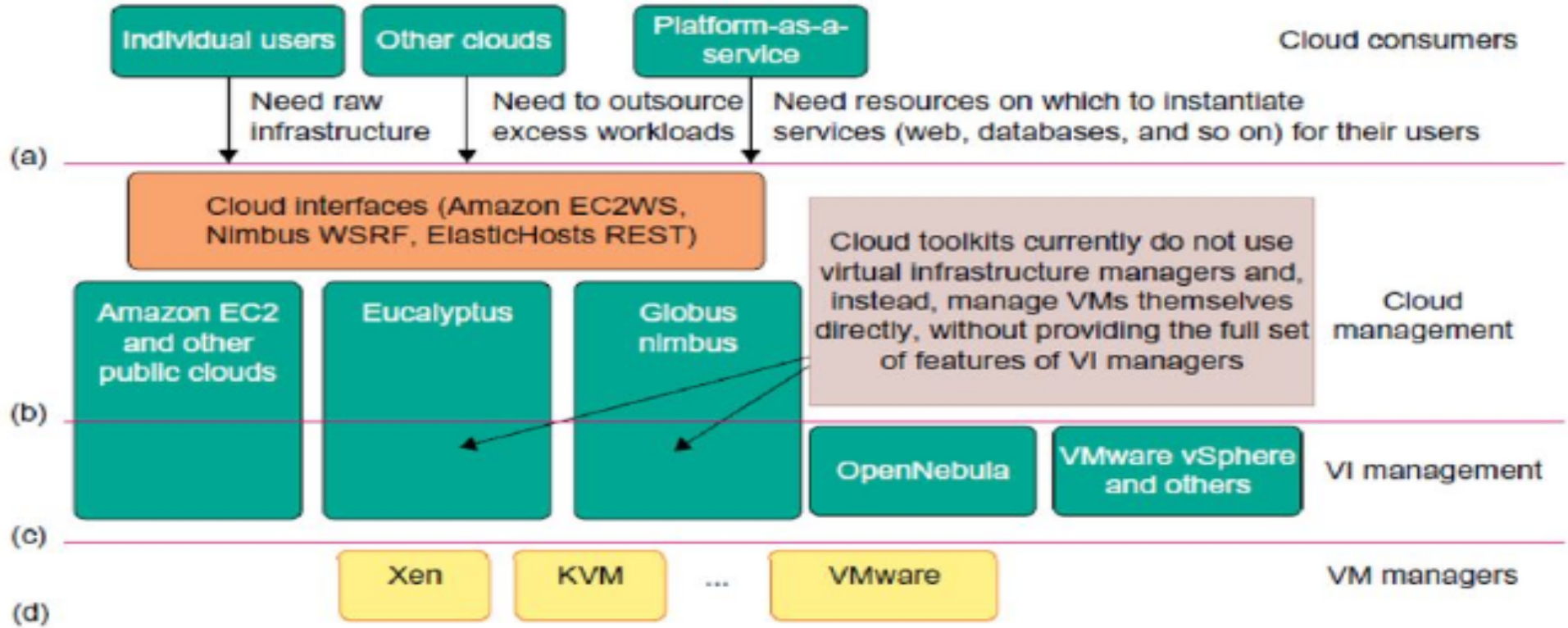


FIGURE 4.4

Cloud ecosystem for building private clouds: (a) Consumers demand a flexible platform; (b) Cloud manager provides virtualized resources over an IaaS platform; (c) VI manager allocates VMs; (d) VM managers handle VMs installed on servers.

The Three Cloud Service Models

- **Cloud computing delivers infrastructure**, **platform**, and **software** (application) as services, which are made available as subscription-based services in a pay-as-you-go model to consumers.
- The services provided over the cloud can be generally categorized into **three different service models**:
  - namely **IaaS**, Platform as a Service (**PaaS**), and Software as a Service (**SaaS**).
  - These form the **three pillars** on top of which cloud computing solutions are delivered to end users.
  - All **three models allow users to access services over the Internet,** relying entirely on the infrastructures of cloud service providers.

The Three Cloud Service Models

- These models are offered based on various **service –level-agreements (SLAs)** between providers and users:
  - The **SLA** for cloud computing is addressed in terms of *service availability*, *performance*, and *data protection and security*.
  - **Figure 4.5** illustrates **three cloud models** at different service levels of the cloud.
    - **Software as a service (SaaS)** model is applied at the application end using special interfaces by users or clients.
    - At the **platform as a service (PaaS)** layer, the cloud platform must perform billing services and handle job queuing, launching, and monitoring services.
    - At the bottom layer , the **infrastructure as a service (IaaS)** services, databases, compute instances, the file system, and storage .

The Three Cloud Service Models

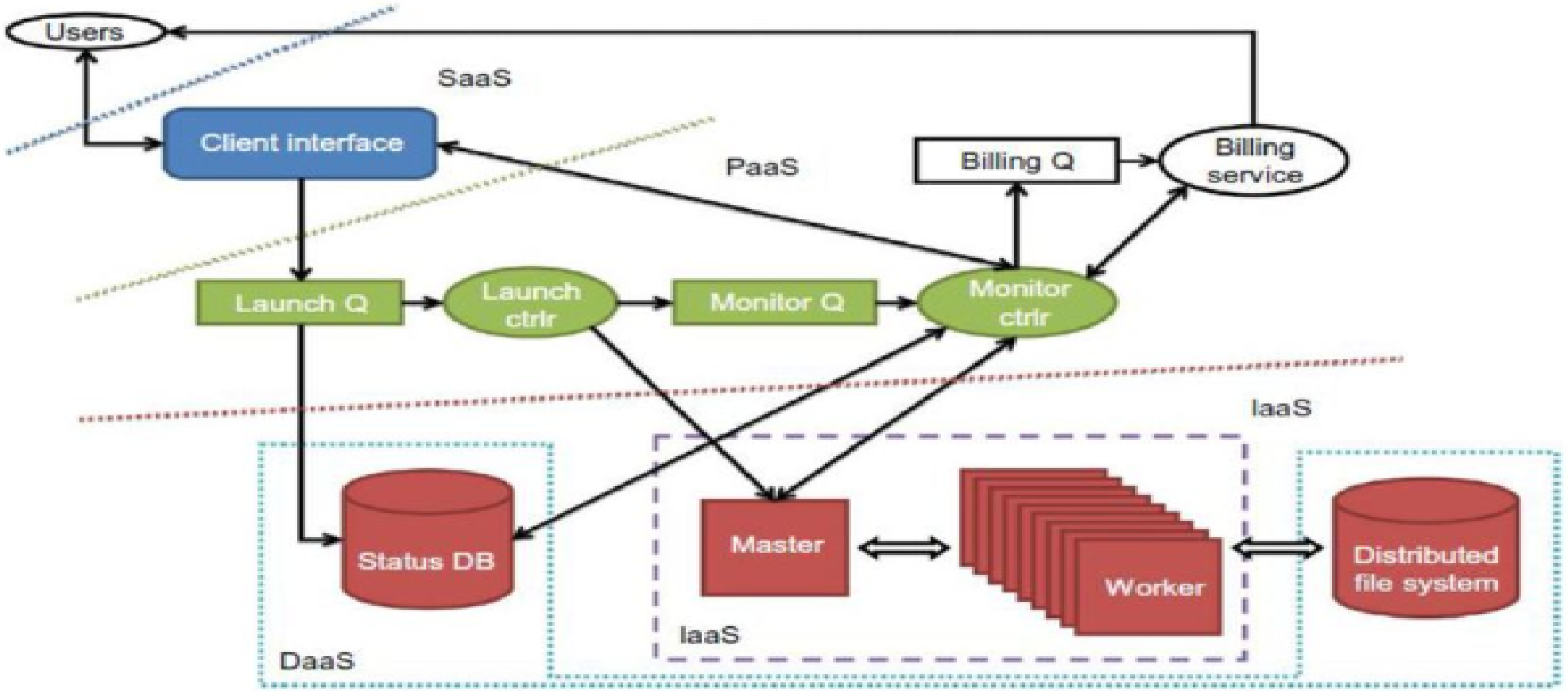


FIGURE 4.5 The IaaS, PaaS, and SaaS cloud service models at different service levels..

Infrastructure as a Service(IaaS)

- The **IaaS** model allows users to use virtualized IT resources for computing, storage, and networking:
  - The user can deploy and run his applications over his chosen OS environment.
    - The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components.
  - This **IaaS** model encompasses storage as a service, compute instances as a service, and communication as a service.
    - The **Virtual Private Cloud** (**VPC**) in **Figure 4.6** shows how to provide **Amazon EC2** clusters and **S3** storage to multiple users.

