



Module 3

CLOUD ARCHITECTURE AND RESOURCE MANAGEMENT

EMERGING CLOUD SOFTWARE ENVIRONMENTS

- ❑ Eucalyptus, Nimbus & OpenNebula are major Open-Source Cloud-Computing Software Platforms
- ❑ Manage the provisioning of virtual machines for a cloud providing IaaS
- ❑ Commercial cloud services charge, by the hour, for CPU time
- ❑ It might be more cost effective for the organization to purchase hardware to create its own private cloud.



- •These software products are designed to allow an organization to set up a private group of machines as their own cloud.



EMERGING CLOUD SOFTWARE ENVIRONMENTS

- Open Source architectures available:
- Eucalyptus
- Nimbus
- Open Stack



EUCALYPTUS- ARCHITECTURE

- ❑ Open software environment
- ❑ Compatible with Amazon Web Services
- ❑ Linux based software architecture
- ❑ Components:- EC2 and S3
- ❑ EC2- Elastic Cloud Compute
- ❑ S3- Simple Storage System
- ❑ Provide High Performance Computing
- ❑ Provide vm image management



COMPONENTS

- ❑ Cluster Controller (CC)
- ❑ Cloud Controller (CLC)
- ❑ Node Controller (NC)
- ❑ Walrus
- ❑ Storage Controller (SC)



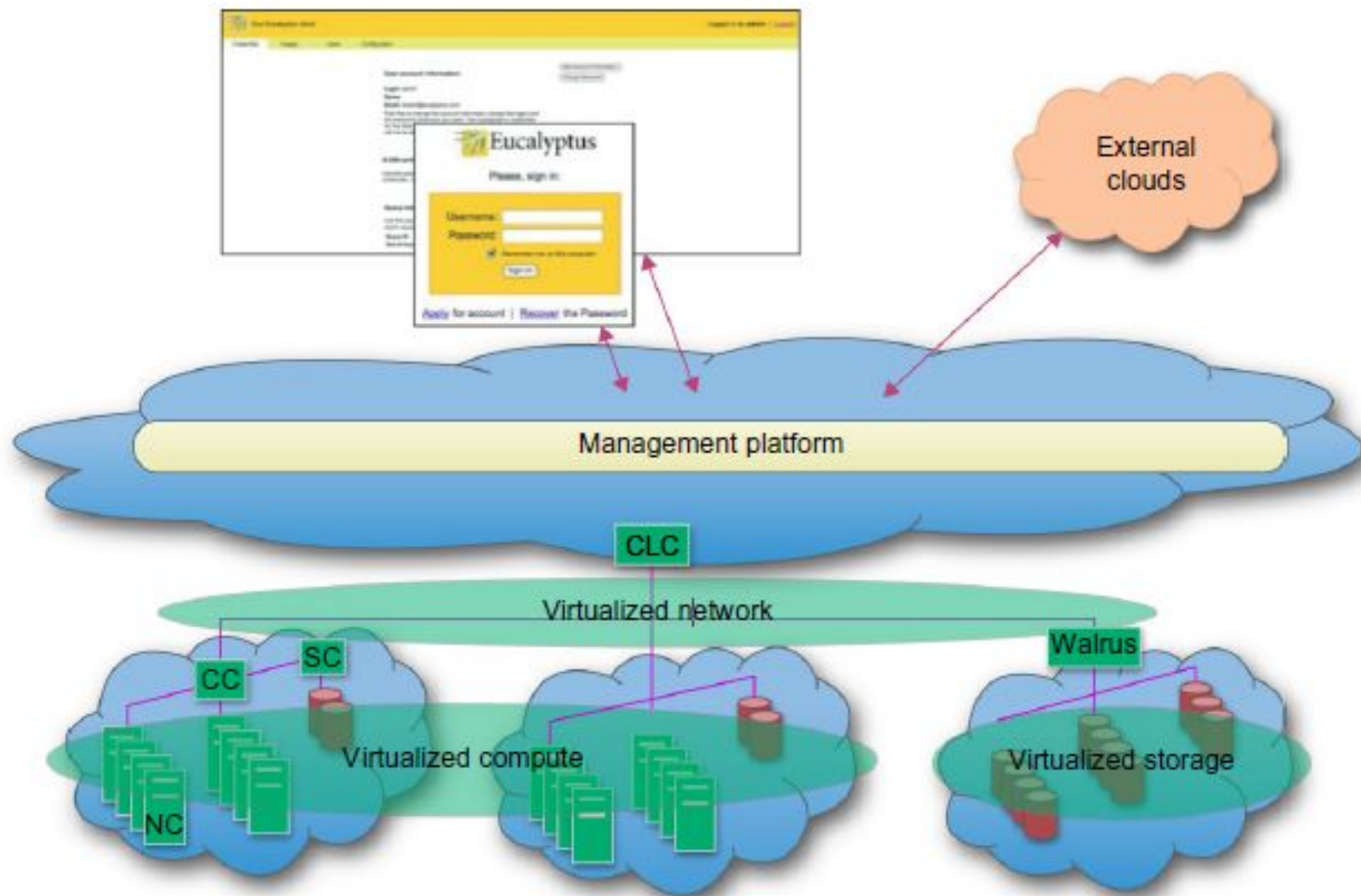


FIGURE 6.26

The Eucalyptus architecture for hybrid cloud computing.

CLUSTER CONTROLLER

- ❑ CC has got many number of node controller
- ❑ No limit
- ❑ NC performs each and every operations it is intended to be
- ❑ CC along with many node controllers make 1 cluster
- ❑ Each cluster will have a cc and a sc



- CC is controlled by CLC
- SC is connected to walrus
- Walrus – storage mechanism



CC

- ❑ Manages the one or more node controllers
- ❑ Communicates with node controller and CLC
- ❑ Manages the networking between them
- ❑ Intermediatery



CLC

- Front end
- Application hits here
- It is compliant to Amazons S3 and EC2
- Connected to client via web services on one side
- Connected to rest of the components of Eucalyptus on other side
- Basically UI



NC

- ❑ Smallest component
- ❑ Granula level
- ❑ Manages the life cycle of all instances initiated on this
- ❑ One NC takes care of entire activity
- ❑ Inspected
- ❑ NC Interacts with OS,Hypervisor,CLC



WALRUS

- ❑ Storage system
- ❑ Used for creating machine images and snapshots
- ❑ Block storage system
- ❑ Snapshots act as Check points used for system recovery in case of failure
- ❑ Images – A Collection of software modules like application softwares , system softwares, system configurations



SC

- ❑ Walrus is a part of SC
- ❑ Persistent block storage



- Available in commercial version and open source
- Any user can send his root file system and upload and register his image and link it with a particular kernel



NIMBUS

- ❑ Nimbus is a set of open source tools that together provide an IaaS cloud computing solution.
- ❑ Figure 6.27 shows the architecture of Nimbus, which allows a client to lease remote resource sby deploying VMs on those resources and configuring them to represent the environment desired by the user.



