Subject : Cloud Computing
Subject Code : RLIMCA381
Course : S5 MCA-REG

Cloud Computing

Module-1

Introduction to Cloud Computing and Examining the Value Proposition

In File..

- Introduction
- Cloud Types and Classification
- Characteristics
- Open Standards
- Value of Cloud for Enterprise
- Cloud Architecture
- Understanding Services and Applications by Type

What is Cloud Computing?

- *Cloud Computing* refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards.
- Resources are virtual, limitless and details of the physical systems are abstracted from the user.
- Cloud computing makes the utility computing possible with a pay-as-you-go, infinitely scalable, universally available system.

- Cloud computing takes the technology, services, and applications that are similar to those on the internet and turns them into a self-service utility.
- Pools physical resources and present them as a virtual resource.
- It is a new model for provisioning resources, for staging applications, and for platform-independent user access to services.
- The services and applications that run on clouds may or may not be delivered by a cloud service provider.

Cloud Computing: Key Concepts

Abstraction

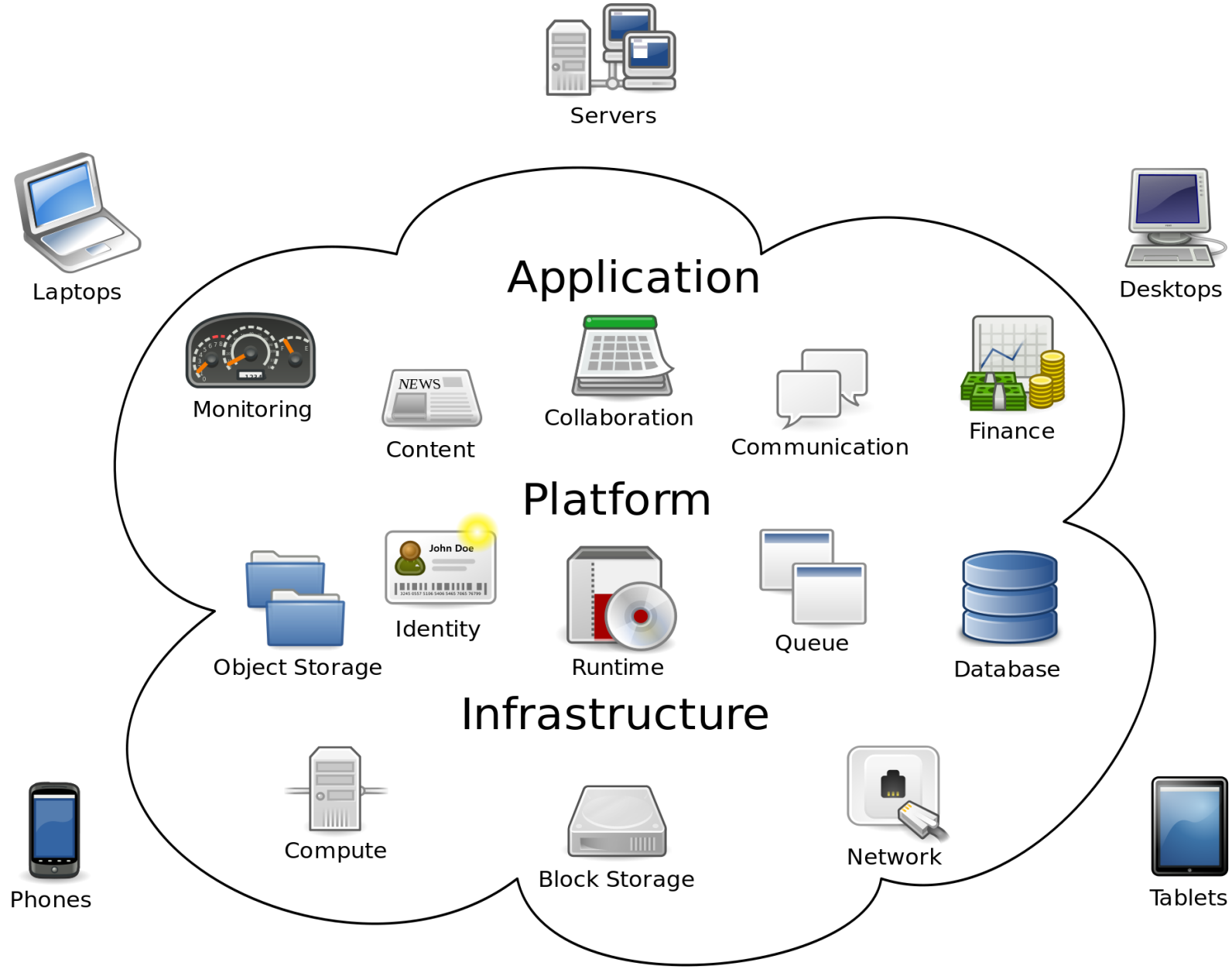
Virtualization

Abstraction

- Cloud computing abstracts the details of system implementation from users and developers.
- Applications run on physical systems that aren't specified.
- Data is stored in locations that are unknown.
- Administration of systems is outsourced to others.
- Access by users is made possible from anywhere.

Virtualization

- Cloud computing virtualizes systems by pooling and sharing resources.
- Systems and storage can be provisioned as needed from a centralized infrastructure.
- Costs are assessed on a metered basis
- Multi-tenancy is enabled
- Resources are scalable and can move quickly and easily.



Cloud Computing

Some Cloud Providers and Applications

Cloud Types

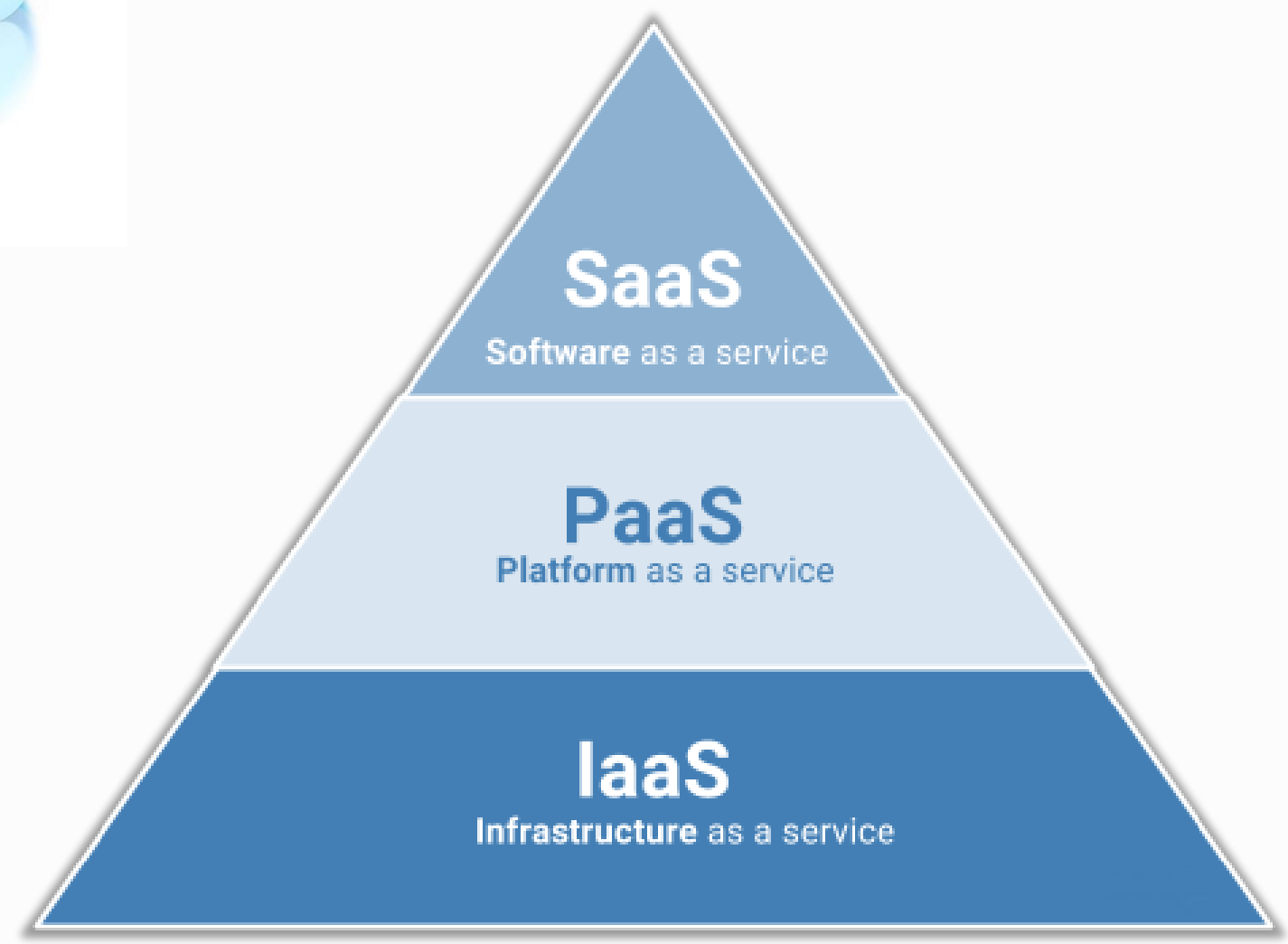
- Cloud Classes
- The NIST Model
- The Cloud Cube Model
- Deployment Models
- Service Models