Infrastructure as a Service

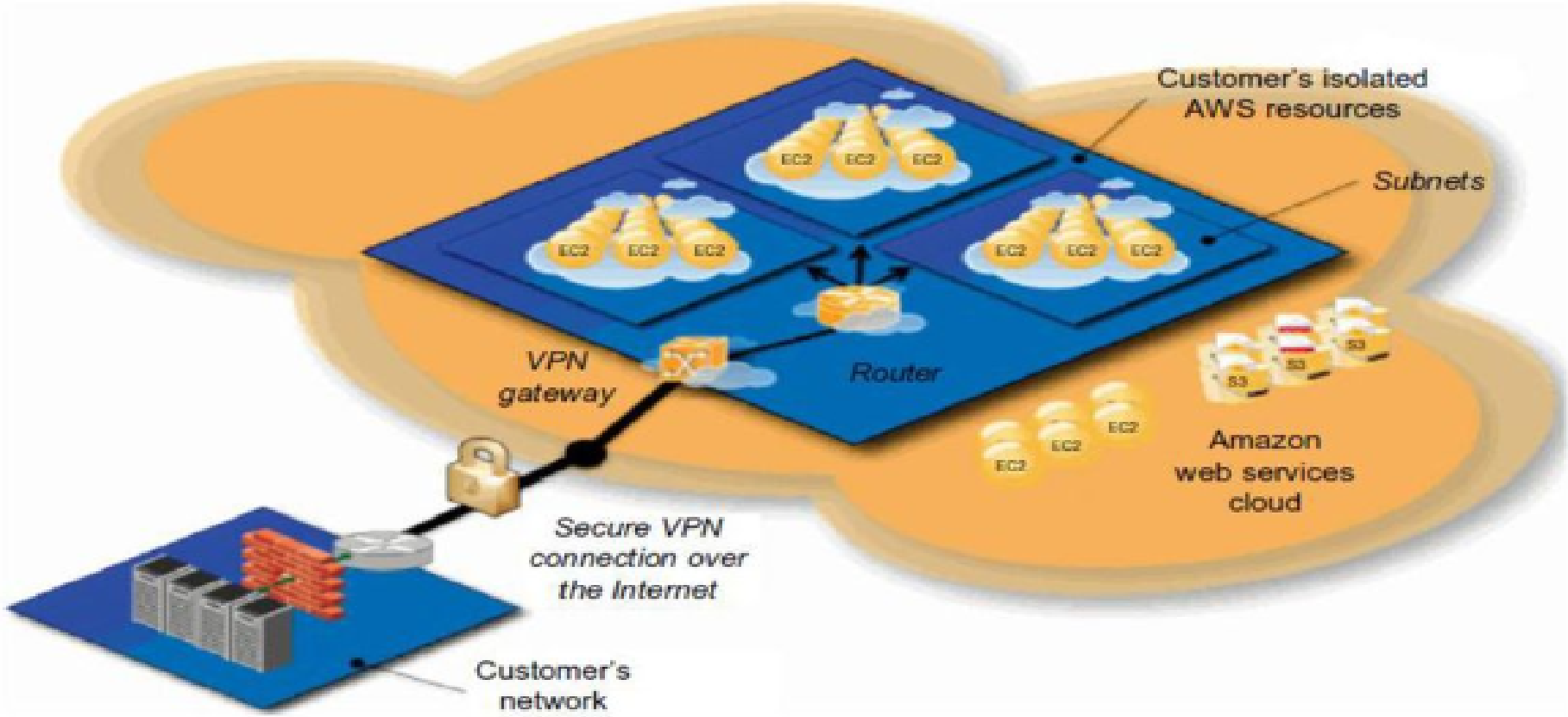


FIGURE 4.6 Amazon VPC (virtual private cloud).

Platform as a Service (PaaS)

- **PaaS** cloud model allows to develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment:
  - Such a platform includes OS and runtime library support.
  - This has triggered the creation of the **PaaS model** to enable users to develop and deploy their user applications.
- **Table 4.2** highlights cloud platform services offered by five **PaaS** services.

Platform as a Service (PaaS)

Table 4.2 Five Public Cloud Offerings of PaaS

Cloud Name	Languages and Developer Tools	Programming Models Supported by Provider	Target Applications and Storage Option
Google App Engine	Python, Java, and Eclipse-based IDE	MapReduce, web programming on demand	Web applications and BigTable storage
Salesforce.com's Force.com	Apex, Eclipse-based IDE, web-based Wizard	Workflow, Excel-like formula, Web programming on demand	Business applications such as CRM
Microsoft Azure	.NET, Azure tools for MS Visual Studio	Unrestricted model	Enterprise and web applications
Amazon Elastic MapReduce	Hive, Pig, Cascading, Java, Ruby, Perl, Python, PHP, R, C++	MapReduce	Data processing and e-commerce
Aneka	.NET, stand-alone SDK	Threads, task, MapReduce	.NET enterprise applications, HPC

Platform as a Service (PaaS)

- The **PaaS cloud** model is an integrated computer system consisting of both HW and SW infrastructure:
  - The user application can be developed on this **virtualized cloud** platform using some programming languages and software tools supported by the provider (e.g., Java, Python, .NET).
  - The user does not manage the underlying cloud infrastructure.
  - The cloud provider supports user application development and testing on a **well-defined** service platform.
- The **PaaS model** enables a collaborated software development platform , SW management, integration, service monitoring solutions for users from different parts of the world.

Google App Engine for PaaS Applications

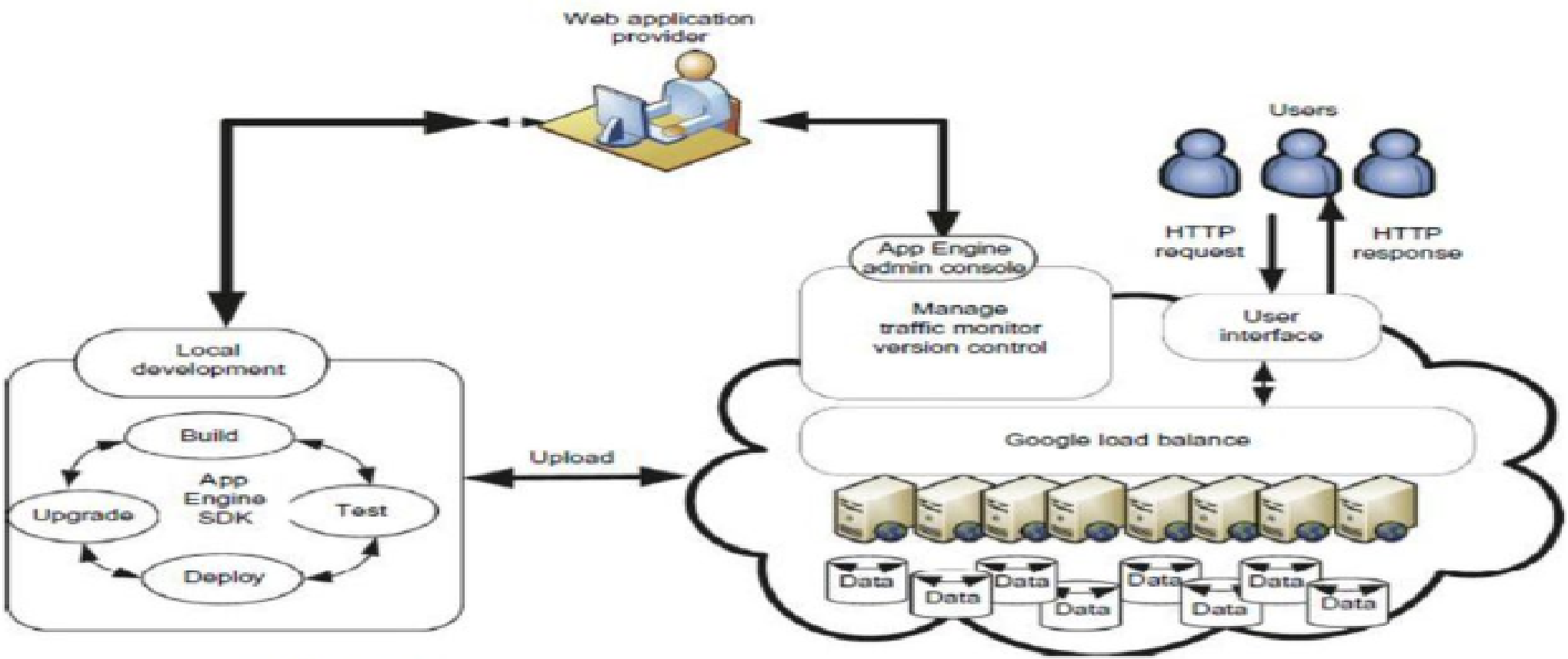


FIGURE 4.7

Google App Engine platform for PaaS operations.

Software as a Service (SaaS)

- **SaaS** refers to browser-initiated application software over thousands of cloud customers:
  - The SaaS model provides SW applications as a service.
  - As a result, on the customer side, there is no upfront investment in servers or SW licensing.
  - On the provider side, costs are kept rather low, compared with conventional hosting of user applications.
  - Customer data is stored in the cloud that is either vendor proprietary or publicly hosted to support **PaaS** and **IaaS**.

