UNIT- III

CLOUD INFRASTRUCTURE

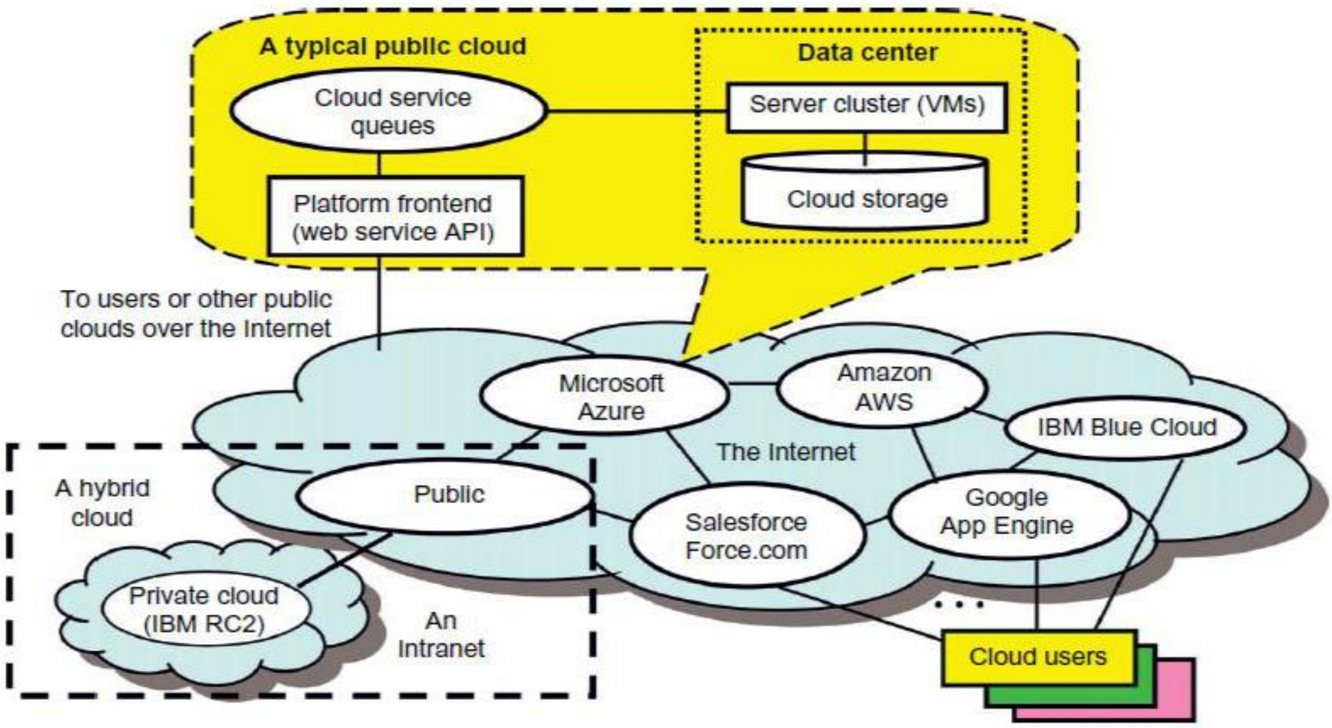
Topics Covered:

- Cloud Computing and Service Models
- Data-Center Design and Interconnection Networks
- Architectural Design of Compute and Storage Clouds
- Layered Cloud Architecture Development
- Architectural Design Challenges
- Inter Cloud Resource Management
- Resource Provisioning and Platform Deployment
- Global Exchange of Cloud Resources

⇒ Cloud Computing and Service Models

- ✓ **Clouds aim to power the next generation of data centers by architecting them as virtual resources over automated hardware, databases, user interfaces, and application environments.**
- ✓ A ***Public cloud*** is built over the ***Internet*** and can be **accessed by any user who has paid for the service**. Public clouds are owned by **service providers** and are accessible through a subscription.
- ✓ A ***Private*** **cloud** is built within the domain of an ***Intranet*** owned by a **single organization**.
- ✓ A ***Hybrid cloud*** is built with both public and private clouds.

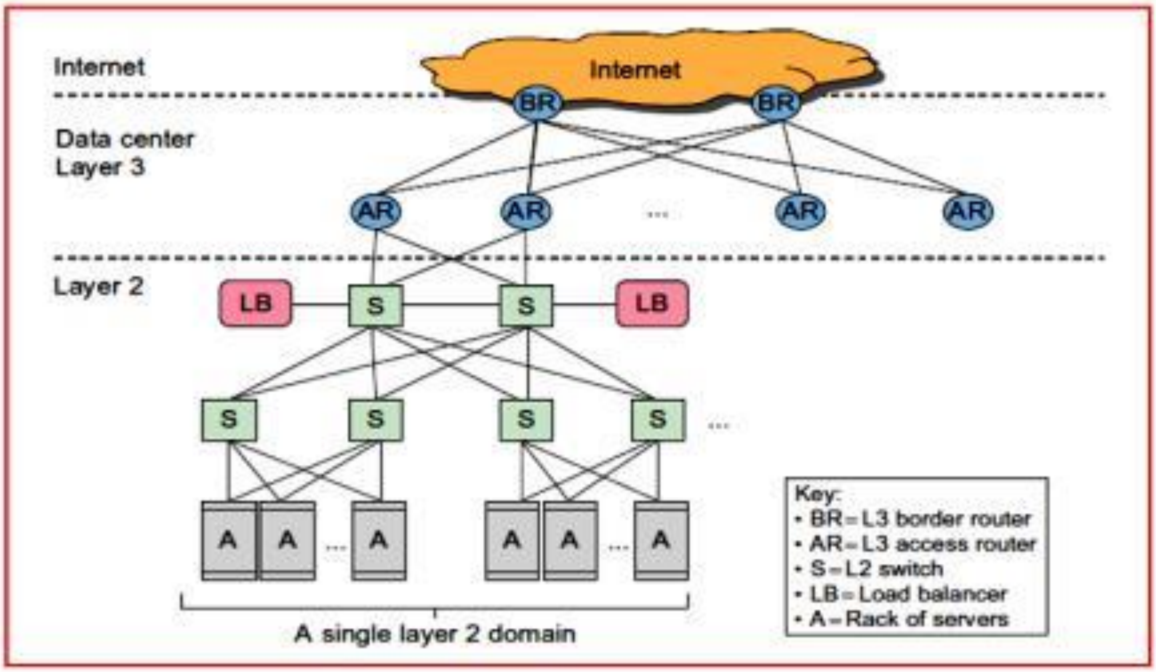
- *Public, Private, and Hybrid clouds illustrated by functional architecture and connectivity of representative clouds available.*



