Cloud Platform Architecture Over Virtualized Datacenters

Distributed and Cloud Systems (SICT 4401)

Dr. Eng. Rebhi Baraka (rbaraka@iugaza.edu.ps)
Faculty of Information Technology
Islamic University of Gaza

Outline

- Cloud Computing and Service Models.
- Data-Center Design and Interconnection Networks.
- Architectural Design of Compute and Storage Clouds.
- Public Cloud Platforms: Google Cloud Platform and GAE.
- Inter-cloud Resource Management.
- Cloud Security and Trust Management.

CLOUD COMPUTING AND SERVICE MODELS

- Developers of innovative cloud applications no longer acquire large capital equipment in advance.
  - They just rent the resources from some large data centers that have been automated for this purpose.
- Virtualized cloud platforms are often built on top of large data centers. These are formed based on large server clusters interconnected with high speed networks.
- Clouds forms the bases of current data centers by architecting them as virtual resources over automated hardware, databases, user interfaces, and application environments.
  - Clouds grow out of the desire to build better data centers through automated resource provisioning.

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 pro- vides 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 and create business value by outsourcing job execution to cloud providers.
  - the computations (programs) are sent to where the data is located, rather than copying the data to millions of desktops as in the traditional approach.
- Cloud computing avoids large data movement, resulting in much better network bandwidth utilization.
- Machine virtualization has enhanced resource utilization, increased application flexibility, and reduced the total cost of using virtualized data-center resources.

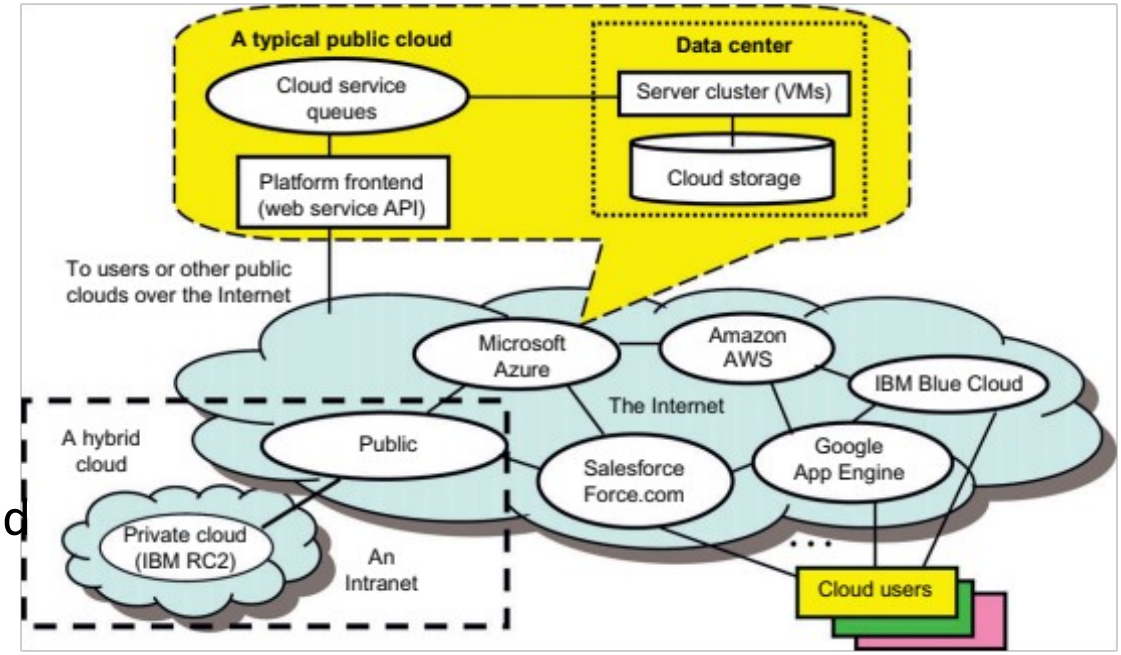
Public, Private, and Hybrid Clouds

- The cloud offers significant benefit to IT companies by freeing them from the low-level task of setting up the hardware (servers) and managing the system software.
- Cloud computing applies a virtual platform with elastic resources put together by on-demand provisioning of hardware, software, and data sets, dynamically.
- The main idea is to move desktop computing to a service-oriented platform using server clusters and huge databases at data centers.
- Cloud computing leverages its low cost and simplicity to both providers and users.

Public, Private, and Hybrid Clouds

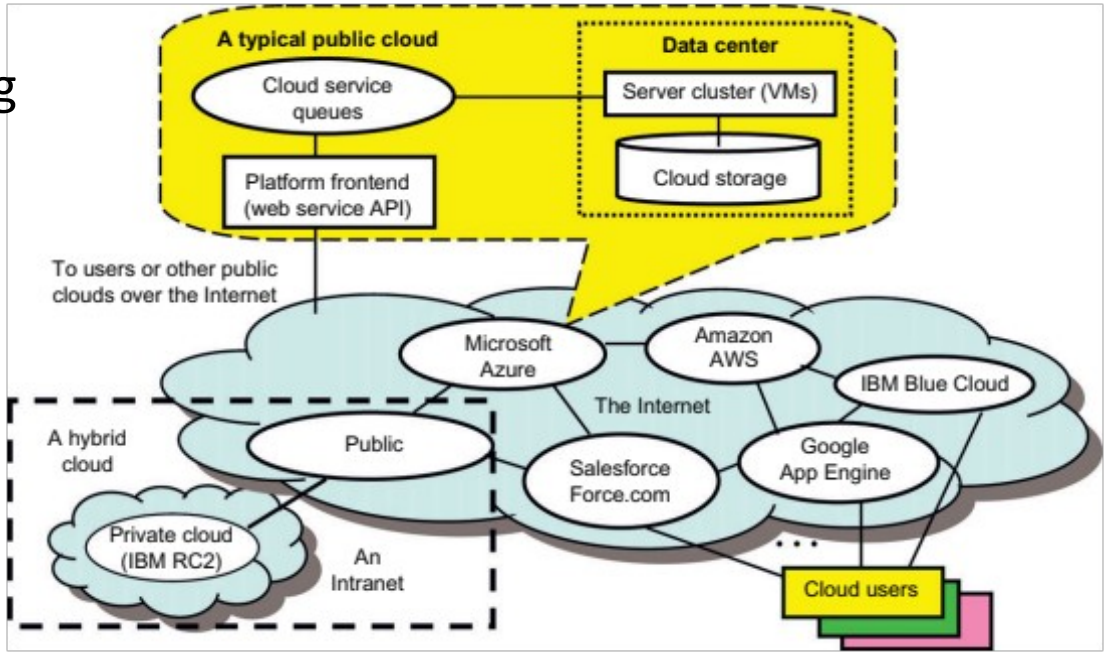
- All computations in cloud applications are distributed to servers in a data center.
  - These are mainly virtual machines (VMs) in virtual clusters created out of data-center resources.
  - Cloud platforms are systems distributed through virtualization.

- Both public clouds and private clouds are developed in the Internet.
- As many clouds are generated by commercial providers or by enterprises in a distributed manner, they will be interconnected over the Internet to achieve scalable and efficient computing services.



Public, Private, and Hybrid Clouds

- 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.
- Intranet-based private clouds are linked to public clouds to get additional resources.
- Nevertheless, users in Europe may not feel comfortable using clouds in the United States, and vice versa, until extensive service-level agreements (SLAs) are developed between the two user communities.

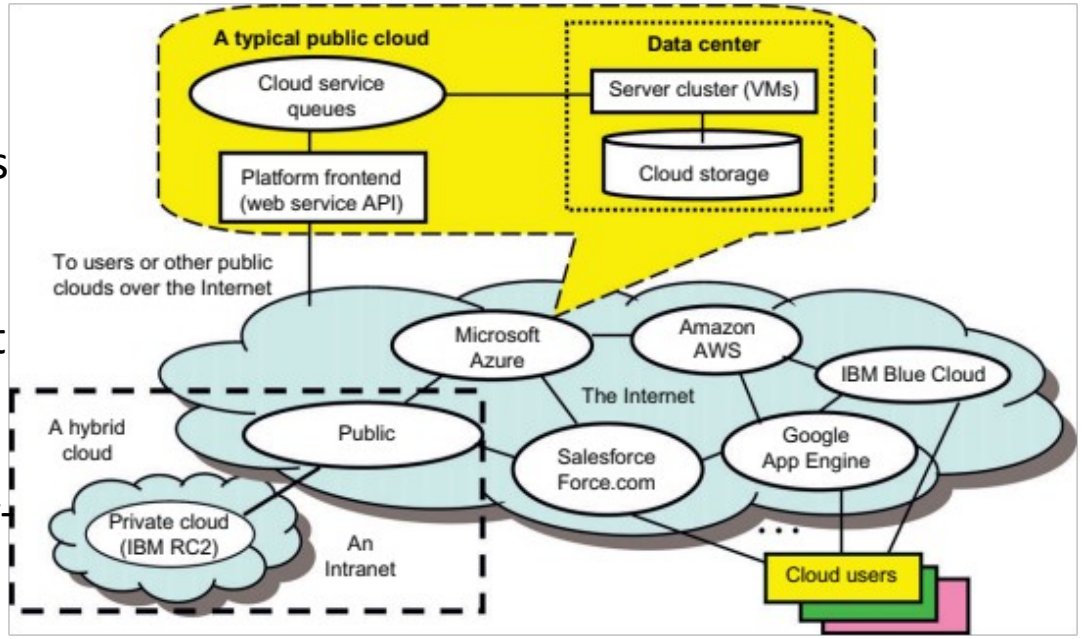


Public, Private, and Hybrid Clouds

Public Clouds

- 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 the figure 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.

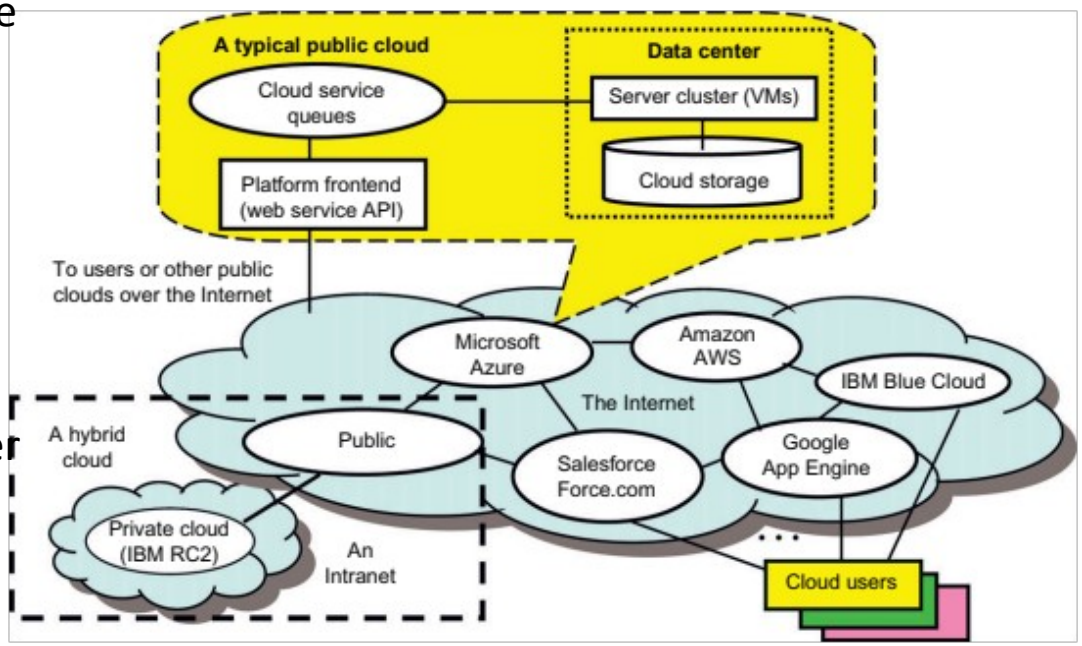
- The providers are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure.
- A public cloud delivers a selected set of business processes.
- The application and infrastructure services are offered on a flexible pay-per-use basis.



Public, Private, and Hybrid Clouds

Private Clouds

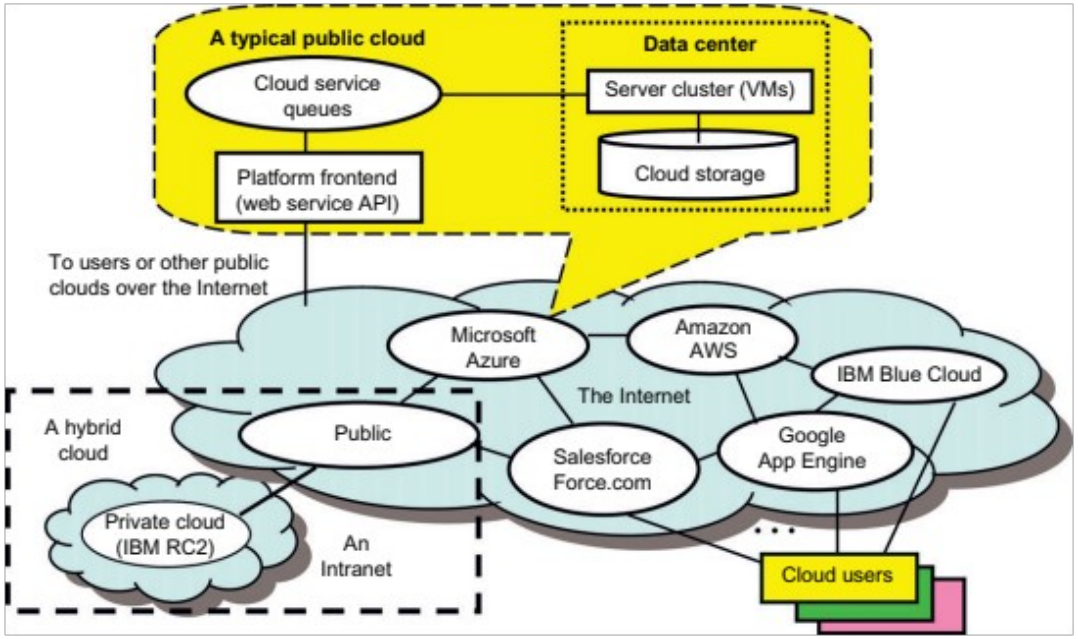
- A private cloud is built within the domain of an intranet owned by a single organization.
  - It is client owned and managed, and its access is limited to the owning clients and their partners.
  - Its deployment was not meant to sell capacity over the Internet through publicly accessible interfaces.
- Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains.
- 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.



Public, Private, and Hybrid Clouds

Hybrid Clouds

- Built with both public and private clouds.
- 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.
- Public clouds promote standardization, preserve capital investment, and offer application flexibility.
- Private clouds attempt to achieve customization and offer higher efficiency, resiliency, security, and privacy.
- Hybrid clouds operate in the middle, with many compromises in terms of resource sharing.



Public, Private, and Hybrid Clouds

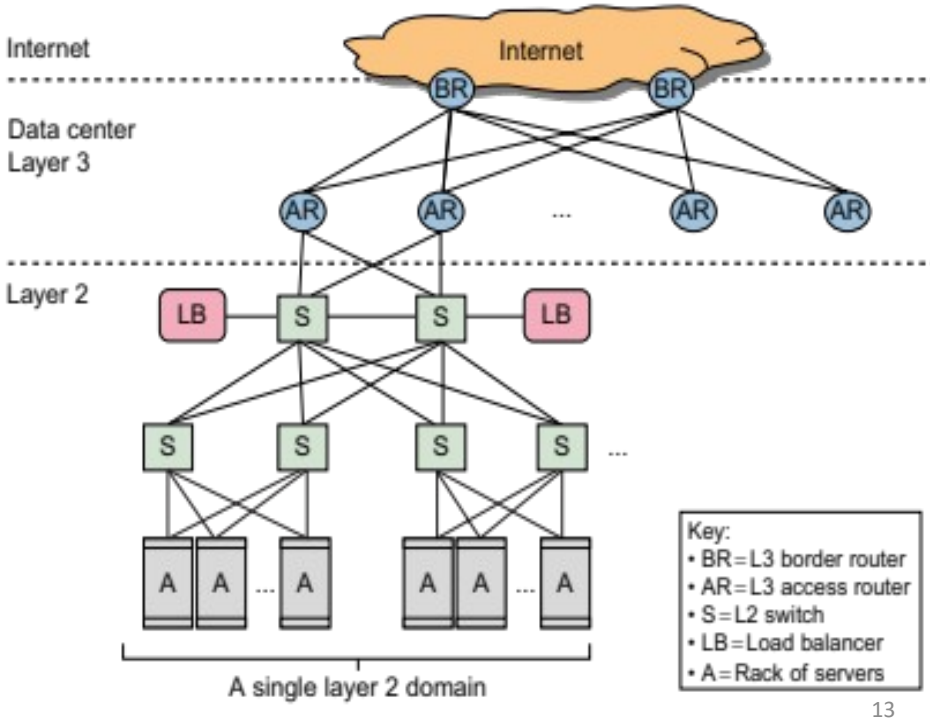
Data Center Networking Structure

- The core of a cloud is the server cluster (or VM cluster).
- Cluster nodes are used as compute nodes.
- A few control nodes are used to manage and monitor cloud activities.
- The scheduling of user jobs requires that you assign work to virtual clusters created for users.
- 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 centers scaling is a fundamental requirement.
- Data center server clusters are typically built with large number of servers, ranging from thousands to millions of servers (nodes).
  - For example, Microsoft has a data center in the Chicago area that has 100,000 eight-core servers, housed in 50 containers.
  - Uses disks on server nodes plus memory cache and databases.

Public, Private, and Hybrid Clouds

Data Center Networking Structure

- Data-center networks are mostly IP-based commodity networks, such as the 10 Gbps Ethernet network, which is optimized for Internet access.
- The figure 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).



Public Clouds vs. Private Clouds:

Characteristics	Public clouds	Private clouds
Technology leverage and ownership	Owned by service providers	Leverage existing IT infrastructure and personnel; owned by individual organization
Management of provisioned resources	Creating and managing VM instances within proprietary infrastructure; promote standardization, preserves capital investment, application flexibility	Client managed; achieve customization and offer higher efficiency
Workload distribution methods and loading policies	Handle workload without communication dependency; distribute data and VM resources; surge workload is off-loaded	Handle workload dynamically, but can better balance workloads; distribute data and VM resources
Security and data privacy enforcement	Publicly accessible through remote interface	Access is limited; provide pre-production testing and enforce data privacy and security policies
Example platforms	Google App Engine, Amazon AWS, Microsoft Azure	IBM RC2

Cloud Ecosystem and Enabling Technologies

- Cloud computing platforms differ from conventional computing platforms in computing paradigms and cost models applied.

Traditional computing model	Cloud computing paradigm
• Involves buying the hardware, acquiring the necessary system software, installing the system, testing the configuration and executing the application code and management of resources. • This cycle repeats itself in about every 18 months, meaning the machine we bought becomes obsolete every 18 months.	• Follows a pay- as-you-go model. Therefore the cost is significantly reduced, because we simply rent computer resources without buying the computer in advance. • All hardware and software resources are leased from the cloud provider without capital investment on the part of the users. Only the execution phase costs some money.

Classical Computing	**Cloud Computing**
(Repeat the following cycle every 18 months)	(Pay as you go per each service provided)
Buy and own	**Subscribe**
Hardware, system software, applications to meet peak needs	- - - -
Install, configure, test, verify, evaluate, manage	**Use** (Save about 80-95% of the total cost)
- - - -	- - - -
Use	(Finally)
- - - -	**$ - Pay for what you use**
Pay $$$$$ (High cost)	based on the QoS

Cloud Ecosystem and Enabling Technologies

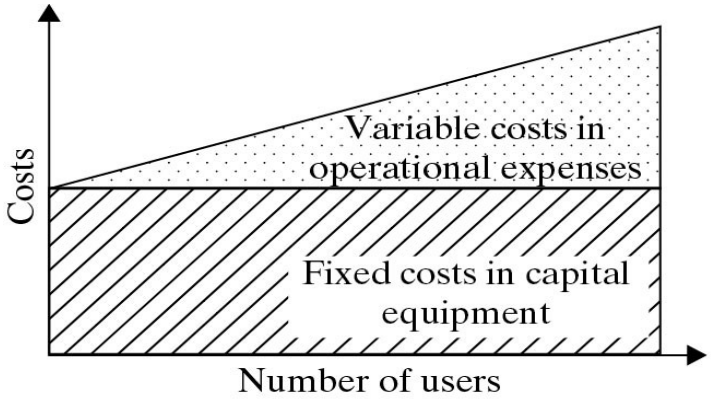
Cloud Design Objectives

- **Shifting computing from desktops to data centers** Computer processing, storage, and software delivery is shifted away from desktops and local servers and toward data centers over the Internet.
- **Service provisioning and cloud economics** Providers supply cloud services by signing SLAs with consumers and end users. The services must be efficient in terms of computing, storage, and power consumption. Pricing is based on a pay-as-you-go policy.
- **Scalability in performance** The cloud platforms and software and infrastructure services must be able to scale in performance as the number of users increases.
- **Data privacy protection** Can you trust data centers to handle your private data and records? This concern must be addressed to make clouds successful as trusted services.
- **High quality of cloud services** The QoS of cloud computing must be standardized to make clouds interoperable among multiple providers.
- **New standards and interfaces** This refers to solving the data lock-in problem associated with data centers or cloud providers. Universally accepted APIs and access protocols are needed to provide high portability and flexibility of virtualized applications.

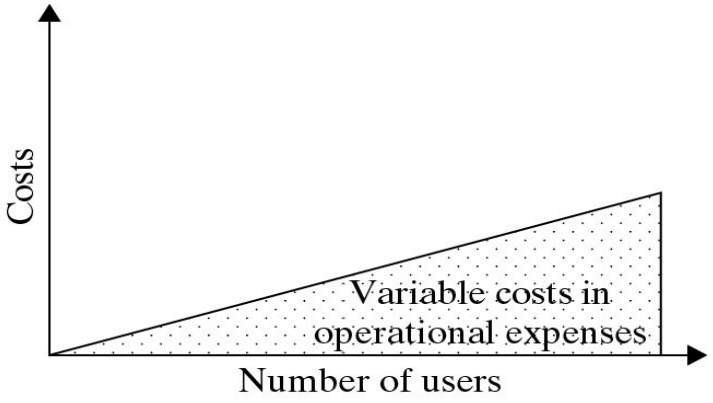
Cloud Ecosystem and Enabling Technologies

Cost Model

- The fixed cost is the main cost, and that it could be reduced slightly as the number of users increases.
  - Operational costs may increase sharply with a larger number of users.
  - The total cost escalates quickly with massive numbers of users.
- Cloud computing applies a pay-per-use business model, in which user jobs are outsourced to data centers.
  - No up-front cost in hardware acquisitions. Only variable costs are experienced by cloud users.



(a) Traditional IT cost model



(b) Cloud computing cost model

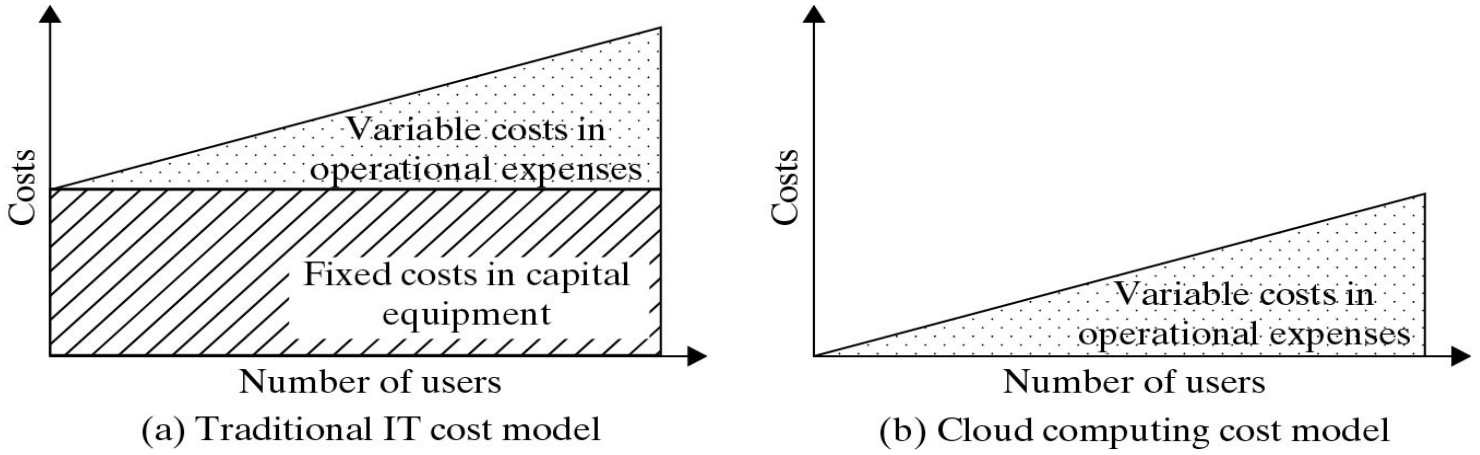
Cloud Ecosystem and Enabling Technologies

Cost Model

- Cloud computing will reduce computing costs significantly for both small users and large enterprises.
- Cloud users only pay for operational expenses and do not have to invest in permanent equipment.
  - This is attractive to massive numbers of small users.