Classes of Clouds

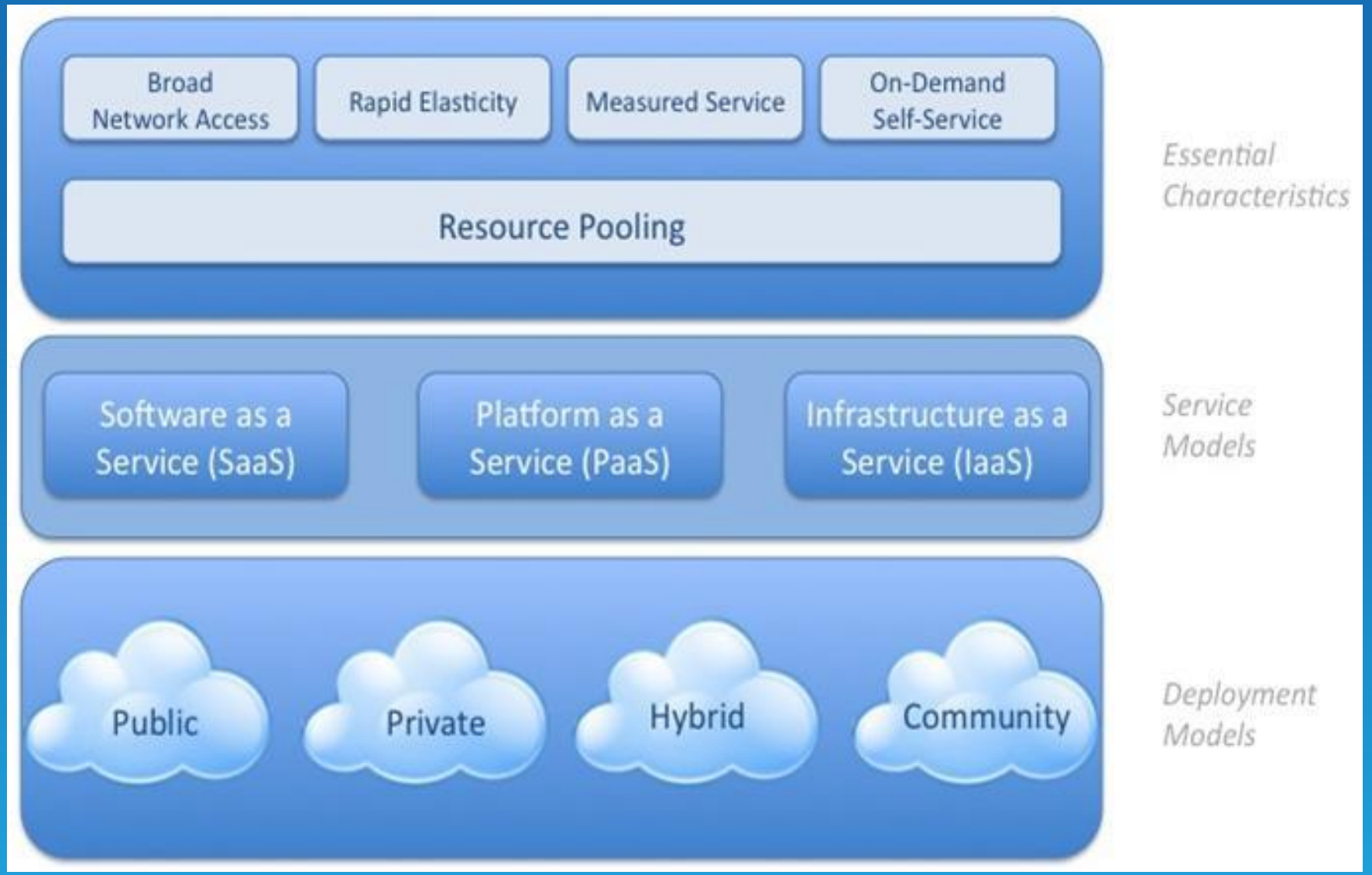
Based on Deployment Model

Classes of Clouds

Based on Service Model

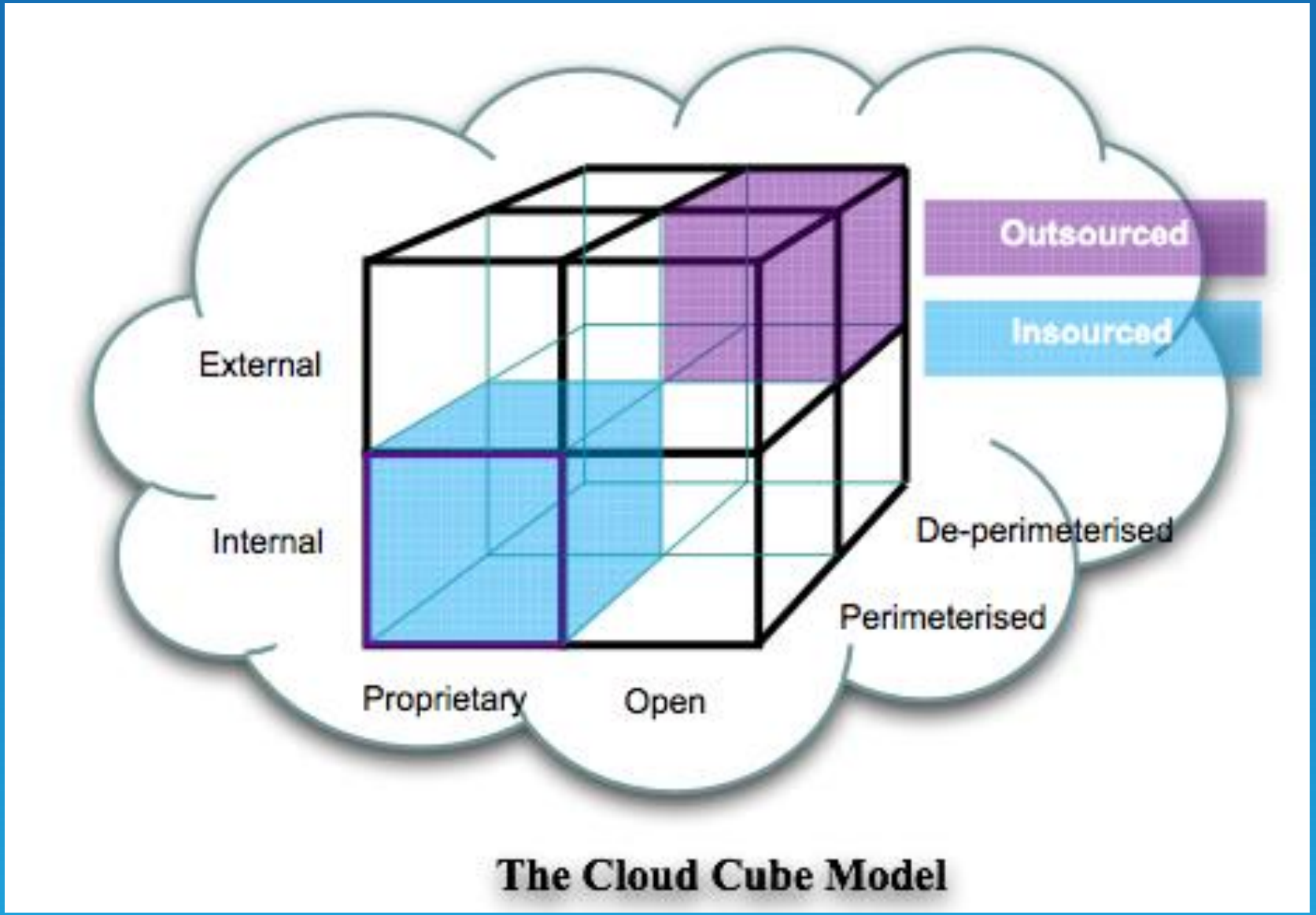


The NIST Model



NIST: National Institute of Standards and Technology

The Cloud Cube Model



The Cloud Cube Model-*Cont…*

- An association called the Jericho Forum focuses on protecting cloud networks.
- They developed a model that categorize a cloud network based on four dimensional factors; which includes:
  - ***Physical Location of the Data:***
    - ✓ *Internal (I)* or *External (E)* determines your organization's boundaries.
  - ***Ownership:***
    - ✓ *Proprietary (P) / Open (O)* is a measure of technology ownership, interoperability, ease of data transfer, and degree of vendor application lock-in.

- *Security Boundary:*
  - ✓ *Perimeterised* (Per) or *De-perimiterised* (D-p) is a measure of whether the operation is inside or outside the security boundary or network firewall.
- *Sourcing:*
  - ✓ *Insourced* or *Outsourced* means whether the service is provided by the customer or the service provider.
  - ✓ Taking all the eight cloud forms together, the *sourcing* corresponds to two different states:
    - Per (IP, IO, EP, EO)
    - D-p (IP, IO, EP, EO)

Cloud Deployment Models

- A deployment model defines the purpose of the cloud and the nature of how the cloud is located.