Data-Center Networking Structure

- 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.

- Data-center networks are mostly **IP-based commodity networks**, such as the 10 Gbps Ethernet networks, which is optimized for Internet access. 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). Figure shows a multilayer structure for accessing the Internet.



Cloud Design Objectives: The following list highlights six design objectives for cloud computing:

- **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.

⇒ Data-Center Design and Interconnection Networks

- **"The cloud is built on massive datacenters".** The data centers are built economics of scale— meaning **lower unit cost for larger data centers.** A small data center could have 1,000 servers. **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. These unit costs are greater than those of a 1,000-server data center. The network cost to operate a small data center is about seven times greater and the storage cost is 5.7 times greater. Microsoft has about 100 data centers, large or small, which are distributed around the globe.

Data-Center 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 **multi core 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**. 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.

- The data-center module is housed in a **truck-towable container**. The modular container design includes the **network, computer, storage, and cooling gear**. A modular data center built in a truck-towed ICE Cube container that can be cooled by chilled air circulation with cold-water heat exchanges.

Data-Center Management Issues: Here are basic requirements for managing the resources of a data center.

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

- **Low cost to both users and providers** The cost to users and providers of the cloud system built over the data centers should be reduced, including all operational costs.
- **Security enforcement and data protection** Data privacy and security defense mechanisms must be deployed to protect the data center against network attacks and system interrupts and to maintain data integrity from user abuses or network attacks.
- **Green information technology** Saving power consumption and upgrading energy efficiency are in high demand when designing and operating current and future data centers.

⇒ ARCHITECTURAL DESIGN OF COMPUTE AND STORAGE CLOUDS

A Generic Cloud Architecture Design

An Internet cloud is envisioned as a **public cluster of servers** provisioned on demand to perform collective web services or distributed applications using **data-center resources**.

❖ Cloud Platform design objectives:

There four major design goals of a cloud computing platform**:**

- o Scalability,
- o Virtualization,
- o Efficiency, and
- o Reliability

❖ Enabling Technologies for Clouds

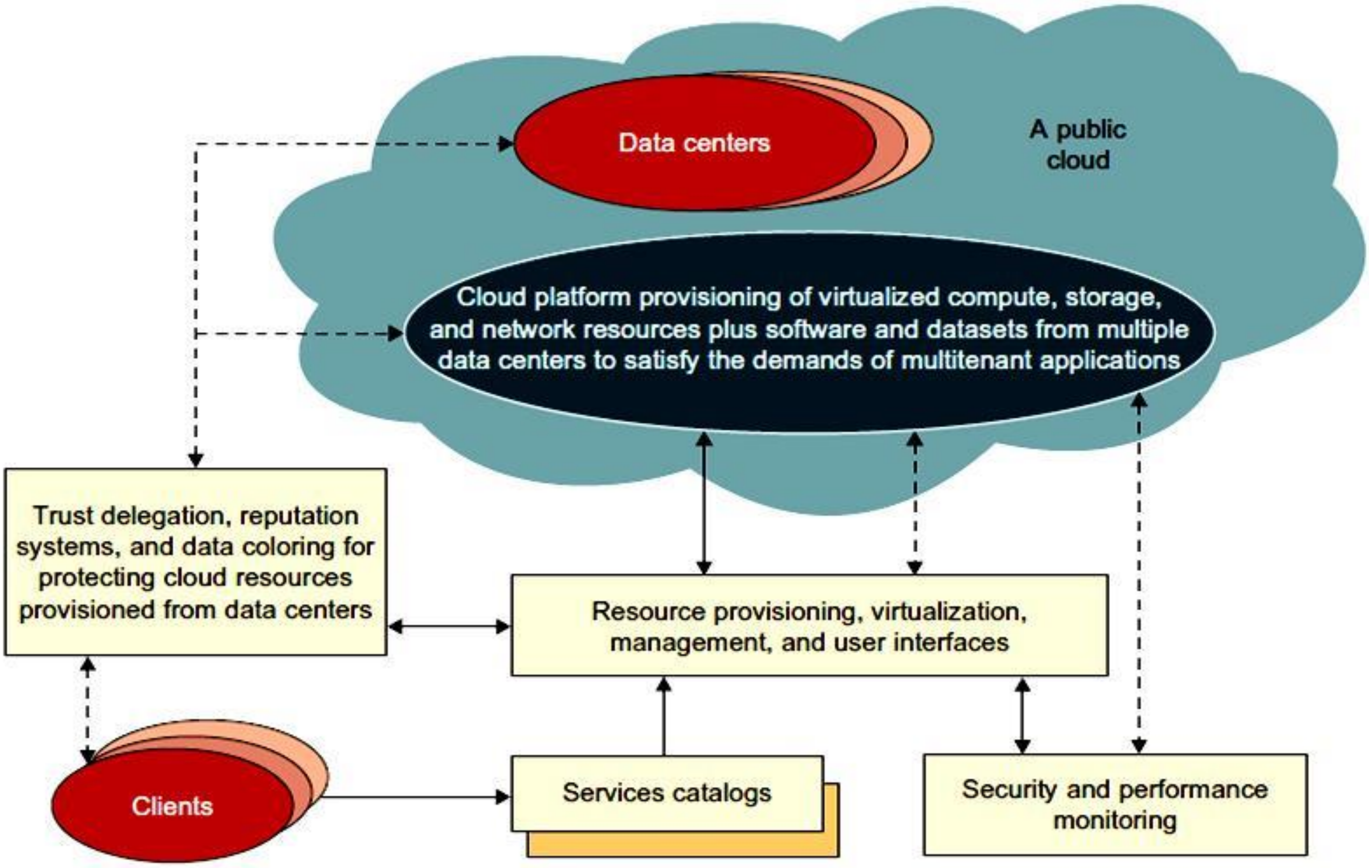
Table 4.3 Cloud-Enabling Technologies in Hardware, Software, and Networking