Cost-Effectiveness in Cloud Computing vs. Datacenter Utilization

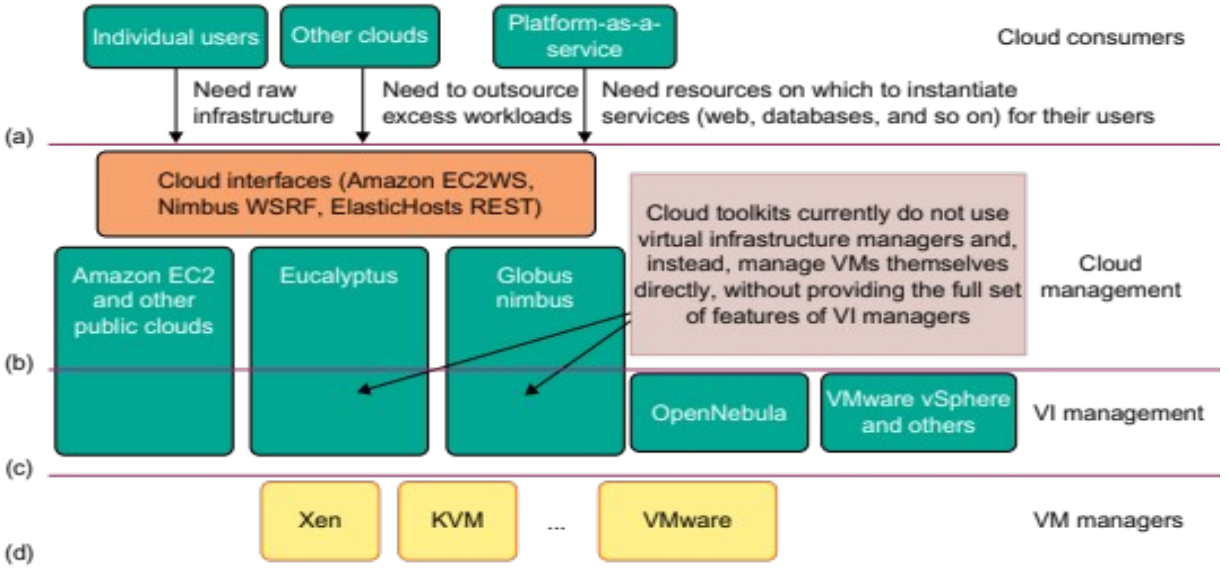
$$\text{UserHours}_{\text{cloud}} \times (\text{revenue} - \text{Cost}_{\text{cloud}}) \geq \text{UserHours}_{\text{datacenter}} \times \left(\text{revenue} - \frac{\text{Cost}_{\text{datacenter}}}{\text{Utilization}}\right)$$



Cloud Ecosystem and Enabling Technologies

Cloud Ecosystems (for private clouds)

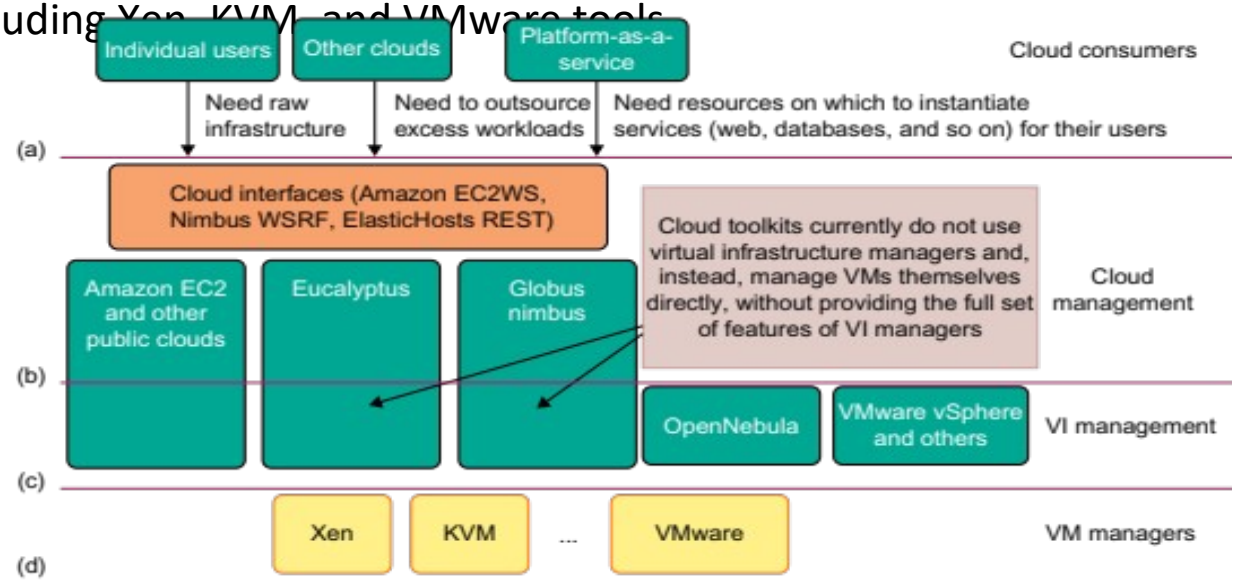
- Four levels of ecosystem development in a private cloud.
  - At the user end, consumers demand a flexible platform.
  - At the cloud management level, the cloud manager provides virtualized resources over an IaaS platform.
  - At the virtual infrastructure (VI) management level, the manager allocates VMs over multiple server clusters.
  - Finally, 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.
  - Integrating these two layers is complicated by the lack of open and standard interfaces between them.



Cloud Ecosystem and Enabling Technologies

Cloud Ecosystems (for private clouds)

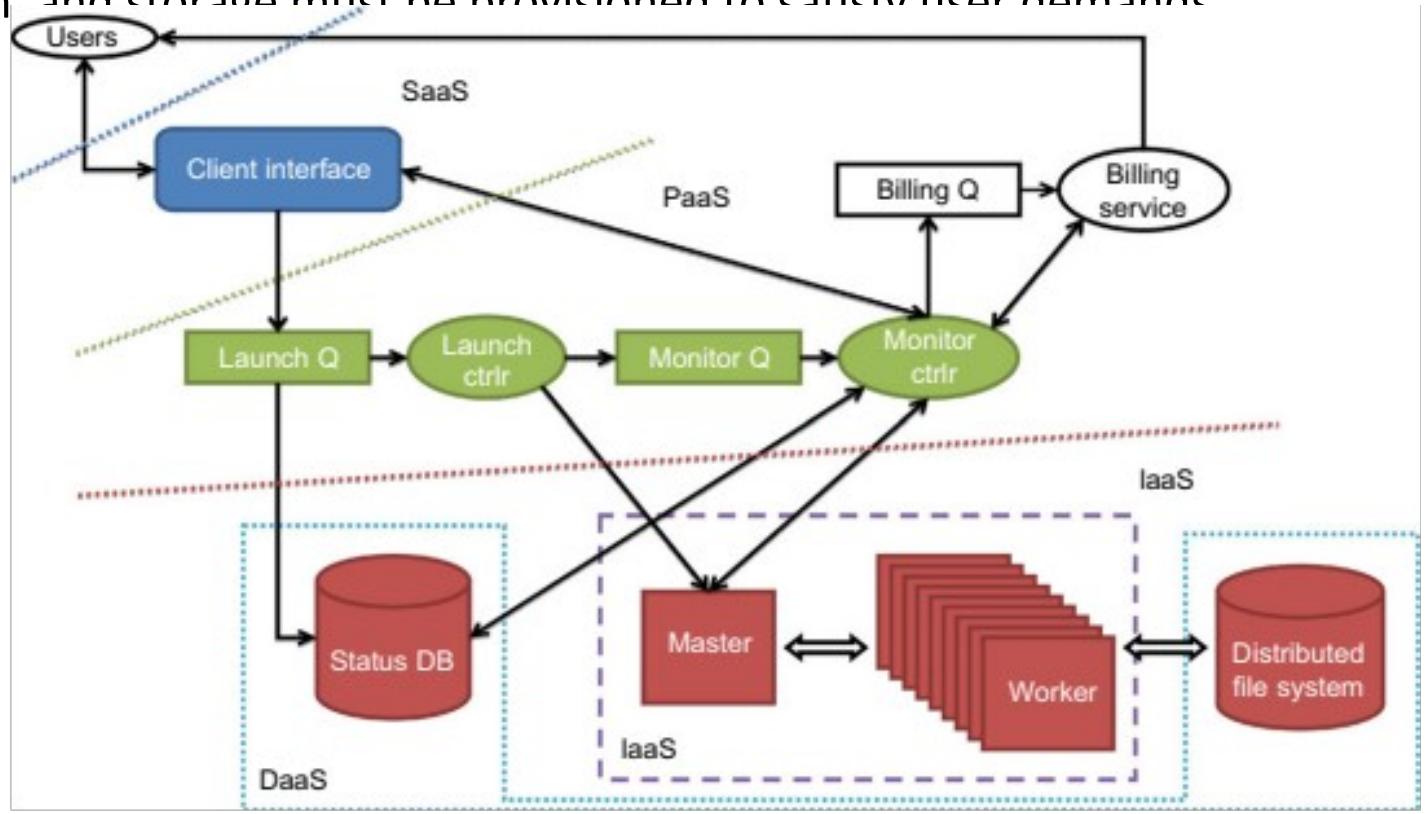
- VI management tools support dynamic placement and VM management on a pool of physical resources, automatic load balancing, server consolidation, and dynamic infrastructure resizing and partitioning.
- In addition to public clouds such as Amazon EC2, Eucalyptus and Globus Nimbus are open source tools for virtualization of cloud infrastructure.
  - To access these cloud management tools, one can use the Amazon EC2WS, Nimbus WSRF, and ElasticHost REST cloud interfaces.
  - For VI management, OpenNebula and VMware vSphere can be used to manage all VM generation including Xen, KVM, and VMware tools.



Infrastructure as a Service (IaaS)

Three cloud models at different service levels of the cloud.

- SaaS is applied at the application end using special interfaces by users or clients.
- At the PaaS layer, the cloud platform must perform billing services and handle job queuing, launching, and monitoring services.
- At the bottom layer of the IaaS services, databases, compute instances, the file system, and storage must be provisioned to satisfy user demands.



Infrastructure as a Service (IaaS)

- Most basic cloud service model.
- Cloud providers offer computers, as physical or more often as virtual machines, and other resources.
- Virtual machines are run as guests by a hypervisor, such as Xen or KVM.
- Cloud users deploy their applications by then installing operating system images on the machines as well as their application software.
- Cloud providers typically bill IaaS services on a utility computing basis, that is, cost will reflect the amount of resources allocated and consumed.
- Examples of IaaS include: Amazon CloudFormation (and underlying services such as Amazon EC2), Rackspace Cloud, Terremark, and Google Compute Engine.

Infrastructure as a Service (IaaS)

Some IaaS Offerings from Public Clouds:

Table 4.1 Public Cloud Offerings of IaaS [10,18]

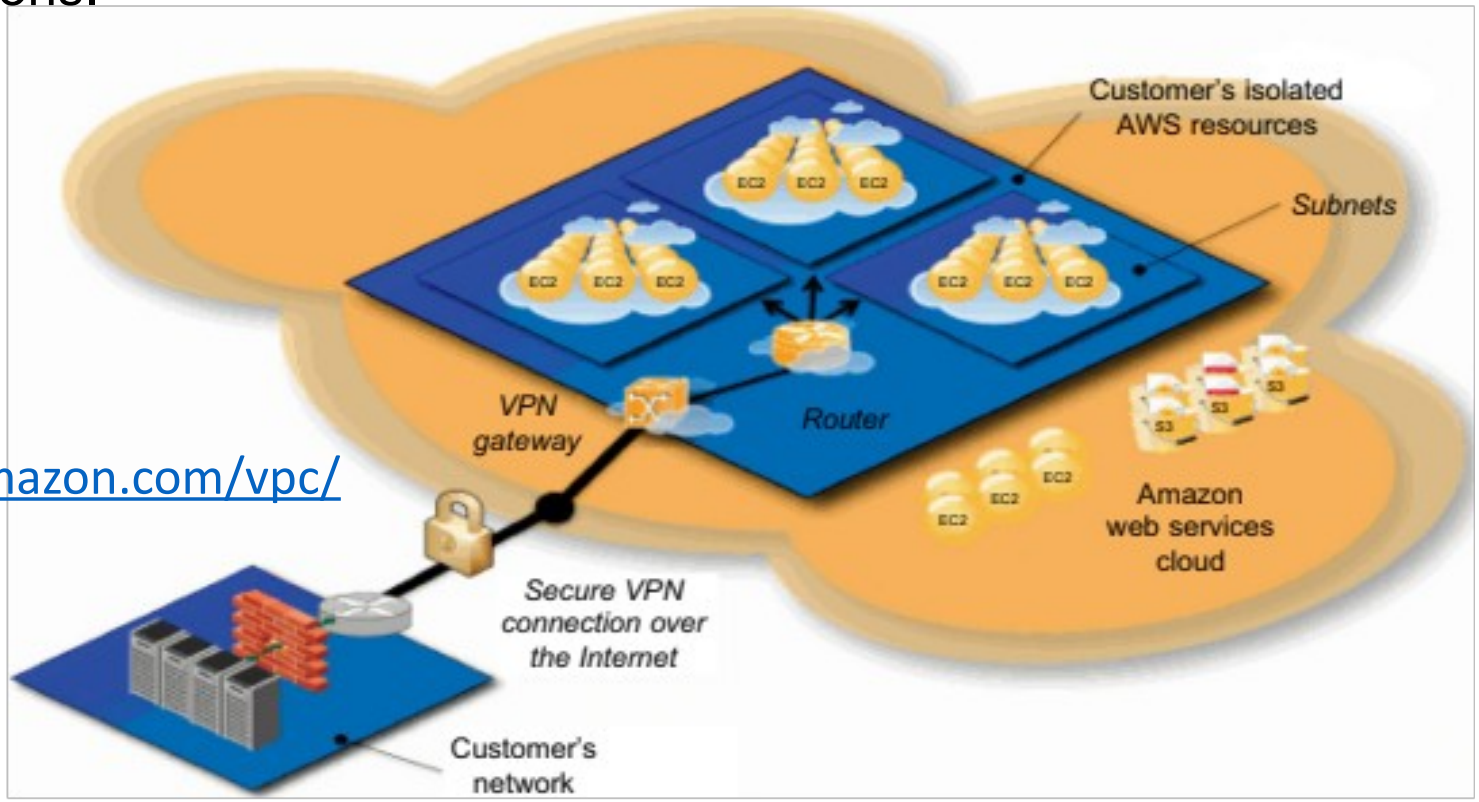
Cloud Name	VM Instance Capacity	API and Access Tools	Hypervisor, Guest OS
Amazon EC2	Each instance has 1–20 EC2 processors, 1.7–15 GB of memory, and 160–1.69 TB of storage.	CLI or Web Service (WS) portal	Xen, Linux, Windows
GoGrid	Each instance has 1–6 CPUs, 0.5–8 GB of memory, and 30–480 GB of storage.	REST, Java, PHP, Python, Ruby	Xen, Linux, Windows
Rackspace Cloud	Each instance has a four-core CPU, 0.25–16 GB of memory, and 10–620 GB of storage.	REST, Python, PHP, Java, C#, .NET	Xen, Linux
FlexiScale in the UK	Each instance has 1–4 CPUs, 0.5–16 GB of memory, and 20–270 GB of storage.	Web console	Xen, Linux, Windows
Joyent Cloud	Each instance has up to eight CPUs, 0.25–32 GB of memory, and 30–480 GB of storage.	No specific API, SSH, Virtual/Min	OS-level virtualization, OpenSolaris

Infrastructure as a Service (IaaS)

Example1: Amazon Virtual Private Cloud (VPC) for Multiple Tenants

- A private cloud for basic computations
- It provides EC2 instances and/or storage (S3) for computations and applications.

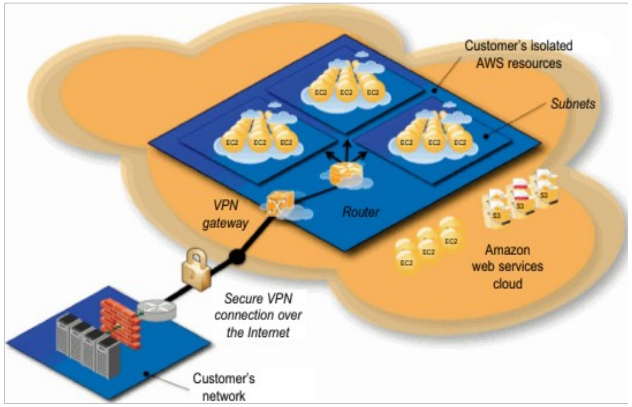
https://aws.amazon.com/vpc/



Infrastructure as a Service (IaaS)

Example 1: Amazon Virtual Private Cloud (VPC) for Multiple Tenants

- Amazon EC2 provides the following services:
  - Resources from multiple data centers globally distributed,
  - Web services (SOAP and Query),
  - Web-based console user interfaces,
  - Access to VM instances
  - Per-hour pricing,
  - Linux and Windows OSes, and
  - automatic scaling and load balancing.

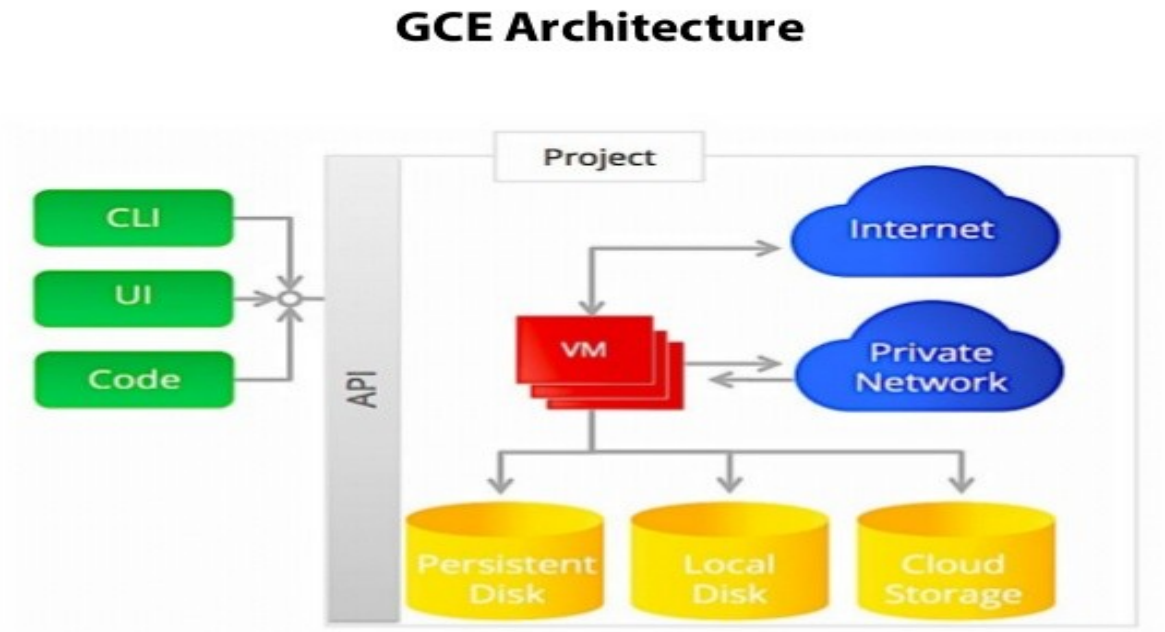


- VPC allows auto-scaling and elastic load balancing services to support related demands.

Infrastructure as a Service (IaaS)

Example 2: Google Compute Engine

https://cloud.google.com/compute



Platform as a service (PaaS)

- Cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server.
- Application developers develop and run their software on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.
- Examples of PaaS include: Amazon Elastic Beanstalk, Cloud Foundry, Heroku, Force.com, EngineYard, Mendix, Google App Engine, Microsoft Azure and OrangeScape.

Platform as a service (PaaS)

PaaS Offerings from Public Clouds:

Table 4.2 Five Public Cloud Offerings of PaaS [10,18]

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)

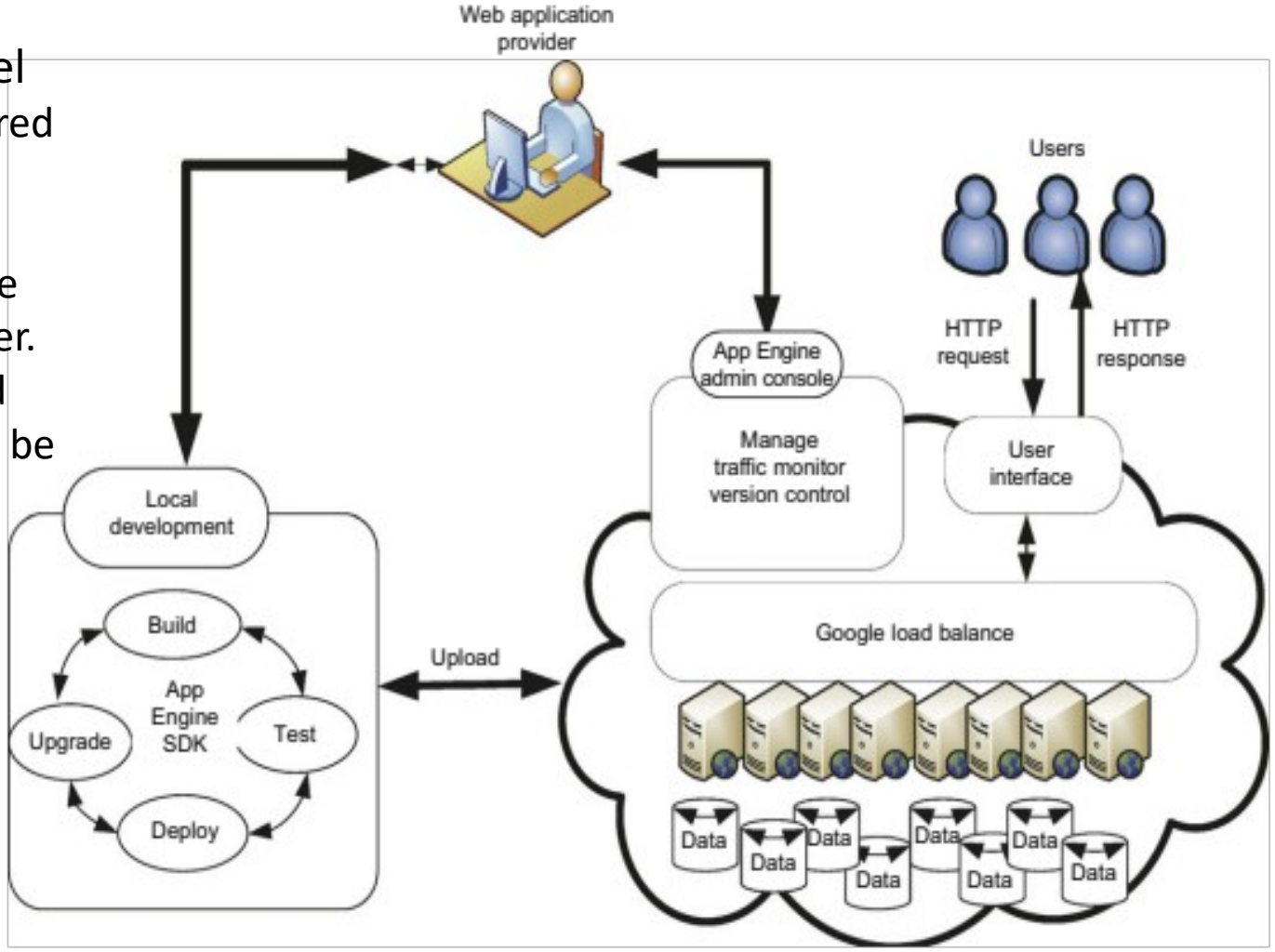
Example: Google App Engine for PaaS Applications

- Allows web applications running on Google's server clusters to share the same capability with many other users.
- The applications have features such as automatic scaling and load balancing which are very convenient while building web applications.
- A distributed scheduler mechanism schedules tasks for triggering events at specified times and regular intervals.
- GAE provides a development environment.
  - Coding and debugging stages can be performed locally as

Platform as a service (PaaS)

Example: Google App Engine for PaaS Applications

- Operational model for GAE Fully featured local development environment that simulates GAE on the developer's computer.
- All the functions and application logic can be implemented locally which similar to traditional software development.