Public Cloud

- The public cloud infrastructure is available for public use.
- Cloud space is shared by a large industry group.
- It is owned by an organization selling cloud services.

Private Cloud

- Private cloud infrastructure is operated for the exclusive use of an organization.
- The cloud may be managed by that organization or a third party.
- Private clouds may be either on or off-premises.

Community Cloud

- Organized to serve a common function or purpose.
- It may be for one or for several organizations, but they share common concerns such as their mission, policies, security, regulatory compliance needs, and so on.
- A community cloud may be managed by the constituent organization(s) or by a third party.

Hybrid Cloud

- Combines multiple clouds such as private, community or public.
- These clouds retain their unique identities, but are bound together as a unit.
- A hybrid cloud may offer standardized or proprietary access to data and applications, as well as application portability.

Which Cloud is Right for You?

Public Cloud

- Hundreds of companies can use simultaneously, but separately
- The cloud provider handles maintenance, security, flexibility, and scalability

Private Cloud

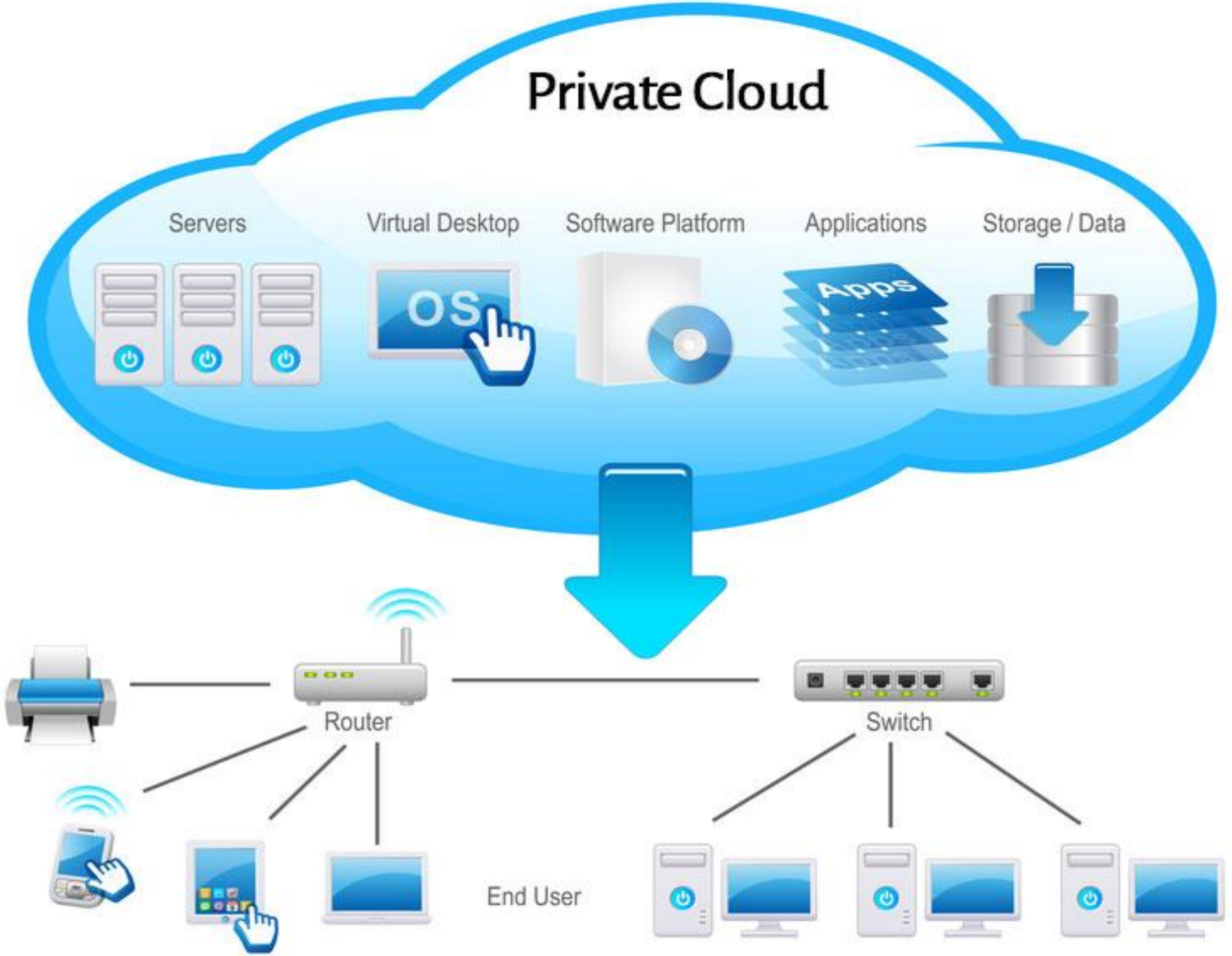
- Consist of a single organization with its own cloud of servers and software to be used without a public access point