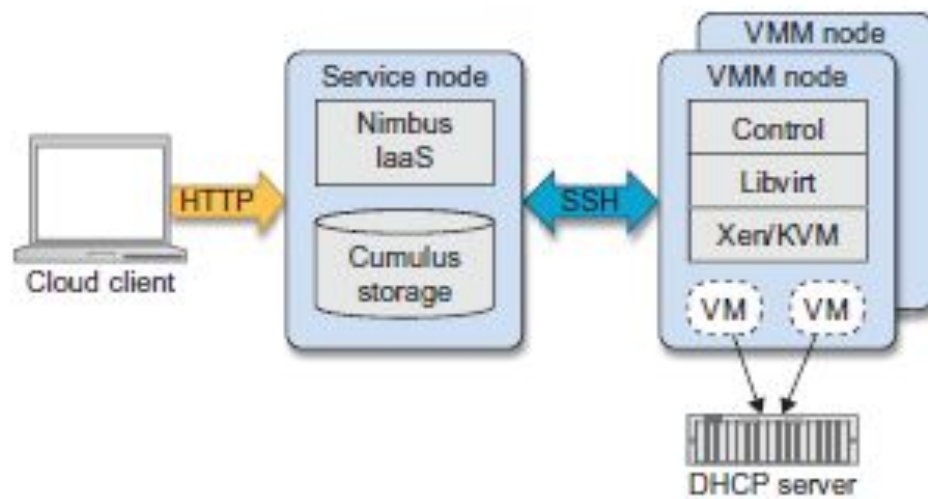


FIGURE 6.27

Nimbus cloud infrastructure.

(Courtesy of Nimbus Project [82])

- To this end, Nimbus provides a special web interface known as Nimbus Web .
- Its aim is to provide administrative and user functions in a friendly interface.
- Nimbus Web is centered around a Python Django web application that is intended to be deployable completely separate from the Nimbus service.



- a storage cloud implementation called Cumulus has been tightly
- integrated with the other central services, although it can also be used stand-alone.
- Cumulus is compatible with the Amazon S3 REST API, but extends its capabilities by including features such as quota management.



- Nimbus supports two resource management strategies.
- Resource Pool Mode
 - In this mode, the service has direct control of a pool of VM manager nodes and it assumes it can start VMs.



- Pilot Mode
- Here, the service makes requests to a
- cluster's Local Resource Management System (LRMS) to get a VM manager available to deploy VMs



OPENSTACK

- ❑ OpenStack was introduced by Rackspace and NASA in July 2010
- ❑ The project is building an open source community, spanning technologists, developers, researchers, and industry to share resources and technologies with the goal of creating a massively scalable and secure cloud infrastructure.
- ❑ The software is open source and limited to just open source APIs such as Amazon.



- Open Stack focuses on the development of two aspects of cloud computing to address compute and storage aspects with the
 - Open Stack Compute
 - Open Stack Storage solutions



- ❑ OpenStack Compute is the internal fabric of the cloud creating and managing large groups of virtual private servers
- ❑ Open Stack Object Storage is software for creating redundant, scalable object storage using clusters of commodity servers to store terabytes or even petabytes of data



- Recently, an image repository was prototyped.
- The Image service enables users to discover, register, and retrieve virtual machine images
- The image repository contains an image registration and discovery service and an image delivery service.
- Together they deliver images to the compute service while obtaining them from the storage service



OPENSTACK COMPUTE

- ❑ OpenStack is developing a cloud computing fabric controller, a component of an IaaS system, known as Nova
- ❑ The architecture for Nova is built on the concepts of **shared-nothing and messaging-based information exchange.**
- ❑ Hence, most communication in Nova is facilitated by message queues.



MESSAGING-BASED INFORMATION EXCHANGE

- To prevent blocking components while waiting for a response from others, deferred objects are introduced.
- Such objects include callbacks that get triggered when a response is received.



SHARED-NOTHING PARADIGM

- The overall system state is kept in a distributed data system.
- State updates are made consistent through atomic transactions.
- Nova is implemented in Python while utilizing a number of externally supported libraries and components
- This includes boto, an Amazon API provided in Python, and Tornado, a fast HTTP server



ARCHITECTURE OF OPEN STACK COMPUTE

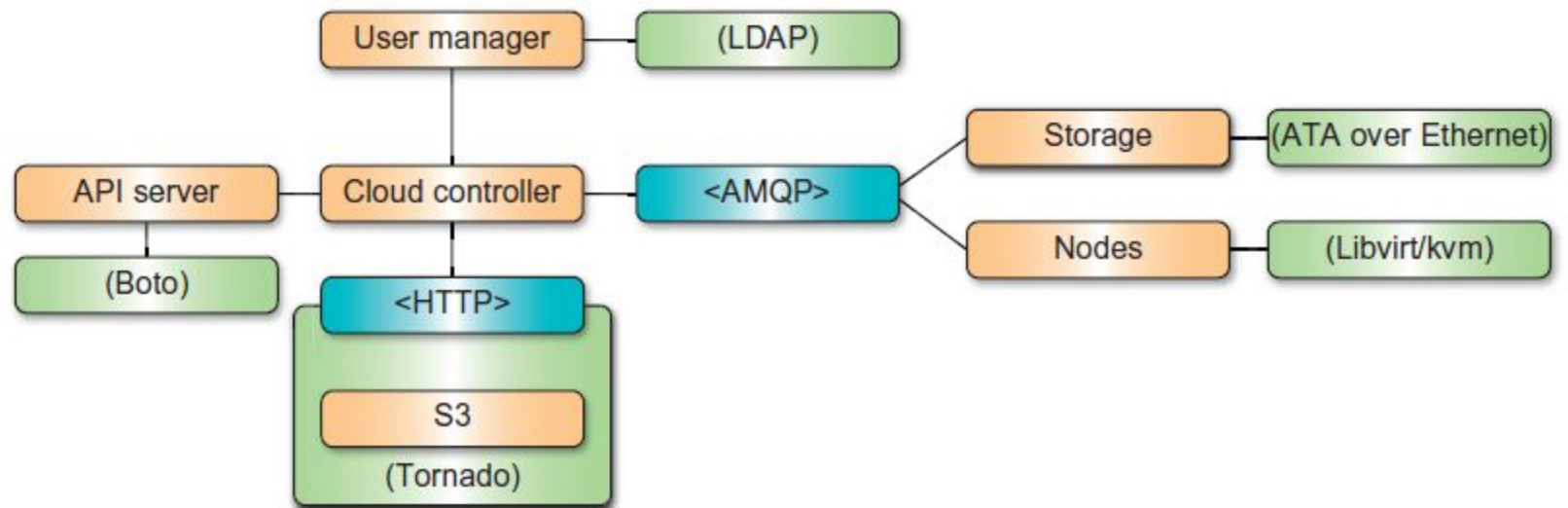
- The API Server receives HTTP requests from boto, converts the commands to and from the API format, and forwards the requests to the cloud controller.
- The cloud controller ensures authorization while interacting with the User Manager via Lightweight Directory Access Protocol (LDAP), interacts with the S3 service, and manages nodes, as well as storage workers through a queue



ARCHITECTURE OF OPEN STACK COMPUTE

- The API Server receives HTTP requests from boto, converts the commands to and from the API format, and forwards the requests to the cloud controller.
- The cloud controller ensures authorization while interacting with the User Manager via Lightweight Directory Access Protocol (LDAP), interacts with the S3 service, and manages nodes, as well as storage workers through a queue





- ❑ S3- Simple Storage System
- ❑ Tornado, a fast HTTP server
- ❑ Lightweight Directory Access Protocol (LDAP)-used to access "Directory Servers"
- ❑ Directory is a special kind of database that holds information in a tree structure
- ❑ AMQP (Advanced Messaging Queuing Protocol)
- ❑ **Ata over internet**
 - a network protocol designed for simple, high-performance access of block storage devices over Ethernet networks



- Nova integrates networking components to manage private networks, public IP addressing, virtual private network (VPN) connectivity, and firewall rules.
- It includes the following types:
 - NetworkController manages address and virtual LAN (VLAN) allocations
 - RoutingNode governs the NAT (network address translation) conversion of public IPs to private IPs, and enforces firewall rules
 - AddressingNode runs Dynamic Host Configuration Protocol (DHCP) services for private networks
 - TunnelingNode provides VPN connectivity



- The network state (managed in the distributed object store) consists of the following:
 - VLAN assignment to a project
 - Private subnet assignment to a security group in a VLAN
 - Private IP assignments to running instances
 - Public IP allocations to a project
 - Public IP associations to a private IP/running instance



OPENSTACK STORAGE

- ❑ The OpenStack storage solution is built around a number of interacting components and concepts, including a proxy server, a ring, an object server, a container server, an account server, replication, updaters, and auditors.
- ❑ Proxy server is to enable lookups to the accounts, containers, or objects in OpenStack storage rings and route the requests
- ❑ Any object is streamed to or from an object server directly through the proxy server to or from the user.