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

A Generic Cloud Architecture

Figure shows a security-aware cloud platform built with a **virtual cluster of VMs, storage, and networking resources over the data-center servers operated by providers**. The Internet cloud is envisioned as a massive cluster of servers. These servers are provisioned on demand to perform collective web services or distributed applications using data-center resources. The cloud platform is formed dynamically by provisioning or de provisioning 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.

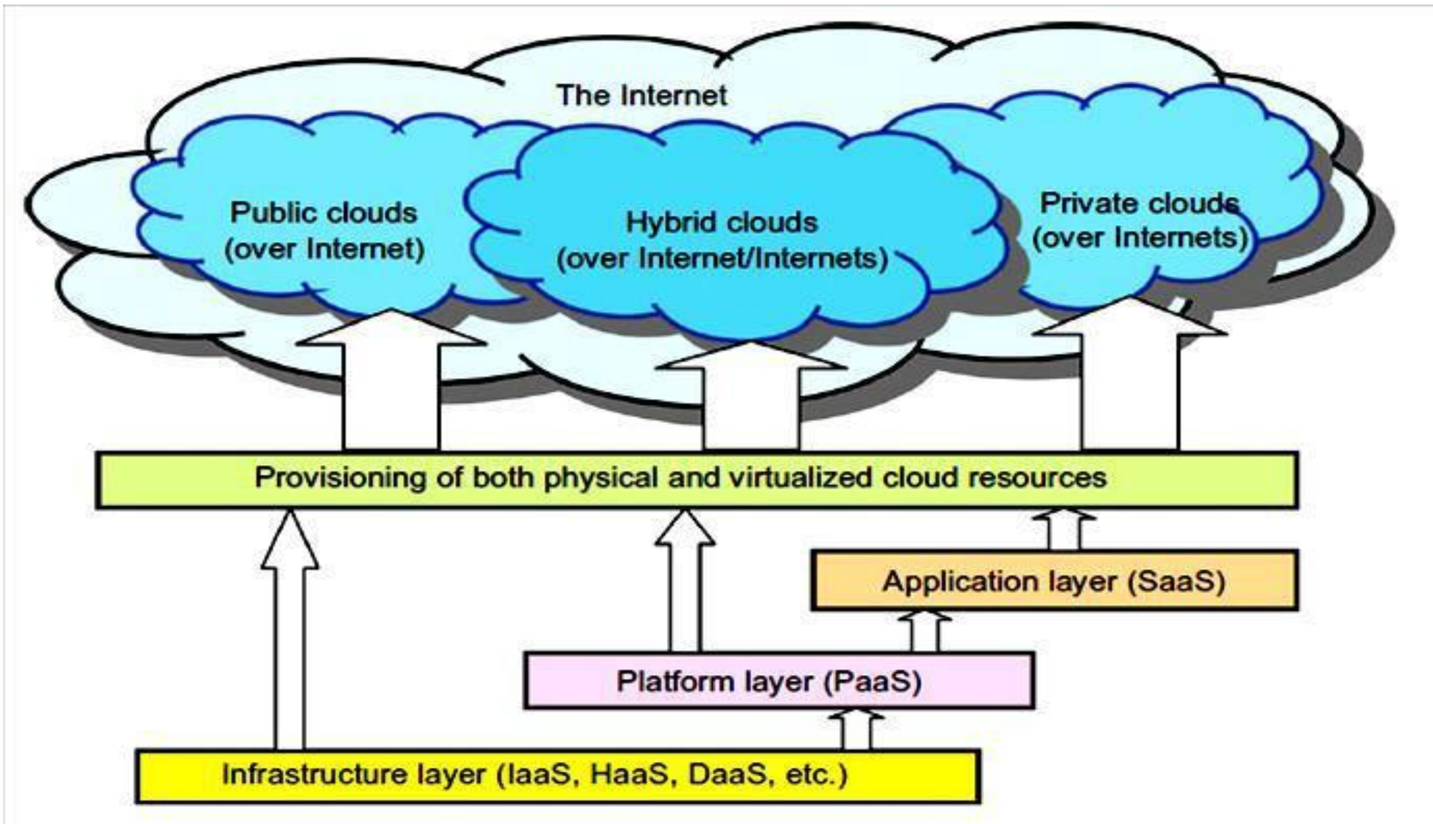
The cloud platform demands distributed storage and accompanying services. Other cloud resources are added into a cloud platform, including storage area networks (SANs), database systems, firewalls, and security devices.