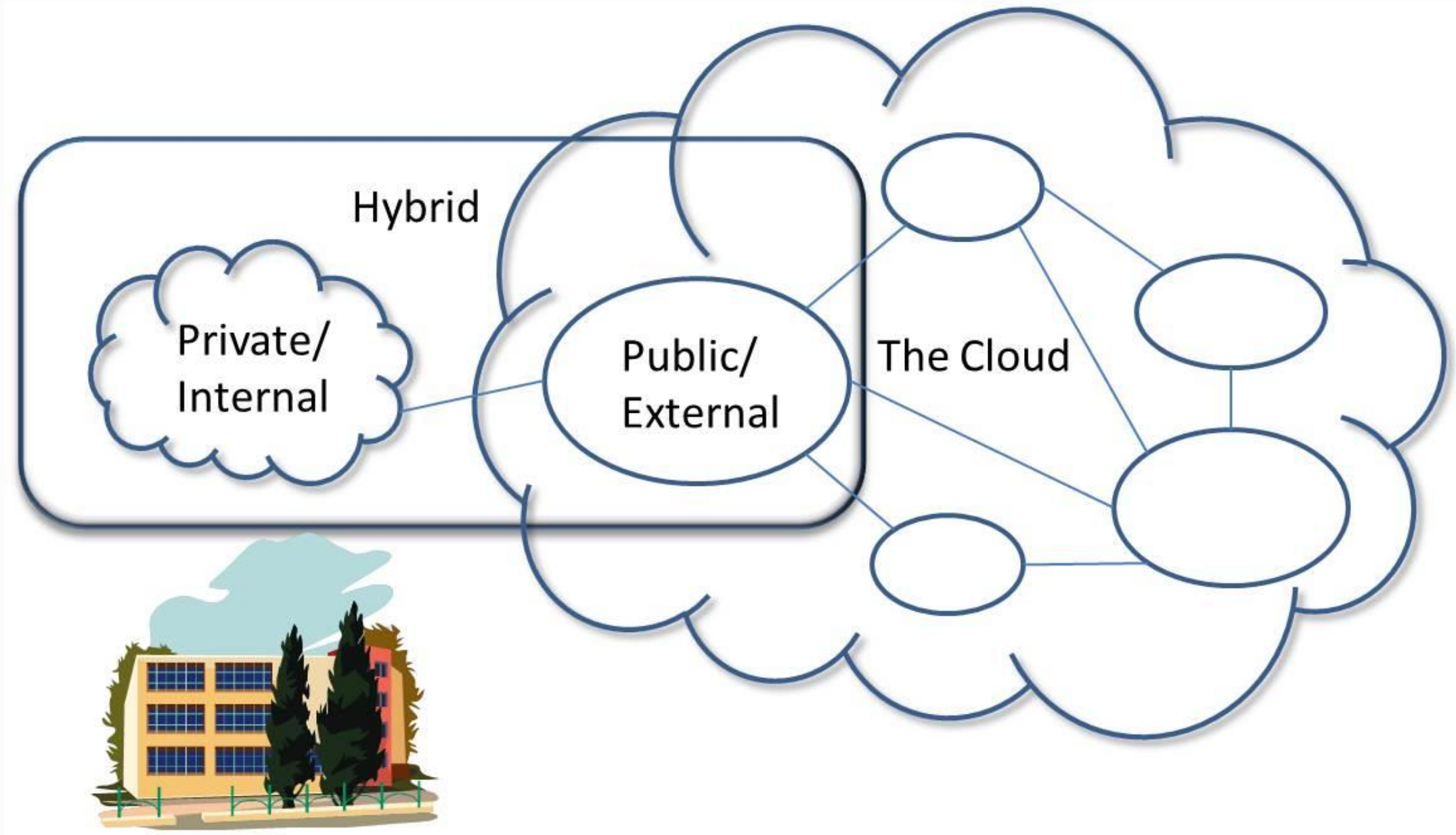
Community Cloud

- Several different companies or organizations pool together their cloud-based resources to solve a shared problem

Hybrid Cloud

- Composed of two or more distinct cloud infrastructures that remain unique entities, but are bound together by standardized or proprietary technology


Private Cloud
Servers
Virtual Desktop
Software Platform
Applications
Storage / Data
OS
Apps
Router
Switch
End User


Hybrid
Private/
Internal
Public/
External
The Cloud
On premise/internal
Off premise/Third party

Cloud Service Models

- Categorization of cloud computing based on Service types are:
  - *SaaS: Software as a Service*
  - *PaaS: Platform as a Service*
  - *IaaS : Infrastructure as a Service*
- These three are combinedly known as the ***SPI model*** of cloud computing.

SaaS: Software as a Service

- SaaS is a complete operating environment with applications, management, and the user interface.
- The application is provided to the client through a client interface like a browser.
- The customer's responsibility begins and ends with entering and managing its data and user interaction.
- Everything from the application down to the infrastructure is the Vendor's responsibility.
- Mainly used by *end users*.
- *Eg: GoogleApps, Gmail, Oracle On Demand, SalesForce.com, SQL Azure*