- A ring represents a mapping between the names of entities stored on disk and their physical locations.
- Separate rings for accounts, containers, and objects exist.
- A ring includes the concept of using zones, devices, partitions, and replicas.
- Hence, it allows the system to deal with failures, and isolation of zones representing a drive, a server, a cabinet, a switch, or even a data center



- ❑ Object Server is a very simple blob storage server that can store, retrieve and delete objects stored on local devices (VM images)
- ❑ Objects are stored as binary files with metadata stored in the file's extended attributes.



ARCHITECTURAL DESIGN OF COMPUTE AND STORAGE CLOUDS

- Basic cloud design principles
- Basic cloud architecture to process massive amounts of data with a high degree of parallelism



CLOUD PLATFORM DESIGN GOALS

- Scalability
- Virtualization
- Efficiency
- reliability



- ❑ Cloud management receives the user request, finds the correct resources, and then calls the provisioning services which invoke the resources in the cloud.
- ❑ The cloud management software needs to support both physical and virtual machines.
- ❑ Security in shared resources and shared access of data centers also pose another design challenge.



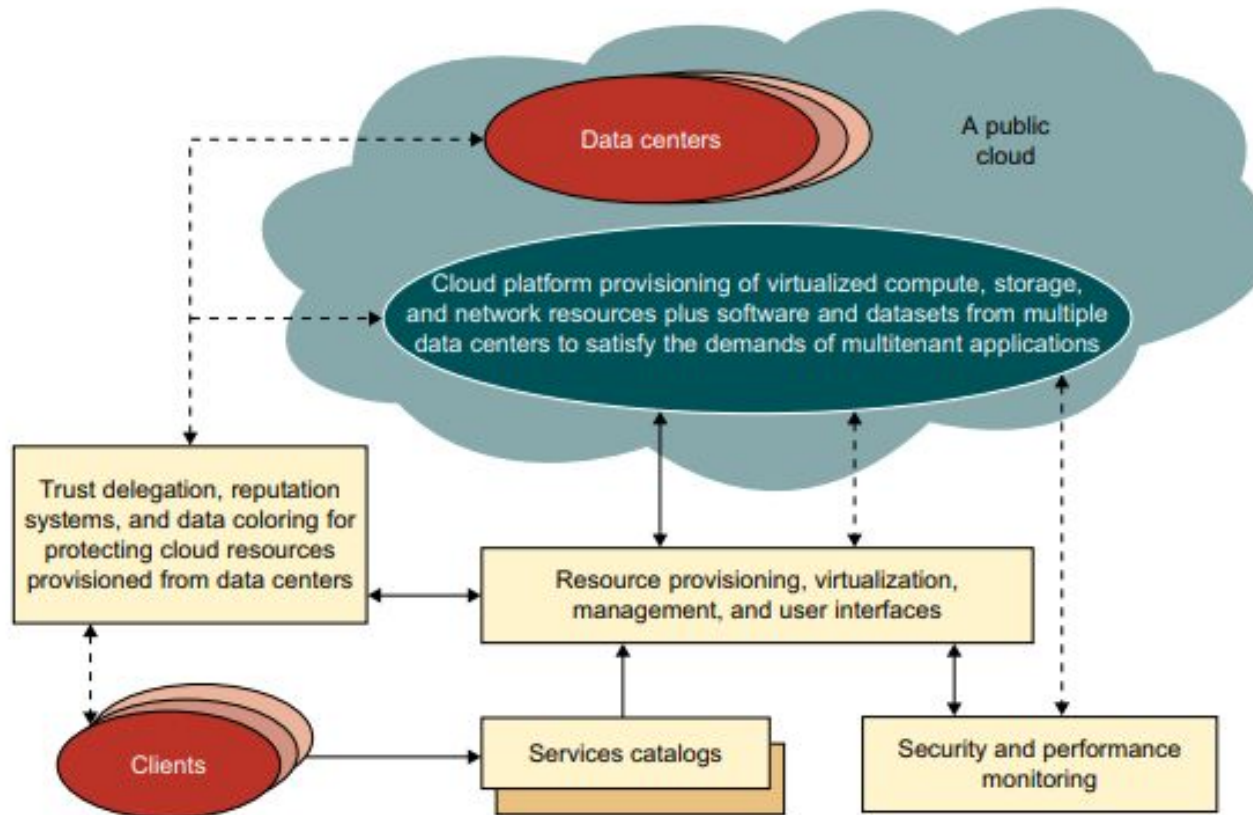
- The platform needs to establish a very large-scale HPC infrastructure.
- The hardware and software systems are combined to make it easy and efficient to operate.
- System scalability can benefit from cluster architecture.
- If one service takes a lot of processing power, storage capacity, or network traffic, it is simple to add more servers and bandwidth. System reliability can benefit from this architecture.
- Data can be put into multiple locations.



- The scale of the cloud architecture can be easily expanded by adding more servers and enlarging the network connectivity accordingly.



A Generic Cloud Architecture



- Figure shows a security-aware cloud architecture.
- Cloud - a massive cluster of servers.
- These servers are provisioned on demand
- The cloud platform is formed dynamically by provisioning or deprovisioning servers, software, and database resources.
- User interfaces are applied to request services



- ❑ Cloud platform demands distributed storage and accompanying services.
- ❑ The cloud computing resources - owned and operated by a third-party provider.
- ❑ Consumers do not need to know the underlying technologies.
- ❑ We need to build a framework to process large-scale data stored in the storage system. This demands a distributed file system over the database system.
- ❑ Other cloud resources are added into a cloud platform, including storage area networks (SANs), database systems, firewalls, and security devices.



- ❑ Monitoring and metering units are used to track the usage and performance of provisioned resources
- ❑ The software infrastructure must handle all resource management and do most of the maintenance automatically.
- ❑ Software must detect the status of each node server joining and leaving, and perform relevant tasks accordingly.
- ❑ The location of the data center in cloud is chosen to reduce power and cooling costs. Thus, the data centers are often built around hydroelectric power.



- The cloud physical platform builder is more concerned about the **performance/price ratio** and reliability issues than sheer speed performance



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



LAYERED CLOUD ARCHITECTURAL DEVELOPMENT

- The architecture of a cloud is developed at three layers: **infrastructure, platform, and application**
- These three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud.
- The infrastructure layer is deployed first to support IaaS services.
- It serves as the foundation for building the platform layer of the cloud for supporting PaaS services.



- The platform layer is a foundation for implementing the application layer for SaaS applications.
- Different types of cloud services demand application of these resources separately.