⇒ Layered Cloud Architectural Development:

- The architecture of a cloud is developed at three layers: **infrastructure, platform, and application**, these three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud. The services to public, private, and hybrid clouds are conveyed to users through networking support over the Internet and Intranets involved.
  - **The infrastructure layer** is deployed first to support **IaaS** services, is built with **virtualized compute, storage, and network resources.**
  - **The platform layer** is a foundation for implementing the application layer for **SaaS** applications. The platform layer is for general-purpose and repeated usage of the collection of software resources. This layer provides users with an **environment to develop their applications**, **to test operation flows, and to monitor execution results and performance.**
    - **The application layer** is formed with a collection of all needed software modules for SaaS applications. Service applications in this layer include **daily office management work, such as information retrieval, document processing, and calendar and authentication services.** The

application layer is also heavily used by enterprises in **business marketing and sales, consumer relationship management (CRM), financial transactions, and supply chain management.**



❖ **Market-Oriented Cloud Architecture**

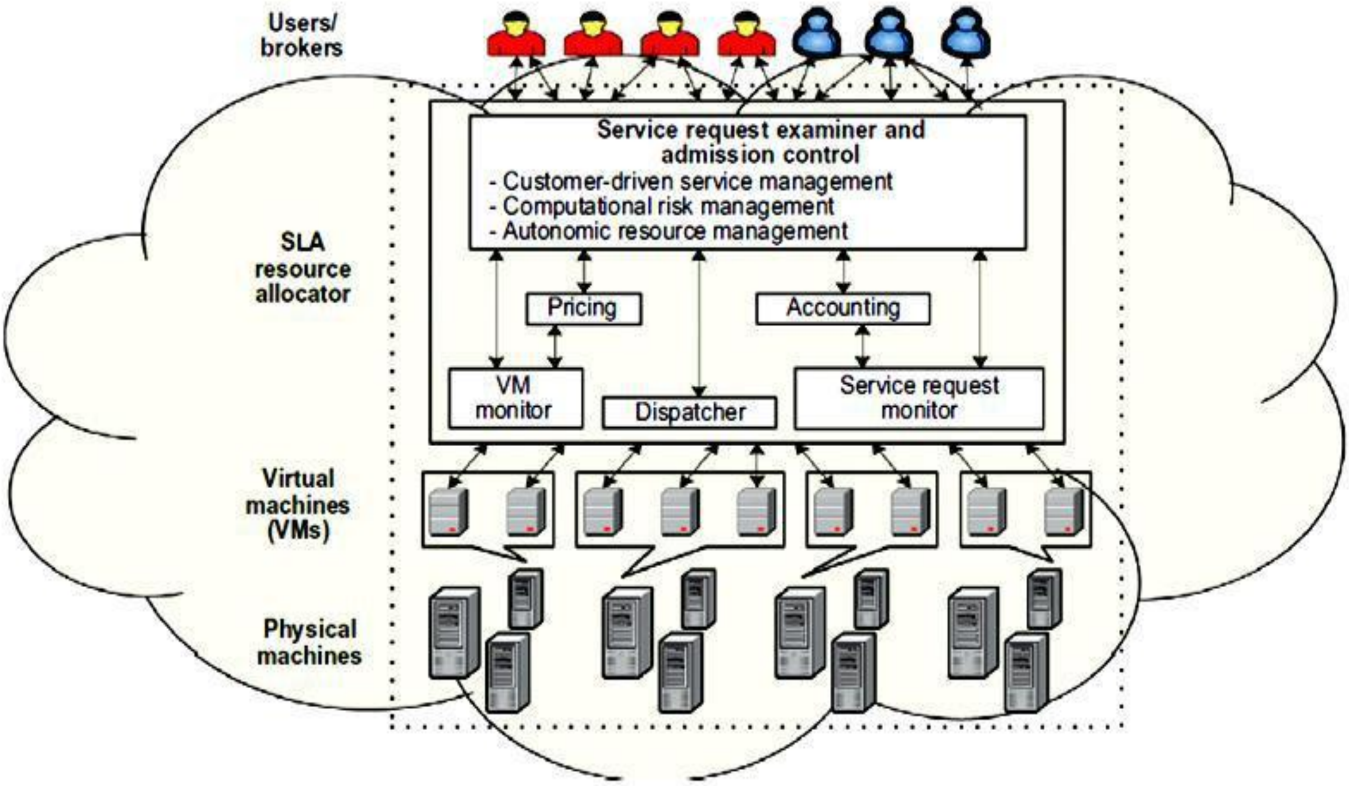
- The below figure shows the high-level architecture for supporting **market-oriented resource allocation** in a cloud computing environment. This cloud is basically built with the following entities:

- **Users or brokers** acting on user's behalf submit service requests from anywhere in the world to the data center 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 whereby many service requests cannot be fulfilled successfully due to limited resources. **The Pricing mechanism** decides
- how service requests are charged.
- **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.

- **Multiple VMs** can be started and stopped on demand on a single physical machine to meet accepted service requests, hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service requests.

- **Quality of Service Factors:** there are critical QOS parameters to consider in a service request, such as time, cost, reliability, and trust/security.

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



❖ Virtualization Support and Disaster Recovery

One very distinguishing feature of cloud computing infrastructure is the use of system **virtualization** and the modification to provisioning tools.

- Hardware Virtualization -Virtualized servers, storage, and network for cloud platform construction.
- Virtualization Support in Public Clouds
- Storage Virtualization for Green Data Centers
- Virtualization for IaaS
- VM Cloning for Disaster Recovery

⇒ **Architectural Design Challenges:** six open challenges in cloud architecture development

1) **Challenge 1—Service Availability and Data Lock-in Problem**

The source of single points of failure, to achieve high availability one can consider using multiple cloud providers. Using multiple cloud providers may provide more protection from failures. Another availability obstacle is distributed denial of service (DDoS) attacks. 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.

2) **Challenge 2—Data Privacy and Security Concerns**

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. Passive attacks steal sensitive data or passwords. Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.

3) **Challenge 3—Unpredictable Performance and Bottlenecks**

Multiple VMs can share CPUs and main memory in cloud computing, but I/O sharing is problematic. One solution is to **improve I/O architectures and operating systems** to efficiently virtualize interrupts and I/O channels. Therefore, data transfer bottlenecks must be removed, bottleneck links must be widened, and weak servers should be removed.