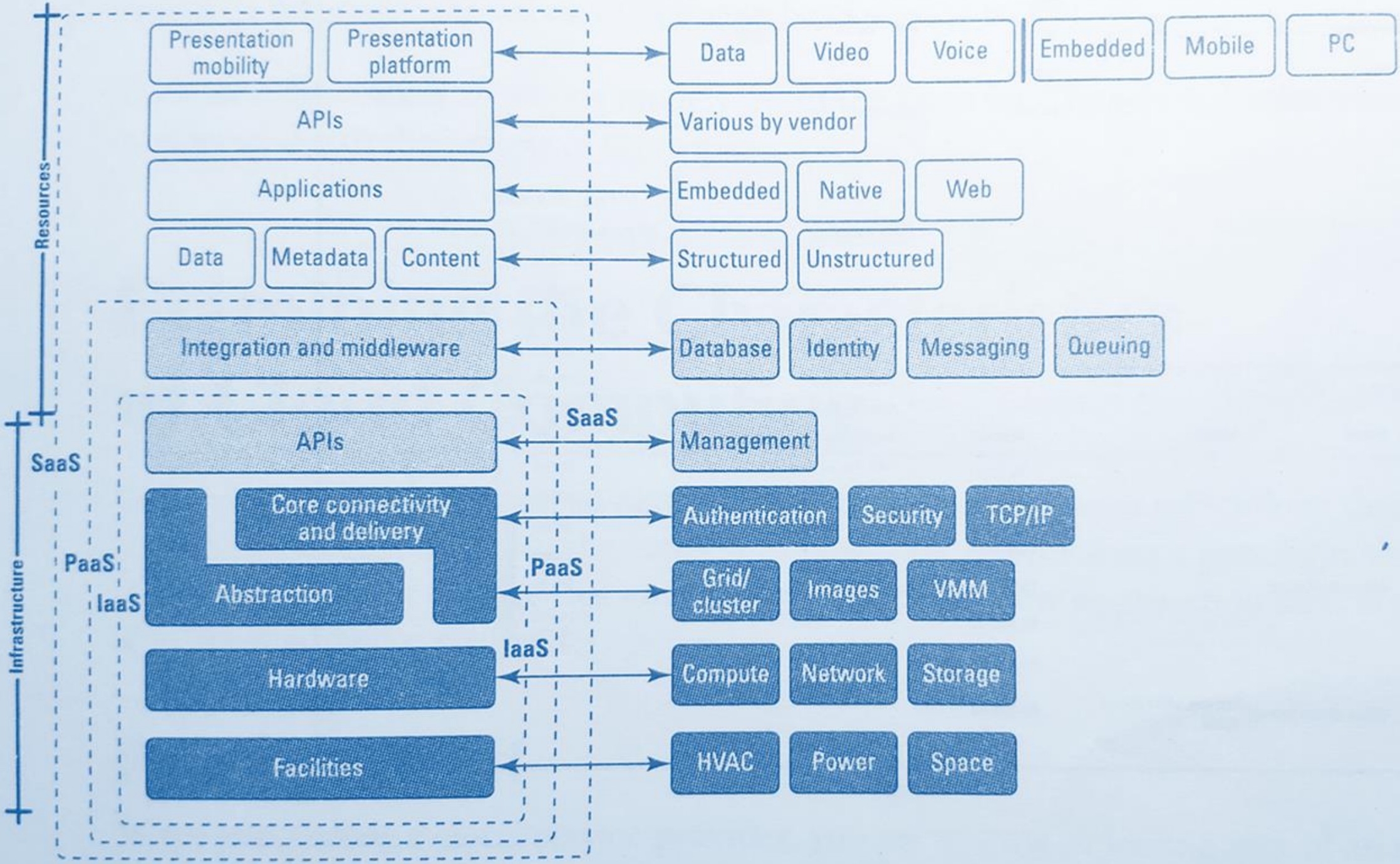
PaaS: Platform as a Service

- Provides virtual machines, operating systems, applications, services, development frameworks, transactions, and control structures.
- The client can deploy its applications on the cloud infrastructure or use applications that were programmed using languages and tools that are supported by the PaaS service provider.
- The service provider manages the cloud infrastructure, the operating systems, and the enabling software.
- The client is responsible for installing and managing the application that it is deploying. Mainly used by Developers.
- Eg: GoGrid CloudCenter, Google AppEngine, Windows Azure Platform

IaaS: Infrastructure as a Service

- IaaS provides virtual machines, virtual storage, virtual infrastructure, and other hardware assets as resources that clients can provision.
- The IaaS service provider manages all the infrastructure, while the client is responsible for all other aspects of the deployment. This can include the operating system, applications, and user interactions with the system.
- Mainly used by System Administrators
- Eg: Amazon Elastic Compute Cloud (EC2), Eucalyptus, GoGrid, FlexiScale, Linode, RackSpace Cloud, Terremark

Cloud Computing Reference Model



Characteristics of Cloud Computing

- Paradigm Shift
- Benefits
- Disadvantages

Paradigm Shift

- A drastic change in the way the computing resources are deployed and used.
- Infrastructure, computers, storage, networking capacity etc are leased or rented.
- Use of advanced technologies helps computing powers accessible through computers, mobile phones etc, with the help of internet.
- Computing has become faster, cheaper, reliable and globally accessible. Due to this, dependency on cloud services increased drastically over years.

Top Cloud Applications in use:

- Collaboration Applications
  - *Eg: Google Drive, Dropbox, Skype, Hangouts*
- Web Applications/Web Serving
  - *Eg: Google Docs, Silverlight, Apache*
- Cloud Backup
- Business Applications
- Personal Productivity Applications
  - *Eg: Google Apps, MS Office, Open Office*

Benefits of Cloud Computing

Five essential characteristics defined by NIST Model:

- *On-Demand Self-Service:* Directly operated by users.
- *Broad Network Access:* Platform independent access to clients. Includes heterogeneous OS, access on devices like laptops, mobile phones, PDA etc.
- *Resource Pooling:* Physical resources are pooled together and abstracted from users. Resources are dynamically allocated and reallocated.
- *Rapid Elasticity:* Automatic or manual scaling of resources as and when required is made possible.
- *Measured services:* Pay-as-per-use system is used for charging the services used by customers.

Additional Advantages of Cloud Computing:

- *Lower Cost*
- *Ease of Utilization:* Simple to use. Users need not worry about HW or SW licenses for implementing our services.
- *Quality of Service:* Offers high quality services.
- *Reliability:* Automatic load balancing and failover facilities implemented.
- *Outsourced IT Management:* Computing infrastructure is managed by someone else. Users need to concentrate only on business. Human service and associated costs can be reduced considerably.
- *Simplified Maintenance and Upgrade*
- *Low Barriers To Entry:* Low capital expenditure only required for new business.

Disadvantages of Cloud Computing

- *Customizing:*
  - Cloud applications and services may not be as customizable as you might require. Applications developed in-house serves better support in this case.
- *Huge data transfer over internet:*
  - Those who run each and every applications on cloud requires huge data transfer. This creates latency in processing speed.
- *Privacy and Security:*
  - Data and applications are managed by third party. Data is transferred via communication medium that are not under direct control of the client. Hence security and privacy is at risk.

- *Dealing with stateless system:*
  - Internet protocols such as HTTP enables communication between the client and provider. The communication methods, transmission routes, data sequences and chances of faulty medium might results in a stateless system. For transactional coherency, the help of service brokers, transaction managers and other middleware must be added to system.
- *Regulatory Compliance Issues:*
  - Data stored in various locations across the world are subject to the jurisdiction of respective countries. Various rules and regulation apply to data-at-rest and data-that-is transmitted. Cloud providers are bound to obey the rules of respective nations.

Open Standards

- Role of Open Standards
- Architectural Standards
- Examples

Role of Open Standards

- *'Open Standards'* are standards made available to the general public.
- Developed and maintained by a collaborative and mutually agreed process.
- It helps interoperability and data exchange among different products or services.
- The cloud computing technology is the result of the convergence of many different standards. It changes the manner in which services and applications are deployed. Hence, the role of open standards becomes crucial.
- Without standards, industry creates proprietary systems with vendor lock-in.
- Businesses require open standards, so that, data is both portable and universally accessible.

- The cloud computing industry is working with the following architectural standards:
  - Platform Virtualization of Resources
  - Service-Oriented Architecture
  - Web-Application Frameworks
  - Deployment of Open-Source Software
  - Standardized Web Services
  - Autonomic Systems
  - Grid Computing

Examples of

Open Source Cloud Platforms

- OpenStack
- EUCALYPTUS

Value of Cloud for Enterprise

- Measuring the Cloud's Value
- Avoiding Capital Expenditures
- Right-Sizing

Value of Cloud for Enterprise

- Cloud computing is very valuable for enterprise as it shifts capital expenditures into operating expenditures.
- Decouples GROWTH from CAPITAL (ability to invest more).
- It shifts several risks away from an organization onto the cloud provider.
- Organization's management can focus more on the development of business rather than concentrating on establishing and maintaining infrastructure.

Measuring the Cloud's Value

- Cloud computing is based on the idea of shared multitenant utility. It presents new opportunities to users and developers.
- Alters traditional IT infrastructure and allows new types of access and business models for user applications.

Who will benefit?

Any application or process that benefits from:

- ✓ Economies of scale.
- ✓ Commoditization of assets.
- ✓ Conformance to programming standards.

Who will NOT benefit?

Any application or process that:

- ✓ Requires a completely customized solution.
- ✓ Imposes a high degree of specialization.
- ✓ Requires access to proprietary technology.

- Companies becomes cloud providers for the following reasons:
  - Profit
  - Optimization: Full utilization of already existing infrastructure.
  - Strategic: For marketing company's products.
  - Extension: Branded cloud computing platform extend the customer relationships by offering additional service options.
  - Presence: To establish a presence in market before competitors can emerge.
  - Platform: A Cloud service provider can become a hub master at the centre of many ISV's (Independent Software Vendor) offerings.

Avoiding Capital Expenditures

- Biggest reason for new business's failure: *Capitalization*.
- Growth will be difficult when capital is not invested on expansion requirements.
- A company wishing to grow would normally be faced with the following options:
  - *Buy the new equipment, and deploy it in-house*
  - *Lease the equipment for a set period of time*
  - *Outsource the operation to a managed-services organization*

- Using and maintaining legacy applications and systems usually costs significantly.
- Cloud computing is a good option when the cost of infrastructure and its management is high.
- Cloud Computing's helps to convert capital expenses (CapEX) to Operating expenses (OpEx). It is made possible through a usage based pricing scheme that is elastic and can be right-sized.
- This allows an organization to transfer risk to their cloud provider.