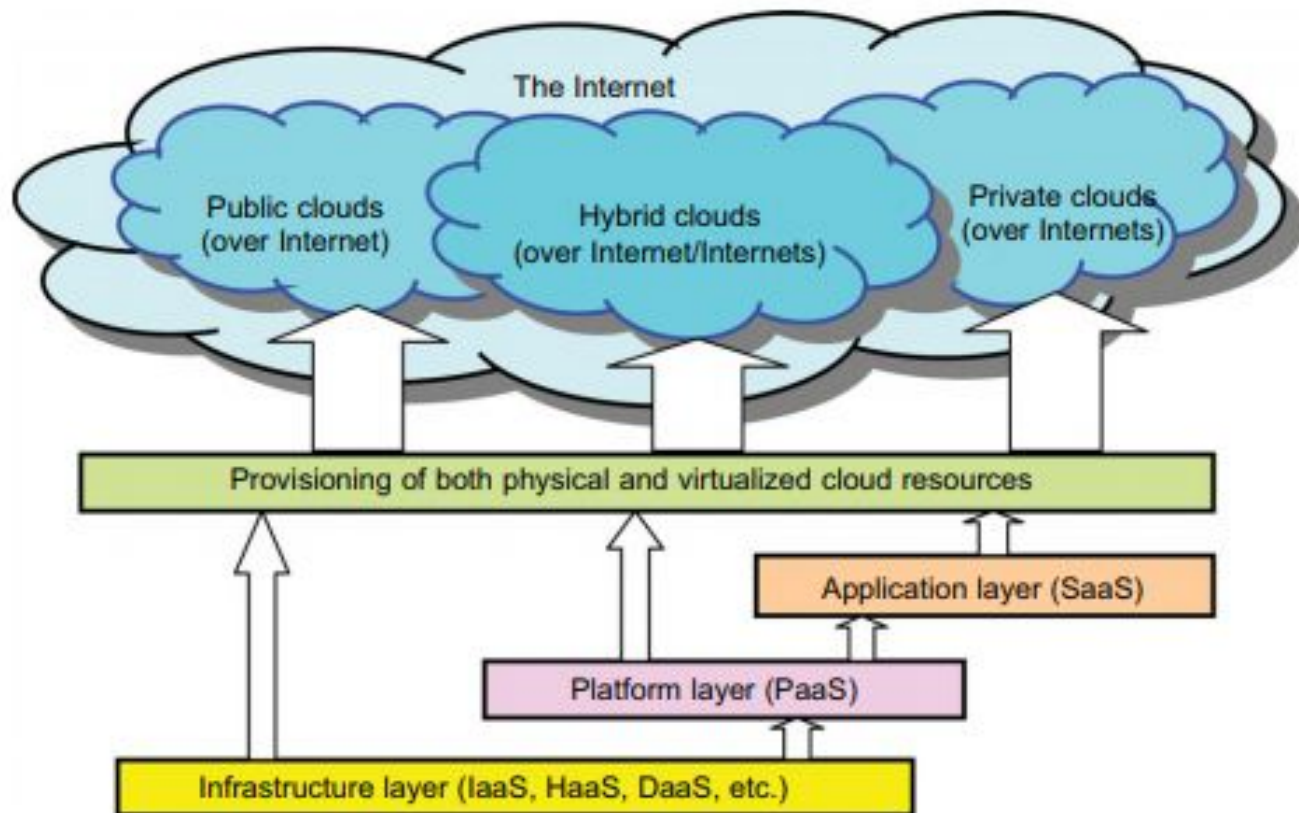


- ❑ The infrastructure layer is built with virtualized compute, storage, and network resources.
- ❑ Internally, virtualization realizes automated provisioning of resources and optimizes the infrastructure management process.
- ❑ The platform layer is for general-purpose and repeated usage of the collection of 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 platform should be able to assure users that they have scalability, dependability, and security protection.
- In a way, the virtualized cloud platform serves as a “system middleware” between the infrastructure and application layers of the cloud.





- 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.



- It should be noted that not all cloud services are restricted to a single layer.
- Many applications may apply resources at mixed layers.
- The three layers are built from the bottom up with a dependence relationship



- From the provider's perspective, the services at various layers demand different amounts of functionality support and resource management by providers.
- In general, SaaS demands the most work from the provider, PaaS is in the middle, and IaaS demands the least
- Services at the application layer demand more work from providers.
- Example of this is the Salesforce.com CRM service-
 - Provider supplies not only the hardware at the bottom layer and the software at the top layer, but also the platform and software tools for user application development and monitoring.



MARKET-ORIENTED CLOUD ARCHITECTURE

- As consumers rely on cloud providers to meet more of their computing needs, they will require a specific level of QoS to be maintained by their providers, in order to meet their objectives and sustain their operations.
- Cloud providers consider and meet the different QoS parameters of each individual 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 between supply and demand.



- 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.
 - It requires the interaction of the following mechanisms to support SLA-oriented resource management.
 - When a service request is first submitted the service request examiner interprets the submitted request for QoS requirements before determining whether to accept or reject the request.



- The request examiner ensures that there is no overloading of resources whereby many service requests cannot be fulfilled successfully due to limited resources
- It also needs the latest status information regarding resource availability and workload processing for efficient resource allocation.
- Finally it assigns requests to VMs and determines resource entitlements for allocated VMs



- Requests can be charged based on submission time (peak/off-peak), pricing rates fixed/changing), or availability of resources (supply/demand)
- Pricing serves as a basis for managing the supply and demand of computing resources within the data center and facilitates in prioritizing resource allocations effectively
- the maintained historical usage information can be utilized by the Service Request Examiner and Admission Control mechanism to improve resource allocation decisions.