4) **Challenge 4—Distributed Storage and Widespread Software Bugs**

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. Data centers must meet programmers' expectations in terms of scalability, data durability, and HA. Large-scale distributed bugs cannot be reproduced, so the debugging must occur at a scale in the production data centers.

5) **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. Google App Engine (GAE) automatically scales in response to load increases and decreases; users are charged by the cycles used. Open Virtualization Format (OVF) describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of VMs. In terms of cloud standardization, we suggest the ability for virtual appliances to run on any virtual platform. We also need to enable VMs to run on heterogeneous hardware platform hypervisors.

6) **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. 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. 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.

⇒ Inter Cloud Resource Management

Here we discuss the various cloud service models and their extensions. The cloud service trends are outlined. Cloud resource management and inter cloud resource exchange schemes are reviewed.

Extended Cloud Computing Services:

Figure shows **six layers of cloud services**, ranging from hardware, network, and collocation to infrastructure, platform, and software applications. We already introduced the top three service layers as **SaaS, PaaS, and IaaS**, respectively. The bottom three layers are more related to physical requirements.

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. Some authors say this layer provides Security as a Service ("SaaS"). 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 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

The cloud players are divided into three classes: **(1) cloud service providers and IT administrators**, (2) **software developers or vendors**, and (3) **end users or business users.** These cloud players vary in their roles under the IaaS, PaaS, and SaaS models. The table entries distinguish the three cloud models as viewed by different players.

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

❖ Cloud Service Tasks and Trends

Cloud services are introduced in five layers.

The top layer is for SaaS applications, as further subdivided into the five application areas in above Figure, mostly for business applications. For example, CRM is heavily practiced in business promotion, direct sales, and marketing services. CRM offered the first SaaS on the cloud successfully.

PaaS is provided by Google, Salesforce.com, and Facebook, among others. IaaS is provided by Amazon, Windows Azure, and RackRack, among others.

Network cloud services provide communications such as those by AT&T, Qwest, and AboveNet.

❖ Software Stack for Cloud Computing

Despite the various types of nodes in the cloud computing cluster, the overall software stacks are built from scratch to meet rigorous goals (see above Table). By using VMs, the platform can be flexible, that is, the running services are not bound to specific hardware platforms. Other layers running on top of the file system are the layers for executing cloud computing applications. They include the **database storage system, programming for large-scale clusters, and data query language support**. The next layers are the components in the software stack.

❖ Runtime Support Services

Cluster monitoring is used to collect the runtime status of the entire cluster. One of the most important facilities is the cluster job management system. The scheduler queues the tasks submitted to the whole cluster and assigns the tasks to the processing nodes according to node availability. Runtime support is software needed in browser-initiated applications applied by thousands of cloud customers. The **SaaS** model provides the **software applications as a service**, rather than letting users purchase the software.

⇒ Resource Provisioning and Platform Deployment

In this section, we will discuss techniques to **provision computer resources** or **VMs**.

➤ Provisioning of Compute Resources (VMs)

Providers supply cloud services by signing **SLAs** with end users. The SLAs must commit sufficient resources such as **CPU, memory, and bandwidth** that the user can use for a preset period. Under provisioning of resources will lead to broken SLAs and penalties. Over provisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider.

Efficient VM provisioning depends on the **cloud architecture and management of cloud infrastructures**. Resource provisioning schemes also demand fast discovery of services and data in cloud computing infrastructures. Public or Private clouds promise to streamline the on-demand provisioning of software, hardware, and data as a service, achieving economies of scale in IT deployment and operation.

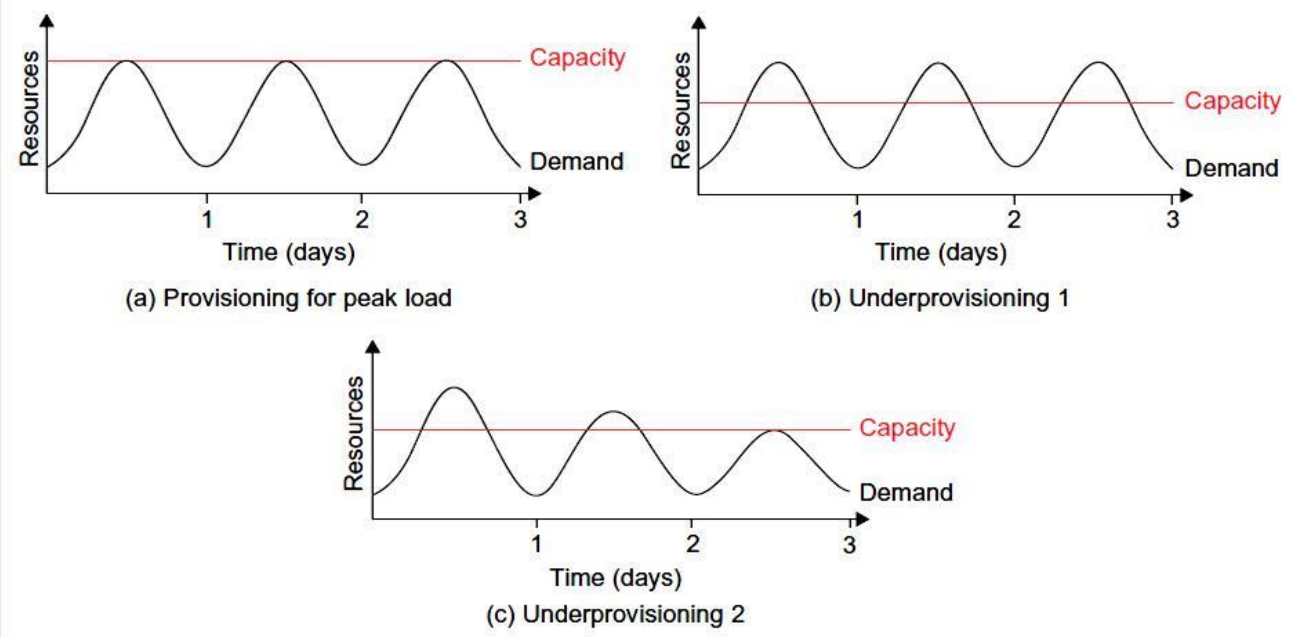
➤ Resource-Provisioning Methods

Figure shows **three cases of static cloud resource provisioning policies**.