Software as a service (SaaS)

- Cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients.
- The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so price is scalable and adjustable if users are added or removed at any point.
- Examples of SaaS include:
  - Google Apps, innkeypos, Quickbooks Online, Limelight Video Platform, Salesforce.com, and Microsoft Office 365.
  - CRM software from Salesforce.com
  - Google Gmail and docs, Microsoft SharePoint,
  - Etc.

DATA-CENTER DESIGN AND INTERCONNECTION NETWORKS

- A data center is often built with a large number of servers through a huge interconnection network. There are:
  - Large-scale data centers and
  - Small modular data centers that can be housed in a 40-ft truck container.
- Issue: Interconnection of modular data centers and their management.

Warehouse-Scale Data-Center Design

- The cloud is built on massive datacenters
- It can house 400,000 to several million servers.
  - Built economics of scale— meaning lower unit cost for larger data centers.
  - A small data center could have 1,000 servers. The larger the data center, the lower the operational cost.
- The approximate monthly cost to operate a huge 400-server data center is estimated by:
  - network cost $13/Mbps;
  - storage cost $0.4/GB; and
  - administration costs.

Warehouse-Scale Data-Center Design

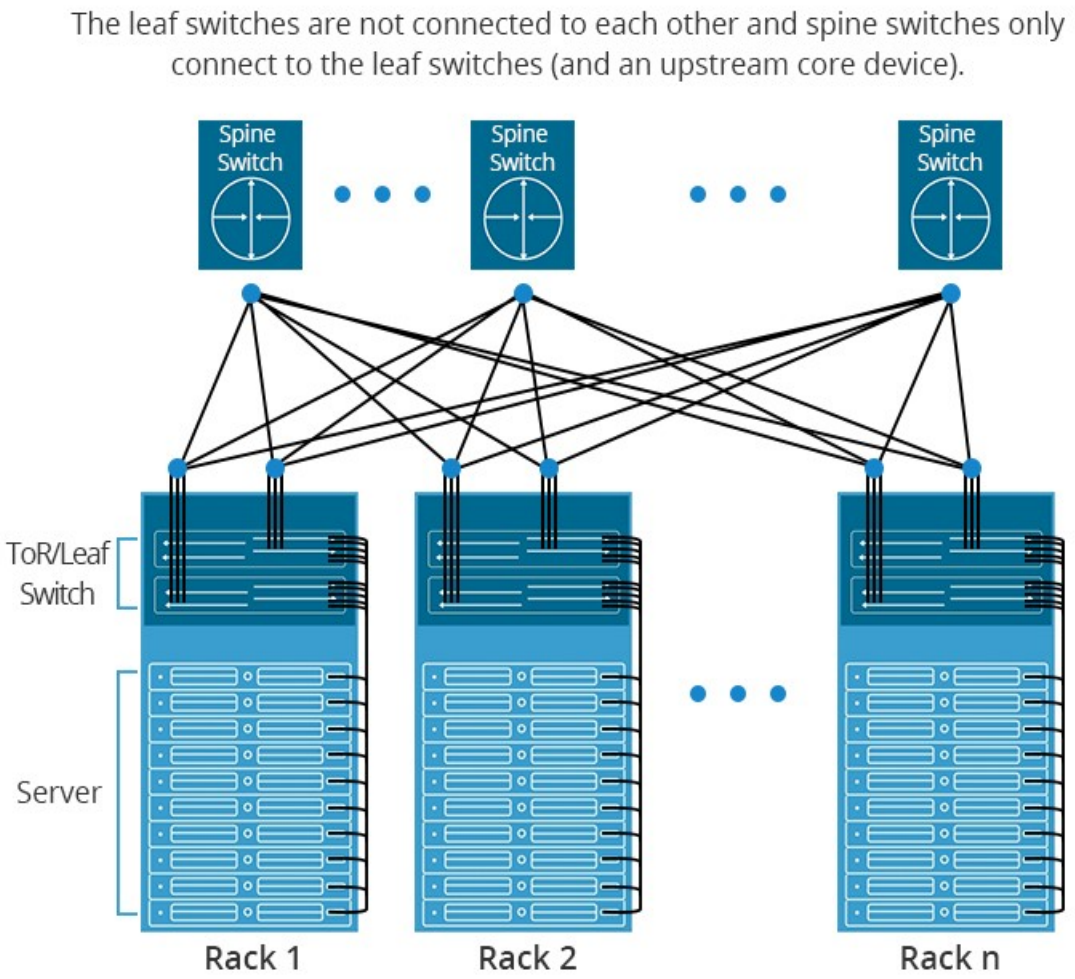
- These unit costs are greater than those of a 1,000-server data center.
- Microsoft, e.g., has about 100 data centers, large or small, which are distributed around the globe.
- Provides Internet services
  - Search, social networking, online maps, video sharing, online shopping, email, cloud computing, etc.
- Datacenters consolidate different machines and software into one location
- Datacenters emphasize virtual machines and hardware heterogeneity in order to serve varied customers

Warehouse-Scale Data-Center Design: Construction Requirements

- Most data centers are built with commercially available components.
- An off-the-shelf server consists of a number of processor sockets, each with a multicore CPU and its internal cache hierarchy, local shared and coherent DRAM, and a number of directly attached disk drives.
- The DRAM and disk resources within the rack are accessible through first-level rack switches and all resources in all racks are accessible via a cluster-level switch.

Warehouse-Scale Data-Center Design: Construction Requirements

- Consider a data center built with 2,000 servers, each with 8 GB of DRAM and four 1 TB disk drives.
- Each group of 40 servers is connected through a 1 Gbps link to a rack-level switch that has an additional eight 1 Gbps ports used for connecting the rack to the cluster-level switch.



Warehouse-Scale Data-Center Design: Construction Requirements

- Approximately, the bandwidth available from local disks is 200 MB/s, whereas the band- width from off-rack disks is 25 MB/s via shared rack uplinks.
- The total disk storage in the cluster is almost 10 million times larger than local DRAM.
- A large application must deal with large discrepancies in latency, bandwidth, and capacity.
- In a very large-scale data center, components are relatively cheaper.
- The components used in data centers are very different from those in building supercomputer systems.

Warehouse-Scale Data-Center Design: Construction Requirements

- With a scale of thousands of servers, concurrent failure, either hardware failure or software failure, of 1 percent of nodes is common.
- Many failures can happen in hardware; for example, CPU failure, disk I/O failure, and network failure.
- It is even quite possible that the whole data center does not work in the case of a power crash.
- Also, some failures are brought on by software.
  - The service and data should not be lost in a failure situation.
- Reliability can be achieved by redundant hardware.
  - The software must keep multiple copies of data in different locations and keep the data accessible while facing hardware or software errors.

Typical Datacenter Layout

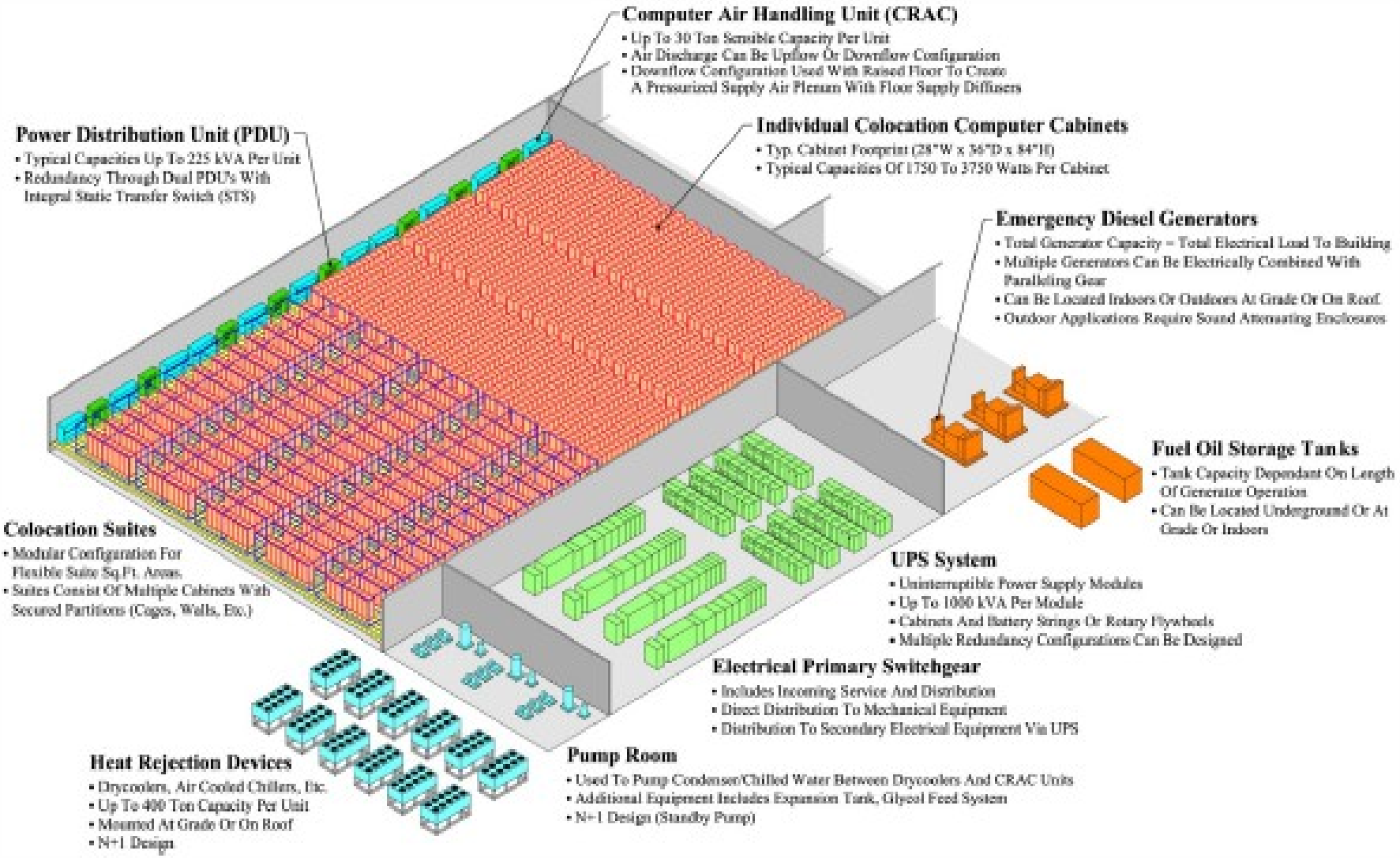


FIGURE 4.1: The main components of a typical datacenter (image courtesy of DLB Associates [23]).

Warehouse-Scale Data-Center Design: Cooling System of a Data-Center Room

- 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.

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

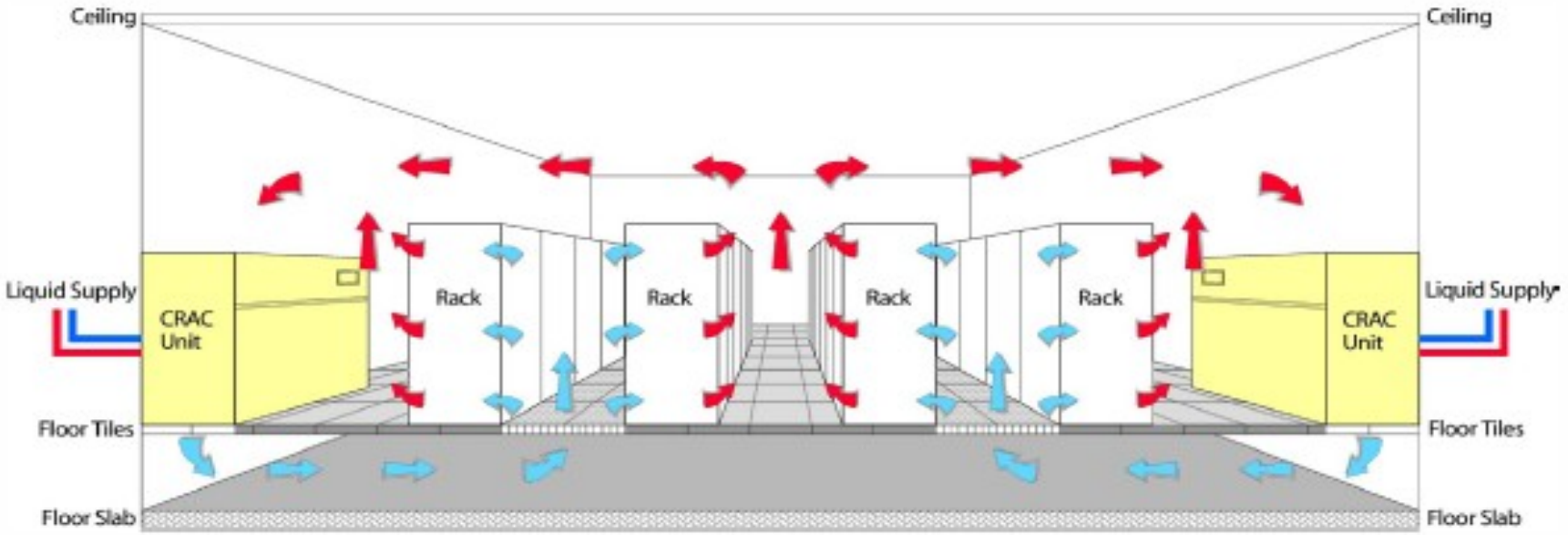


FIGURE 4.2: Datacenter raised floor with hot–cold aisle setup (image courtesy of DLB Associates [23]).

Power and Cooling Requirements

- Cooling system also uses water (evaporation and spills)
  - E.g. 70,000 to 200,000 gallons per day for an 8 MW facility

- Power cost breakdown:
  - Chillers (coolers): 30-50% of the power used by the IT equipment
  - Air conditioning: 10-20% of the IT power, mostly due to fans

Measuring Efficiency

- Power Utilization Effectiveness (PEU)
  - = Total facility power / IT equipment power
  - Median PUE on 2006 study was 1.69
- Performance
  - Latency is important metric because it is seen by users
  - Bing study: users will use search less as response time increases
  - Service Level Objectives (SLOs)/Service Level Agreements (SLAs)
    - E.g. 99% of requests be below 100 ms

$$\text{Efficiency} = \frac{\text{Computation}}{\text{Total Energy}} = \underbrace{\left(\frac{1}{\text{PUE}}\right)}_{\text{(a)}} \times \underbrace{\left(\frac{1}{\text{SPUE}}\right)}_{\text{(b)}} \times \underbrace{\left(\frac{\text{Computation}}{\text{Total Energy to Electronic Components}}\right)}_{\text{(c)}}$$

Data-Center Interconnection Networks

- Interconnection among all servers in the data- center cluster.
- This network design must meet five special requirements:
  - low latency,
  - high bandwidth,
  - low cost,
  - message-passing interface (MPI) communication support, and
  - fault tolerance.
- The design of an inter-server network must satisfy both point-to-point and collective communication patterns among all server nodes.
- Specific design considerations are given next.

Data-Center Interconnection Networks: Application Traffic Support

- The network topology should support all communication patterns.
  - Both point-to-point and collective communications must be supported.
  - The network should have high bisection bandwidth to meet this requirement.
  - For example, one-to-many communications are used for supporting distributed file access.
  - One can use one or a few servers as metadata master servers which need to communicate with slave server nodes in the cluster.
- To support the MapReduce programming paradigm, the network must be designed to perform the map and reduce functions at a high speed.
- The underlying network structure should support various network traffic patterns demanded by user applications.

Data-Center Interconnection Networks: 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.
- Implemented with low-cost Ethernet switches.

Data-Center Interconnection Networks: Network Expandability

- The most critical issue regarding expandability is support of modular network growth for building data-center containers.
- One single data-center container contains hundreds of servers and is considered to be the building block of large-scale data centers.
- The network interconnection should be established among many containers
  - Cluster networks need to be designed for data-center containers.
  - Cable connections are then needed among multiple data-center containers.
- Each container contains several hundred or even thousands of server nodes.
- Plug in the power supply, outside connection link, and cooling water, and the whole system will just work.
  - This is quite efficient and reduces the cost of purchasing and maintaining servers.
  - One approach is to establish the connection backbone first and then extend the backbone links to reach the end servers.
  - One can also connect multiple containers through external switching and cabling.

Data-Center Interconnection Networks: 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.

Data-Center Interconnection Networks: Fault Tolerance and Graceful Degradation

- There should be no critical paths or critical points which may become a single point of failure that pulls down the entire system.
- Most design innovations are in the topology structure of the network.
- The network structure is often divided into two layers.
  - The lower layer is close to the end servers, and
  - the upper layer establishes the backbone connections among the server groups or sub-clusters.
- This hierarchical interconnection approach appeals to building data centers with modular containers.

Data-Center Interconnection Networks

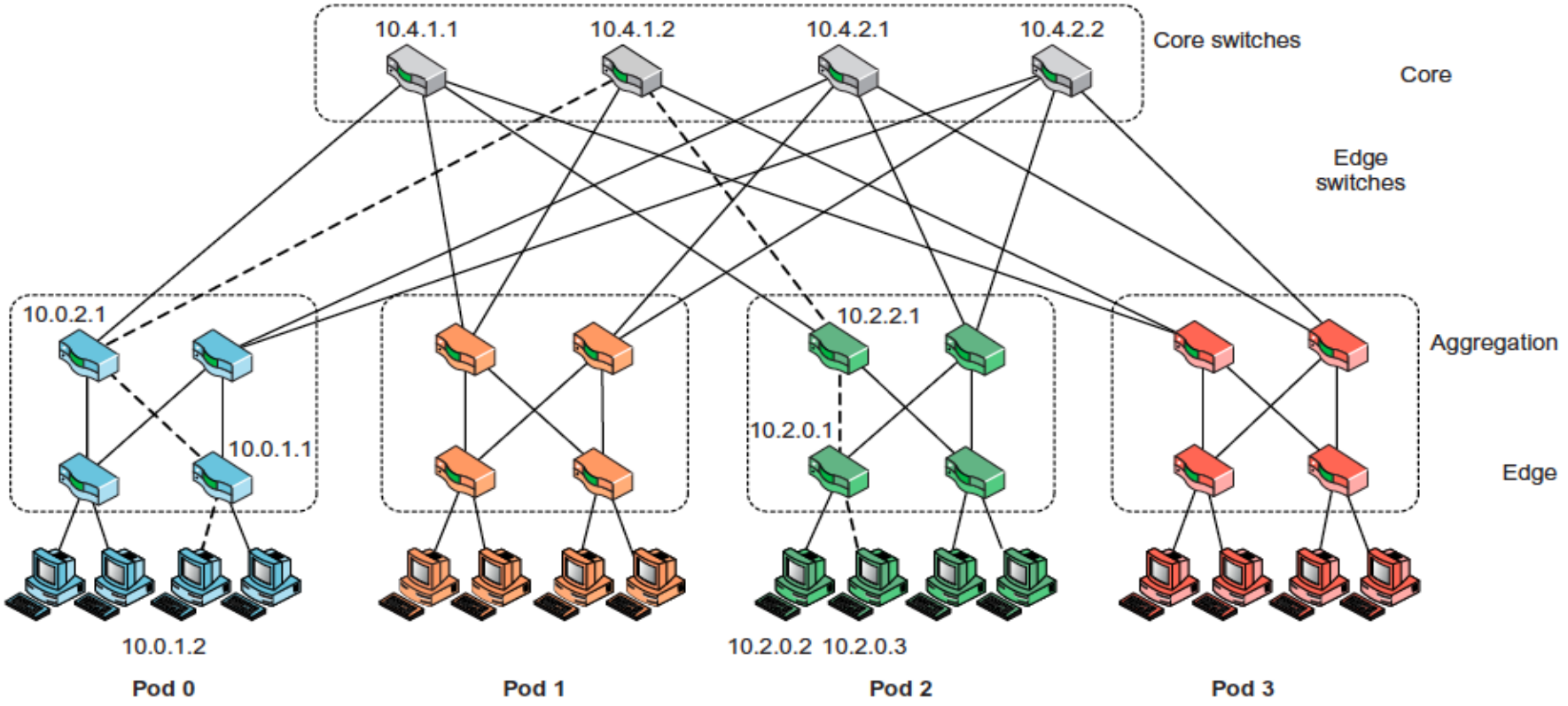
Example: A Fat-Tree Interconnection Network for Data Centers

- The fat-tree topology is applied to interconnect the server nodes.
- The topology is organized into two layers.
  - Server nodes are in the bottom layer, and
    - Edge switches are used to connect the nodes in the bottom layer.
  - The upper layer aggregates the lower-layer edge switches.
- A group of aggregation switches, edge switches, and their leaf nodes form a pod.
  - Core switches provide paths among different pods.
- The fat-tree structure provides multiple paths between any two server nodes.
- This provides fault-tolerant capability with an alternate path in case of some isolated link failures.
- The failure of any edge switch can only affect a small number of end server nodes.
  - The extra switches in a pod provide higher bandwidth to support cloud applications in massive data movement.
- Routing algorithms are built inside the switches.

Data-Center Interconnection Networks

Example: A Fat-Tree Interconnection Network for Data Centers

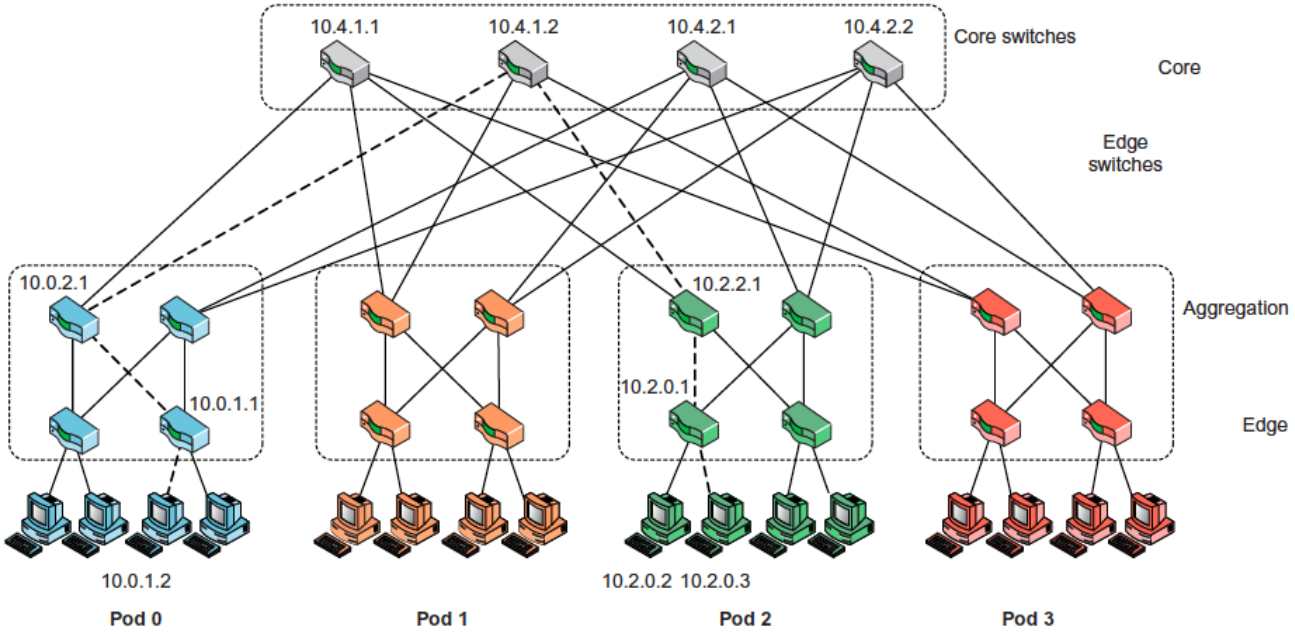
- Server nodes are in the bottom layer, and edge switches are used to connect the nodes in the bottom layer.
- The upper layer aggregates the lower-layer edge switches.
- A group of aggregation switches, edge switches, and their leaf nodes form a pod



Data-Center Interconnection Networks

Example: A Fat-Tree Interconnection Network for Data Centers

- Core switches provide paths among different pods.
- The fat-tree structure provides multiple paths between any two server nodes.
- This provides fault-tolerant capability with an alternate path in case of some isolated link failures.
- The failure of an aggregation switch and core switch will not affect the connectivity of the whole network.
- The failure of any edge switch can only affect a small number of end server nodes.