Software as a Service (SaaS)

- The best examples of **SaaS services** include **Google Gmail and docs**, **Microsoft SharePoint**, and the **CRM software** from Salesforce.com.
- Providers such as **Google** and **Microsoft** offer integrated **IaaS** and **PaaS** services.
- Whereas others such as **Amazon** and **GoGrid** offer pure **IaaS** services and expect third-party **PaaS** providers such as **Manjrasoft** to offer application development and deployment services on top of their infrastructure services.

DATA-CENTER DESIGN AND INTERCONNECTION NETWORKS

- A data center is often built with a large number of servers through a huge interconnection network

Warehouse-Scale Data-Center Design:

- Dennis Gannon claims: “The cloud is built on massive datacenters” . Figure 4.8 shows a datacenter that is as large as a shopping mall (11 times the size of a football field) under one roof.
- Such adata center can house 400,000 to 1 million servers. The data centers are built economics of scale—meaning lower unit cost for larger data centers.
- A small data center could have 1,000 servers. Thelarger the data center, the lower the operational cost. The approximate monthly cost to operate a huge400-server data center is estimated by network cost $13/Mbps; storage cost $0.4/GB; and administration costs. These unit costs are greater than those of a 1,000-server data center.
- The network cost to operate a small data center is about seven times greater and the storage cost is 5.7 times greater.Microsoft has about 100 data centers, large or small, which are distributed around the globe.

FIGURE 4.8

A huge data center that is 11 times the size of a football field, housing 400,000 to 1 million servers.

Cooling System of a Data-Center Room

- Figure 4.9 shows the layout and cooling facility of a warehouse in a data center. The data-center room has raised floors for hiding cables, power lines, and cooling supplies.
- The cooling system is somewhat simpler than the power system. The raised floor has a steel grid resting on stanchions about 2–4 ft above the concrete floor.
- The under-floor area is often used to route power cables to racks, but its primary use is to distribute cool air to the server rack.
- The CRAC (computer room air conditioning) unit pressurizes the raised floor plenum by blowing cold air into the plenum.

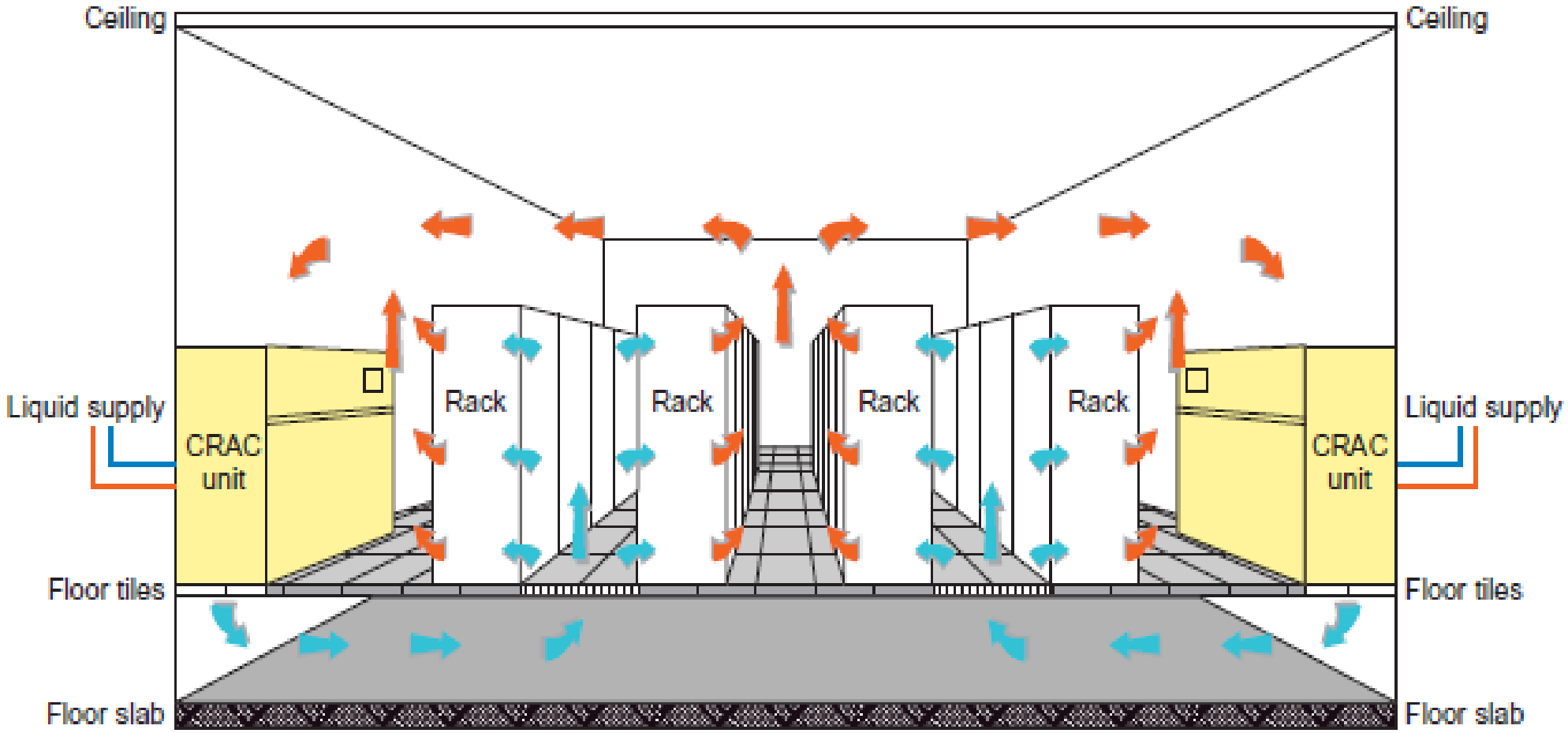


FIGURE 4.9

The cooling system in a raised-floor data center with hot-cold air circulation supporting water heat exchange facilities.

(Courtesy of DLB Associates, D. Dyer [23])

Data-Center Interconnection Networks

- **Application Traffic Support**: map-reduce.
- **Network Expandability**: The interconnection network should be expandable.
- With thousands or even hundreds of thousands of server nodes, the cluster network interconnection should be allowed to expand once more servers are added to the data center.
- The network topology should be restructured while facing such expected growth in the future.
- Also, the network should be designed to support load balancing and data movement among the servers.
- None of the links should become a bottleneck that slows down application performance. The topology of the interconnection should avoid such bottlenecks

- **Fault Tolerance and Graceful Degradation:** The interconnection network should provide some mechanism to tolerate link or switch failures.
- In addition, multiple paths should be established between any two server nodes in a data center.
- Fault tolerance of servers is achieved by replicating data and computing among redundant servers.
- Similar redundancy technology should apply to the network structure. Both software and hardware network redundancy apply to cope with potential failures.
- One the software side, the software layer should be aware of network failures. Packet forwarding should avoid using broken links.

- The network support software drivers should handle this transparently without affecting cloud operations.

Modular Data Center in Shipping Containers

FIGURE 4.11

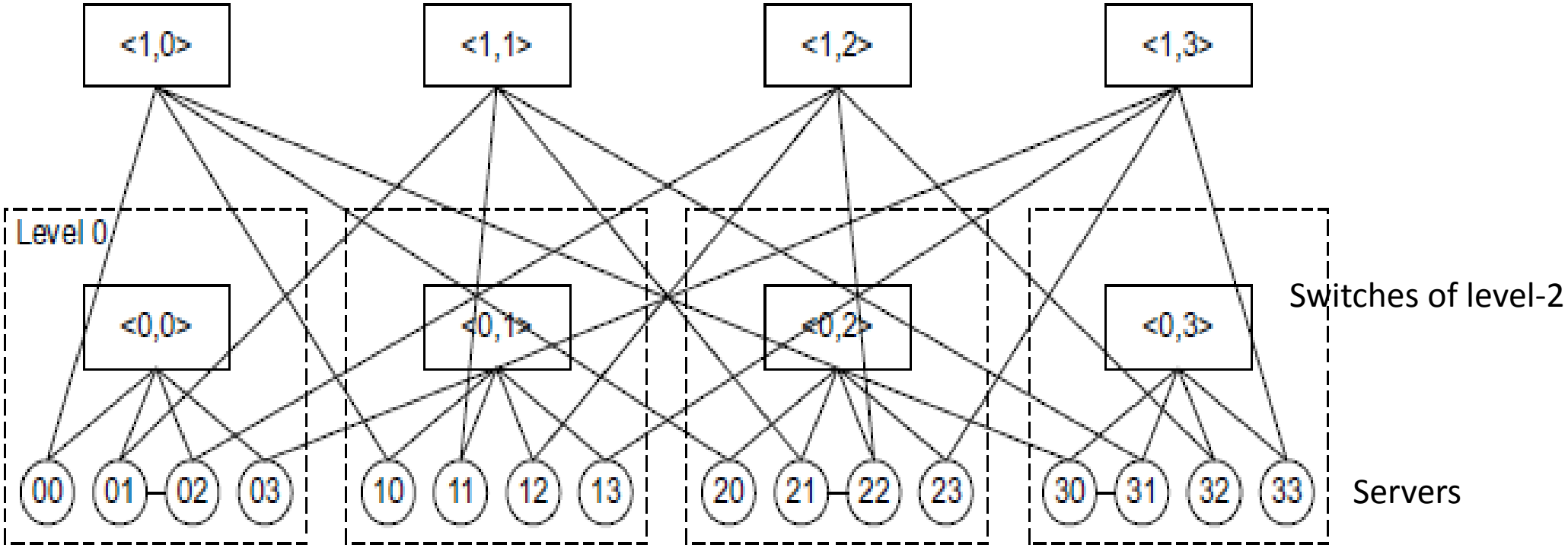
A modular data center built in a truck-towed ICE Cube container, that can be cooled by chilled air circulation with cold-water heat exchanges.