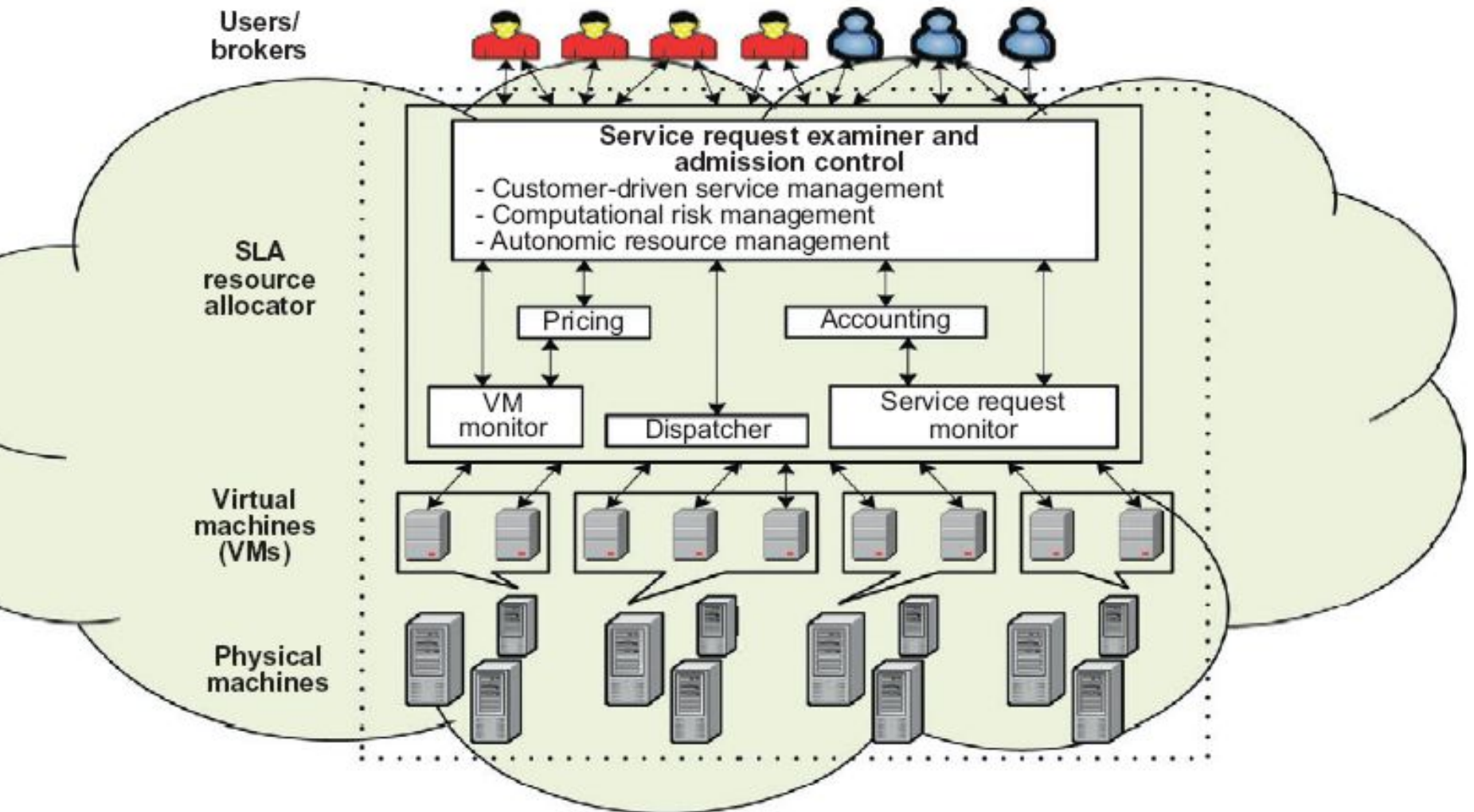


- The **VM Monitor mechanism** keeps track of the availability of VMs and their resource entitlements.
- The **Dispatcher mechanism** starts the execution of accepted service requests on allocated VMs.
- The **Service Request Monitor mechanism** keeps track of the execution progress of service requests.



- ❑ Multiple VMs can be started and stopped on demand on a single physical
- ❑ Multiple VMs can concurrently run applications based on different operating system environments on a single physical machine





QUALITY OF SERVICE FACTORS

- There should be greater importance on customers since they pay to access services in clouds
- **Negotiation mechanisms** are needed to respond to alternate offers protocol for establishing SLAs



- Commercial cloud offerings must be able to 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.
- The cloud also derives **appropriate market-based resource management strategies** that encompass both customer-driven service management and computational risk management to sustain SLA-oriented resource allocation.



- The system incorporates autonomic resource management models



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.
- ❑ 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.
- ❑ In cloud computing, virtualization also means the resources and fundamental infrastructure are virtualized.



- The user will not care about the computing resources that are used for providing the services.
- Application developers focus on service logic.



HARDWARE VIRTUALIZATION

- ❑ Virtualization software is used to virtualize the hardware.
- ❑ System virtualization software is a special kind of software which simulates the execution of hardware and runs even unmodified operating systems.
- ❑ VMs provide flexible runtime services to free users from worrying about the system environment.



- As the computing resources are shared by many users, a method is required to maximize the users' privileges and still keep them separated safely.
- Virtualization allows users to have full privileges while keeping them separate.
- The virtualized resources form a resource pool.
- The virtualization is carried out by special servers dedicated to generating the virtualized resource pool.



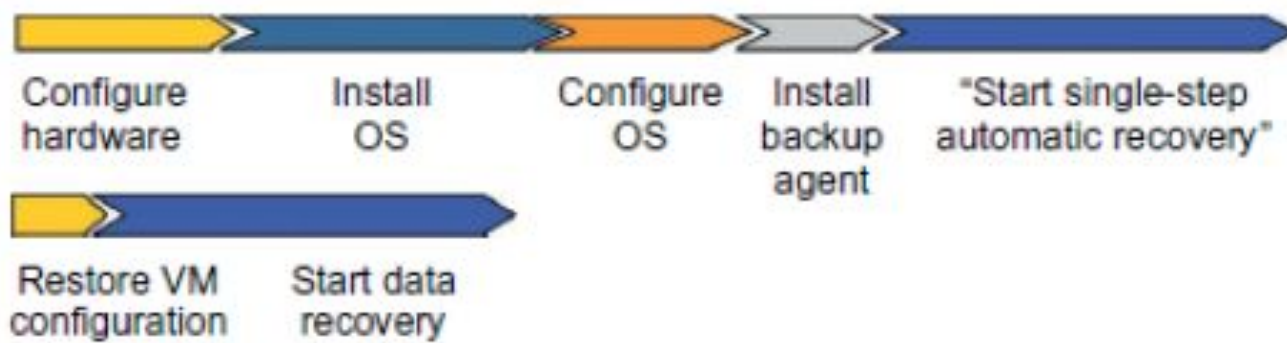
- The virtualized infrastructure (black box in the middle) is built with many virtualizing integration managers.
- These managers handle loads, resources, security, data, and provisioning functions.
- Each platform carries out a virtual solution to a user job.
- All cloud services are managed in the boxes at the top.



VM CLONING FOR DISASTER RECOVERY:

- ❑ VM technology requires an advanced disaster recovery scheme.
- ❑ One scheme is to recover one physical machine by another physical machine.
- ❑ The second scheme is to recover one VM by another.
- ❑ Traditional disaster recovery from one physical machine to another is rather slow, complex, and expensive.
- ❑ To recover a VM platform, the installation and configuration times for the OS and backup agents are eliminated.





- This results in a much shorter disaster recovery time, about 40 percent of that to recover the physical machines.
- 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.
- Only updated data and modified states are sent to the suspended VM to update its state.
- 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.
- 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.
- Software stacks have improved interoperability among different cloud platforms, but the APIs itself are still proprietary.
- The vendor **lock-in problem in cloud computing** is the situation where customers are dependent (i.e. **locked-in**) on a single **cloud** provider technology implementation and cannot easily move in the future to a different vendor without substantial costs, legal constraints, or technical incompatibilities



- So customers cannot easily extract their data and programs from one site to run on another.
- The 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.
- 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.



□ 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.(Mimicking a server to gain unauthorized access)
- In general, passive attacks steal sensitive data or passwords.
- Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.



- 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.



- Cloud users and providers have to think about the implications of placement and traffic at every level of the system, if they want to minimize costs.
- Therefore, data transfer bottlenecks must be removed, bottleneck links must be widened, and weak servers should be removed.



□ Challenge 4—Distributed Storage and Widespread Software Bugs

- The database is always growing in cloud applications.
- A storage system should 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 High Availability.
- 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.



- Debugging over simulators is a solution to attacking the problem, if the simulator is well designed



- 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



- 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.



- 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.
- OVF also defines a transport mechanism for VM templates, and can apply to different virtualization platforms with different levels of virtualization.



- In terms of cloud standardization:- the ability for virtual appliances to run on any virtual platform and the ability of VMs to run on heterogeneous hardware platform hypervisors is needed



□ 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 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.
- Hence reputation-guarding services similar to the “trusted e-mail” services may be given to service 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 30