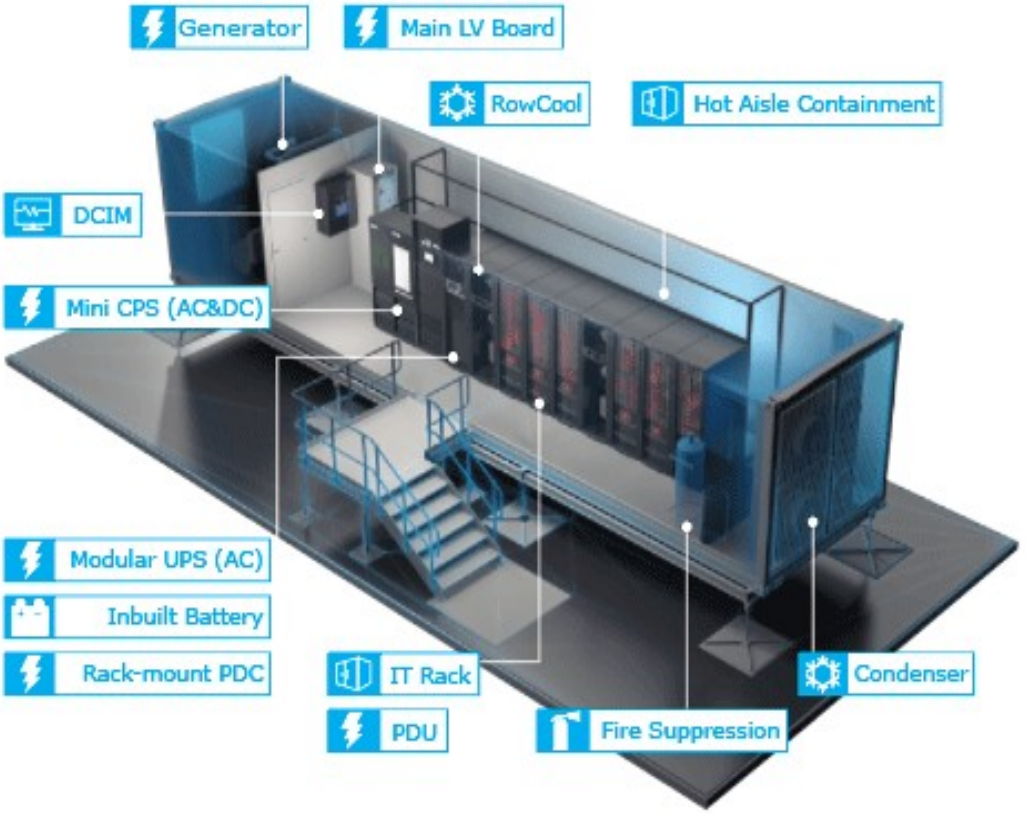


MODULAR DATA CENTER IN SHIPPING CONTAINERS

- A modern data center is structured as a shipyard of server clusters housed in truck-towed containers.
- Inside the container, hundreds of blade servers are housed in racks surrounding the container walls.
- An array of fans forces the heated air generated by the server racks to go through a heat exchanger, which cools the air for the next rack (detail in callout) on a continuous loop.
- For example: The SGI ICE Cube container can house 46,080 processing cores or 30 PB of storage per container.
- Large-scale data center built with modular containers appear as a big shipping yard of container trucks.
- This container-based data center was motivated by demand for:
  - Lower power consumption, higher computer density, and mobility to relocate data centers to better locations with lower electricity costs, better cooling water supplies, and cheaper housing for maintenance engineers.
- Sophisticated cooling technology enables up to 80% reduction in cooling costs compared with traditional warehouse data centers.

MODULAR DATA CENTER IN SHIPPING CONTAINERS

- The figure shows the housing of multiple sever racks in a truck-towed container.
- Example is the SGI ICE Cube modular data center.



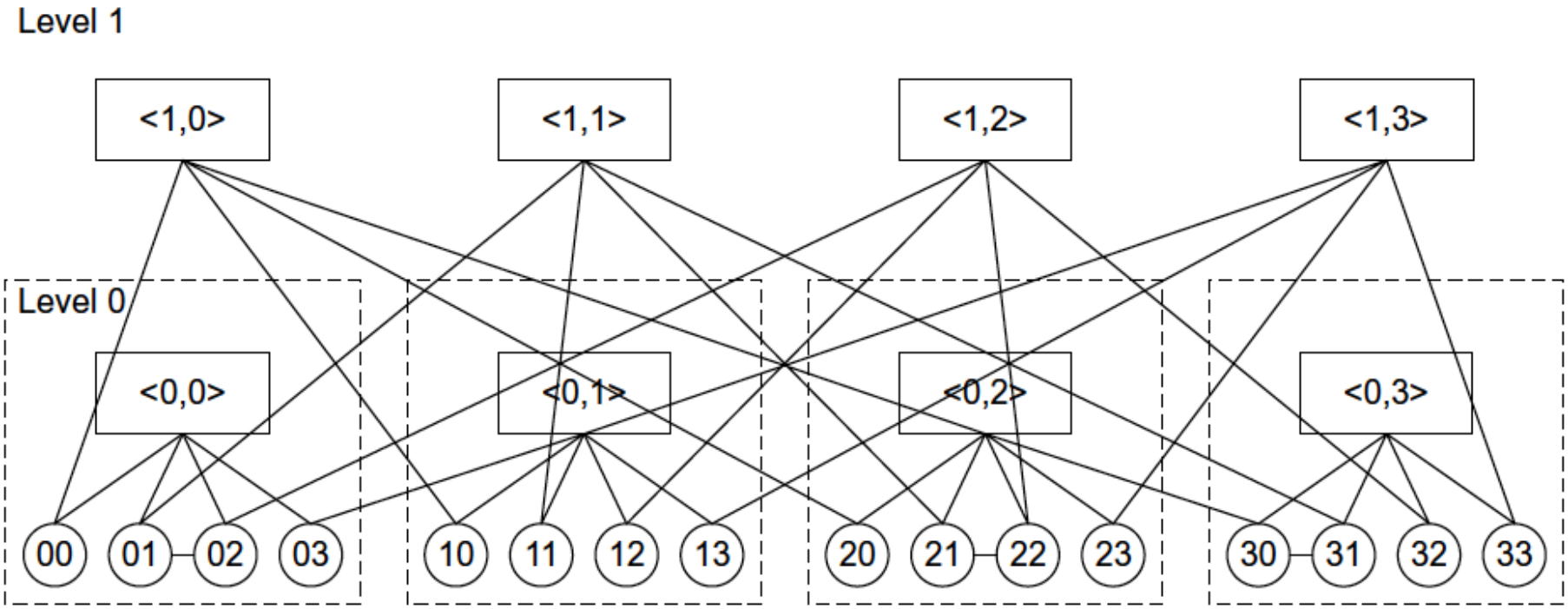
MODULAR DATA CENTER IN SHIPPING CONTAINERS

Container Data-Center Construction

- Start with one system (server), then move to a rack system design, and finally to a container system with multiple racks for 1,000 servers.
  - This requires the layout of the floor space with power, networking, cooling, and complete testing.
- The modular data-center approach supports many cloud service applications.
  - For example, the health care industry will benefit by installing a data center at all clinic sites.
- However, how to exchange information with the central database and maintain periodic consistency becomes a rather challenging design issue in a hierarchically structured data center.
- The security of collocation cloud services may involve multiple data centers.

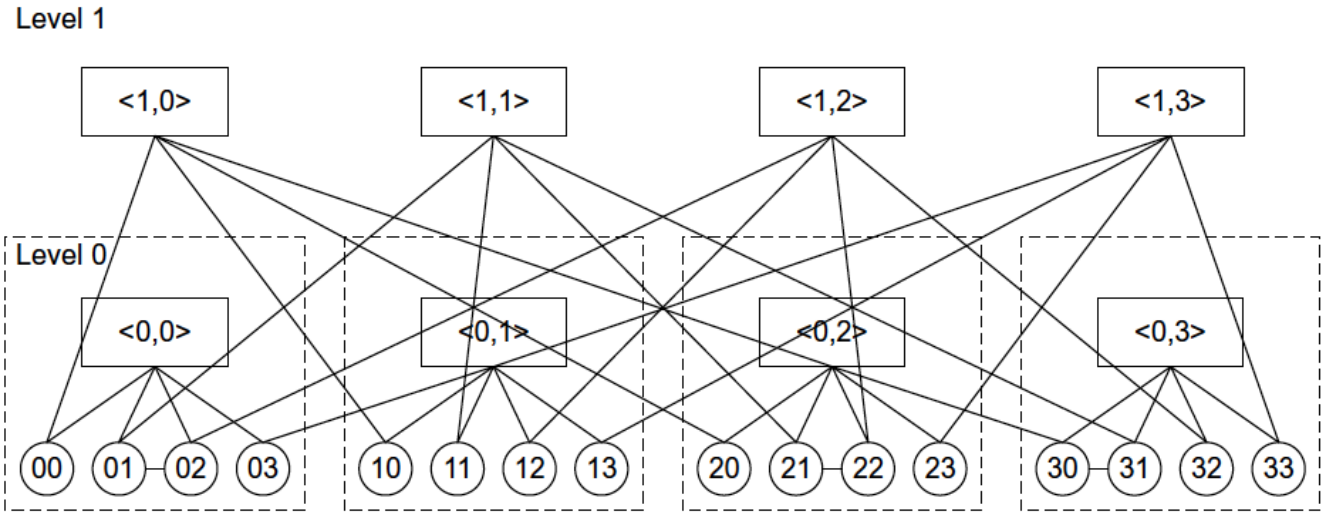
INTERCONNECTION OF MODULAR DATA CENTERS

- Construction of larger data centers using a farm of container modules.
- Example: A Server-Centric Network for a Modular Data Centre (BCube network)
  - The bottom layer contains all the server nodes and they form Level 0.
  - Level 1 switches form the top layer of BCube0.
  - BCube is a recursively constructed structure. The BCube0 consists of n servers connecting to an n-port switch.



INTERCONNECTION OF MODULAR DATA CENTERS

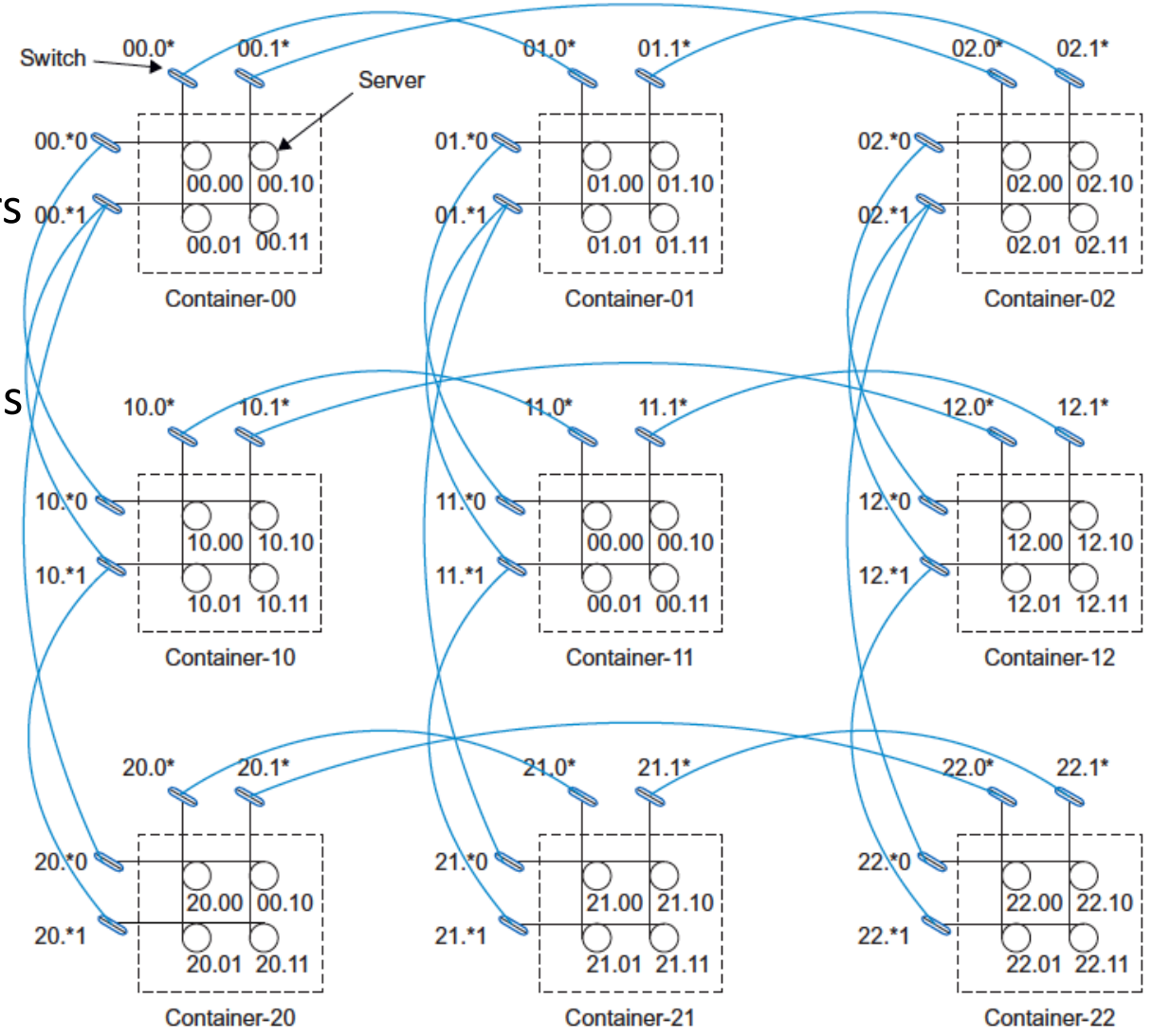
- Example: A Server-Centric Network for a Modular Data Centre (BCube network)
- The BCube provides multiple paths between any two nodes.
  - Multiple paths provide extra bandwidth to support communication patterns in different cloud applications.
- The BCube provides a kernel module in the server OS to perform routing operations.
- The kernel module supports packet forwarding while the incoming packets are not destined to the current node.



INTERCONNECTION OF MODULAR DATA CENTERS

Inter-Module Connection Networks

- MDCube (Modularized Datacenter Cube)is a network connecting multiple BCube containers by using high-speed switches in the BCube.
- Example: a 2D MDCube is constructed from nine BCube1 containers.
- The MDCube is used to build a large-scale data center for supporting cloud application communication patterns.



DATA-CENTER RESOURCE MANAGEMENT ISSUES

- Making common users satisfied
  - The data center should be designed to provide quality service to the majority of users.
- 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.

DATA-CENTER RESOURCE MANAGEMENT ISSUES

- 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

Topics covered:

- Basic cloud architecture to process massive amounts of data with a high degree of parallelism.
- Virtualization support,
- Resource provisioning,
- Infrastructure management, and
- Performance modeling.

A Generic Cloud Architecture Design

Cloud Platform Design Goals

- Scalability,
- virtualization,
- efficiency, and
- reliability
- computing platform.
- Security in shared resources and shared access of data centers also pose another design challenge.

❑ 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.

❑ System reliability can benefit from this architecture.

- 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.
- In such a situation, even if one of the data centers crashes, the user data is still accessible.

A Generic Cloud Architecture Design

Enabling Technologies for Clouds

- The key driving forces behind cloud computing are:
  - the ubiquity of broadband and wireless networking,
  - falling storage costs, and
  - progressive improvements in Internet computing software.
- Cloud users are able to demand:
  - more capacity at peak demand,
  - reduce costs,
  - experiment with new services, and
  - remove unneeded capacity,
- Service providers can increase system utilization via multiplexing, virtualization, and dynamic resource provisioning.
- Clouds are enabled by the progress in hardware, software, and networking technologies summarized in the table in the next slide.

Enabling Technologies for The 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

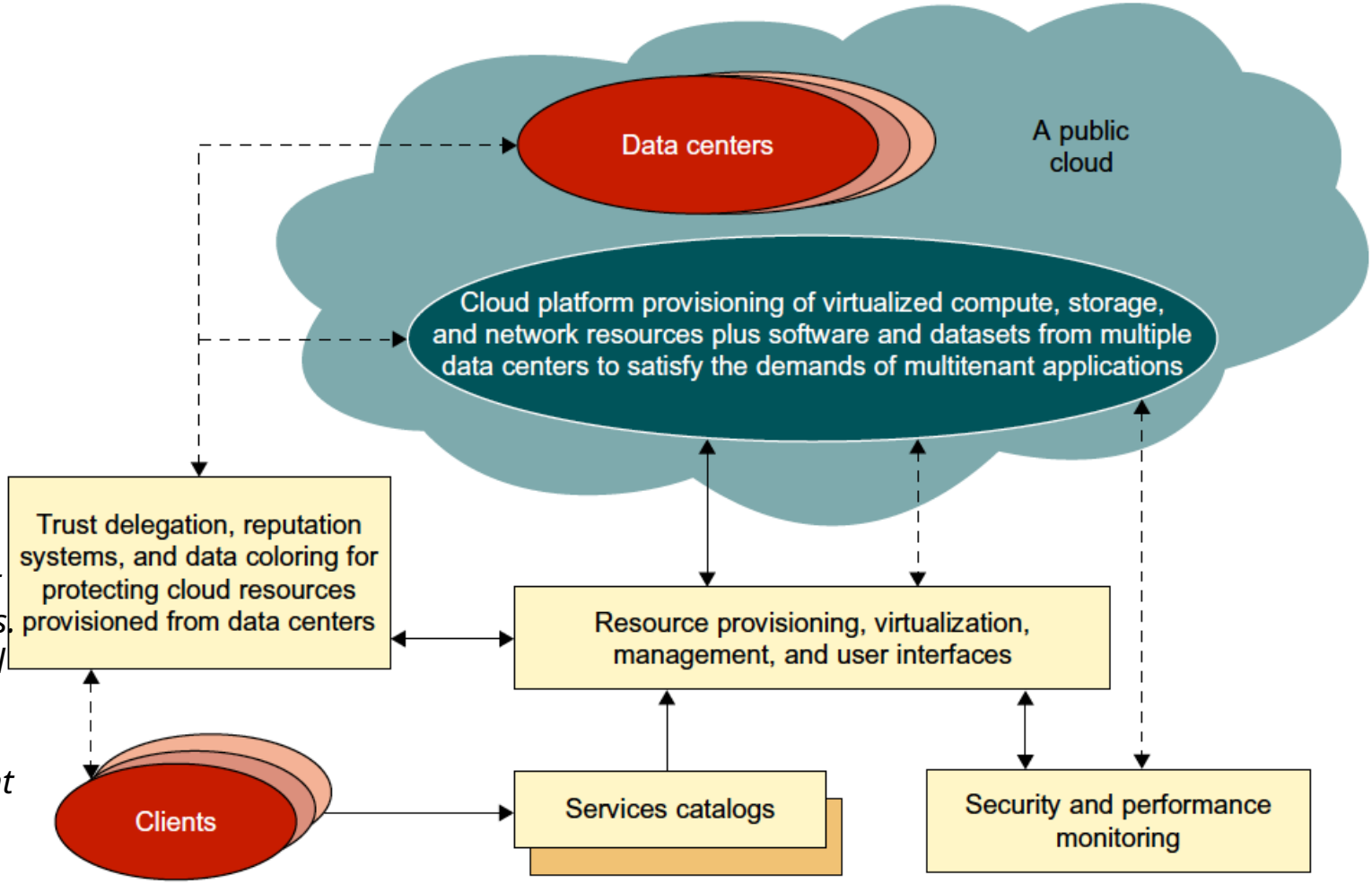
A Generic Cloud Architecture Design

A Generic Cloud Architecture

- The Internet cloud is envisioned as a massive cluster of servers.
- 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.
- The cloud platform demands distributed storage and accompanying services.
- The cloud computing resources are built into the data centers,
  - Data centers are typically owned and operated by a third-party provider.

A Generic Cloud Architecture Design

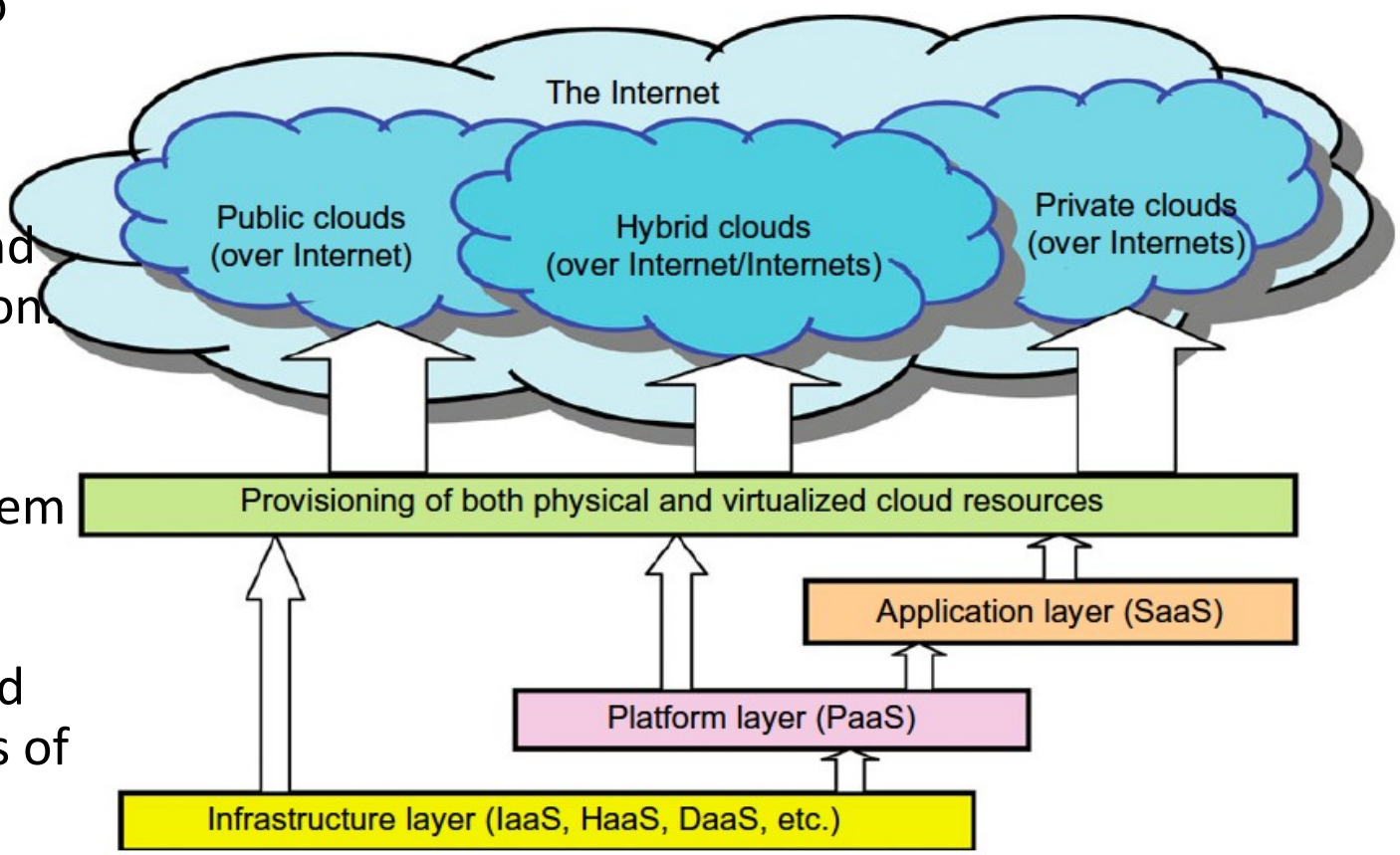
A Generic Cloud Architecture



Solid lines represent data or service flows. Dashed lines control flows in trust management and security enforcement operations.