Interconnection of Modular Data Centers

- Container-based data-center modules are meant for construction of even larger data centers using a farm of container modules. Some proposed designs of container modules are presented in this section.
- Their interconnections are shown for building scalable data centers. The following example is a server-centric design of the data-center module

- The BCube provides multiple paths between any two nodes. Multiple paths provide extra bandwidth to support communication patterns in different cloud applications.
- The BCube provides a kernel module in the server OS to perform routing operations.
- The kernel module supports packet forwarding while the incoming packets are not destined to the current node.
- Such modification of the kernel will not influence the upper layer applications. Thus, the cloud application can still run on top of the BCube network structure without any modification

Level 1 Switches of level-1, for level 0 connect;level-1 0 to 3…….so on



Switches of level-2

Servers

FIGURE 4.12

BCube, a high-performance, server-centric network for building modular data centers.

A 2D MDCube constructed from nine BCube containers.

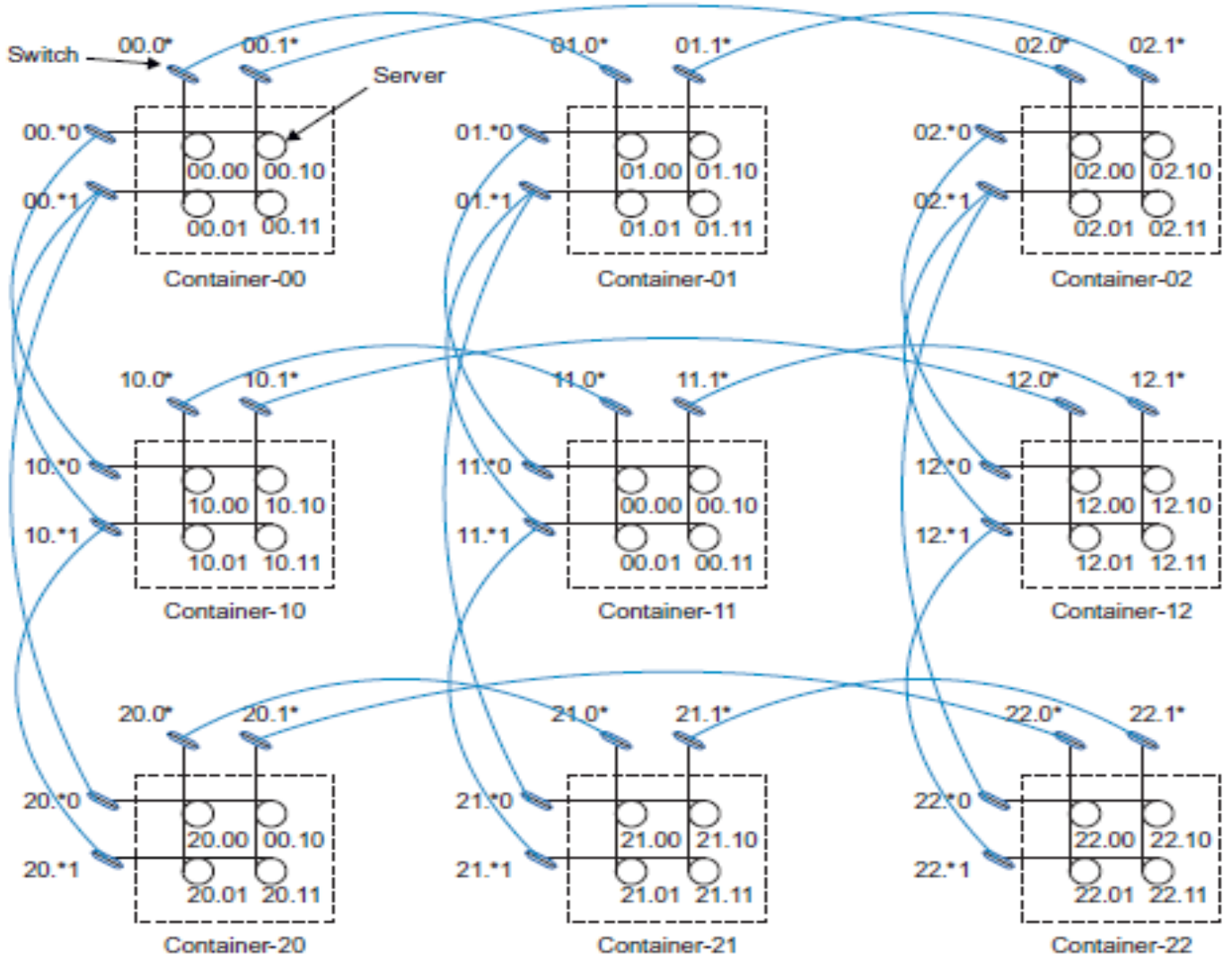


FIGURE 4.13

A 2D MDCube constructed from nine BCube containers.

(Courtesy of Wu, et al. [82])

Data-Center Management Issues

- **Making common users happy**: The data center should be designed to provide quality service to the majority of users for at least 30 years.
- **Controlled information flow** Information flow should be streamlined. Sustained services and high availability (HA) are the primary goals.
- **Multiuser manageability** The system must be managed to support all functions of a data center, including traffic flow, database updating, and server maintenance.
- **Scalability to prepare for database growth** The system should allow growth as workload increases. The storage, processing, I/O, power, and cooling subsystems should be scalable.
- **Reliability in virtualized infrastructure** Failover, fault tolerance, and VM live migration should be integrated to enable *recovery of critical applications from failures or disasters.*

- **Low cost to both users and providers** The cost to users and providers of the cloud system built over the data centers should be reduced, including all operational costs.

- **Security enforcement and data protection** Data privacy and security defense mechanisms must be deployed to protect the data center against *network attacks and system interrupts and to maintain data integrity from user abuses or network attacks*.

- **Green information technology** Saving power consumption and upgrading energy efficiency are in high demand when designing and operating current and future data centers.

Architectural Design of Compute and Storage Clouds

- 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.
- *Scalability*, *virtualization*, *efficiency*, and *reliability* are **four major design goals** of a cloud computing platform:
  - **Cloud management** receives the user request, finds the correct resources, and then calls the provisioning services which invoke the resources in the cloud.
  - The cloud management software needs to support both physical and **virtual machines** (**VMs**).
  - **Security** in shared resources and shared access of data centers also pose another design challenge.

Architectural Design of Compute and Storage Clouds

- The platform needs to establish a very large-scale **HPC** infrastructure.
  - The HW and SW systems are combined to make it easy and efficient to operate.
- **System scalability** can benefit from cluster architecture: If one service takes a lot of processing power, storage capacity, or network traffic, it is simple to add more servers and bandwidth, or data can be put into multiple locations.
  - For example, user e-mail can be put in three disks which expand to different geographically separate data centers.

Architectural Design of Compute and Storage Clouds

- The key driving forces behind **cloud computing** are the ubiquity of broadband and wireless networking, and progressive improvements in Internet computing SW:
  - **Cloud users** are able to demand more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers can increase system utilization via *multiplexing*, *virtualization*, and *dynamic resource provisioning*.
  - Clouds are enabled by the progress in HW, SW, and networking technologies summarized in **Table 4.3**.

Multiplexing is a technique used to combine and send the multiple data streams over a single medium

Architectural Design of Compute and Storage 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** shows a **security-aware** cloud architecture.
- The Internet cloud is envisioned as a massive cluster of servers:
  - These servers are provisioned on demand to perform ***collective web services*** or ***distributed applications*** using **data-center** resources.
- The **cloud platform** is formed dynamically by provisioning or deprovisioning servers, software, and database resources.
  - **Servers** in the cloud can be physical machines or VMs.
  - User interfaces are applied to request services.
  - The provisioning tool carves out the cloud system to deliver the requested service.