- ✓ In case (a), **over provisioning** with the peak load causes heavy resource waste (shaded area).

- ✓ In case (b), **under provisioning** (along the capacity line) of resources results in losses by both user and provider in that paid demand by the users (the shaded area above the capacity) is not served and wasted resources still exist for those demanded areas below the provisioned capacity.

- ✓ In case (c), **under- and then over provisioning**, the constant provisioning of resources with fixed capacity to a declining user demand could result in even worse resource waste. The user may give up the service by canceling the demand, resulting in reduced revenue for the provider. Both the user and provider may be losers in resource provisioning without elasticity.



(a) Provisioning for peak load
(b) Underprovisioning 1
(c) Underprovisioning 2

Three resource-provisioning methods are presented in the following sections.

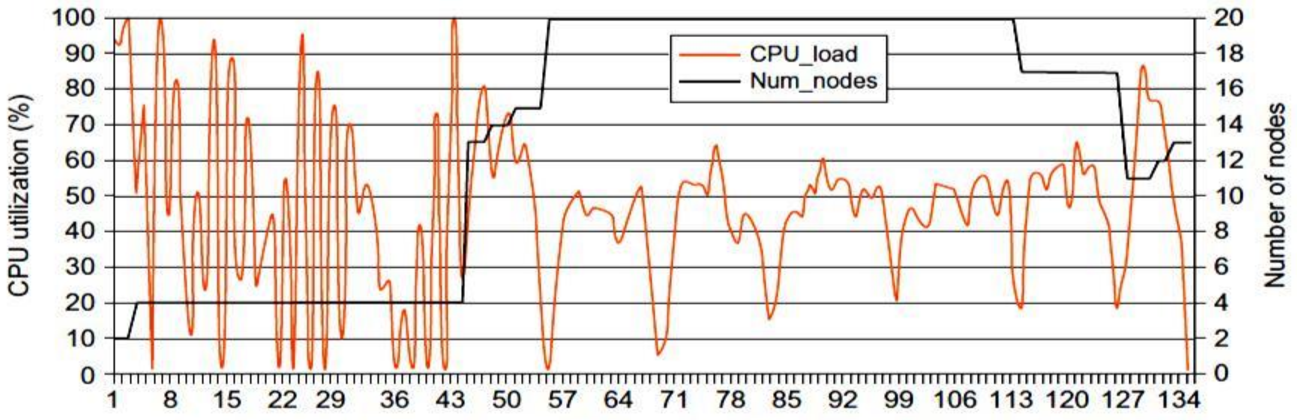
- The **demand-driven method** provides static resources and has been used in grid computing for many years.
- The **event driven method** is based on predicted workload by time.
- The **popularity-driven method** is based on Internet traffic monitored.

1) Demand-Driven Resource Provisioning

This method adds or removes computing instances **based on the current utilization level of the allocated resources.** The demand-driven method automatically allocates two Xeon processors for the user application, when the user was using one Xeon processor more than 60 percent of the time for an extended period. In general, when a resource has surpassed a threshold for a certain amount of time, the scheme increases that resource based on demand.

The x-axis in Figure is the time scale in milliseconds. In the beginning, **heavy fluctuations of CPU load** are encountered. All three methods have demanded a few VM instances initially.

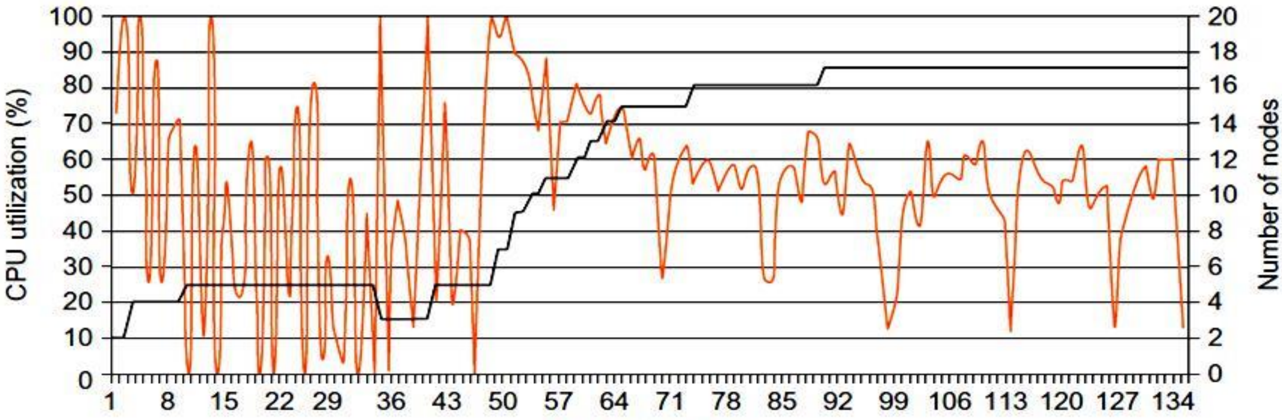
Gradually, the utilization rate becomes more stabilized with a maximum of 20 VMs (100 percent utilization) provided for demand-driven provisioning in Figure (a). However, the event-driven method reaches a stable peak of 17 VMs toward the end of the event and drops quickly in Figure (b). The popularity provisioning shown in Figure (c) leads to a similar fluctuation with peak VM utilization in the middle of the plot.