Layered Cloud Architectural Development

- The architecture of a cloud is developed at three layers: infrastructure, platform, and application.
- They are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud.
- The platform should be able to assure users that they have scalability, dependability, and security protection.
- the virtualized cloud platform serves as a " system middleware" between the infrastructure and application layers of the cloud.



Layered Cloud Architectural Development

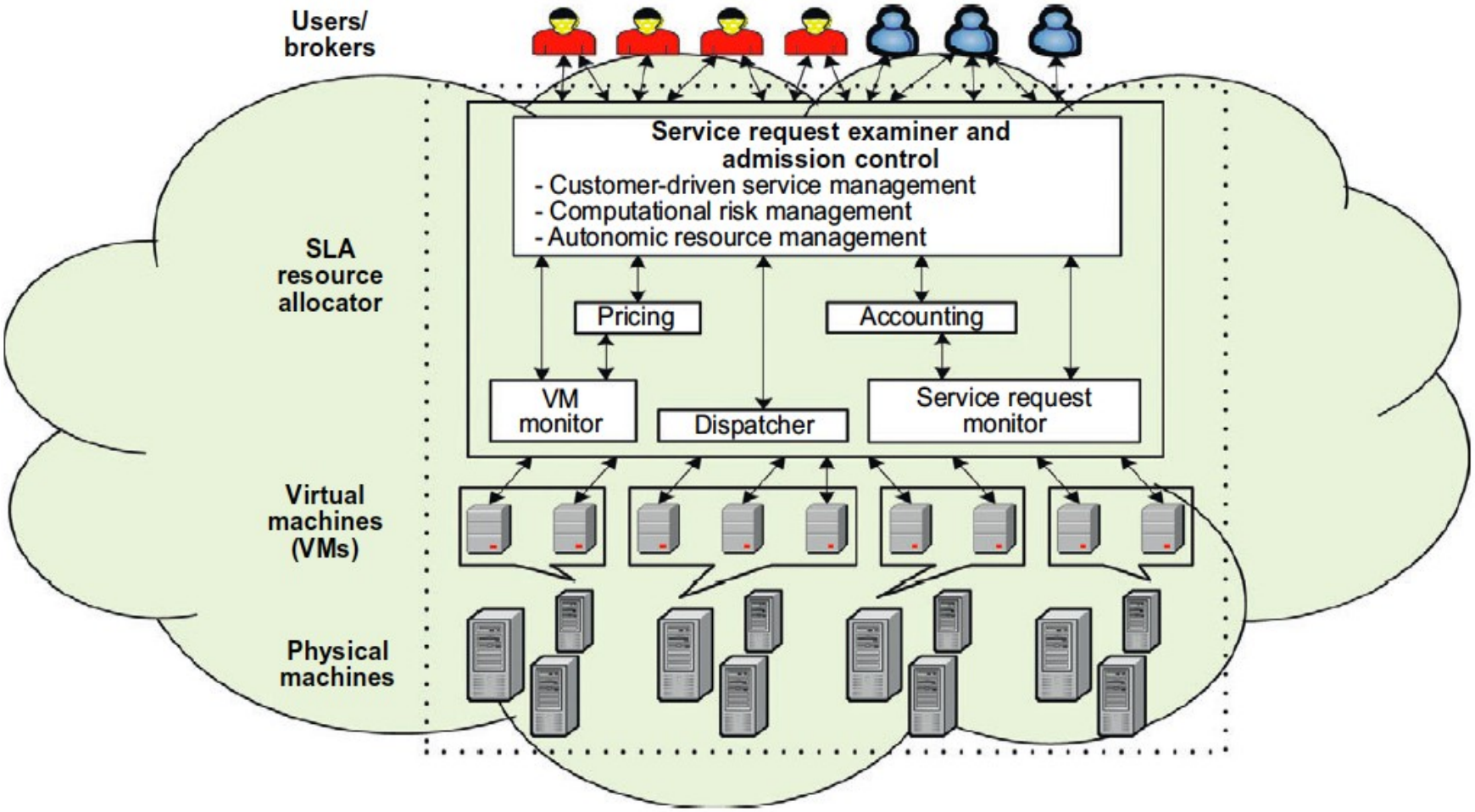
Market-Oriented Cloud Architecture

- Users require a specific level of QoS to be maintained by their providers.
- Providers consider and meet the different QoS parameters of each consumer as negotiated in specific SLAs.
- To achieve this:
  - market-oriented resource management is necessary to regulate the supply and demand of cloud resources to achieve market equilibrium.
  - promote QoS-based resource allocation mechanisms.
- Based on the high level architecture on the next slide:
  - Users submit service requests from anywhere to the data center and cloud to be processed.
  - The SLA resource allocator acts as the interface between the data center/cloud service provider and external users/brokers.
  - The request examiner
    - ensures that there is no overloading of resources.
    - gets latest status information regarding resource availability from VM Monitor mechanism and
    - workload processing from Service Request Monitor mechanism.
    - assigns requests to VMs and determines resource entitlements for allocated VMs.
- Pricing mechanism decides how service requests are charged.
  - For instance, requests can be charged based on submission time (peak/off-peak), pricing rates (fixed/changing), or availability of resources (supply/demand).
- The VM Monitor mechanism keeps track of the availability of VMs and their resource entitlements.
- 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.

Layered Cloud Architectural Development

Market-Oriented Cloud Architecture

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



Layered Cloud Architectural Development

Quality of Service Factors

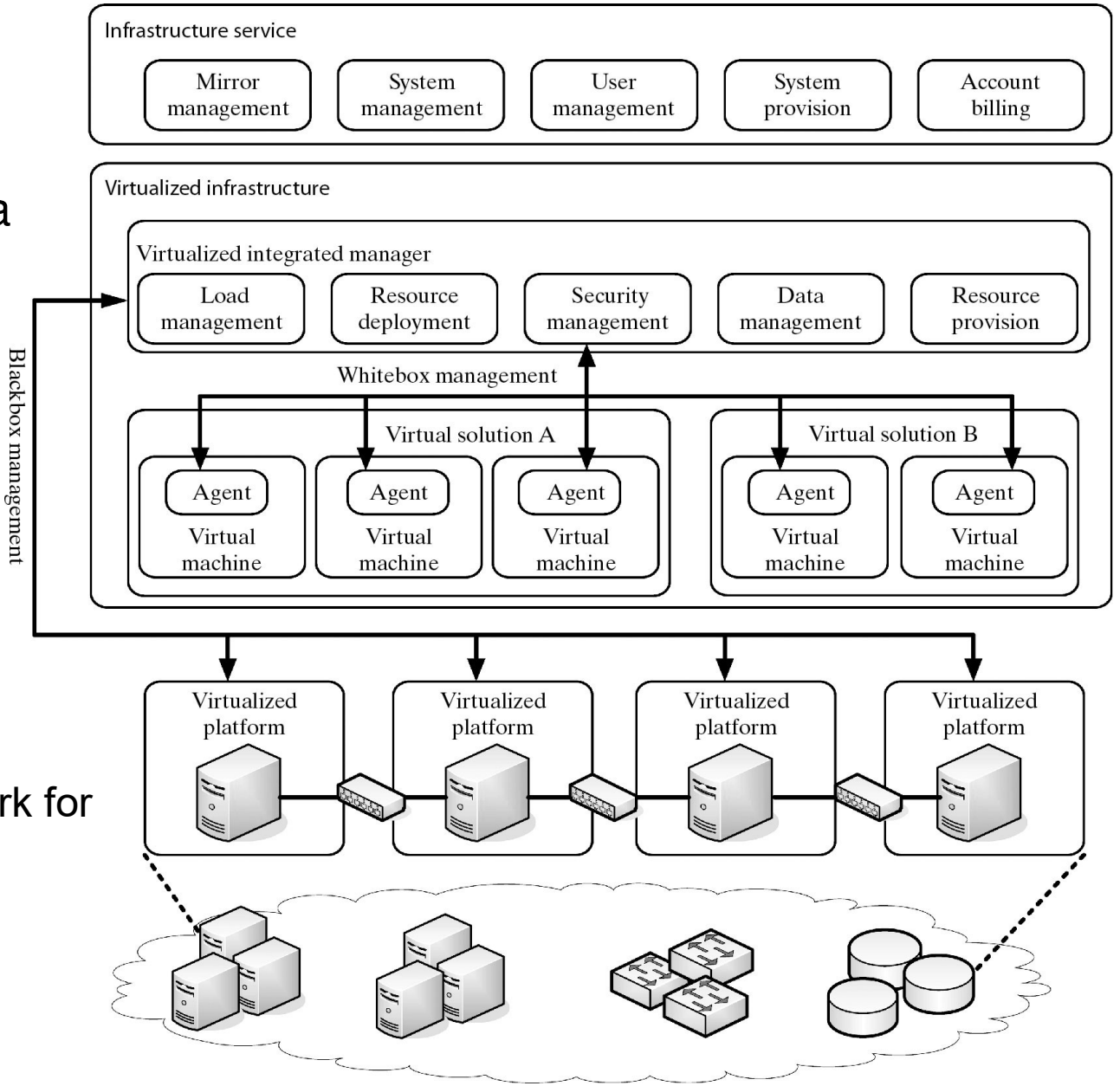
- time, cost, reliability, and trust/security.
- QoS requirements cannot be static and may change over time due to continuing changes in business operations and operating environments.
- Mechanisms for Service Level Agreement (SLA) negotiation dynamically.
- Support customer-driven service management based on customer profiles and requested service requirements.
- Commercial clouds define computational risk management tactics to identify, assess, and manage risks involved in the execution of applications with regard to service requirements and customer needs.
- Use appropriate market-based resource management strategies
  - To encompass both customer-driven service management and
  - computational risk management
- Sustain SLA-oriented resource allocation.
- Incorporate autonomic resource management models.
- Leverage VM technology to dynamically assign resource shares according to service requirements.

Virtualization Support and Disaster Recovery

- System virtualization is a very distinguishing feature of cloud computing infrastructure.
- 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.
- Virtualization also means the resources and fundamental infrastructure are virtualized.
- User does not care about the computing resources that are used for providing the services.
- Application developers do not care about some infrastructure issues such as scalability and fault tolerance because they are virtualized.
  - They focus on service logic.

Virtualization Support and Disaster Recovery

- Infrastructure needed to virtualize the servers in a data center for implementing specific cloud applications.



Virtualized servers, storage , and network for cloud platform construction

Virtualization Support and Disaster Recovery

Hardware Virtualization

Based on the previous slide:

- 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 OSes.
- Establish the virtual disk storage and virtual networks needed by the VMs.
- The virtualized resources form a resource pool.
- The virtualization is carried out by special servers dedicated to generating the virtualized resource pool.
- The virtualized infrastructure (black box in the middle of the fig.) is built with many virtualizing integration managers .
  - These managers handle loads, resources, security, data, and provisioning functions.
- The figure shows two VM platforms.
  - Each platform carries out a virtual solution to a user job. All cloud services are managed in the boxes at the top.

Virtualization Support and Disaster Recovery

Hardware Virtualization

- Virtualization software is used to virtualize the hardware.
  - It a special kind of software which simulates the execution of hardware and runs even unmodified operating systems.
  - Cloud computing systems use virtualization software as the running environment for legacy software such as old operating systems and unusual applications.
- Virtualization software is also used as the platform for developing new cloud applications that enable developers to use any operating systems and programming environments they like.
  - The development environment and deployment environment can now be the same, which eliminates some runtime problems.
- system virtualization software is considered the hardware analog mechanism to run an unmodified operating system, usually on bare hardware directly, on top of software.
- The table in the next slide lists some of the system virtualization software in wide use.
- Currently, the VMs installed on a cloud computing platform are mainly used for hosting third-party programs.

Virtualization Support and Disaster Recovery

Hardware Virtualization

- VMs provide flexible runtime services to free users from worrying about the system environment.

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

Virtualization Support and Disaster Recovery

Virtualization Support in Public Clouds

- AWS provides extreme flexibility (VMs) for users to execute their own applications.
- GAE provides limited application-level virtualization for users to build applications only based on the services that are created by Google.
- Microsoft provides programming-level virtualization (.NET virtualization) for users to build their applications.
- Cloud computing leverage the benefits of virtualization to provide a scalable and autonomous computing environment.

Virtualization Support and Disaster Recovery

Storage Virtualization for Green Data Centers

- Virtualization had a great impact on cost reduction from reduced power consumption in physical computing systems.
- Virtualization and server consolidation have already proven handy in this aspect.
- Green data centers and benefits of storage virtualization are considered to further strengthen the synergy of green computing.

Virtualization Support and Disaster Recovery

Virtualization for IaaS

- Use of VMs in clouds has the following distinct benefits:

(1) System administrators consolidate workloads of underutilized servers in fewer servers;

(2) VMs have the ability to run legacy code without interfering with other APIs;

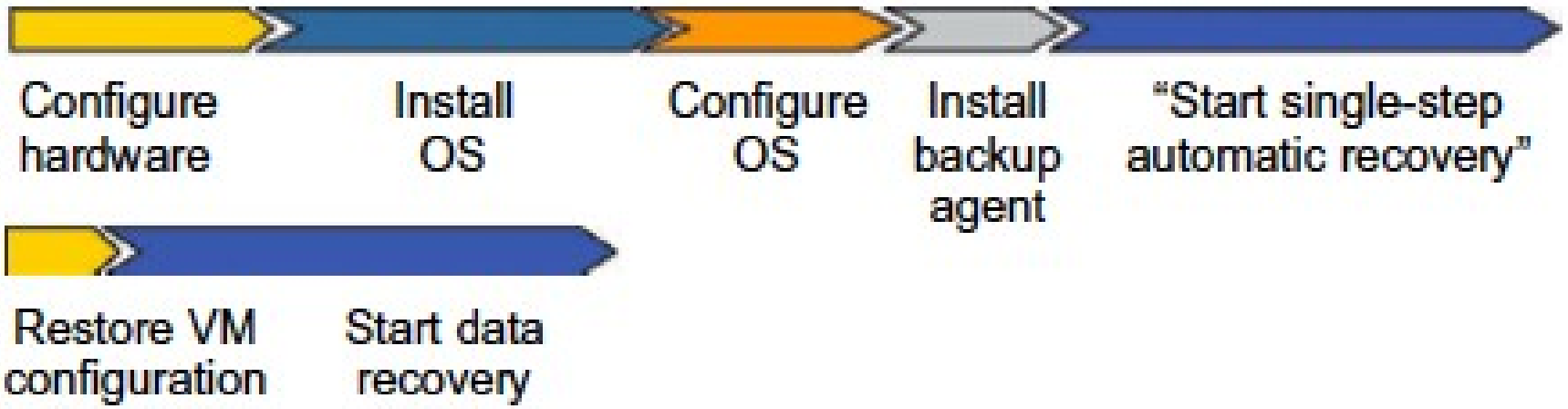
(3) VMs can be used to improve security through creation of sandboxes for running applications with questionable reliability;

(4) virtualized cloud platforms can apply performance isolation, letting providers offer some guarantees and better QoS to customer applications.

Virtualization Support and Disaster Recovery

VM Cloning for Disaster Recovery

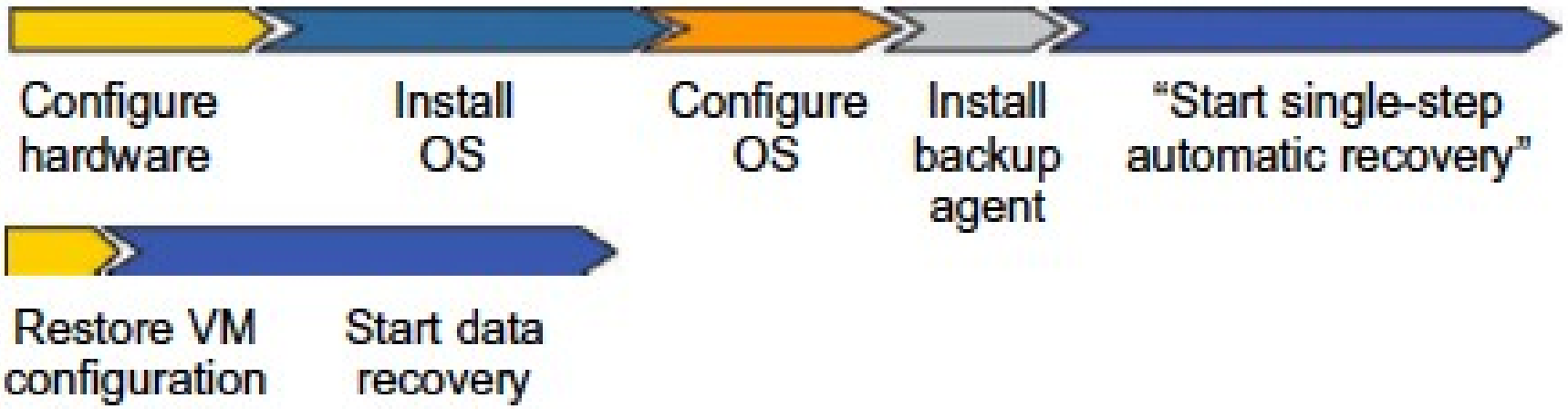
- Two schemes for disaster recovery.
  - Recover one physical machine by another physical machine (traditional scheme).
  - Recover one VM by another VM.
- As shown in the top timeline, traditional disaster recovery is rather slow, complex, and expensive.
  - Total recovery time is attributed to the hardware configuration, installing and configuring the OS, installing the backup agents, and the long time to restart the physical machine.



Virtualization Support and Disaster Recovery

VM Cloning for Disaster Recovery

- To recover a VM platform, the installation and configuration times for the OS and backup agents are eliminated.
  - Therefore, we end up with a much shorter disaster recovery time, about 40 percent of that to recover the physical machines.
- Virtualization aids in fast disaster recovery by VM encapsulation.



Virtualization Support and Disaster Recovery

VM Cloning for Disaster Recovery

- The cloning of VMs offers an effective solution.
- The idea is to make a clone VM on a remote server for every running VM on a local server.
- Among all the clone VMs, only one needs to be active.
  - The remote VM should be in a suspended mode.
- A cloud control center should be able to activate this clone VM in case of failure of the original VM,
  - taking a snapshot of the VM to enable live migration in a minimal amount of time.
- The migrated VM can run on a shared Internet connection.
- Only updated data and modified states are sent to the suspended VM to update its state.
- The Recovery Property Objective (RPO) and Recovery Time Objective (RTO) are affected by the number of snapshots taken.
- Security of the VMs should be enforced during live migration of VMs.

Architectural Design Challenges

Challenge 1—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 software infrastructure and accounting systems.
- Using multiple cloud providers may provide more protection from failures.
- Another availability obstacle is distributed denial of service (DDoS) attacks.
- Criminals threaten to cut off the incomes of SaaS providers by making their services unavailable.
- Some utility computing services offer SaaS providers the opportunity to defend against DDoS attacks by using quick scale-ups.

Architectural Design Challenges

Challenge 1—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 software infrastructure and accounting systems.
- Using multiple cloud providers may provide more protection from failures.
- Another availability obstacle is distributed denial of service (DDoS) attacks.
- Criminals threaten to cut off the incomes of SaaS providers by making their services unavailable.
- Some utility computing services offer SaaS providers the opportunity to defend against DDoS attacks by using quick scale-ups.

Architectural Design Challenges

Challenge 1—Service Availability and Data Lock-in Problem

- Software stacks have improved interoperability among different cloud platforms, but the APIs itself are still proprietary.
  - Thus, customers cannot easily extract their data and programs from one site to run on another.
- The obvious solution is to standardize the APIs so that a SaaS developer can deploy services and data across multiple cloud providers.
  - This will rescue the loss of all data due to the failure of a single company.
- In addition to mitigating data lock-in concerns,
  - standardization of APIs enables a new usage model in which the same software infrastructure can be used in both public and private clouds.
- Such an option could enable " surge computing," in which the public cloud is used to capture the extra tasks that cannot be easily run in the data center of a private cloud.

Architectural Design Challenges

Challenge 2—Data Privacy and Security Concerns

- Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks.
- Many obstacles can be overcome immediately with well-understood 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.
- Many nations have laws requiring SaaS providers to keep customer data and copyrighted material within national boundaries.
- Traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms.
- In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits.
- Another type of attack is the man-in-the-middle attack for VM migrations.
- Passive attacks steal sensitive data or passwords.
- Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.

Architectural Design Challenges

Challenge 3—Unpredictable Performance and Bottlenecks

- Multiple VMs can share CPUs and main memory in cloud computing, but I/O sharing is problematic.
  - For example, to run 75 EC2 instances with the STREAM benchmark requires a mean bandwidth of 1,355 MB/second.
  - However, for each of the 75 EC2 instances to write 1 GB files to the local disk requires a mean disk write bandwidth of only 55 MB/second.
- This demonstrates the problem of I/O interference between VMs.
- One solution is to improve I/O architectures and operating systems to efficiently virtualize 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.
- Therefore, data transfer bottlenecks must be removed, bottleneck links must be widened, and weak servers should be removed.

Architectural Design Challenges

Challenge 4—Distributed Storage and Widespread Software Bugs

- The database is always growing in cloud applications.
- The opportunity is to create a storage system that will not only meet this growth, but also combine it with the cloud advantage of scaling arbitrarily up and down on demand.
  - This demands the design of efficient distributed SANs.
- Data centers must meet programmers' expectations in terms of scalability, data durability, and HA.
- Data consistence checking in SAN-connected data centers is a major challenge in cloud computing.
- Large-scale distributed bugs cannot be reproduced, so the debugging must occur at a scale in the production 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 in ways that are impossible without using VMs.
  - Debugging over simulators is another approach to attacking the problem, if the simulator is well designed.

Architectural Design Challenges

Challenge 5—Cloud Scalability, Interoperability, and Standardization

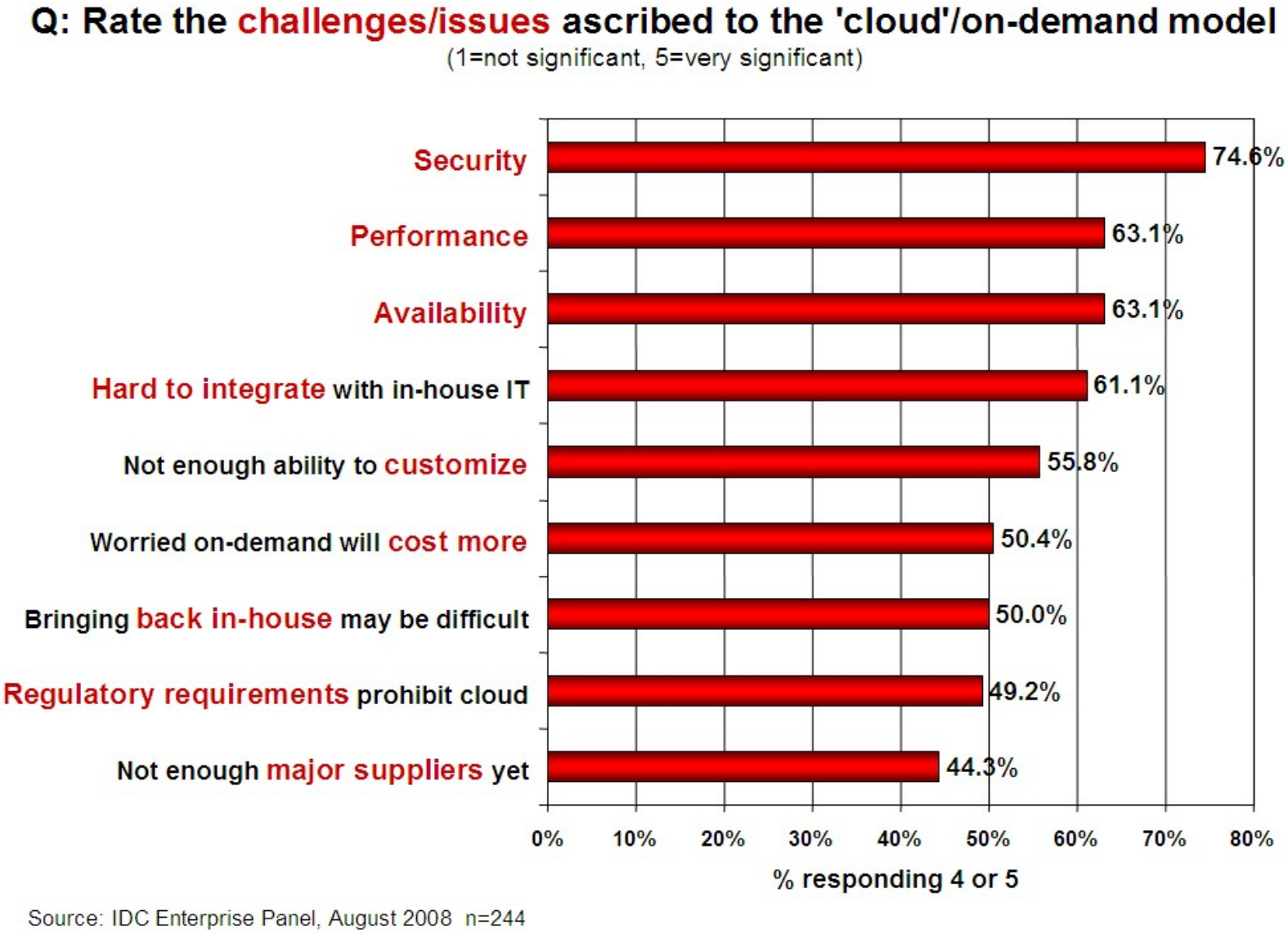
- The pay-as-you-go model applies to storage and network bandwidth;
  - both are counted in terms of the number of bytes used.
- Computation is different depending on virtualization level.
  - GAE automatically scales in response to load increases and decreases; users are charged by the cycles used.
  - 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 SLAs.
- 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 software to be deployed in VMs. This VM format does not rely on the use of a specific host platform, virtualization platform, or guest operating system.
- The approach is to address virtual platform-agnostic packaging with certification and integrity of packaged software.
- The package supports virtual appliances to span more than one VM.