A Generic Cloud Architecture

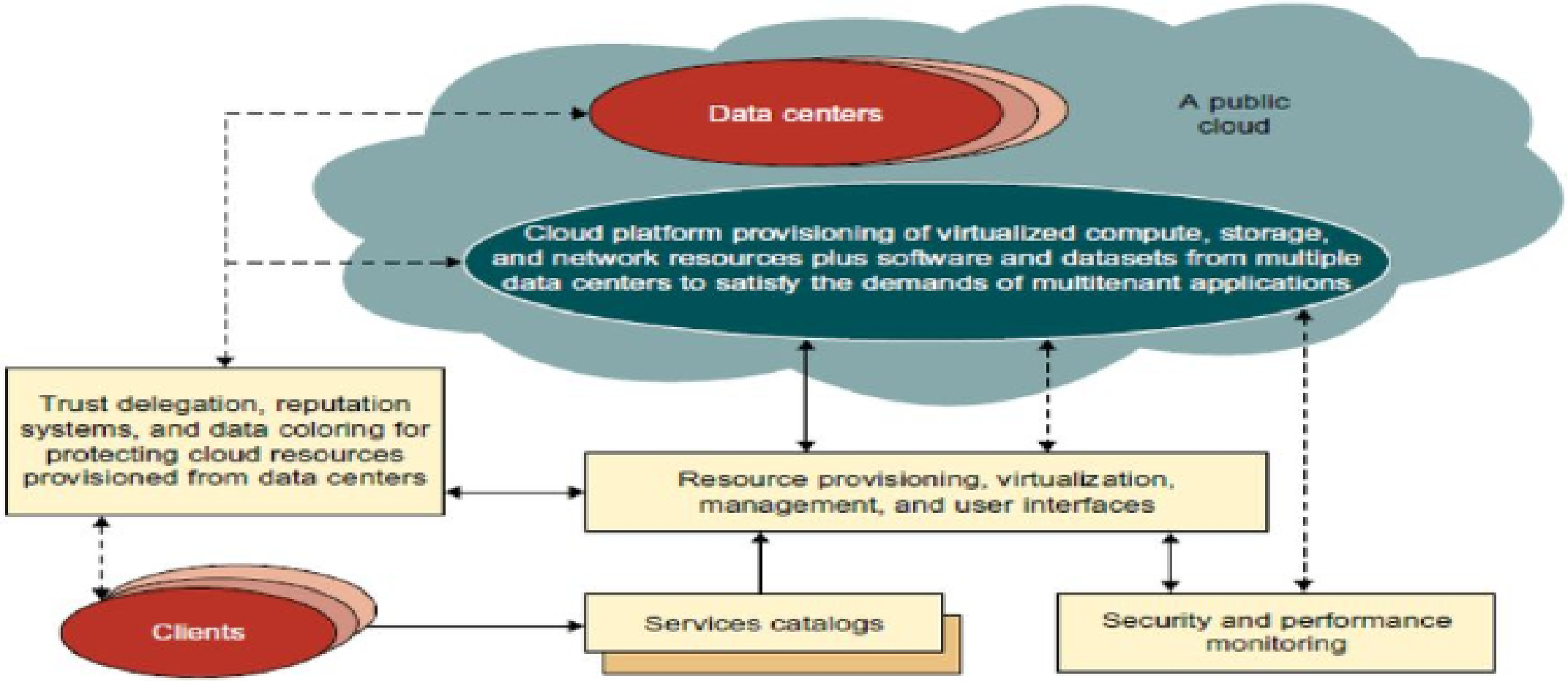


FIGURE 4.14

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

A Generic Cloud Architecture

- The cloud computing resources are built into the **data centers**, which are typically owned and operated by a third-party provider:
- The cloud demands a high degree of trust of massive amounts of data retrieved from large data centers.
- To build a framework to process large-scale data stored in the storage system:
  - This demands a **distributed file system** over the **database system**.
  - Other **cloud resources** are added into a cloud platform, including **storage area networks (SANs)**, database systems, firewalls, and security devices.

A Generic Cloud Architecture

- **Web service** providers offer special **APIs** that enable developers to exploit Internet clouds.
- The SW infrastructure of a cloud platform must handle all **resource management** and do most of the maintenance automatically:
  - SW must detect the status of each node server joining and leaving, and perform relevant tasks accordingly.
  - **Cloud computing providers**, such as Google and Microsoft, have built a large number of data centers all over the world.
  - Each **data center** may have thousands of servers.

A Generic Cloud Architecture

- In general, **private clouds** are easier to manage, and **public clouds** are easier to access.
- The trends in cloud development are that more and more clouds will be **hybrid**.
  - This is because many cloud applications must go beyond the boundary of an intranet.
- One must learn how to create a **private cloud** and how to interact with **public clouds** in the open Internet.
- **Security** becomes a critical issue in safeguarding the operation of all cloud types.

Layered Cloud Architectural Development

- The architecture of a cloud is developed at **three layers**: infrastructure, platform, and application as demonstrated in **Figure 4.15**).
- The services to **public**, **private**, and **hybrid** clouds are conveyed to users through networking support over the Internet and intranets involved.
- It is clear that the **infrastructure layer** is deployed first to support **IaaS** services.
- This infrastructure layer serves as the foundation for building the platform layer of the cloud for supporting **PaaS** services.
- The **platform layer** is a foundation for implementing the application layer for **SaaS** applications.

Layered Cloud Architectural Development

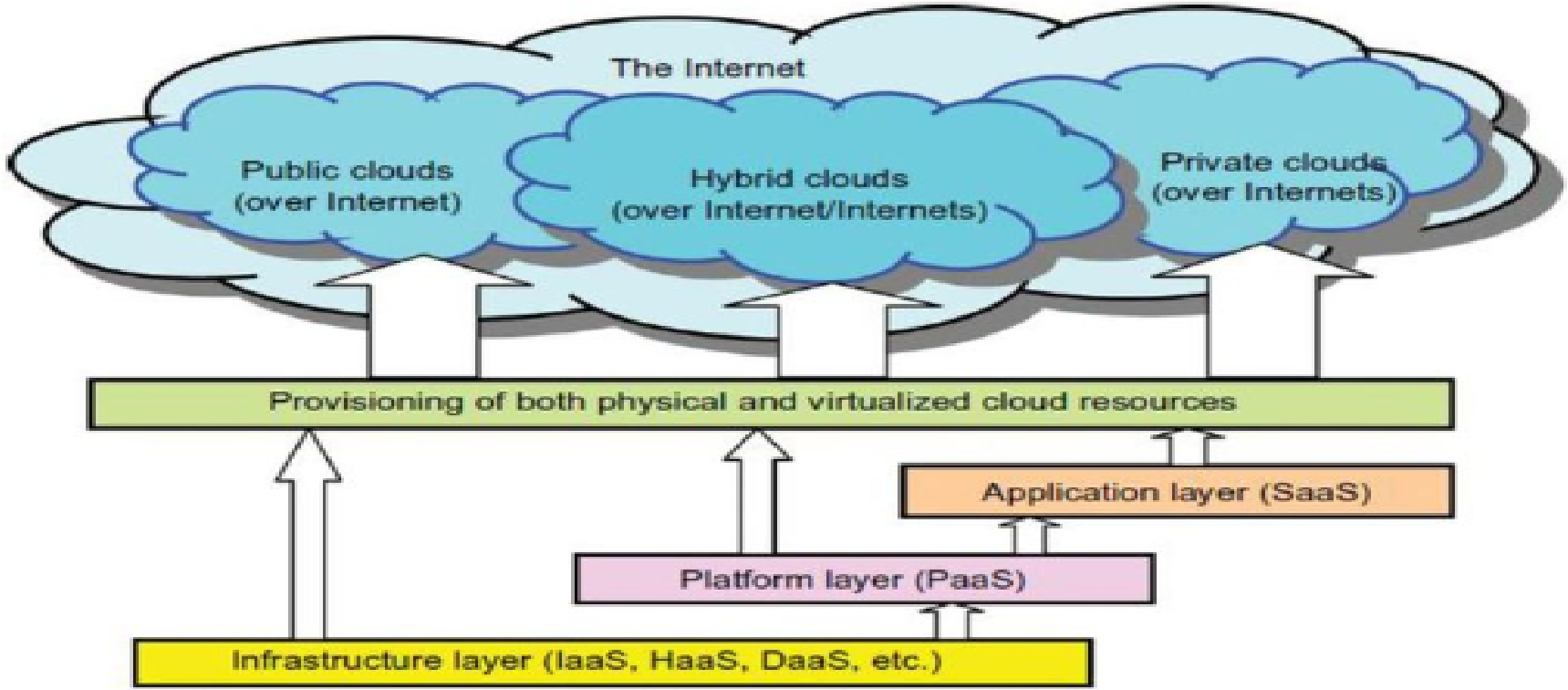


FIGURE 4.15 Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet.