PUBLIC CLOUD PLATFORMS

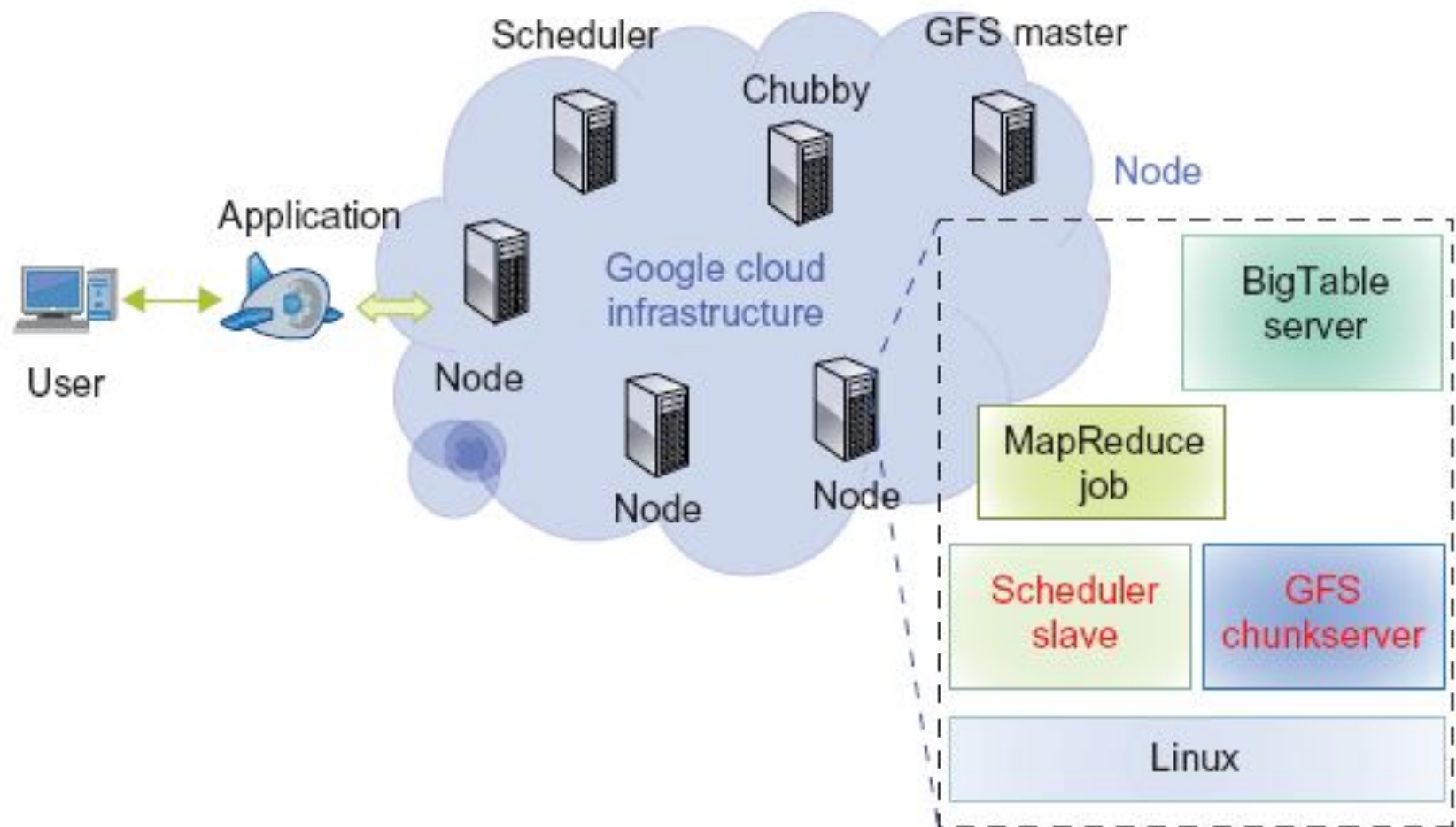
▣ **Google App Engine: (IMP)**

- ▣ Offers a PaaS platform supporting various cloud and web applications.
- ▣ Google pioneered cloud services are Gmail, Google Docs, and Google Earth
- ▣ Notable technology achievements include the Google File System (GFS), MapReduce, BigTable, and Chubby.



- In 2008, Google announced the GAE web application platform which is becoming a common platform for many small cloud service providers.
- This platform specializes in supporting scalable (elastic) web applications.
- GAE enables users to run their applications on a large number of data centers associated with Google's search engine operations.





- The figure shows the major building blocks of the Google cloud platform
- **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.



- In **Distributed**, **lock** is a mechanism that allows only one of the innumerable nodes to access and modify a resource or data that is being shared commonly to maintain data integrity.
- **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.
- GAE runs the user program on Google's infrastructure.
- The frontend is an application framework which is similar to other web application frameworks such as ASP, J2EE, and JSP



FUNCTIONAL MODULES OF GAE:

- 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



- Google offers essentially free GAE services to all Gmail account owners within a quota.
- If you exceed the quota, the page instructs you on how to pay for the service.
- GAE only accepts Python, Ruby, and Java programming languages.
- The platform does not provide any IaaS services, unlike Amazon, which offers IaaS and PaaS.



- This model allows the user to deploy user-built applications on top of the cloud infrastructure that are built using the programming languages and software tools supported by the provider (e.g., Java, Python).
- The user does not manage the underlying cloud infrastructure.
- The cloud provider facilitates support of application development, testing, and operation support on a well-defined service platform.



AMAZON WEB SERVICES

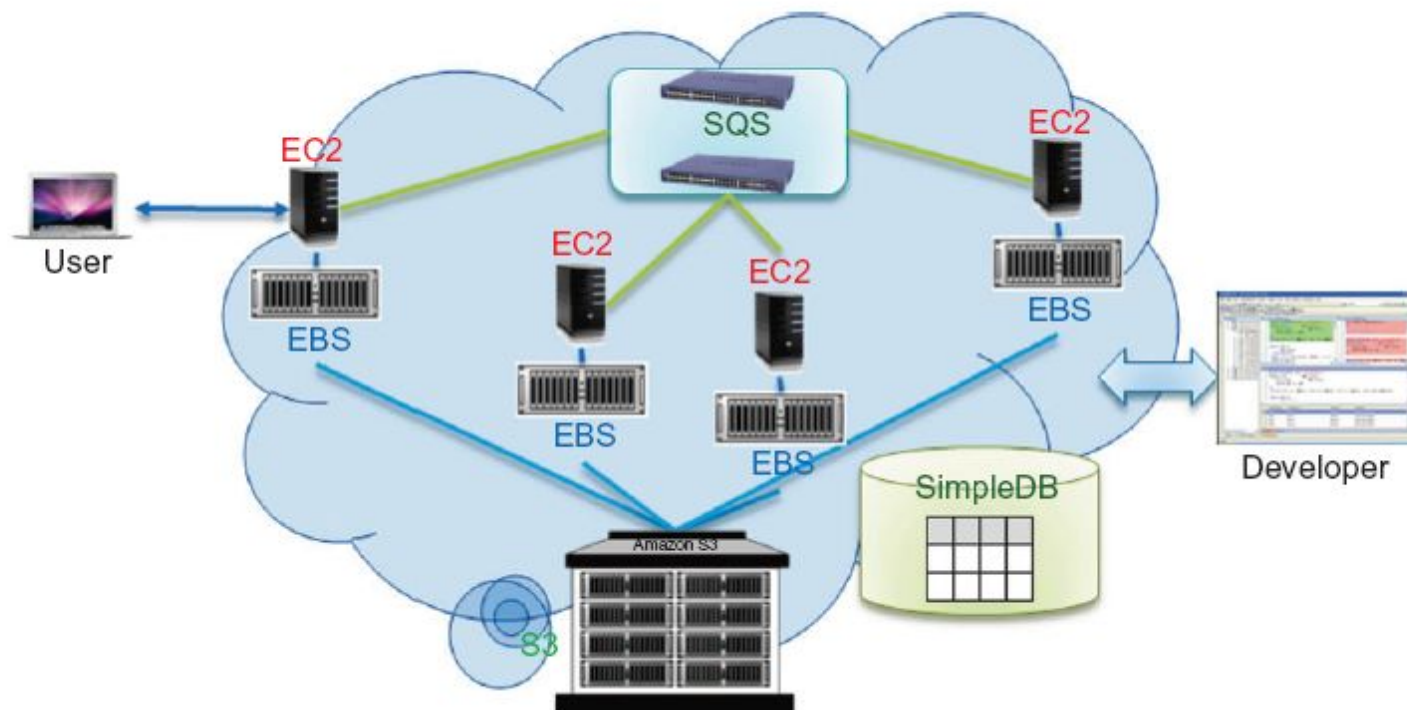
- Amazon has been a leader in providing public cloud services and applies the IaaS model in providing its services.
- Amazon Web Services (AWS) provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis.