Architectural Design Challenges

Challenge 6—Software Licensing and Reputation Sharing

- Many cloud computing providers originally relied on open source software 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.
- One customer's bad behavior can affect the reputation of the entire cloud.
  - For instance, blacklisting of EC2 IP addresses by spam-prevention services may limit smooth VM installation.
- An opportunity would be to create reputation-guarding services similar to the "trusted e-mail" services currently offered (for a fee) to services hosted on smaller ISPs.
- Another legal issue concerns the transfer of legal liability.
  - Cloud providers want legal liability to remain with the customer, and vice versa.
  - This problem must be solved at the SLA level.

Challenges/Issues in Cloud Computing



Source: IDC Enterprise Panel, August 2008 n=244

Challenges in Cloud Computing (1)

Concerns from The Industry (Providers):

- Replacement Cost
  - Exponential increase in cost to maintain the infrastructure
- Vendor Lock-in
  - No standard API or protocol can be very serious
- Standardization
  - No standard metric for QoS is limiting the popularity
- Security and Confidentiality
  - Trust model for cloud computing
- Control Mechanism
  - Users do not have any control over infrastructures

Challenges in Cloud Computing (2)

Concerns from Research Community:

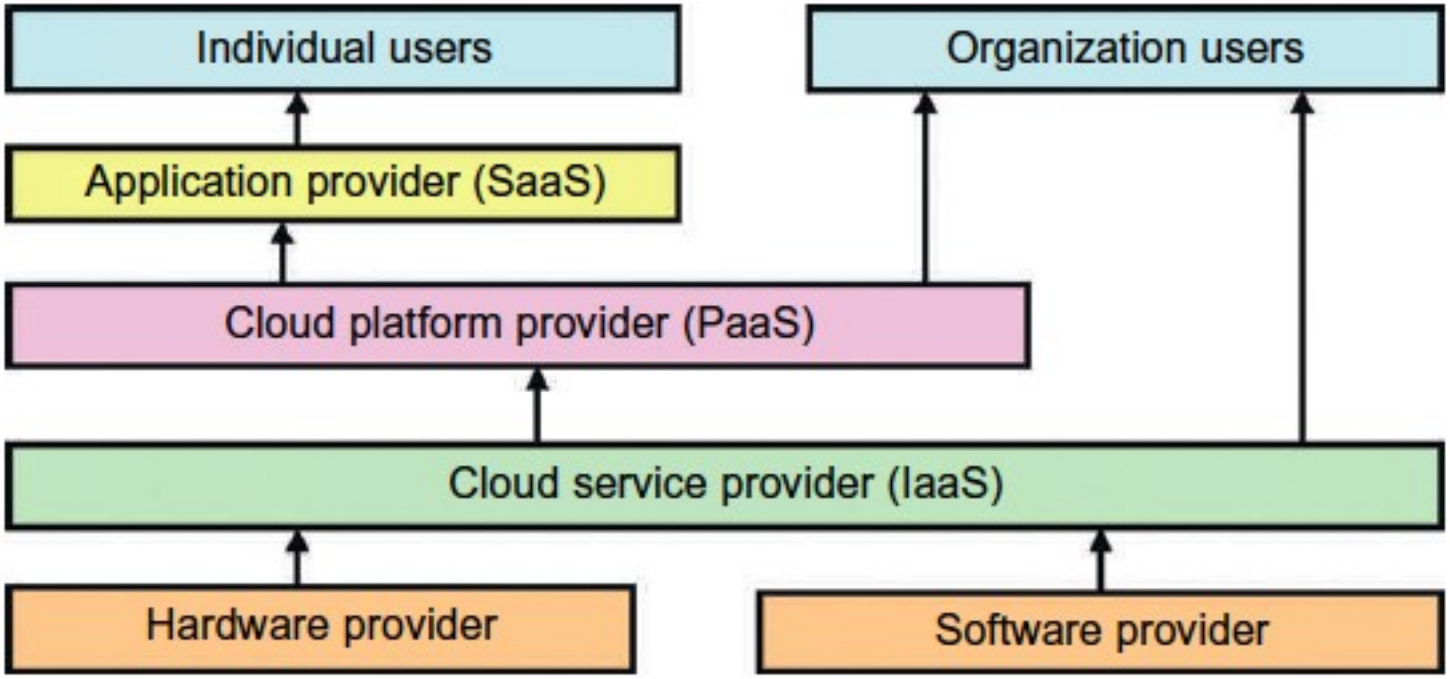
- Conflict to legacy programs
  - With difficulty in developing a new application due to lack of control
- Provenance
  - How to reproduce results in different infrastructures
- Reduction in Latency
  - No specially designed interconnect used
  - Very low controllability in layout of interconnect due to abstraction
- Programming Model
  - Hard to debug where programming naturally error-prone
  - Details about infrastructure are hidden
- QoS Measurement
  - Especially for ubiquitous computing where context changes

PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE

Note: AWS and Azure are self-study

Public Clouds and Service Offerings

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



PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE

Note: AWS and Azure are self-study

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 as described earlier.
- Also, many cloud entrepreneurs are selling value-added utility services to massive numbers of users.
- 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.

PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE

Public Clouds and Service Offerings

Table 4.5 Five Major Cloud Platforms and Their Service Offerings [30]

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	

Note: *WCA: WebSphere CloudBurst Appliance; RC2: Research Compute Cloud; RAD: Rational Application Developer; SOA: Service-Oriented Architecture; TSAM: Tivoli Service Automation Manager; EC2: Elastic Compute Cloud; S3: Simple Storage Service; SQS: Simple Queue Service; GAE: Google App Engine; AWS: Amazon Web Services; SQL: Structured Query Language; EBS: Elastic Block Store; CRM: Consumer Relationship Management.*

PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE

Note: AWS and Azure are self-study

Public Clouds and Service Offerings

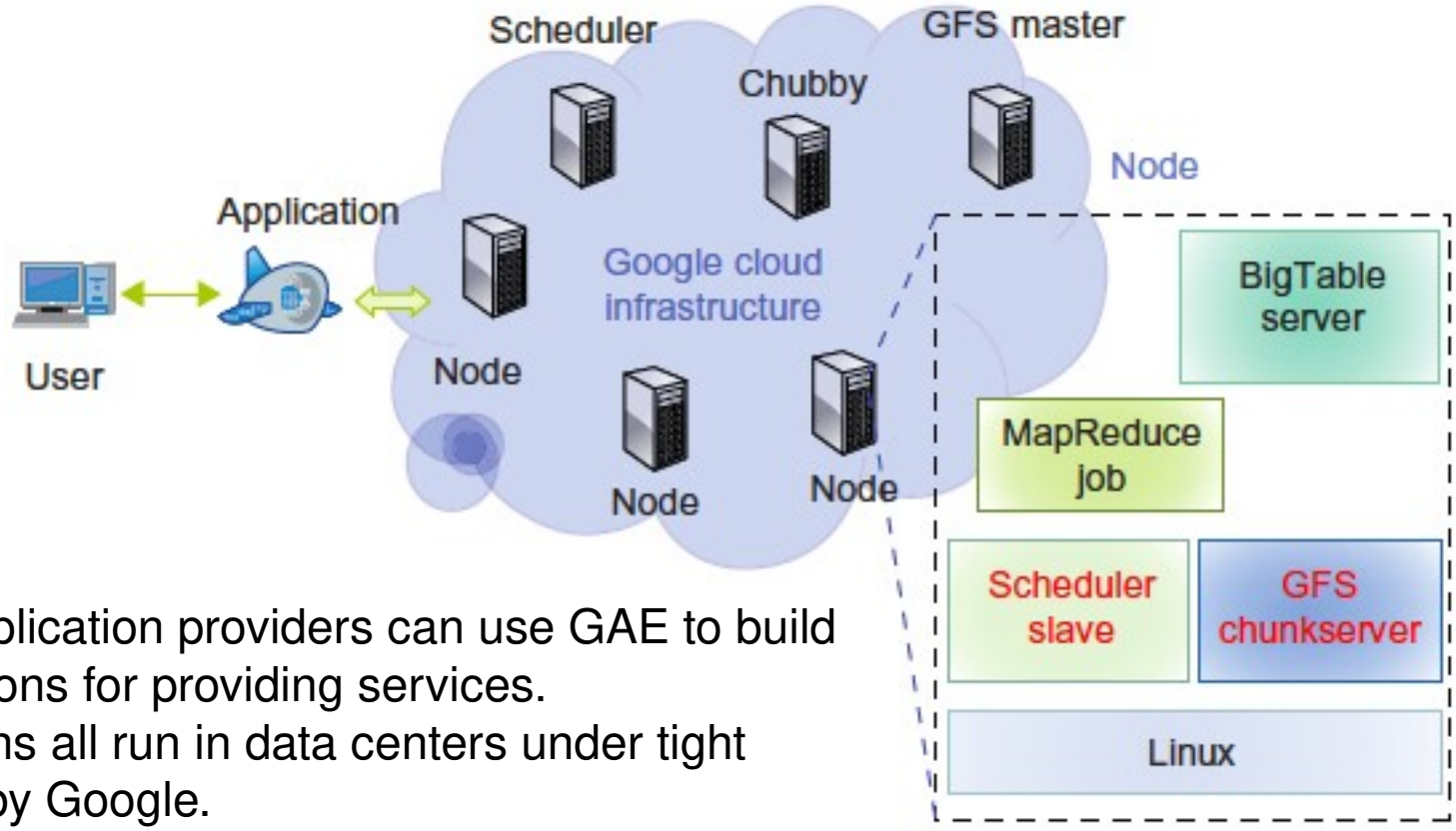
- These models (in the previous table) are offered based on various SLAs between the providers and the users.
  - SLAs are more common in network services as they account for the QoS characteristics of network services.
- For cloud computing services, it is difficult to find a reasonable precedent for negotiating an SLA.
- In a broader sense, the SLAs for cloud computing address service availability, data integrity, privacy, and security protection.

Platform as a Service (PaaS): Google App Engine

- This platform allows users to develop and host web application in Google datacenters with automatic scaling according to the demand.
- It is a free service for a certain limit and it only requires a Gmail account to access the services. After the free limit is exceeded the customers are charged for additional storage, bandwidth and instance hours.
- The current version supports Java, Python and Go as the programming languages and Google plans to add more languages in the future.
- All billed App Engine applications have a 99.95% uptime SLA. App Engine is designed to sustain multiple datacenter outages without any downtime.
- The app engine has a few restrictions - can only execute code called from an HTTP request, Java applications may only use a subset from the JRE standard edition and Java application cannot create new threads.

Platform as a Service (PaaS): Google App Engine (GAE) Architecture

- GFS is used for storing large amounts of data.
- MapReduce is for use in application program development.
- Chubby is used for distributed application lock services.
- BigTable offers a storage service for accessing structured data.
- Users can interact with Google applications via the web interface provided by each application.

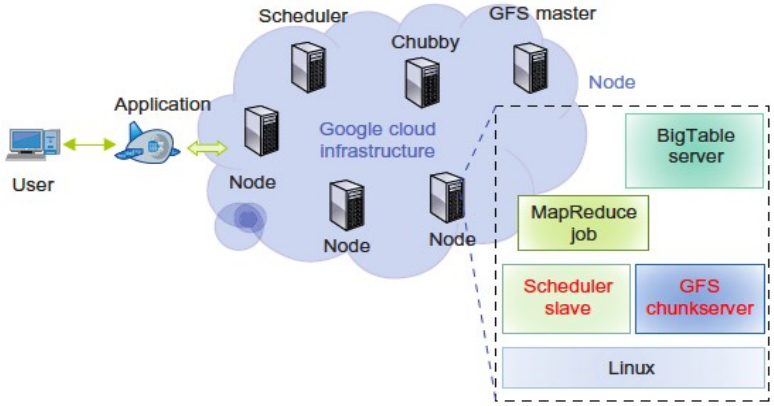


- Third-party application providers can use GAE to build cloud applications for providing services.
- The applications all run in data centers under tight management by Google.
- Inside each data center, there are thousands of servers forming different clusters.

Platform as a Service (PaaS): Google App Engine (GAE) Functional Modules

The GAE platform comprises the following five major components:

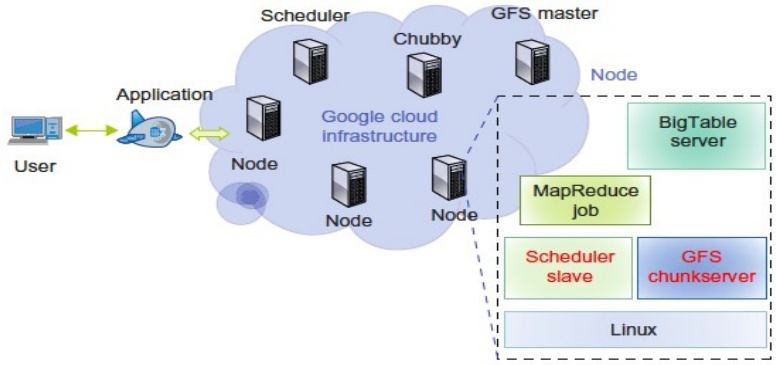
- The datastore offers object-oriented, distributed, structured data storage services based on BigTable techniques.
  - The datastore secures data management operations.
- The application runtime environment offers a platform for scalable web programming and execution.
  - It supports two development languages: Python and Java.
- The software development kit (SDK) is used for local application development.
  - The SDK allows users to execute test runs of local applications and upload application code.
- The administration console is used for easy management of user application development cycles, instead of for physical resource management.
- The GAE web service infrastructure provides special interfaces to guarantee flexible use and management of storage and network resources by GAE.



Platform as a Service (PaaS): Google App Engine (GAE) GAE Applications

Well-known GAE applications include:

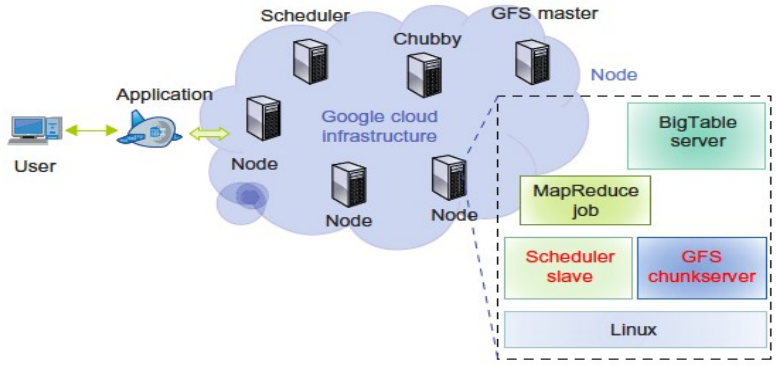
- Google Search Engine, Google Docs, Google Earth, and Gmail.
- These applications can support large numbers of users simultaneously.
- Users can interact with Google applications via the web interface provided by each application.
- Third-party application providers can use GAE to build cloud applications for providing services.
  - The applications are all run in the Google data centers.
- Inside each data center, there might be thousands of server nodes to form different clusters.
  - Each cluster can run multipurpose servers.



Platform as a Service (PaaS): Google App Engine (GAE) GAE Applications

GAE supports many web applications:

- One is a storage service to store application-specific data in the Google infrastructure.
  - The data can be persistently stored in the backend storage server while still providing the facility for queries, sorting, and even transactions similar to traditional database systems.
- GAE also provides Google-specific services, such as the Gmail account service
  - which is the login service, that is, applications can use the Gmail account directly.
  - This can eliminate the tedious work of building customized user management components in web applications.
- Thus, web applications built on top of GAE can use the APIs authenticating users and sending e-mail using Google accounts.



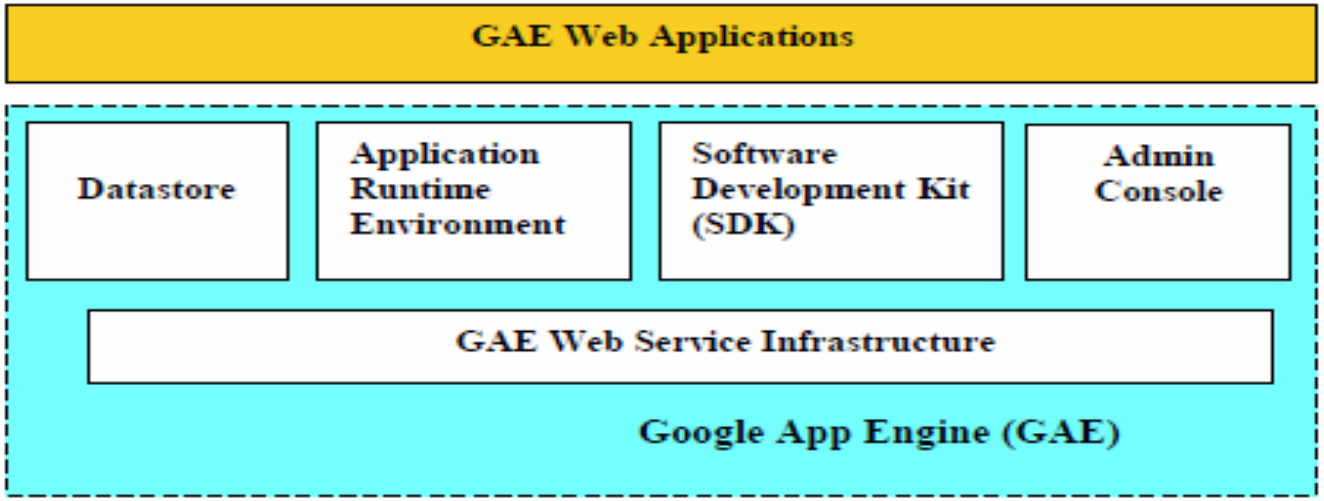


Figure 7.24 Functional components in the Google App Engine (GAE)
(Courtesy of Google, http://code.google.com/appengine/)

Google App Engine Front Page: http://code.google.com/appengine/

Signing up for an account or use your gmail account name : https://appengine.google.com/

Downloading GAE SDK : http://code.google.com/appengine/downloads.html

Python Getting Started Guide: http://code.google.com/appengine/docs/python/gettingstarted/

Java Getting Started Guide: http://code.google.com/appengine/docs/java/gettingstarted/

Quota page for free service: http://code.google.com/appengine/docs/quotas.html#Resources

Billing page if you go over the quota:
http://code.google.com/appengine/docs/billing.html#Billable_Quota_Unit_Cost

AWS – a leader in providing public IaaS services.

- EC2 (Elastic compute cloud allows users to rent virtual computers to run their own computer applications. It allows scalable deployment. A user can create, launch, and terminate server instances as needed, paying by the hour for active servers.
- S3 (simple storage service) provides the object-oriented storage service for users.
- EBS (Elastic block service) provides the block storage interface which can be used to support traditional applications.
- Amazon DevPay is a simple to use online billing and account management service that makes it easy for businesses
- MPI clusters uses hardware-assisted virtualization instead of para-virtualization and users are free to create a new AMIs
- AWS import/export allows one to ship large volumes of data to and from EC2 by shipping physical discs.
- Brokering systems offer a striking model for controlling sensors and providing office support of smartphones and tablets.
- Small-business companies can put their business on the Amazon cloud platform. Using AWS they can service a large number of internet users and make profits through those paid services.

Amazon Web Services (AWS)

Compute

Amazon Elastic Compute Cloud (EC2)

Amazon Elastic MapReduce

Auto Scaling

Content Delivery

Amazon CloudFront

Database

Amazon SimpleDB

Amazon Relational Database Service (RDS)

E-Commerce

Amazon Fulfillment Web Service (FWS)

Messaging

Amazon Simple Queue Service (SQS)

Amazon Simple Notification Service (SNS)

Monitoring

Amazon CloudWatch

Networking

Amazon Virtual Private Cloud (VPC)

Elastic Load Balancing

Payments & Billing

Amazon Flexible Payments Service (FPS)

Amazon DevPay

Storage

Amazon Simple Storage Service (S3)

Amazon Elastic Block Storage (EBS)

AWS Import/Export

Support

AWS Premium Support

Web Traffic

Alexa Web Information Service

Alexa Top Sites

Workforce

Amazon Mechanical Turk

Amazon's Lesson

- Down for 3 days since 4/22/2011
- 1000x of businesses went offline. E.g. Pfizer, Netflix, Quora, Foursquare,Reddit
- SLA contract
  - 99.95% availability (<4.5hour down)
  - 10% penalty, otherwise

CNNMoney
A Service of CNN, Fortune & Money
FORTUNE
Home | Video | Business News | Markets | Term Sheet

Why Amazon's cloud Titanic went down

By David Goldman, staff writer April 22, 2011: 5:37 PM ET

NEW YORK (CNNMoney) -- This was never supposed to happen.

Amazon Web Services is the Titanic of cloud hosting, designed with backups to the backups' backups that prevent hosted websites and applications from failing.

Microsoft Azure Cloud :

This is essentially a PaaS Cloud.

- Windows Azure run its cluster hosted at Microsoft's datacenters that manages computing and storage resources.
  - One can download Azure development kit to run a local version of Azure. It allows Azure applications to be developed and debugged one the windows 7 hosts.
- All cloud services can interact with traditional MS software applications such as Windows Live, Office Live, Exchange Online, etc.
- If offers a Windows-based cloud platform using Microsoft virtualization technology.
  - Applications are built on VM's deployed on the data-center services.
    - Azure manages all servers, storage and network resources of the data center.