Layered Cloud Architectural Development

- The **infrastructure layer** is built with virtualized compute, storage, and network resources.
  - Internally, **virtualization** realizes automated provisioning of resources and optimizes the infrastructure management process.
- The **platform layer** is for general-purpose and repeated usage of the collection of SW resources:
  - This layer provides users with an environment to develop their applications, to test operation flows, and to monitor execution results and performance.
  - The **virtualized cloud platform** serves as a "**system middleware**" between the infrastructure and application layers of the cloud.

Layered Cloud Architectural Development

- The **application layer** is formed with a collection of all needed SW modules for **SaaS** applications.
  - **Service applications** in this layer include daily office management work, such as information retrieval, document processing, and calendar and authentication services.
- The **application layer** is also heavily used by enterprises in business marketing and sales, **consumer relationship management (CRM)**, financial transactions, and supply chain management.
  - It should be noted that not all cloud services are restricted to a single layer.
  - Many applications may apply resources at mixed layers.

Market-Oriented Cloud Architecture

- As consumers rely on **cloud providers**, they will require a specific level of **QoS** to be maintained by their providers:
  - **Cloud providers** consider and meet the different QoS parameters of each individual consumer as negotiated in specific **SLAs**.
  - To achieve this, the providers deploy **market-oriented resource management** to regulate the supply and demand of cloud resources to achieve market equilibrium between supply and demand instead of traditional s**ystem-centric resource management** architecture.
  - **Figure 4.16** shows the high-level architecture for **market-oriented resource allocation** in a cloud computing environment.

Market-Oriented Cloud Architecture

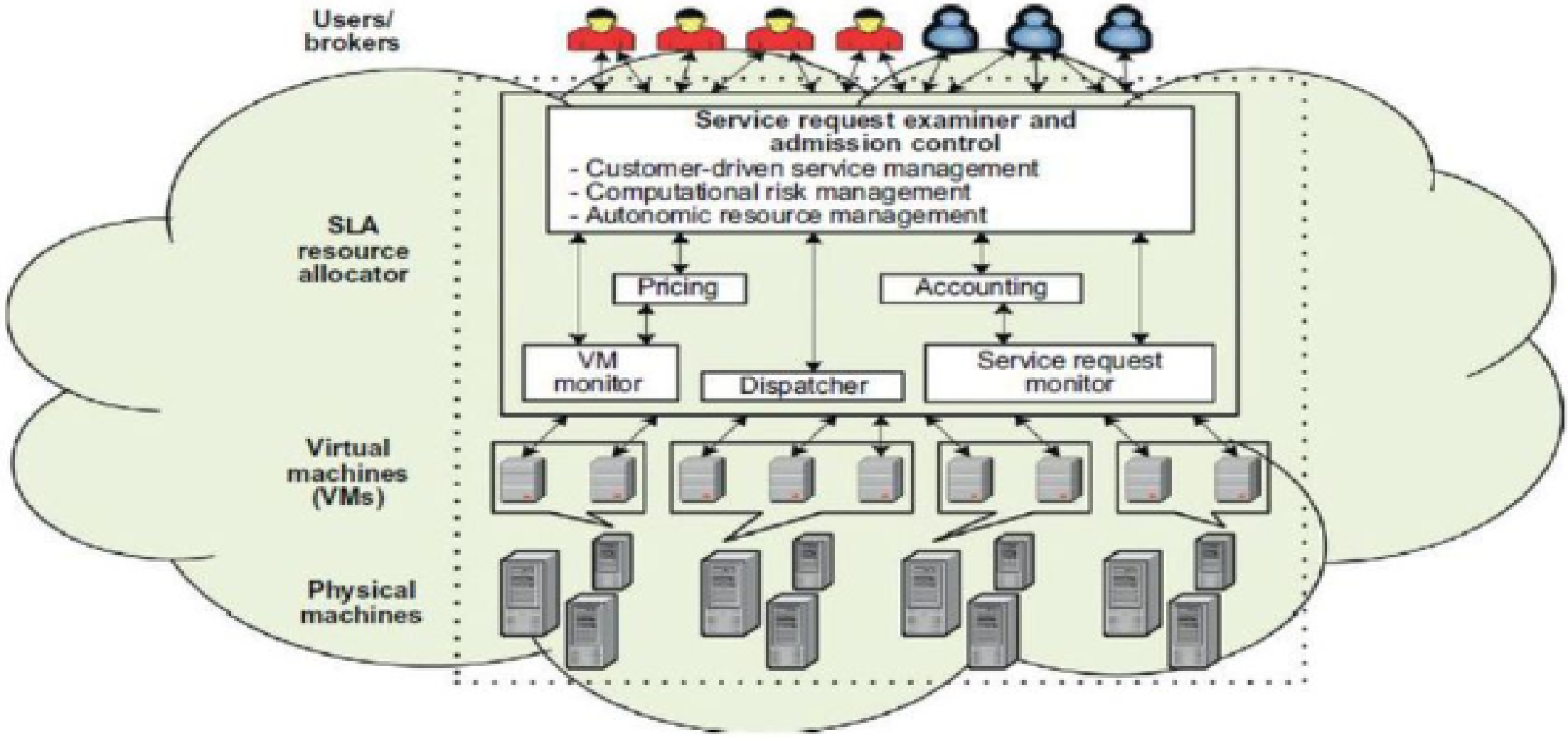


FIGURE 4.16

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

Market-Oriented Cloud Architecture

- The **market-oriented cloud** is basically built with the following entities:
  - **Users** or **brokers** acting on user's behalf submit service requests from anywhere in the world to the data center.
  - The **SLA resource allocator** acts as the interface between the data center/cloud service provider and external users/brokers.
  - When a **service request** is first submitted, the service request examiner interprets the request for **QoS** requirements before to accept or reject the request.

Market-Oriented Cloud Architecture

- The **service request examiner** ensures that there is **no overloading** of resources whereby many service requests cannot be fulfilled due to limited resources.
- It also needs the latest status information regarding **resource availability** (*from the VM Monitor mechanism*) and **workload processing** (*from the Service Request Monitor mechanism*) in order to make **resource allocation** decisions effectively.
- The **VM Monitor mechanism** keeps track of the availability of VMs and their resource entitlements.

Market-Oriented Cloud Architecture

- The **dispatcher mechanism** starts the execution of accepted service requests on allocated VMs.
- The **Service Request Monitor mechanism** keeps track of the execution progress of service requests.
- **Multiple VMs** can be started and stopped on demand on a single physical machine to meet accepted service requests.
- **Multiple VMs** can concurrently run applications based on different OS environments on a single physical machine.

Virtualization Support and Disaster Recovery

- One of the distinguishing features of cloud computing infrastructure is the use of **system virtualization**:
  - **Virtualization of servers** on a shared cluster can consolidate web services.
- As the VMs are the containers of cloud services, the **provisioning tools** will first find the corresponding physical machines and deploy the VMs to those nodes before scheduling the service to run on the virtual nodes:
  - The user will not care about the computing resources that are used for providing the services.

Hardware Virtualization

- In many cloud computing systems**, virtualization SW** is used to **virtualization of the HW**:
  - **Virtualization SW** is a special kind of SW which simulates the execution of HW and runs even unmodified OSs.
  - **Virtualization software** is also used as the platform for developing new cloud applications that enable developers to use any OSs and programming environments they like.
  - The development environment and deployment environment can now be the same, which eliminates some runtime problems.

Hardware Virtualization

- As mentioned before, system **virtualization SW** is considered the HW mechanism to run an **unmodified OS**, usually on bare HW directly, on top of SW.
- **Table 4.4** lists some of the system **virtualization software** in wide use at the time of this writing.
  - Currently, the **VMs** installed on a cloud computing platform are mainly used for hosting third-party programs.
  - VMs provide flexible runtime services to free users from worrying about the system environment.

Hardware Virtualization

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

Hardware Virtualization

- Users have full access to their own **VMs**, which are completely separate from other users' **VMs**.
  - **Multiple VMs can be mounted on the same physical server.**
  - **Different VMs may run with different OSs.**
- Also needs to establish the **virtual disk storage** and **virtual networks** needed by the VMs:
  - The **virtualization** is carried out by special servers dedicated to generating the **virtualized resource pool**.
  - The virtualized infrastructure is built with ***virtualizing integration managers***.
    - **These managers** handle loads, resources, security, data, and provisioning functions.

Cloud 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.
- **Challenge 4**: Distributed Storage and Widespread Software Bugs.
- **Challenge 5**: Cloud Scalability, Interoperability, and Standardization.
- **Challenge 6**: Software Licensing and Reputation Sharing.

Service Availability and Data Lock-in Problem

- The management of a cloud service by a single company is often the source of single points of failure.
- To achieve HA, one can consider using **multiple cloud providers**.
- Even if a company has **multiple data centers** located in different geographic regions, it may have common SW infrastructure and accounting systems.
- **Therefore, using multiple cloud providers may provide more protection from failures**.
- Another availability obstacle is **distributed denial of service** (**DDoS**) attacks. Some utility computing services offer **SaaS** providers the opportunity to defend against **DDoS attacks** by using quick scale-ups.