Right-Sizing

- It is the method of providing the organization with adequate resources for its smooth operation.
- In situations where demand is unpredictable and change can be rapid, right-sizing a cloud computing solution demands automated solutions.
  - Eg: Amazon Web Services' Auto Scaling feature for its EC2 service

Understanding Cloud Architecture

- Exploring Cloud Computing Stack
- Composability
- Infrastructure
- Platforms
- Virtual Appliances
- Communication Protocols
- Applications
- Connecting to Cloud

1. Exploring the Cloud Computing Stack

- Cloud computing is build over distributed network's architecture. To its standard networking protocols, system virtualization is added.
- The cloud creates a system where resources can be pooled and partitioned as needed.
- Cloud architecture can couple software running on virtualized hardwares (in multiple locations) to provide an on-demand service to the user.
- Compared to usual internet applications, cloud computing architecture provides abstraction and metered services.

- General descriptions of cloud computing in terms of two architectural layers:
  - A "client" as a front end
  - The “cloud” as a backend
- These two components are composed of several layers, functionalities, and a mixture of standard and proprietary protocols.
- Cloud computing model delivers various IT services, usually controlled through an Application Programming Interface (API).

- A organization’s cloud can be created in-house or outsourced to another datacenter.
- Resources in a cloud are virtualized, hence, easier to modify and optimize.
- From a user’s perspective, resources must be infinitely scalable, service be measurable, and the pricing be metered.

2. Composability

- Applications built in the cloud are often built from a collection of components; a feature referred to as composability.
- A composable system uses standard components to assemble services that can be tailored for a specific purpose.
- Composability of hardware and software makes the system design easier to implement and solutions will be more portable and interoperable.

- A composable component must be:
  - Modular: It is a self-contained and independent unit that is cooperative, reusable, and replaceable.
  - Stateless: A transaction is executed without regard to other transactions or requests. Some cloud computing applications provide managed states through brokers, transaction monitors, and service buses.

- Benefits of composability:
  - Easier to assemble systems
  - Cheaper system development
  - More reliable operation
  - A larger pool of qualified developers
  - A logical design methodology

IaaS : Highly Composable

PaaS : Moderately Composable

SaaS : Less Composable

- Designing of composable systems in cloud computing is called the Service Oriented Architecture (SOA).
- In service oriented design, the services are constructed from a set of modules using standard communications and service interfaces.
- Example of a SOA for a specific system:
  - Services designed using Web Services Description Language (WSDL).
  - Data exchange between its services using some form of Extensible Markup Language (XML).
  - And, the communications between the services using the Simple Object Access Protocol (SOAP).

3. Infrastructure

- IaaS providers mostly rely on virtual machine technology to deliver servers that can run applications.
- Virtual servers behaves like real servers, delivering a certain number of microprocessor (CPU) cycles, memory access and network bandwidth to customers.
- Virtual machines are containers that are assigned specific resources.
- Software runs on virtual machines which enables utility computing.

- The Virtual Machine Monitor(VMM), also called a hypervisor, is the low-level software that allows different operating systems to run in their own memory space and manages I/O for the virtual machines.
- As required, applications are custom developed by developers which can create additional threads of execution and can scale as and when required.

4. Platforms

- A platform in the cloud is a software layer that is used to create higher levels of service.
  - Eg: Salesforce.com's *Force.com* Platform, Windows's *Azure* Platform, Google Apps and the Google AppEngine
- They offer all the hosted hardware and software needed to build and deploy web applications or services that are custom built by the developer.
- The software is constructed using various components and services and are controlled through the API provided by vendor.

- For example, Windows's Azure Platform allows developers to run on a Hyper-V VM, support ASP.NET application framework, SQL Server, and can be programmable within Visual Studio.

- Depending upon the vendor, different tools will be available such as, collaboration tools, testing tools, program's performance measuring tools, database and Web service integration tools, storage tools etc.
- Platforms often provides tools and utilities to aid application design and deployment.
- Users interact with the platform, uses the services through that API. An API can control data flow, communications, and other important aspects of the cloud application.

5. Virtual Appliances

- Refers to a software that installs as a middleware onto a virtual machine. It is a platform instance, occupying the middle of the cloud computing stack.
  - Eg: Virtual servers: application modules that run a particular machine instance or image type.
- Since, the appliance being one of the standardized components, it is used as the basis for assembling more complex services.

- Virtual appliances remove the need for application configuration and maintenance by cloud users.
- File formats like *Open Virtualization Format (OVF)* helps converting a virtual appliance from one platform to another. It is supported by major virtualization platform vendors like VMware, Microsoft, Oracle and Citrix.
  - Examples of Virtual appliances: HelpdeskLive, Jumpbox, VirtualBox, ThoughtPolice.

6. Communication Protocols

- Enables Inter-Process Communication (IPC) between clients and server over distributed networking.
- Various forms of *Remote Procedure Calls* (RPC) are used for engaging services and managing transactions over a stateless network.
- *XML-RPC* (a web-based RPC technology), uses platform independent XML data to encode program calls that are transported over HTTP.
- Most widely used message-passing standard is the Simple Object Access Protocol(SOAP) which uses XML for its messages and uses RPC and HTTP for message passing.

- *XML: Extensible MarkUp Language*

- Web Services Description Language (WSDL) is commonly used for discovery and description of Web-based resources which uses SOAP messaging.
- Most commonly used Data Exchange Standards are:
  - *REST (REpresentational State Transfer)*
    - ✓ It is a method for standardizing resources on the web.
    - ✓ REST assigns a global identifier to a resource so there is a uniform method for accessing information sources.
    - ✓ Currently, the basis for cloud computing transactions such as initiation, processing and completion are based on the rules of REST.
  - *ATOM*
    - ✓ Also called Atom Publishing Protocol[APP]

7. Applications

- The applications of cloud computing are practically limitless.
- With the right middleware and proper applications for execution, a cloud computing system could execute all the programs a normal computer could run.
- Potentially, everything from generic word processing software to customized computer programs designed for a specific company could work on a cloud computing system.

8. Connecting to Cloud

- Most common connecting methods:
  - A web browser
  - A proprietary application
- Methods for securely connecting over these connections:
  - Use a secure protocol for data transfer such as HTTPS, FTP etc.
  - Using VPN or remote transfer protocols like MS-RDP where data protected by tunneling mechanism.
  - Through encrypted data transmission.

Understanding Services and Applications by Type

- IaaS
  - ✓ IaaS Workloads
  - ✓ Pods, Aggregation, Silo
- PaaS
- SaaS
  - ✓ Open SaaS and SOA
  - ✓ Mashups