(a) Demand-driven

2) Event-Driven Resource Provisioning

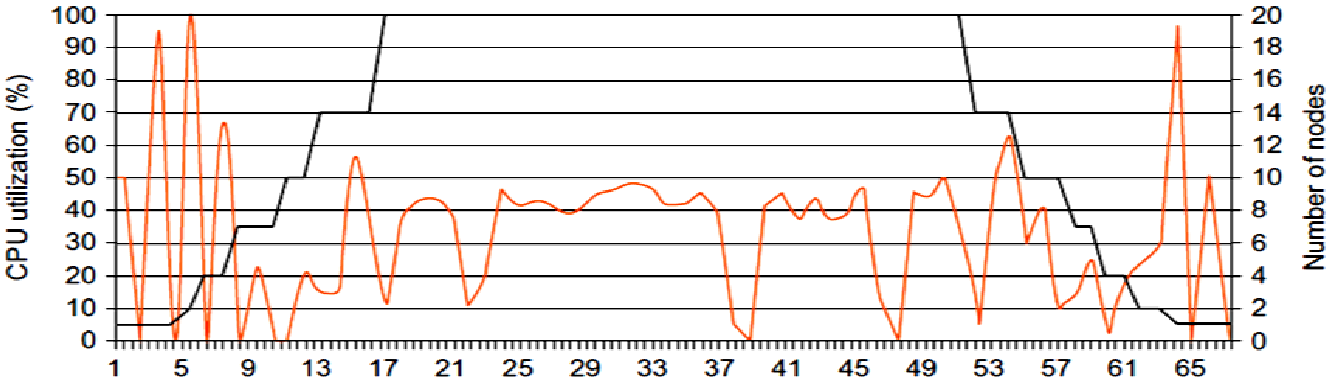
This scheme **adds or removes machine instances based on a specific time event**. This scheme anticipates peak traffic before it happens. The method results in a minimal loss of QOS, if the event is predicted correctly. Otherwise, wasted resources are even greater due to events that do not follow a fixed pattern.



(b) Event-driven

3) Popularity-Driven Resource Provisioning

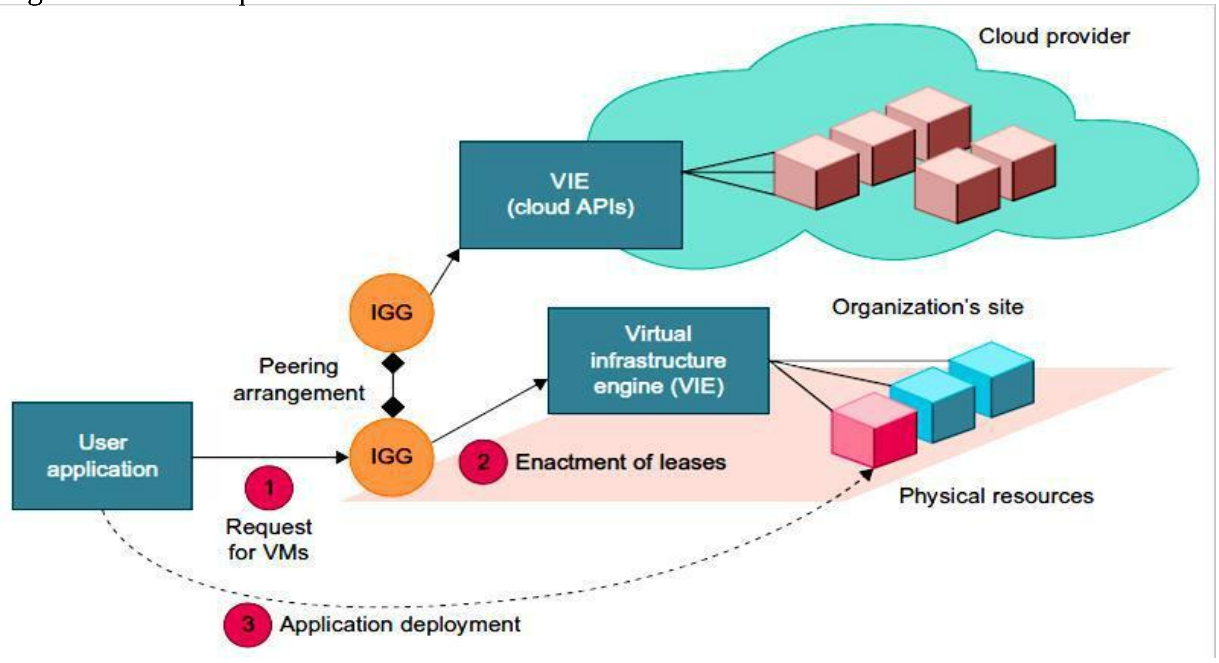
In this method, the **Internet searches for popularity of certain applications and creates the instances by popularity demand.** The scheme anticipates increased traffic with popularity. Again, the scheme has a minimal loss of QoS, if the predicted popularity is correct. Resources may be wasted if traffic does not occur as expected.



(c) Popularity-driven

Dynamic Resource Deployment

The cloud uses VMs as building blocks to create an execution environment across multiple resource sites. The **Inter-Grid is a Java-implemented software system that lets users create execution cloud environments on top of all participating grid resources.** Peering arrangements established between gateways enable the allocation of resources from multiple grids to establish the execution environment. In Figure, a scenario is illustrated by which an Inter Grid Gateway (IGG) allocates resources from a local cluster to deploy applications in three steps: (1) requesting the VMs, (2) enacting the leases, and (3) deploying the VMs as requested.



Provisioning of Storage Resources

The data storage layer is built on top of the physical or virtual servers. As the cloud computing applications often provide service to users, it is unavoidable that the data is stored in the clusters of the cloud provider. One example is **e-mail systems**.