Data Privacy and Security Concerns

- Cloud offerings through **public** networks, exposing the system to more attacks.
- Many attacks can be overcome with technologies such as **encrypted storage**, **virtual LANs**, and **network middleboxes** (e.g., firewalls, packet filters).
  - For example, you could encrypt your data before placing it in a cloud.
- Impose laws requiring **SaaS** providers to keep customer data and copyrighted material within national boundaries.
  - **In a cloud environment, attacks may result from *hypervisor malware*, *guest hopping and hijacking*, or *VM rootkits*.**
- In general, **passive attacks** steal sensitive data or passwords.
- **Active attacks** may manipulate kernel data structures which will cause major damage to cloud servers.

VM rootkit

- **Rootkit** is a term applied to a type of malware that is designed to infect a target PC and allow an attacker to install a set of tools that grant him persistent remote access to the computer

Unpredictable Performance and Bottlenecks

- Multiple **VMs** can share CPUs and **main memory** in cloud computing, but **I/O sharing** is problematic:
  - **This issue is due to the problem of I/O interference between VMs**.
- One solution is to improve I/O architectures and operating systems to efficiently **virtualizes interrupts and I/O channels**.
  - Internet applications continue to become more data-intensive.
  - If we assume applications to be “pulled apart” across the boundaries of clouds, this may complicate data placement and transport.

Distributed Storage and Widespread Software Bugs

- The design of efficient distributed **storage area networks** (**SANs**) is based on the database and storage system of the cloud network.
- **Data consistence checking** in **SAN**-connected data centers is a **major challenge** in cloud computing.
- Large-scale **distributed bugs** cause debugging and that must occur at the data centers.
  - **No data center will provide such a convenience**.
- One solution may be a reliance on using **VMs** in cloud computing.
  - The level of **virtualization** may make it possible to capture valuable information.

Cloud Scalability, Interoperability, and Standardization

- Computation in a cloud is different depending on **virtualization level**.
  - **Google App Engine**(**GAE**) automatically scales in response to load increases and decreases; users are charged by the cycles used.
  - **Amazon Web Service** (**AWS**) charges by the hour for the number of **VM** instances used, even if the machine is idle.
- The opportunity here is to scale quickly up and down in response to **load variation**, in order to save money, but without violating **SLA**s.

Cloud Scalability, Interoperability, and Standardization

- **Open Virtualization Format** (**OVF**) describes an ***open***, ***secure***, ***portable***, ***efficient***, and ***extensible format*** for the packaging and distribution of **VMs**.
  - It also defines a **format for distributing SW** to be deployed in **VMs**.
  - This **VM format** does not rely on the use of a specific *host platform*, *virtualization platform*, or *guest OS*.
- The approach is to address v**irtual platform-agnostic packaging** with certification and integrity of packaged software.
  - **The package supports virtual appliances to span more than one VM**.

Software Licensing and Reputation Sharing

- Many cloud computing providers originally relied on **open source SW** because the licensing model for commercial software is not ideal for utility computing.
- The primary opportunity is either for **open source** to remain popular or simply for commercial software companies to change their licensing structure to better fit cloud computing.
- One can consider using both **pay-for-use** and **bulk-use** licensing schemes to widen the business coverage.

Public Clouds and Service Offerings

- Cloud services are demanded by computing and IT administrators, software vendors, and end users (**Figure 4.19** introduces five levels of cloud players):
  - **At the top level**, *individual users* and *organizational users* demand very different services.
  - The application providers at the **SaaS** level serve mainly individual users.
  - Most business organizations are serviced by **IaaS** and **PaaS** providers.
  - The **IaaS** provide compute, storage, and communication resources to both applications and organizational users.
  - The cloud environment is defined by the **PaaS** or platform providers.
    - Note that the **platform providers** support both infrastructure services and organizational users directly.

Public Clouds and Service Offerings

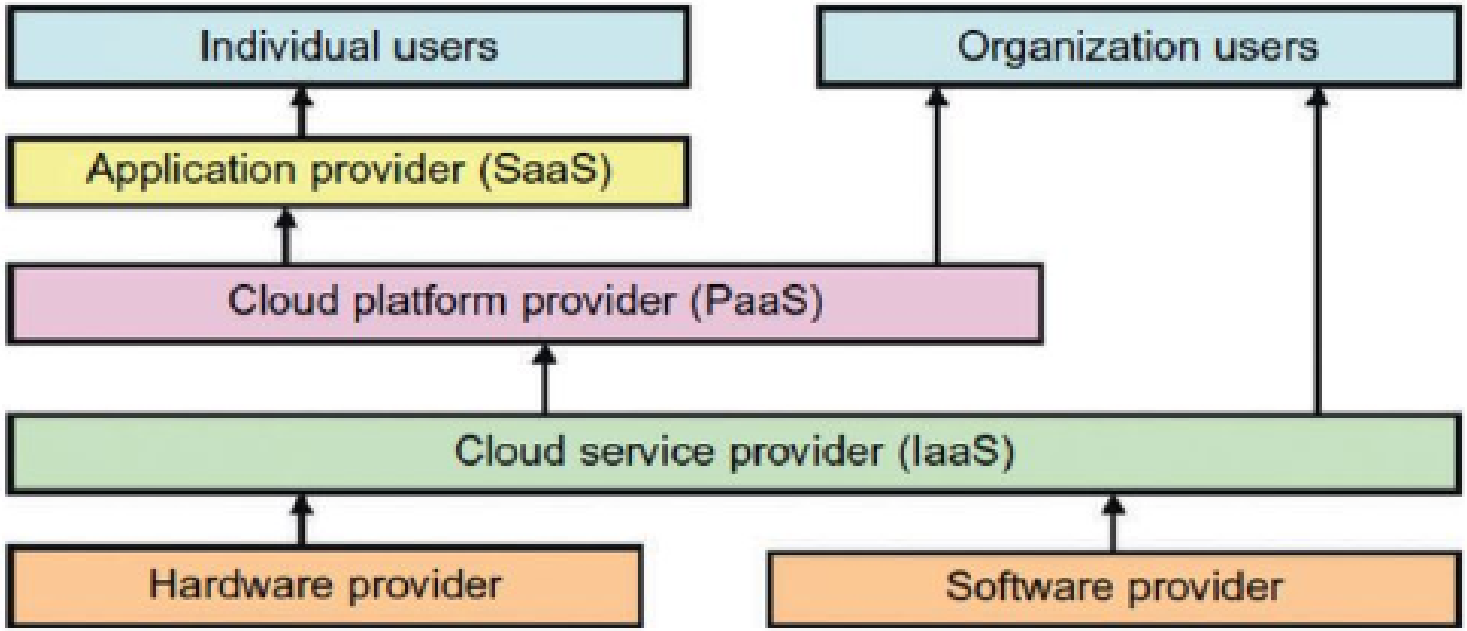


FIGURE 4.19

Roles of individual and organizational users and their interaction with cloud providers under various cloud service models.

Public Clouds and Service Offerings

- Cloud services rely on new advances in **machine virtualization**, **SOA, grid infrastructure management**, and **power efficiency**.
  - Consumers purchase such services in the form of **IaaS**, **PaaS**, or **SaaS**.
- The cloud industry leverages the growing demand by many enterprises and business users **to outsource their computing and storage jobs** to professional providers.
  - **The provider service charges are often much lower than the cost for users to replace their obsolete servers frequently**.
- **Table 4.5** summarizes the profiles of **five major cloud** providers by 2010 standards.

Table 4.5 Five Major Cloud Platforms and Their Service Offerings

Model	IBM	Amazon	Google	Microsoft	Salesforce
PaaS	BlueCloud, WCA, RC2		App Engine (GAE)	Windows Azure	Force.com
IaaS	Ensembles	AWS		Windows Azure	
SaaS	Lotus Live		Gmail, Docs	.NET service, Dynamic CRM	Online CRM, Gifttag
Virtualization		OS and Xen	Application Container	OS level/ Hypel-V	
Service Offerings	SOA, B2, TSAM, RAD, Web 2.0	EC2, S3, SQS, SimpleDB	GFS, Chubby, BigTable, MapReduce	Live, SQL Hotmail	Apex, visual force, record security
Security Features	WebSphere2 and PowerVM tuned for protection	PKI, VPN, EBS to recover from failure	Chubby locks for security enforcement	Replicated data, rule-based access control	Admin./record security, uses metadata API
User Interfaces		EC2 command-line tools	Web-based admin. console	Windows Azure portal	
Web API	Yes	Yes	Yes	Yes	Yes
Programming Support	AMI		Python	.NET Framework	

Case study:MS WINDOW AZURE self study

Microsoft Windows Azure

- In 2008, Microsoft launched a **Windows Azure** platform to meet the challenges in cloud computing.
- This platform is built over Microsoft data centers.
- **Figure 4.22** shows the overall architecture of Microsoft's cloud platform.
- The platform is divided into **three** major component platforms:
  - **Windows Azure** offers a cloud platform built on Windows OS and based on Microsoft virtualization technology.
  - Applications are installed on **VMs deployed** on the **data-center servers**.
  - Azure manages all servers, storage, and network resources of the data center.
  - On top of the infrastructure are the various services for building different cloud applications.