Table 4.6 AWS Offerings in 2011

| Service Area | Service Modules and Abbreviated Names |
|-----------------------------|--|
| Compute | Elastic Compute Cloud (EC2), Elastic MapReduce, Auto Scaling |
| Messaging | Simple Queue Service (SQS), Simple Notification Service (SNS) |
| Storage | Simple Storage Service (S3), Elastic Block Storage (EBS), AWS Import/Export |
| Content Delivery | Amazon CloudFront |
| Monitoring | Amazon CloudWatch |
| Support | AWS Premium Support |
| Database | Amazon SimpleDB, Relational Database Service (RDS) |
| Networking | Virtual Private Cloud (VPC) (Example 4.1 , Figure 4.6), Elastic Load Balancing |
| Web Traffic | Alexa Web Information Service, Alexa Web Sites |
| E-Commerce | Fulfillment Web Service (FWS) |
| Payments and Billing | Flexible Payments Service (FPS), Amazon DevPay |
| Workforce | Amazon Mechanical Turk |





- ❑ EC2 provides the virtualized platforms to the host virtual machines where the cloud application can run.
- ❑ 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.



- SQS stands for Simple Queue Service, and its job is to ensure a reliable message service between two processes.
- Amazon offers queuing and notification services (SQS and SNS), which are implemented in the AWS cloud.
- Amazon offers a Relational Database Service (RDS) with a messaging interface.



- Amazon CloudFront implements a content distribution network.
- Amazon DevPay is a simple-to-use online billing and account management service that makes it easy for businesses to sell applications that are built into or run on top of AWS.
- There are many businesses like Netflix which have been built on the **AWS**



MICROSOFT:

- Microsoft launched a Windows Azure platform built over Microsoft data centers to meet the challenges in cloud computing.



OVERALL ARCHITECTURE OF MICROSOFT'S CLOUD PLATFORM

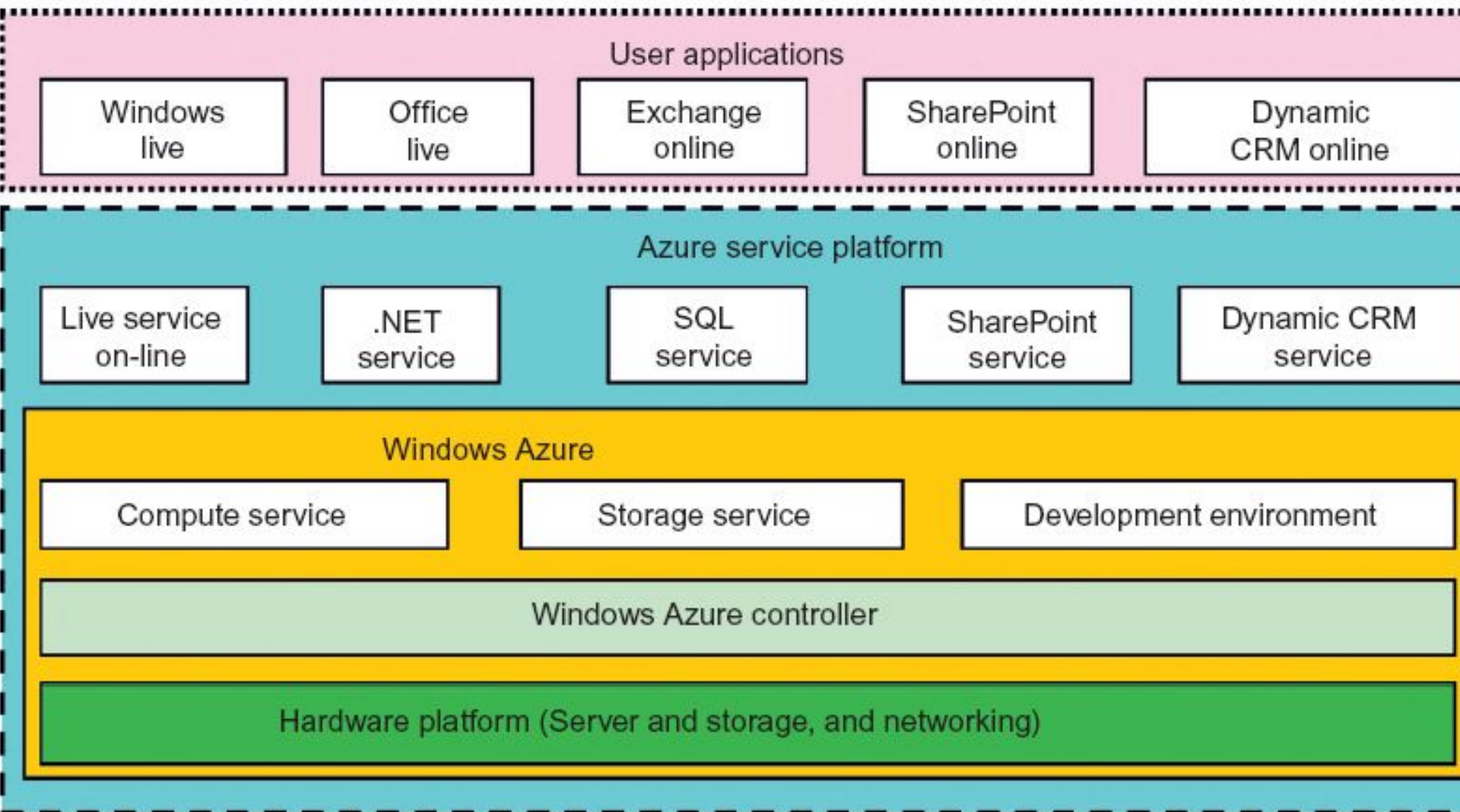


FIGURE 4.22

- Windows Azure offers a cloud platform built on Windows OS and based on Microsoft virtualization technology.
- Applications are installed on VMs deployed on the data-center servers.
- Azure manages all servers, storage, and network resources of the data center.
- On top of the infrastructure are the various services for building different cloud applications.



- ❑ Cloud-level services provided by the Azure platform are introduced below:
- ❑ **Live service** - Users can visit Microsoft Live applications and apply the data involved across multiple machines concurrently.
- ❑ **.NET service** - This package supports application development on local hosts and execution on cloud machines.



- ❑ **SQL Azure** - This function makes it easier for users to visit and use the relational database associated with the SQL server in the cloud.
- ❑ **SharePoint service** - This provides a scalable and manageable platform for users to develop their special business applications in upgraded web services.
- ❑ **Dynamic CRM service** - This provides software developers a business platform in managing CRM applications in financing, marketing, and sales and promotions.(Customer Relationship Management CRM)



- All these cloud services in Azure can interact with traditional Microsoft software applications, such as Windows Live, Office Live, Exchange online, SharePoint online, and dynamic CRM online.
- The Azure platform applies the standard web communication protocols SOAP and REST.
- The Azure service applications allow users to integrate the cloud application with other platforms or third-party clouds.



EXTENDED CLOUD COMPUTING SERVICES

- ❑ Six layers of cloud services, ranging from hardware, network, and collocation to infrastructure, platform, and software applications.
- ❑ The top three service layers are SaaS, PaaS, and IaaS, respectively.
- ❑ The cloud platform provides PaaS, which sits on top of the IaaS infrastructure. The top layer offers SaaS.



- 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).



| | | | |
|---|----------------|-----------------------|---|
| 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 | | | Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth |
| Computational resources (IaaS) | Storage (DaaS) | Communications (CaaS) | |
| Collocation cloud services (LaaS) | | | Sawis, 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 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.
- Also known as 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 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 |



SOFTWARE STACK FOR CLOUD COMPUTING

- The overall software stack structure of cloud computing software can be viewed as layers.
- Each layer has its own purpose and provides the interface for the upper layers just as the traditional software stack does.
- The platform for running cloud computing services can be either physical servers or virtual servers.