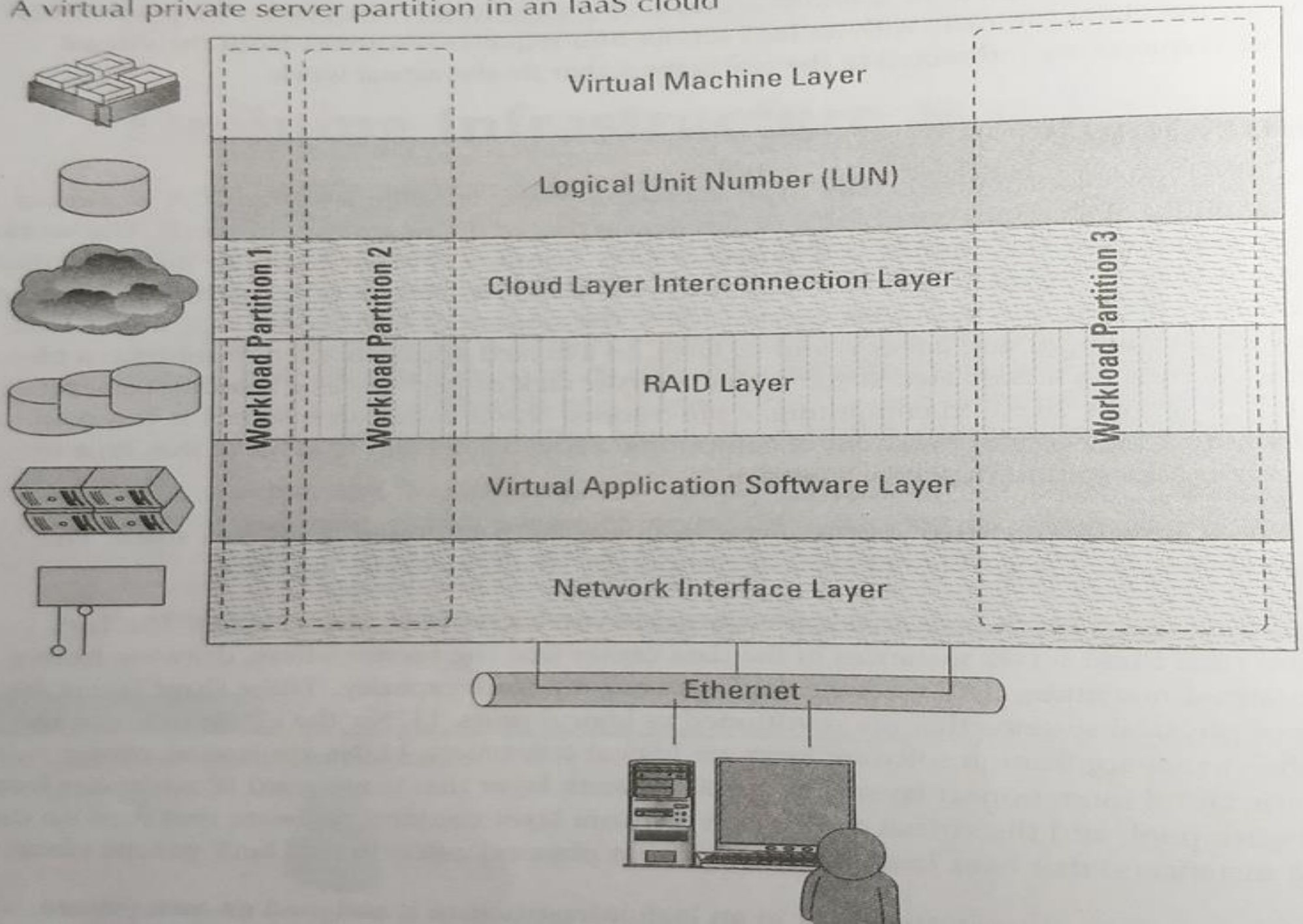
Defining Infrastructure as a Service (IaaS)

- A utility cloud computing service model in which, hardware is virtualized in the cloud.
- The service vendor owns the basic facilities such as servers, storage, network infrastructure etc.
- The developer interacts with the IaaS model to create virtual private servers, storage systems, networks etc., and then populates these virtual systems with the applications and services it needs to complete its solution.
- In IaaS, the virtualized resources are mapped to real systems. When the client interacts with an IaaS service and requests resources from the virtual systems, those requests are redirected to the real servers that do the actual work.

IaaS Workloads

- In an IaaS deployment, the basic unit of virtualized client is called a workload. A workload works like a real server and performs certain amount of work.
- In cloud computing, a provisioned server called an instance is reserved by a customer, and the necessary amount of computing resources are allocated to the client's needs.
- A client would reserve machine resources that is required to run each of these workloads.

A virtual private server partition in an IaaS cloud



The three different workloads require three different sizes of computers: small, medium, and large

- Common Workload Attributes:
  - Work done measured in number of Transactions Per Minute (TPM)
  - Throughput
  - Disk I/Os measured in Input/Output Per Second (IOPS)
  - Amount of RAM used in MB
  - Network Throughput
  - Latency etc.
- LUNs and the virtual application software layers are logical constructs.
- A client using an IaaS infrastructure is assigned with its own private network, where, the workloads are configured as per the requirement and can be monitored and controlled through APIs.

Pods

- Every workload support a certain number of users.
- When you reach the limit of the largest virtual machine instance possible, you must make a copy or clone of the instance to support additional users.
- A group of users within a particular instance is called a *POD*.
- Sizing limitations for pods is serious concern while building large cloud-based applications.
- Pods are managed by a Cloud Control System (CCS). For example; in AWS, the CCS is the AWS Management Console.

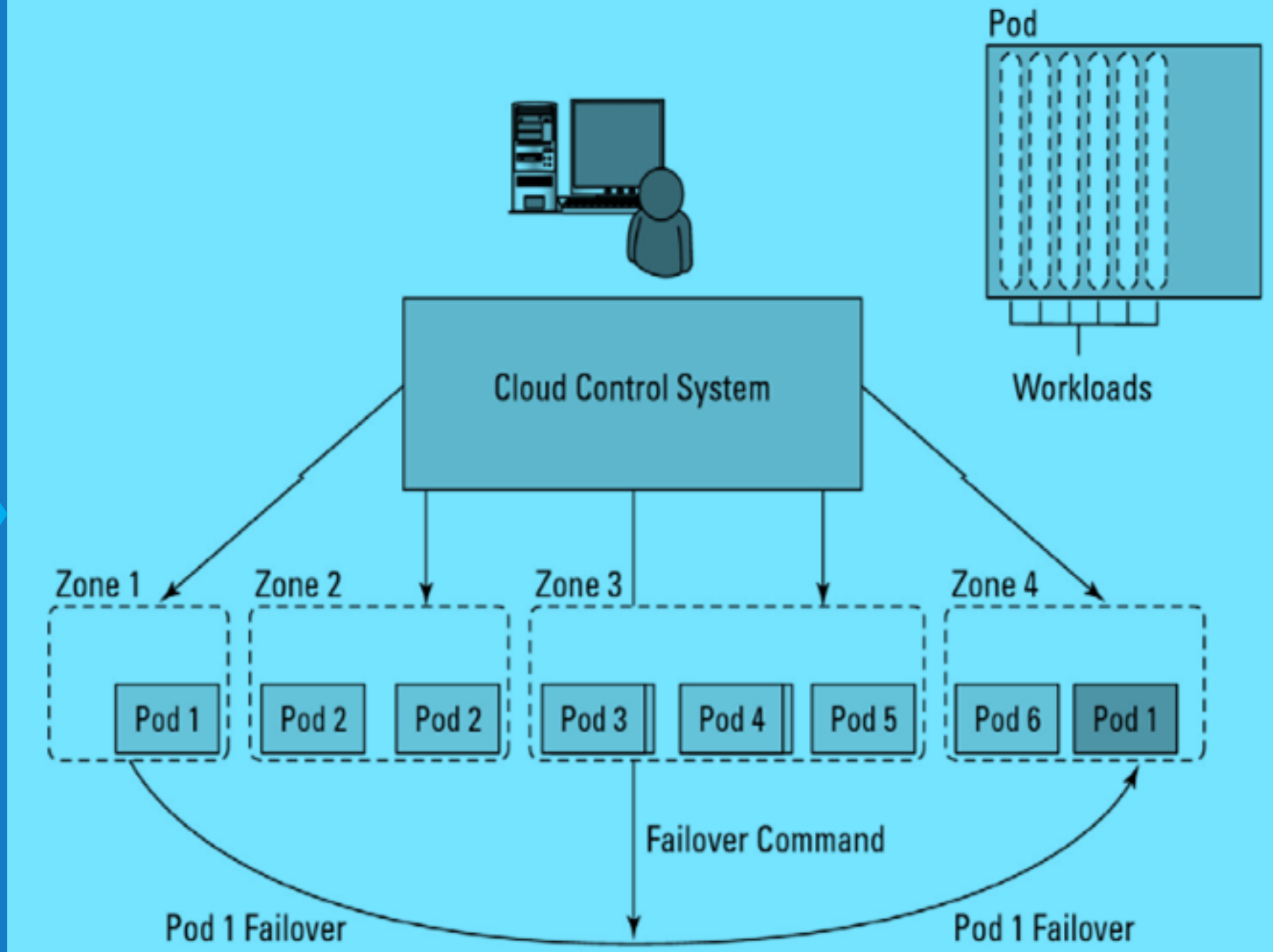
Aggregation

- Pods are ***aggregated*** into pools within an IaaS region or site called an availability zone.
- In very large cloud computing networks, when systems fail, they fail on a pod-by-pod basis, or on a zone-by-zone basis.
- A failover system between zones gives IaaS private clouds a very high degree of availability.