A distributed file system is very important for **storing large-scale data**. Many cloud computing companies have developed large-scale data storage systems to keep huge amount of data collected every day. For example, see the below table:

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.

⇒ Global Exchange of Cloud Resources

- In order to support a large number of application service consumers from around the world, cloud infrastructure providers (i.e., IaaS providers) have established data centers in multiple geographical locations to provide redundancy and ensure reliability in case of site failures.

- For example, **Amazon** has data centers in the United States (e.g., one on the East Coast and another on the West Coast) and Europe. However, currently Amazon expects its cloud customers (i.e., SaaS providers) to express a preference regarding where they want their application services to be hosted. Amazon does not provide seamless/automatic mechanisms for scaling its hosted services across multiple geographically distributed data centers.

- This approach has many shortcomings. First, it is difficult for cloud customers to determine in advance the best location for hosting their services as they may not know the origin of consumers of their services. Second, SaaS providers may not be able to meet the QoS expectations of their service consumers originating from multiple geographical locations.

- This necessitates building mechanisms for seamless federation of data centers of a cloud provider or providers supporting dynamic scaling of applications across multiple domains in order to meet QoS targets of cloud customers.

- Cloud infrastructure provider would like to make use of services of multiple cloud infrastructure service providers who can provide better support for their specific consumer needs. This kind of requirement often arises in enterprises with global operations and applications such as Internet services, media hosting, and Web 2.0 applications.

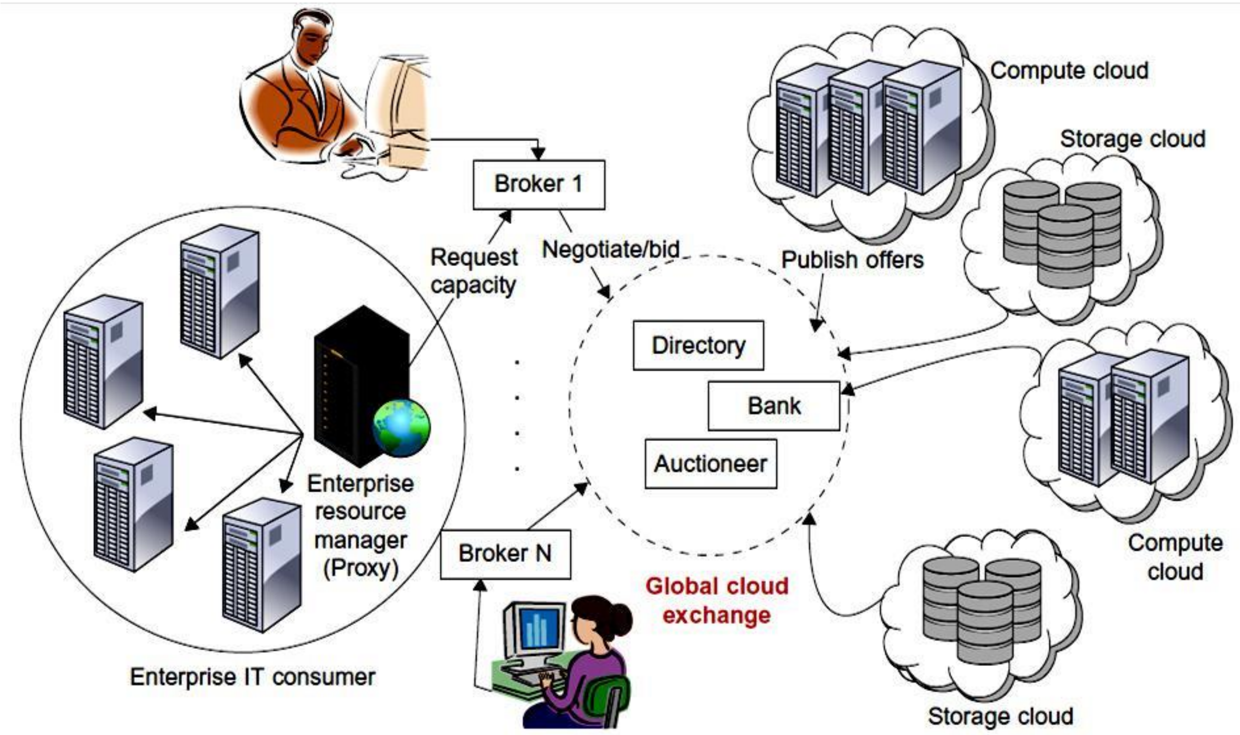


Fig: Inter-cloud exchange of cloud resources through brokering

- Cloud providers will be able to dynamically expand or resize their provisioning capability based on sudden spikes in workload demands by leasing available computational and storage capabilities from other cloud service providers; operate as part of a market-driven resource leasing federation, where application service providers such as Salesforce.com host their services based on negotiated SLA contracts driven by competitive market prices; and deliver on-demand, reliable, cost-effective, and QoS-aware services based on virtualization technologies while ensuring high QoS standards and minimizing service costs. They need to be able to utilize market-based utility models as the basis for provisioning of virtualized software services and federated hardware infrastructure among users with heterogeneous applications.

- They consist of client brokering and coordinator services that support utility-driven federation of clouds: application scheduling, resource allocation, and migration of workloads. The architecture cohesively couples the administratively and topologically distributed storage and compute capabilities of clouds as part of a single resource leasing abstraction. The system will ease the cross domain capability integration for on-demand, flexible, energy-efficient, and reliable access to the infrastructure based on virtualization technology.

- The Cloud Exchange (CEx) acts as a market maker for bringing together service producers and consumers. It aggregates the infrastructure demands from application brokers and evaluates them against the available supply currently published by the cloud coordinators.

- An SLA specifies the details of the service to be provided in terms of metrics agreed upon by all parties, and incentives and penalties for meeting and violating the expectations, respectively.