- Using VMs, the platform can be flexible, that is, the running services are not bound to specific hardware platforms. This brings flexibility to cloud computing platforms.
- The software layer on top of the platform is the layer for storing massive amounts of data. This layer acts like the file system in a traditional single machine.



- 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.



- Runtime Support Services:
- The distributed scheduler for the cloud application has special characteristics that can support cloud applications, such as scheduling the programs written in MapReduce style.
- The runtime support system keeps the cloud cluster working properly with high efficiency.



- 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.
- As a result, on the customer side, there is no upfront investment in servers or software licensing.
- On the provider side, costs are rather low, compared with conventional hosting of user applications.
- The customer data is stored in the cloud that is either vendor proprietary or a publicly hosted cloud supporting PaaS and IaaS



RESOURCE PROVISIONING AND PLATFORM DEPLOYMENT

- ❑ **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.
- ❑ Overprovisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider.



- Deploying an autonomous system to efficiently provision resources to users is a challenging problem.
- The difficulty comes from the unpredictability of consumer demand, software and hardware failures, heterogeneity of services, power management, and conflicts in signed SLAs between consumers and service providers.



- ❑ Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures.
- ❑ In a virtualized cluster of servers, this demands efficient installation of VMs, live VM migration, and fast recovery from failures.
- ❑ In the EC2 platform, some predefined VM templates are also provided. Users can choose different kinds of VMs from the templates.



- The provider should offer resource-economic services.
- Power-efficient schemes for caching, query processing, and thermal management are mandatory due to increasing energy waste by heat dissipation from data centers.



- ❑ **Resource Provisioning Methods:**
- ❑ 3 cases of static cloud resource provisioning policies.
- ❑ Overprovisioning with the peak load causes heavy resource waste.
- ❑ Underprovisioning of resources results in losses by both user and provider.
- ❑ Demand by the users is not served and wasted resources still exist for those demanded areas below the provisioned capacity.
- ❑ 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.



DEMAND-DRIVEN RESOURCE PROVISIONING

- This method adds or removes computing instances based on the current utilization level of the allocated resources
- When a resource has surpassed a threshold for a certain amount of time, the scheme increases that resource based on demand.
- When a resource is below a threshold for a certain amount of time, that resource could be decreased accordingly.



- Amazon implements such an auto-scale feature in its EC2 platform.
- This method is easy to implement. The scheme does not work out right if the workload changes abruptly.



EVENT-DRIVEN RESOURCE PROVISIONING

- This scheme adds or removes machine instances based on a specific time event.
- The scheme works better for seasonal or predicted events such as Christmastime in the West and the Lunar New Year in the East.
- During these events, the number of users grows before the event period and then decreases during the event period.
- 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.



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.



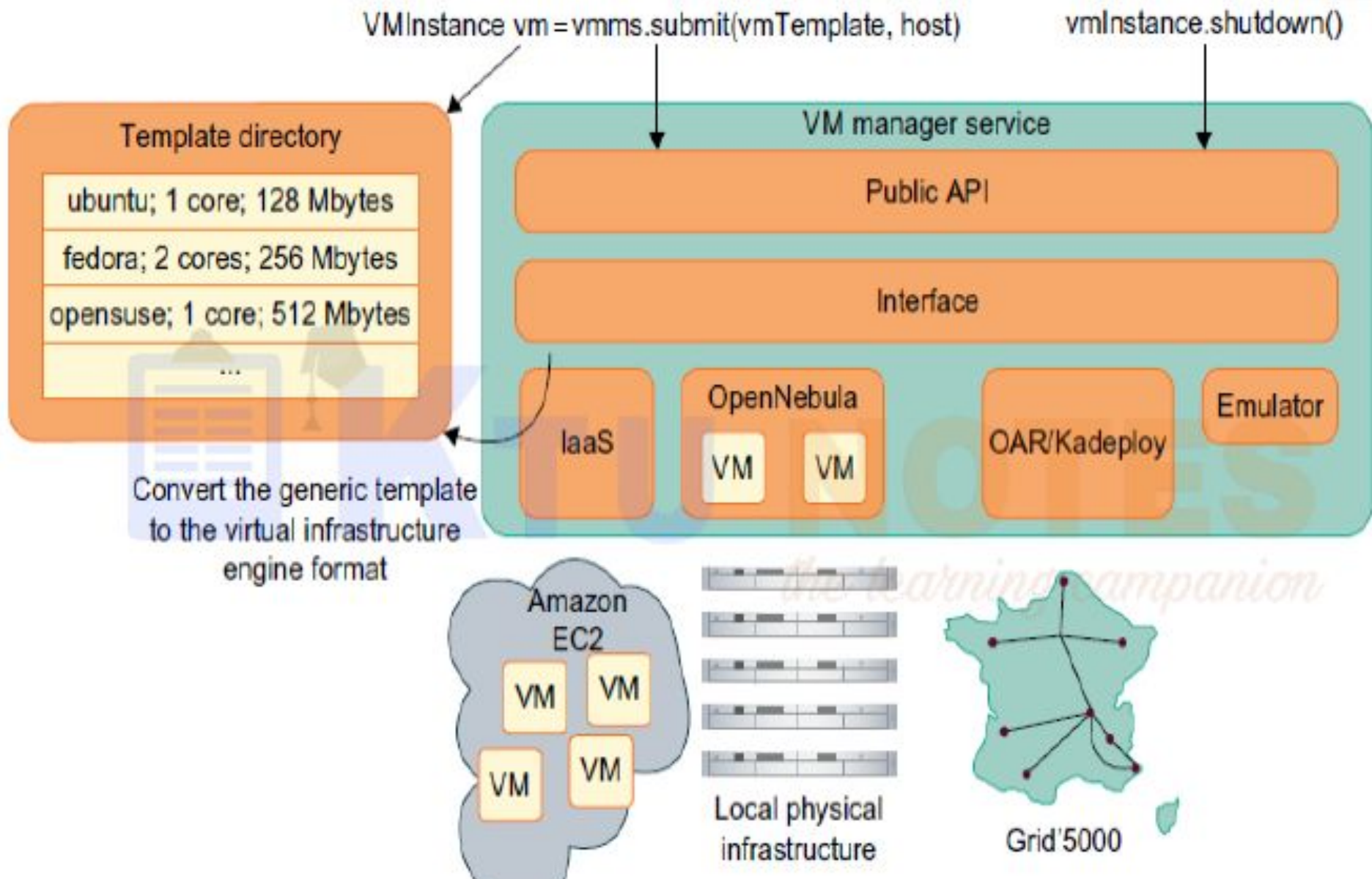
VIRTUAL MACHINE CREATION AND MANAGEMENT

- Cloud-loading experiments are used by a Melbourne research group on the French Grid'5000 system.
- This experimental setting illustrates VM creation and management.
- The figure shows the interactions among VM managers for cloud creation and management.
- The managers provide a public API for users to submit and control the VMs.



- The VM manager is the link between the gateway and resources.
- The manager manage VMs deployed on a set of physical resources.





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 VMThe gateway administrator provides the VM template information when the infrastructure is set up.
- The administrator can update, add, and delete templates at any time.
- To deploy an instance of a given VM, the VMM generates a descriptor from the template.
- This descriptor contains the same fields as the template and additional information related to a specific VM instance.



- Typically the additional information includes:
 - The disk image that contains the VM's file system
 - The address of the physical machine hosting the VM
 - The VM's network configuration
 - The required information for deployment on an IaaS provider
 - Before starting an instance, the scheduler gives the network configuration and the host's address; it then allocates MAC and IP addresses for that instance.



- The template specifies the disk image field.
- To deploy several instances of the same VM template in parallel, each instance uses a temporary copy of the disk image.
- The descriptor's fields are different for deploying a VM on an IaaS provider.
- Network information is not needed, because Amazon EC2 automatically assigns a public IP to the instances.