Microsoft Windows Azure

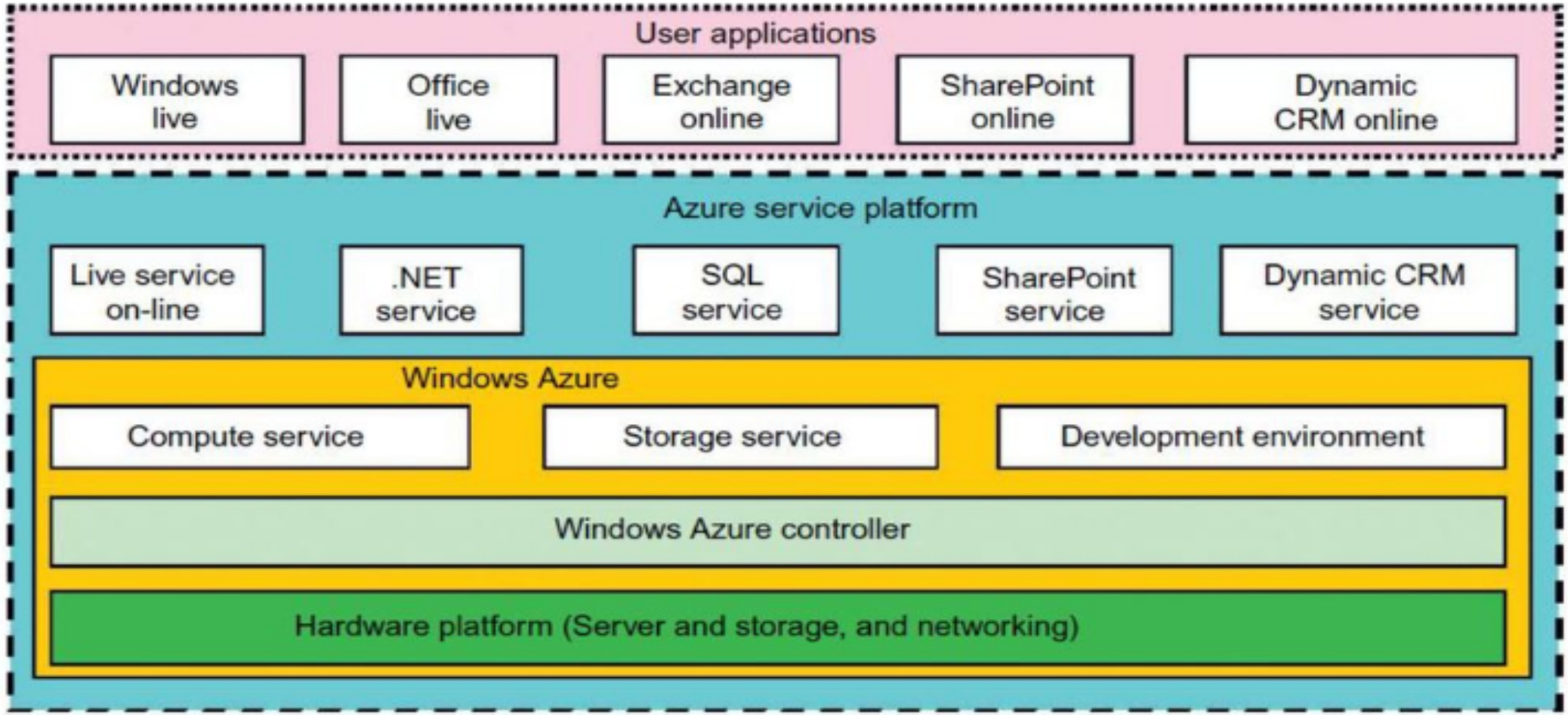


FIGURE 4.22

Microsoft Windows Azure platform for cloud computing.

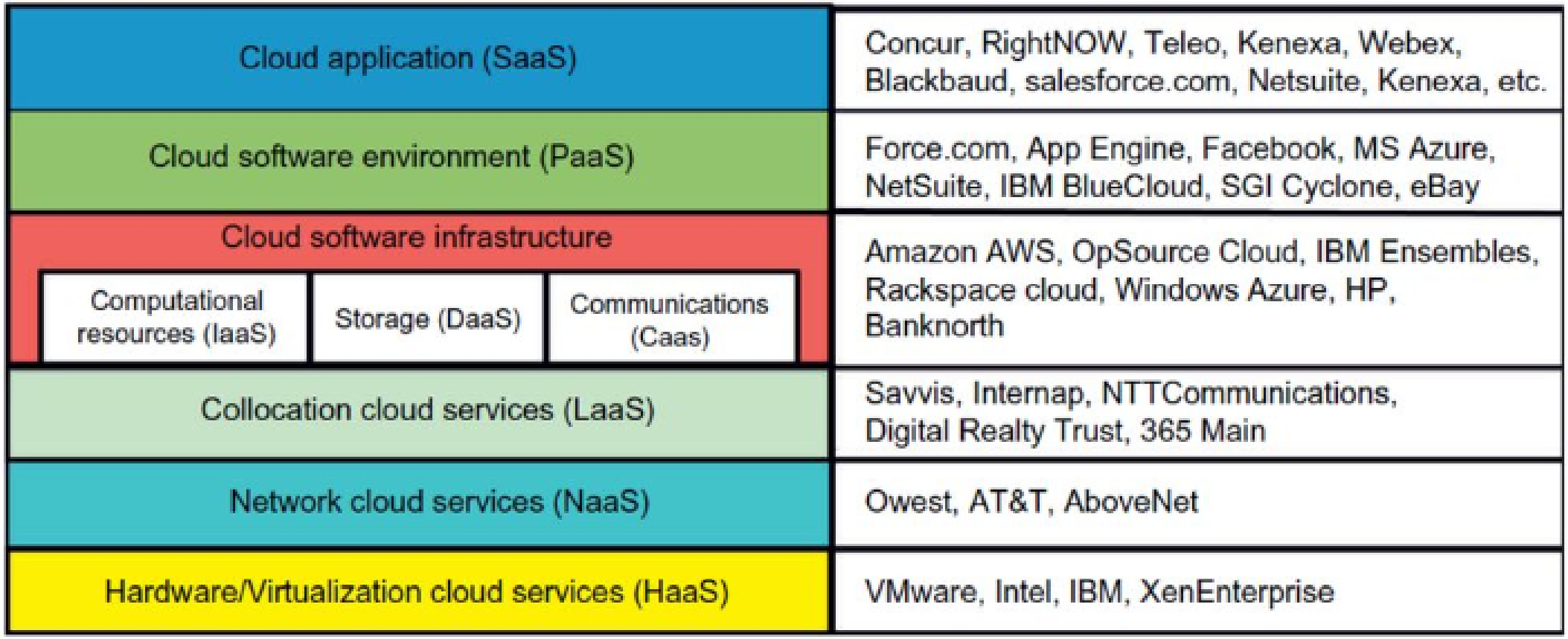
Microsoft Windows Azure

- **Cloud-level services** provided by the Azure platform are introduced below:
  - **Live service:** Users can visit Microsoft Live applications and access multiple machines concurrently.
  - **.NET service:** This package supports application development on local hosts and execution on cloud machines.
  - **SQL Azure:** This function makes it easier for users to visit and use the relational database with the SQL server in the cloud.
  - **SharePoint service:** This provides a scalable and manageable platform for users to develop business applications in upgraded web services.
  - **Dynamic CRM service:** This provides SW developers a business platform in managing CRM applications in financing, marketing, sales, etc.

Extended Cloud Computing Services

- **Figure 4.23** shows **six layers of cloud services**, ranging from hardware, network, and collocation to infrastructure, platform, and software applications.
- We already introduced the top three service layers as **SaaS**, **PaaS**, and **IaaS**, respectively.
- The cloud platform provides PaaS, which sits on top of the **IaaS** infrastructure.
- The top layer offers **SaaS**.
- These must be implemented on the cloud platforms provided.
- Although the three basic models are dissimilar in usage, as shown in **Table 4.7**, they are built one on top of another.

Extended Cloud Computing Services



Extended Cloud Computing Services

Table 4.7 Cloud Differences in Perspectives of Providers, Vendors, and Users

Cloud Players	IaaS	PaaS	SaaS
IT administrators/cloud providers	Monitor SLAs	Monitor SLAs and enable service platforms	Monitor SLAs and deploy software
Software developers (vendors)	To deploy and store data	Enabling platforms via configurators and APIs	Develop and deploy software
End users or business users	To deploy and store data	To develop and test web software	Use business software

- Part-I ends here