Microsoft Windows Azure

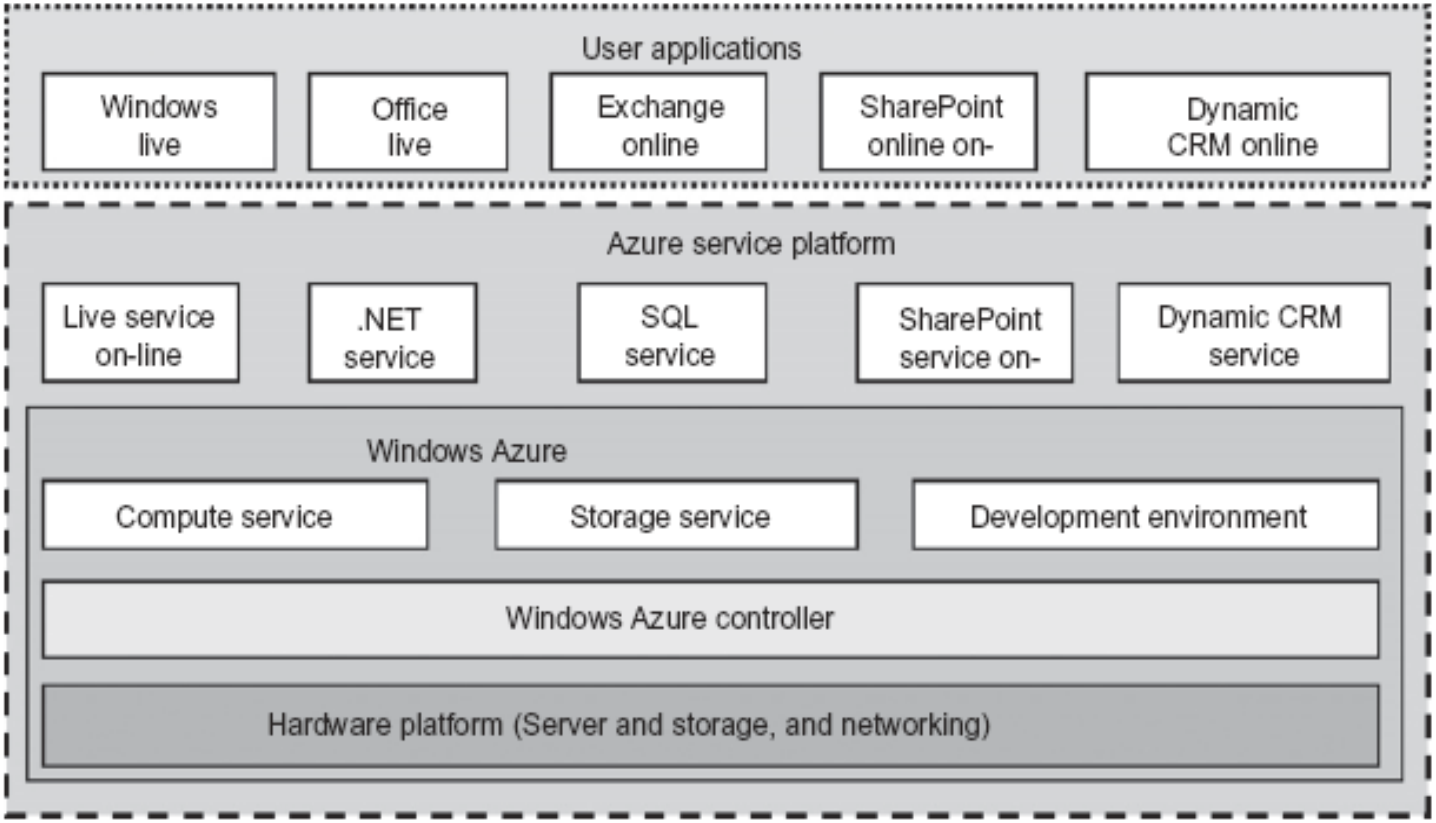


FIGURE 4.22

Microsoft Windows Azure platform for cloud computing.

(Courtesy of Microsoft, 2010, http://www.microsoft.com/windowsazure)

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services

- includes six layers of cloud services, ranging from hardware, network, and collocation to infrastructure, platform, and software applications.

Cloud application (SaaS)	Concur, RightNOW, Teleo, Kenexa, Webex, Blackbaud, salesforce.com, Netsuite, Kenexa, etc.
Cloud software environment (PaaS)	Force.com, App Engine, Facebook, MS Azure, NetSuite, IBM BlueCloud, SGI Cyclone, eBay
Cloud software infrastructure: Computational resources (IaaS); Storage (DaaS); Communications (Caas)	Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth
Collocation cloud services (LaaS)	Savvis, Internap, NTTCommunications, Digital Realty Trust, 365 Main
Network cloud services (NaaS)	Owest, AT&T, AboveNet
Hardware/Virtualization cloud services (HaaS)	VMware, Intel, IBM, XenEnterprise

Cloud Services and Major Providers

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services

- Although the three basic models are dissimilar in usage, as shown in the table, they are built one on top of another.
- The implication is that one cannot launch SaaS applications with a cloud platform.
  - The cloud platform cannot be built if compute and storage infrastructures are not there.

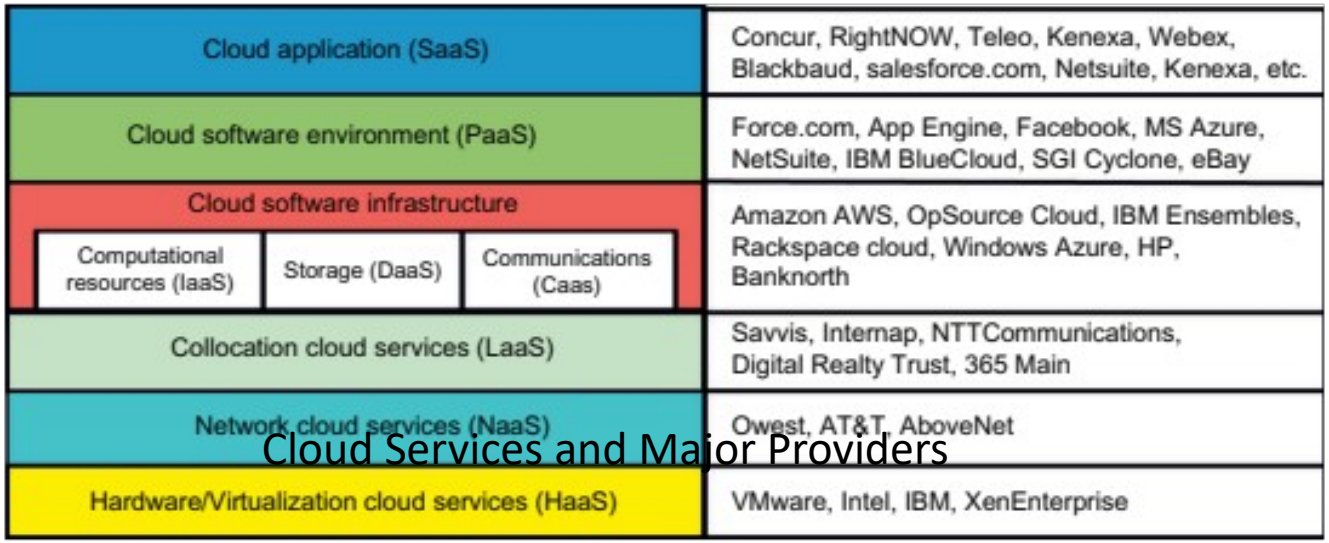
Table 4.7 Cloud Differences in Perspectives of Providers, Vendors, and Users

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

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services

- The bottommost layer provides Hardware as a Service (HaaS).
- The next layer is for interconnecting all the hardware components, and is simply called Network as a Service (NaaS).
  - Virtual LANs fall within the scope of NaaS.
- The next layer up offers Location as a Service (LaaS), which provides a collocation service to house, power, and secure all the physical hardware and network resources.
  - It may provides Security as a Service ("SaaS").

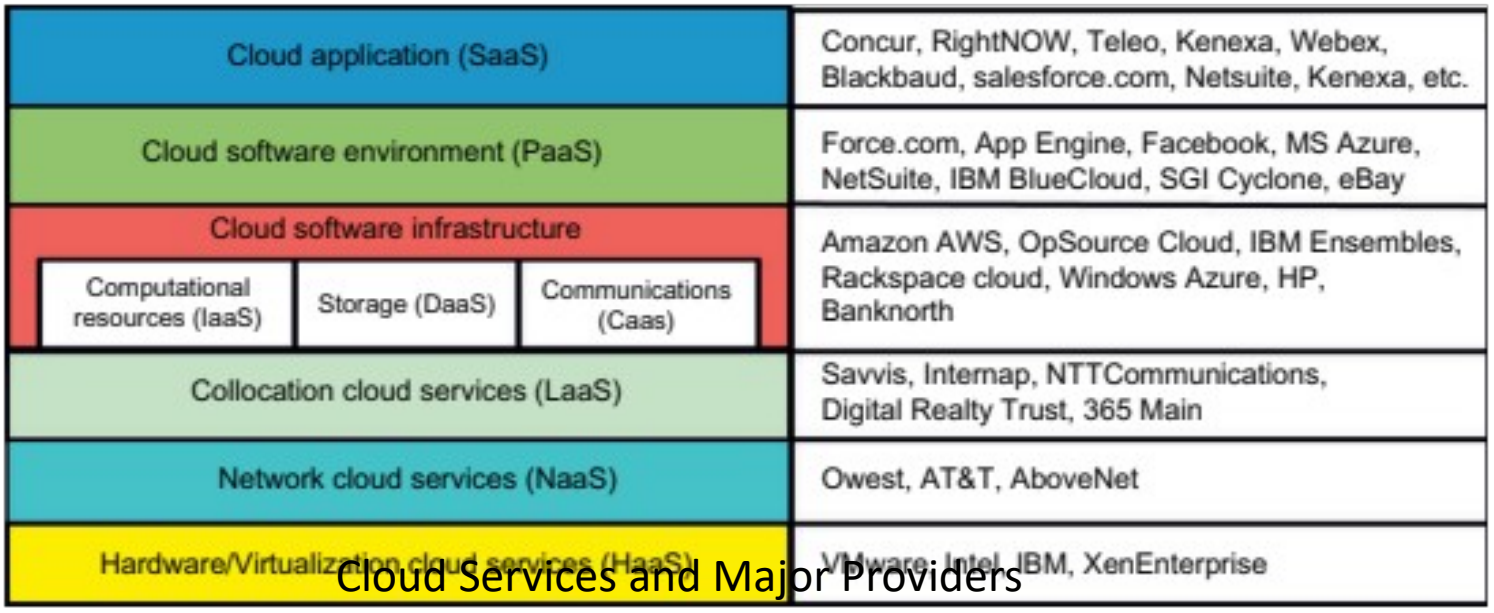


Cloud Services and Major Providers

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services

- The cloud infrastructure layer can be further subdivided as Data as a Service (DaaS) and
- Communication as a Service (CaaS) in addition to compute and storage in IaaS.



Cloud Services and Major Providers

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

- Provisioning of Compute Resources (VMs)
  - Efficient VM provisioning depends on the cloud architecture and management.
  - Resource provisioning schemes also demand fast discovery of services and data.
  - In a virtualized cluster of servers, this demands efficient installation of VMs, live VM migration, and fast recovery from failures.
- (Static) cloud resource provisioning policies (3 policies):
  - Overprovisioning with the peak load causes heavy resource waste.
  - Underprovisioning of resources results in losses by both user and provider in that paid demand by the users is not served and wasted resources still exist for those demanded areas below the provisioned capacity.
  - Constant provisioning of resources with fixed capacity to a declining user demand could result in even worse resource waste.
- Resource Provisioning Methods (3 methods):
  - The **demand-driven method** provides static resources and has been used in grid computing for many years.
  - The **event- driven method** is based on predicted workload by time.
  - The **popularity-driven method** is based on Internet traffic monitored.

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

- Provisioning of Storage Resources
  - The data storage layer is built on top of the physical or virtual servers.
  - Data is stored in the clusters of the cloud provider.
  - Currently hard disk drives are augmented with solid-state drives in the data center.
  - A distributed file system is very important for storing large-scale data.
  - Key/Value stores are also common in data centers.
  - The table outlines three cloud storage services provided by Google, Hadoop, and Amazon.

Table 4.8 Storage Services in Three Cloud Computing Systems

Storage System	Features
GFS: Google File System	Very large sustainable reading and writing bandwidth, mostly continuous accessing instead of random accessing. The programming interface is similar to that of the POSIX file system accessing interface.
HDFS: Hadoop Distributed File System	The open source clone of GFS. Written in Java. The programming interfaces are similar to POSIX but not identical.
Amazon S3 and EBS	S3 is used for retrieving and storing data from/to remote servers. EBS is built on top of S3 for using virtual disks in running EC2 instances.

INTER-CLOUD RESOURCE MANAGEMENT

Virtual Machine Creation and Management

- Independent Service Management
  - Independent services request facilities to execute many unrelated tasks.
  - By using independent service providers, the cloud applications can run different services at the same time.
  - Some other services are used for providing data other than the compute or storage services.
- Running Third-Party Applications
  - The cloud computing platform provides the extra capabilities of accessing backend services or underlying data.
- Virtual Machine Manager
  - Manages VMs deployed on a set of physical resources.
  - The VM manager implementation is generic so that it can create and stop VMs on a physical cluster.
  - To deploy a VM, the manager needs to use its templat.

INTER-CLOUD RESOURCE MANAGEMENT

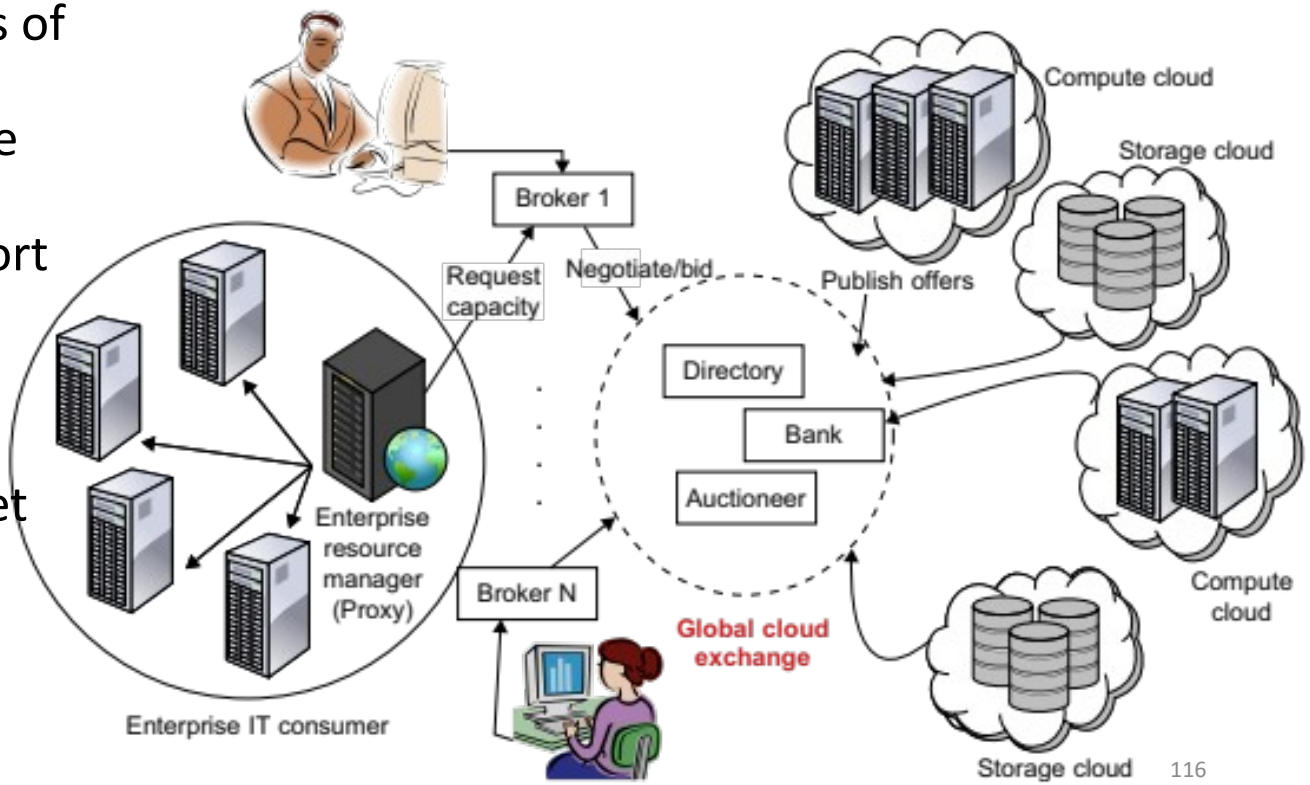
Virtual Machine Creation and Management

- Virtual Machine Templates
  - A VM template is analogous to a computer's configuration and contains a description for a VM with the following static information:
    - The number of cores or processors to be assigned to the VM
    - The amount of memory the VM requires
    - The kernel used to boot the VM's operating system
    - The disk image containing the VM's file system
    - The price per hour of using a VM
- Distributed VM Management
  - A distributed VM manager makes requests for VMs and queries their status.
  - This manager requests VMs from a gateway, e. g., on behalf of the user application.

INTER-CLOUD RESOURCE MANAGEMENT

Global Exchange of Cloud Resources

- Cloud providers (e.g., IaaS providers) establish data centers in multiple geographical locations to provide redundancy and ensure reliability in case of site failures.
  - figure shows a high-level components of a proposed InterCloud architecture.
- Make use of services of multiple cloud infrastructure service providers who can provide better support for their specific consumer needs.
- The Cloud Exchange (CEx) acts as a market maker for bringing together service producers and consumers.



CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

Basic Cloud Security

Three basic cloud security enforcements are expected:

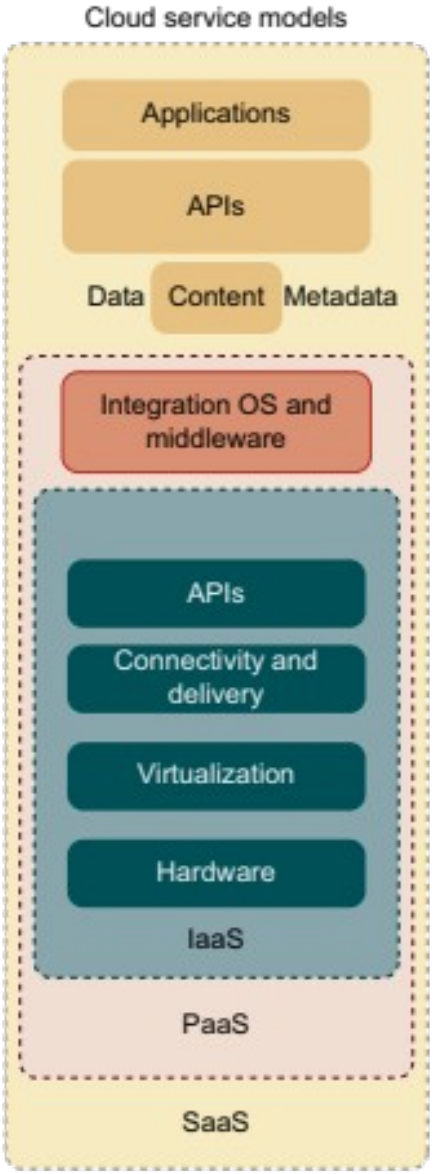
- First, facility security in data centers demands on-site security year round. Biometric readers, CCTV (close-circuit TV), motion detection, and man traps are often deployed.
- Network security demands fault-tolerant external firewalls, intrusion detection systems (IDSes), and third-party vulnerability assessment.
- Platform security demands SSL and data decryption, strict password policies, and system trust certification.

Cloud Service Models & Security Measures

(a) Cloud service models.

(b) corresponding security measures.

- The IaaS is at the innermost level,
- PaaS is at the middle level, and
- SaaS is at the outermost level, including all hardware, software, datasets, and networking resources.



(a) Cloud service models

Security, privacy, and copyright protection measures needed at various cloud service levels

Applications	Binary analysis, scanners, WebApp firewalls, transactional security, copyright protection
Data/information	Data loss protection, common log file, database activity, monitoring, encryption, data coloring (watermarking)
Management	Government risk management and compliance, identity and access management, virtual machines (VMs), patch management
Networking	Network IDS/IPS, firewalls, data processing information, Anti-DDoS, QoS, DNSSEC
Trusted computing	Hardware and software RoT and APIs, trust-overlay and reputation systems
Compute and storage	IDS/IPS, host-based firewalls, integrity and fire/log management, encryption, masking

Acronyms:	
IPS:	Intrusion-prevention system
RoT:	Root of trust
DDoS:	Distributed denial of service
DNSSEC:	Domain Name System Security Extensions
QoS:	Quality of service

(b) Security, privacy, and copyright protection measures

CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

Basic Cloud Security

Cloud components that demand special security protection:

- Protection of servers from malicious software attacks such as worms, viruses, and malware
- Protection of hypervisors or VM monitors from software-based attacks and vulnerabilities
- Protection of VMs and monitors from service disruption and DoS attacks
- Protection of data and information from theft, corruption, and natural disasters
- Providing authenticated and authorized access to critical data and services

CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

Security Challenges in VMs

- Hypervisor malware, guest hopping and hijacking, or VM rootkits.
- Man-in-the-middle attack for VM migrations.
- Passive attacks steal sensitive data or passwords.
- Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.
- An IDS can be a NIDS or a HIDS. Program shepherding can be applied to control and verify code execution.
- Other defense technologies include using the RIO dynamic optimization infrastructure, or VMware's vSafe and vShield tools, security compliance for hypervisors, and Intel vPro technology.
- Others apply a hardened OS environment or use isolated execution and sandboxing.

CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

Cloud Defense Methods

- Virtualization enhances cloud security.
  - But VMs add an additional layer of software that could become a single point of failure.
- With virtualization, a single physical machine can be divided or partitioned into multiple VMs (e.g., server consolidation).
  - This provides each VM with better security isolation and each partition is protected from DoS attacks by other partitions.
  - Security attacks in one VM are isolated and contained from affecting the other VMs.
- The table in the next slide lists eight protection schemes to secure public clouds and data centers.

CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

Cloud Defense Methods

Table 4.9 Physical and Cyber Security Protection at Cloud/Data Centers

Protection Schemes	Brief Description and Deployment Suggestions
Secure data centers and computer buildings	Choose hazard-free location, enforce building safety. Avoid windows, keep buffer zone around the site, bomb detection, camera surveillance, earthquake-proof, etc.
Use redundant utilities at multiple sites	Multiple power and supplies, alternate network connections, multiple databases at separate sites, data consistency, data watermarking, user authentication, etc.
Trust delegation and negotiation	Cross certificates to delegate trust across PKI domains for various data centers, trust negotiation among certificate authorities (CAs) to resolve policy conflicts
Worm containment and DDoS defense	Internet worm containment and distributed defense against DDoS attacks to secure all data centers and cloud platforms
Reputation system for data centers	Reputation system could be built with P2P technology; one can build a hierarchy of reputation systems from data centers to distributed file systems
Fine-grained file access control	Fine-grained access control at the file or object level; this adds to security protection beyond firewalls and IDSes
Copyright protection and piracy prevention	Piracy prevention achieved with peer collusion prevention, filtering of poisoned content, nondestructive read, alteration detection, etc.
Privacy protection	Uses double authentication, biometric identification, intrusion detection and disaster recovery, privacy enforcement by data watermarking, data classification, etc.

CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

Cloud Defense Methods

- VM failures do not propagate to other VMs.
- Hypervisor provides visibility of the guest OS, with complete guest isolation.
- Fault containment and failure isolation of VMs provide a more secure and robust environment.
- Malicious intrusions may destroy valuable hosts, networks, and storage resources.
- Internet anomalies found in routers, gateways, and distributed hosts may stop cloud services.
- Trust negotiation is often done at the SLA level. Public Key Infrastructure (PKI) services could be augmented with data-center reputation systems.
- Worm and DDoS attacks must be contained. It is harder to establish security in the cloud because all data and software are shared by default.

CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

Defense with Virtualization

- The VM is decoupled from the physical hardware.
  - The entire VM can be represented as a software component and can be regarded as binary or digital data.
- The VM can be saved, cloned, encrypted, moved, or restored with ease.
  - VMs enable HA and faster disaster recovery.
  - Live migration of VMs used building distributed intrusion detection systems (DIDSes).
- Multiple IDS VMs can be deployed at various resource sites including data centers.
- DIDS design demands trust among PKI domains. Security policy conflicts must be resolved at design time and updated periodically.

CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

Privacy and Copyright Protection

Here are several security features desired in a secure cloud:

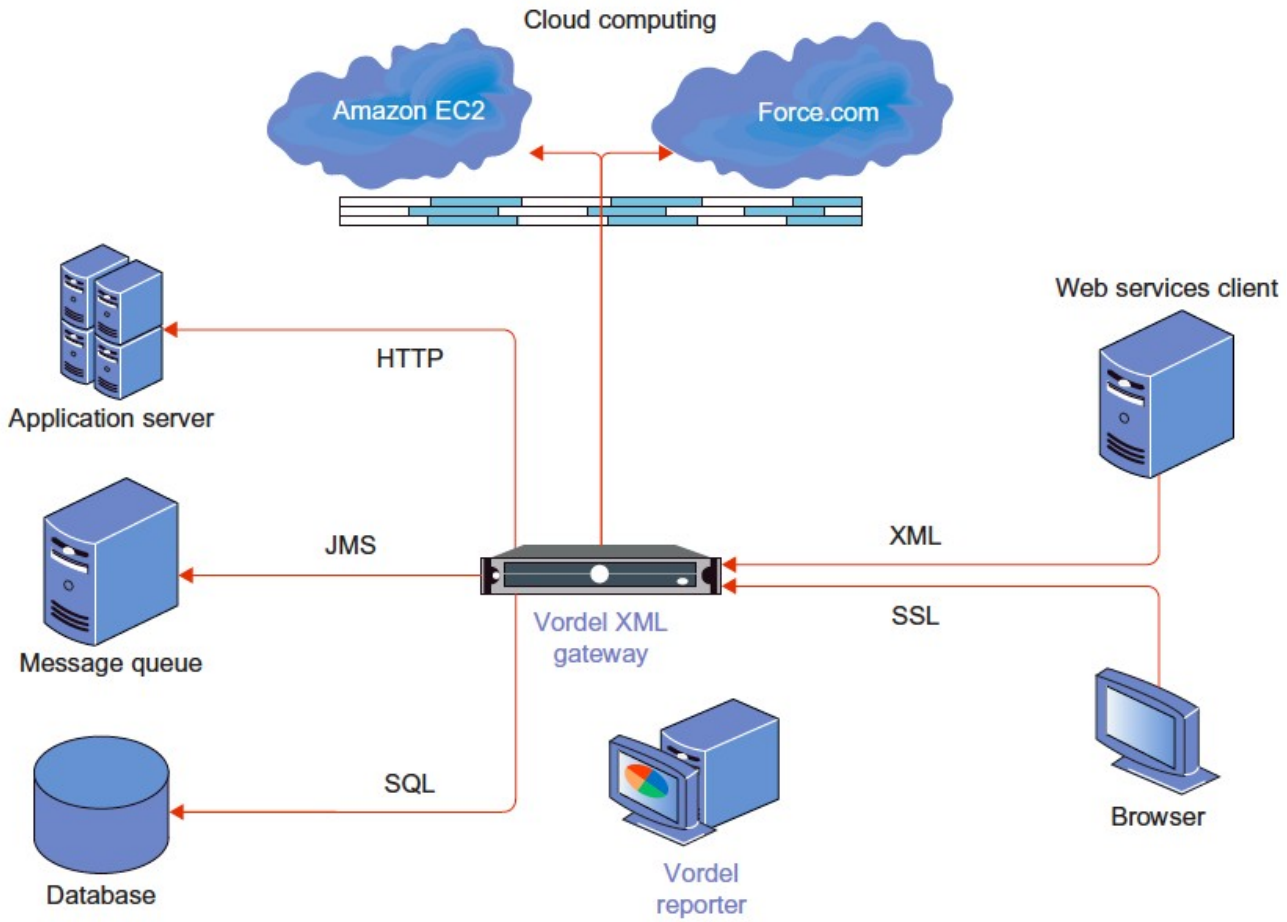
- ❑Dynamic web services with full support from secure web technologies.
- ❑Established trust between users and providers through SLAs and reputation systems.
- ❑Effective user identity management and data-access management.
- ❑Single sign-on and single sign-off to reduce security enforcement overhead.
- ❑Auditing and copyright compliance through proactive enforcement.
- ❑Shifting of control of data operations from the client environment to cloud providers.
- ❑Protection of sensitive and regulated information in a shared environment.

CLOUD SECURITY AND TRUST MANAGEMENT

Cloud Security Defense Strategies

A security defense system for a typical private cloud environment

- The firewall provides an external shield.
- The gateway secures the application server, message queue, database, web service client, and browser with HTTP, JMS, SQL, XML, and SSL security protocols, etc.



CLOUD SECURITY AND TRUST MANAGEMENT

Distributed Intrusion/Anomaly Detection

Distributed Defense against DDoS Flooding Attacks

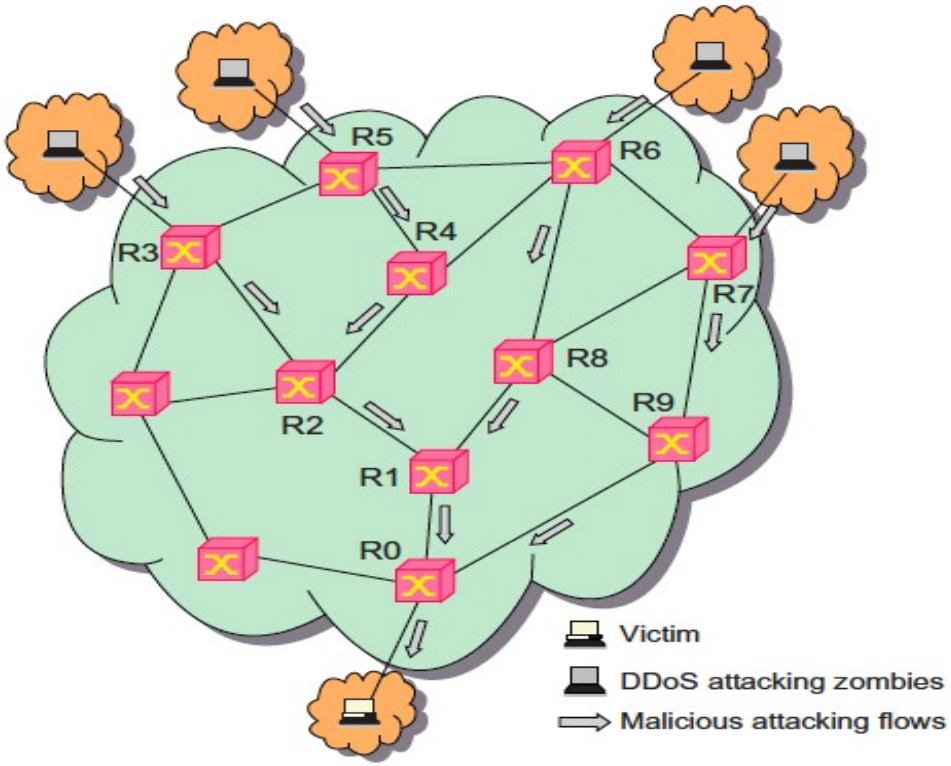
- A DDoS defense system must be designed to cover multiple network domains spanned by a given cloud platform.
- These network domains cover the edge networks where cloud resources are connected.
- DDoS attacks come with widespread worms.
- The flooding traffic is large enough to crash the victim server by buffer overflow, disk exhaustion, or connection saturation.

CLOUD SECURITY AND TRUST MANAGEMENT

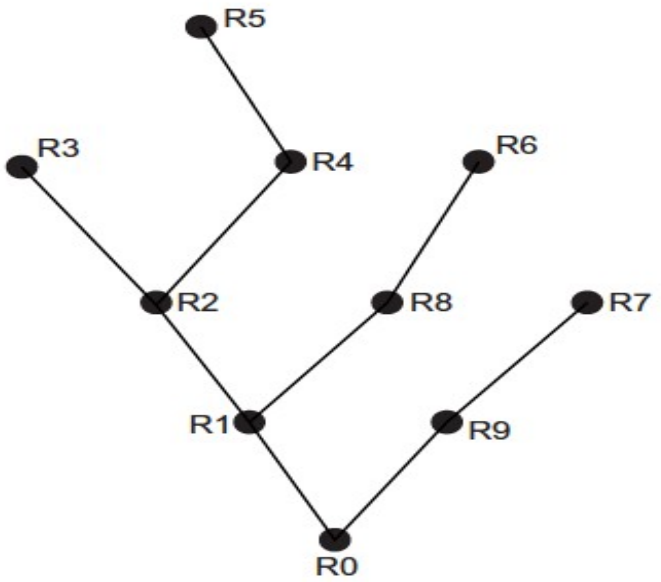
Distributed Intrusion/Anomaly Detection

Distributed Defense against DDoS Flooding Attacks

- Based on the anomaly pattern detected in covered
- network domains, the scheme detects a DDoS attack before the victim is overwhelmed.



(a) Traffic flow pattern of a DDoS attack

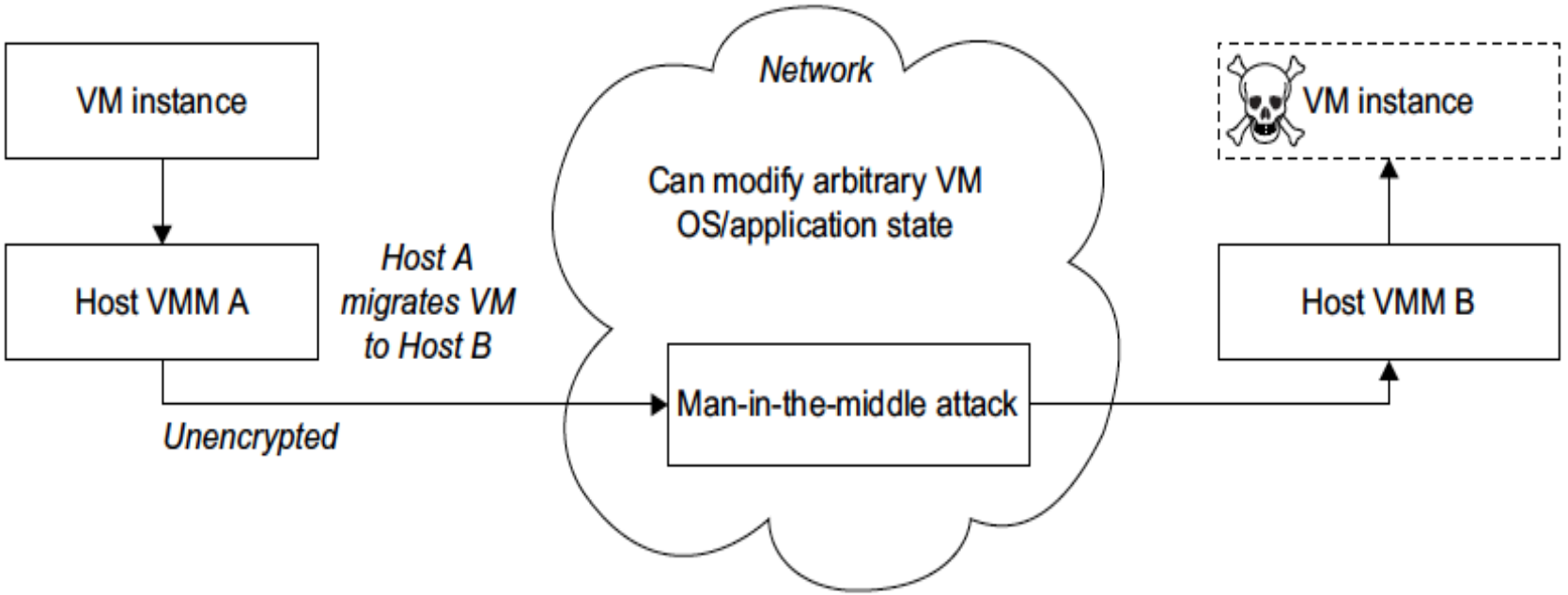


(b) The attack traffic flow tree over 10 routers

CLOUD SECURITY AND TRUST MANAGEMENT

Man-in-the-Middle Attacks

- The attacker can view the VM contents being migrated, steal sensitive data, or even modify the VM-specific contents including the OS and application states.
- The attacker can launch an active attack to insert a VM-based rootkit into the migrating VM, which can subvert the entire operation of the migration process without the knowledge of the guest OS and embedded application.



CLOUD SECURITY AND TRUST MANAGEMENT

Data and Software Protection Techniques

Data Integrity and Privacy Protection

- Users desire a software environment that provides many useful tools to build cloud applications over large data sets.
- In addition to application software for MapReduce, BigTable, EC2, S3, Hadoop, AWS, GAE, and WebSphere2, users need some security and privacy protection software for using the cloud.
- Such software should offer the following features:
  - Special APIs for authenticating users and sending e-mail using commercial accounts
  - Fine-grained access control to protect data integrity and deter intruders or hackers
  - Shared data sets protected from malicious alteration, deletion, or copyright violation
  - Ability to secure the ISP or cloud service provider from invading users' privacy
  - Personal firewalls at user ends to keep shared data sets from Java, JavaScript, and ActiveX applets
  - A privacy policy consistent with the cloud service provider' s policy, to protect against identity theft, spyware, and web bugs
  - VPN channels between resource sites to secure transmission of critical data objects

CLOUD SECURITY AND TRUST MANAGEMENT

Data and Software Protection Techniques

Data Coloring and Cloud Watermarking

- Data coloring means labeling each data object by a unique color.
  - Differently colored data objects are thus distinguishable.
- The user identification is also colored to be matched with the data colors.
- This color matching process can be applied to implement different trust management events.
- Cloud storage provides a process for the generation, embedding, and extraction of the watermarks in colored objects.
- In general, data protection was done by encryption or decryption which is computationally expensive.
- The data coloring takes a minimal number of calculations to color or decolor the data objects.
- Cryptography and watermarking or coloring can be used jointly in a cloud environment.

CLOUD SECURITY AND TRUST MANAGEMENT

Data and Software Protection Techniques

Data Lock-in Problem and Proactive Solutions

- Cloud computing moves both the computation and the data to the server clusters maintained by cloud service providers.
- Once the data is moved into the cloud, users cannot easily extract their data and programs from cloud servers to run on another platform.
  - This leads to a data lock-in problem.
- Data lock-in is attributed to two causes:
  - lack of interoperability, whereby each cloud vendor has its proprietary API that limits users to extract data once submitted; and
  - lack of application compatibility, in that most computing clouds expect users to write new applications from scratch, when they switch cloud platforms.
- One possible solution to data lock-in is the use of standardized cloud APIs.
- This requires building standardized virtual platforms that adhere to Open Virtualization Format (OVF), a platform-independent, efficient, extensible, and open format for VMs.
  - This will enable efficient, secure software distribution, facilitating the mobility of VMs.
  - Using OVF one can move data from one application to another.
- This will enhance QoS, and thus enable cross-cloud applications, allowing workload migration among data centers to user-specific storage.
- By deploying applications, users can access and intermix applications across different cloud services.

CLOUD SECURITY AND TRUST MANAGEMENT

Reputation-Guided Protection of Data Centers

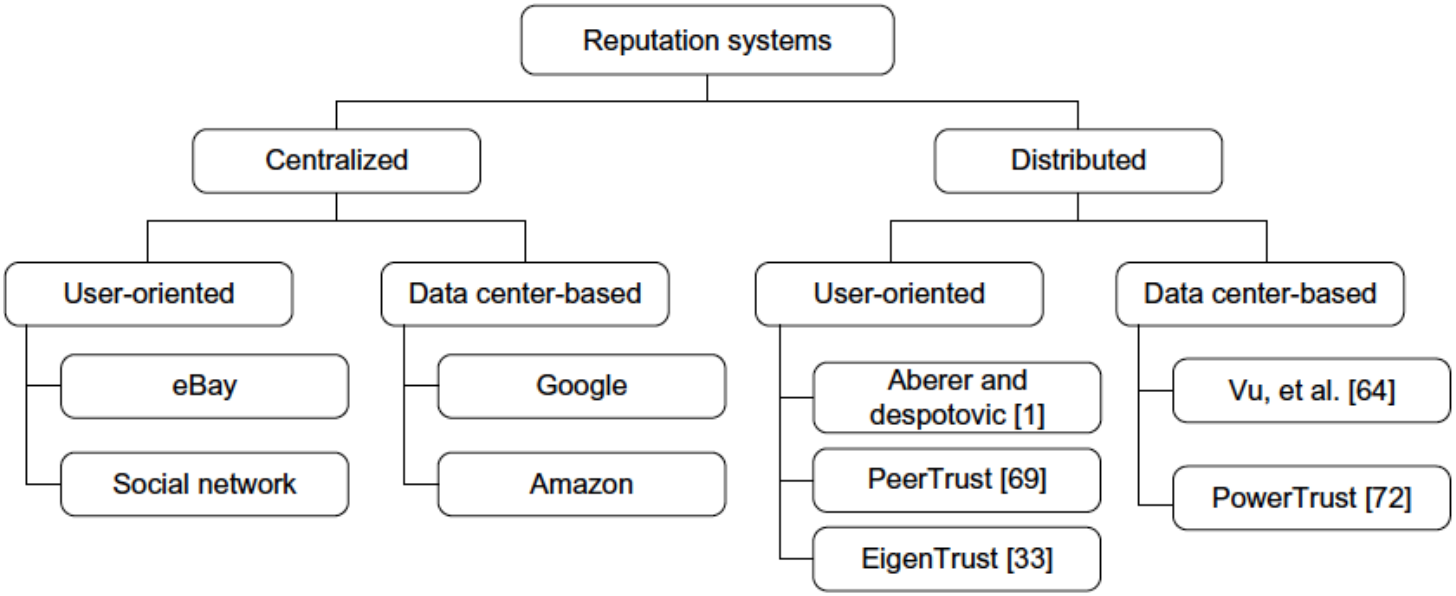
- Trust is a personal opinion, which is very subjective and often biased.
- Trust can be transitive but not necessarily symmetric between two parties.
- Reputation is a public opinion, which is more objective and often relies on a large opinion aggregation process to evaluate.
- Reputation may change or decay over time.
- Recent reputation should be given more preference than past reputation.
- In this section, we review the reputation systems for protecting data centers or cloud user communities.

CLOUD SECURITY AND TRUST MANAGEMENT

Reputation-Guided Protection of Data Centers

Reputation System Design Options

- The figure provides an overview of reputation system design options. Public opinion on the character or standing (such as honest behavior or reliability) of an entity could be the reputation of a person, an agent, a product, or a service.
- It represents a collective evaluation by a group of people/ agents and resource owners. Many reputation systems have been proposed in the past mainly for P2P, multiagent, or e-commerce systems.



CLOUD SECURITY AND TRUST MANAGEMENT

Reputation-Guided Protection of Data Centers

Reputation Systems for Clouds

- Redesigning the aforementioned reputation systems for protecting data centers offers new opportunities for expanded applications beyond P2P networks.
  - Data consistency is checked across multiple databases.
  - Copyright protection secures wide-area content distributions.
  - To separate user data from specific SaaS programs, providers take the most responsibility in maintaining data integrity and consistency.
  - Users can switch among different services using their own data.
  - Only the users have the keys to access the requested data.
- The data objects must be uniquely named to ensure global consistency.
  - Unauthorized updates of data objects by other cloud users are prohibited.
- Reputation systems can be used to support safe cloning of VMs.
- Sandboxes provide a safe execution platform for running programs.
  - They provide a tightly controlled set of resources for guest operating systems, which allows a security test bed to test the application code from third-party vendors.

CLOUD SECURITY AND TRUST MANAGEMENT

Reputation-Guided Protection of Data Centers

Trust Overlay Networks

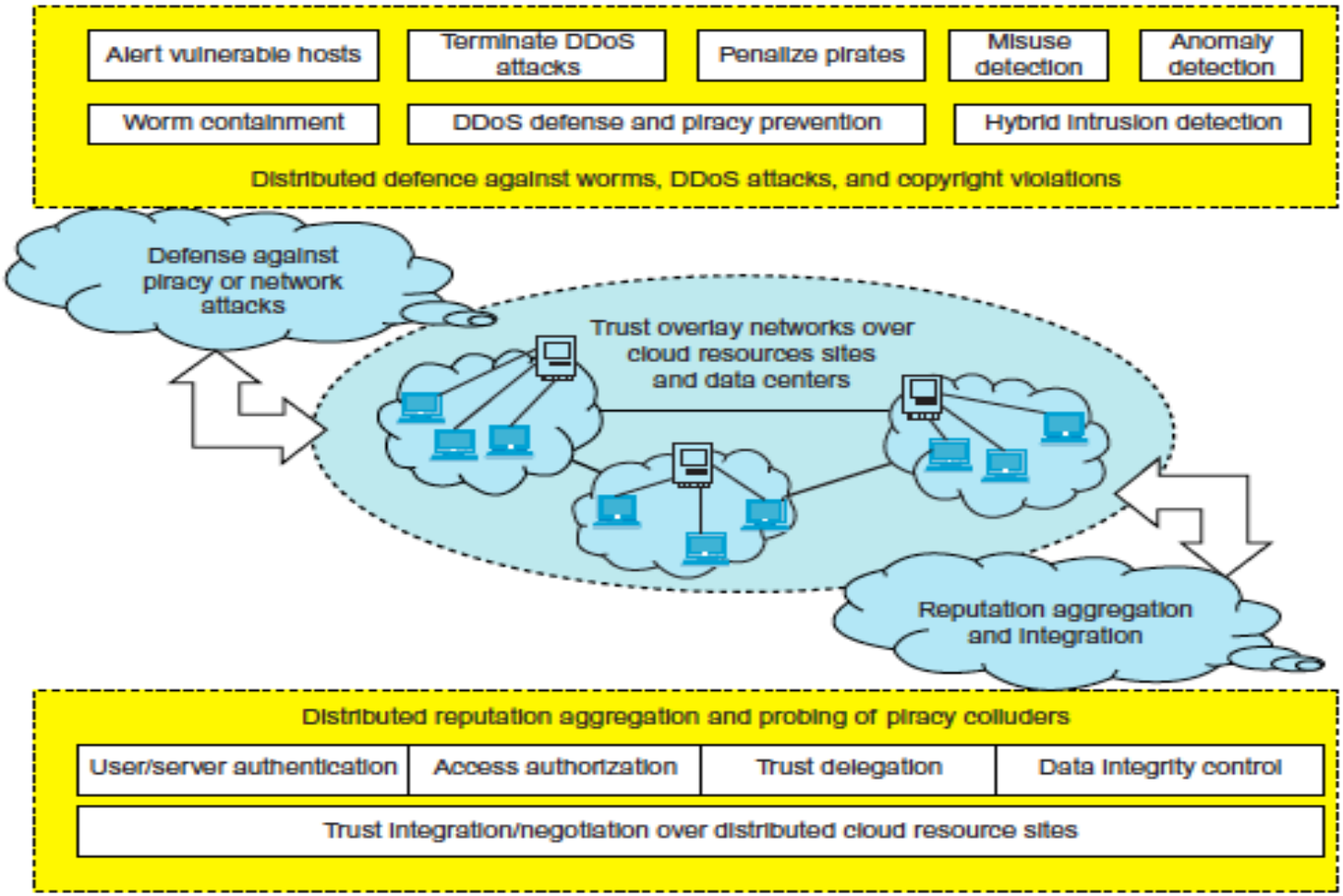
- Models trust relationships among data-center modules.
  - This trust overlay could be structured with a distributed hash table (DHT) to achieve fast aggregation of global reputations from a large number of local reputation scores.
- The designer needs to have two layers for fast reputation aggregation, updating, and dissemination to all users.

CLOUD SECURITY AND TRUST MANAGEMENT

Reputation-Guided Protection of Data Centers

Trust Overlay Networks

- Construction of the two layers of the trust overlay network.



CLOUD SECURITY AND TRUST MANAGEMENT

Reputation-Guided Protection of Data Centers

Trust Overlay Networks

- At the bottom layer is the trust overlay for distributed trust negotiation and reputation aggregation over multiple resource sites.
  - This layer handles user/server authentication, access authorization, trust delegation, and data integrity control.
- At the top layer is an overlay for fast virus/worm signature generation and dissemination and for piracy detection.
  - This overlay facilitates worm containment and IDSes against viruses, worms, and DDoS attacks.
- The reputation system enables trusted interactions between cloud users and data-center owners.
- Privacy is enforced by matching colored user identifications with the colored data objects.

CLOUD SECURITY AND TRUST MANAGEMENT

- The cloud security trend is to apply virtualization support for security enforcement in data centers.
- Both reputation systems and data watermarking mechanisms can:
  - protect data-center access at the coarse-grained level and
  - to limit data access at the fine-grained file level.
- In the long run, a new Security as a Service is desired.
  - This “SaaS” is crucial to the universal acceptance of web-scale cloud computing in personal, business, community, and government applications.
- Internet clouds are certainly in line with IT globalization and efficient computer outsourcing.
- Interoperability among different clouds relies on a common operational standard by building a healthy cloud ecosystem.

End of Slides