Silo

- When a cloud computing infrastructure isolates user clouds from each other, the management system is incapable of interoperating it with other private clouds. This creates an *"information silo"*, or simply a *"silo"*.
- Silos are processing domains that are sealed off from the outside, thus imposing restrictions on interoperability. They are created, when you create a private virtual network within an IaaS framework.
- Silos aren't flexible like open systems and are subject to vendor lock-in. But they are protected and secured in many ways.

Pods, Aggregation and Failover in IaaS



Defining Platform as a Service (PaaS)

- It is a software environment in which a developer can create customized solutions using the development tools provided by the platform/vendor.
- Platforms provides development languages, application frameworks, or other constructs. It provides the tools and development environment to deploy applications.
- PaaS systems must offer a way to create user interfaces, and thus support standards such as HTLM, JavaScript, or other rich media technologies.

- The vendor is responsible for all the operational aspects of the service, maintenance of hardware and software, and managing the product's lifecycle.
  - *Eg: Google's App Engine platform. Developers program on App Engine using APIs published by Google's Development framework tools. Structure of the file system and data stores are defined by Google.*

Defining Software as a Service (SaaS)

- The most complete cloud computing service model.
- The complete infrastructure including computing hardware and software, as well as the solution itself, are provided by a vendor as a complete service offering. Eg: Gmail.
- SaaS may be described as software that is deployed on a hosted service and can be accessed globally over the internet, most often in a browser.
- Except user interaction with the software, all other aspects of the service are abstracted away.

- SaaS applications may be:
  - Customizable (Eg: Salesforce.com with custom API).
  - Non customizable (Eg:office suite).
- SaaS applications include custom software such as billing and invoicing systems, Customer Relationship Management (CRM) applications, Help Desk applications, Human Resource (HR) solutions etc.

SaaS characteristics

1) Available globally on demand, over the internet, through a browser.
2) Subscription-based or usage-based license and is billed on a recurring basis.
3) The software and the service are monitored and maintained by the vendor.
4) Cheaper due to reduced distribution and maintenance costs and minimal end-user system costs.

Continue…

5) Automated upgrades, updates, and patch management and much faster rollout of changes.
6) SaaS applications often have a much lower barrier to entry and they scale on demand.
7) All users have the same version of the software so each user's software is compatible with another's.
8) SaaS supports multiple users and provides a shared data model through a single-instance, multi-tenancy model.

Open SaaS and Service Oriented Architecture (SOA)

- SaaS software is largely based on open source software, which is referred to as Open SaaS.
- The advantages of using open source software are that systems are much cheaper to deploy because you don't have to purchase the operating system or software. It has less vendor lock-in, and more portable applications.
- The popularity of open source software (like Linux, APACHE, MySQL), and the number of people who are trained in it, make Open SaaS an attractive proposition.
- Open source software companies earns better profit, resulting in lower development costs and more robust solutions.

Mashups

- The componentized nature of SaaS solutions support a feature called *Mashups*.
- *A mashup is an application that can display a Web page that shows data and supports features from two or more sources.*
  - Eg: Adding user / location details to Google maps. Live traffic updates in Maps.
- Mashups are key features of Web 2.0. This technology is capable of supporting social network systems. They are incredibly useful hybrid Web applications.

- A mashup requires three separate components:
  - An interactive user interface (created with HTML/XHTML, Ajax, javaScript, or CSS).
  - Web services that can be accessed using an API, and whose data can be bound and transported by Web service protocols (such as SOAP, REST, XML/HTTP, XML/RFC).
  - Data transfer in the form of XML, KML, JSON etc.

3 MARKS QUESTIONS

Define Cloud Computing and explain the key concepts.
Cloud Computing Definition-1 Mark
Abstraction and Virtualization-1 Mark each
#Pages-3, 4

Explain Cloud Cube Model with the help of a diagram.
Definition-1 Marks
Four dimensions-1 Marks
Diagram-1 Marks
#Page-6

Explain benefits and disadvantages of cloud computing.
Benefits-1.5 Marks
Disadvantages-1.5 Marks
#Page-16, 18

Define the term Composability. Explain its benefits.
Definition-1.5 Marks
Benefits-1.5 Marks
#Page-46

Explain IaaS Workloads with the help of a diagram.
Definition-1.5 Marks
Diagram-1.5 Marks
#Page-67

Define Mashup.
Explanation-2 Marks
Required components-1 Marks

Explain the terms PODS, Aggregation and Silos.
Explanation-1 Marks each

6 MARKS QUESTIONS

Explain NIST model of cloud computing with the help of a diagram.
Introduction and Diagram-1.5 Marks
Explanation based on deployment models-1.5 Marks
Explanation based on service models-1.5 Marks
Service attributes-1.5 Marks
#Page-5

Explain the categorization of cloud computing based on service types.
IaaS-2 Marks
PaaS-2 Marks
SaaS-2 Marks

Explain the architecture of IaaS, PaaS and SaaS with the help of a diagram.
Diagram-1.5 marks
Architecture of IaaS, PaaS and SaaS based on diagram-1.5 marks each
#Pages-11, 49, 50, 51

What are the behavioral factors related to cloud adoption?
Introduction-1 Marks
10 Factors-0.5 Marks each
#Page-31

If you want to implement cloud computing services for your business, how will you decide which class of cloud serves your purpose?
Categorization based on Service models and its features-2 Marks
Categorization based on Deployment models and its features-2 Marks
Justification and selection of a cloud model based on user requirement-2 Marks

Explain the value of Cloud computing for an enterprise?
Definition and advantages of cloud-2 Marks
Beneficieries-2 Marks
Behavioral factors-1 Mark
Measuring cloud computing costs-1 Mark
#Pages-16, 24, 31, 33

Explain Cloud Architecture.
Specify all components of architecture.

Subject : Cloud Computing
Subject Code : RLIMCA381
Course : S5 MCA-REG

Cloud Computing

Module-2

Using Cloud Computing Platforms

In File..

- Understanding Abstraction and Virtualization
- Capacity Planning
- Exploring Platform as a Service

Abstraction and Virtualization

- Virtualization Technologies
- Load Balancing and Virtualization
- Hypervisors
- Machine Imaging
- Porting Applications

Virtualization Technologies

- Most cloud-based systems combine their resources into pools that can be assigned on-demand to users. This resource pooling, helps to attain efficient utilization, provide reasonable costs to users, and proactively react to demand.
- Pooled resources can be accessed using a technique called *Virtualization*.

- Virtualization assigns a logical name for a physical resource and then provides a pointer to that physical resource when a request is made.
- The mapping of virtual resources to physical resources can be both dynamic and simple; which helps efficient resource management.

- Virtualized *components* for cloud computing:
  - *Access*: A client can request access to a cloud service from any location.
  - *Application*: A cloud has multiple application instances and directs requests to an instance based on conditions.
  - *CPU*: Computers can be partitioned into a set of virtual machines with each machine being assigned a workload. Alternatively, systems can be virtualized through load-balancing technologies.
  - *Storage*: Data is stored across storage devices and often replicated for redundancy.

- Virtualization *enables* the following *attributes* of cloud computing:
  - *Service-based:* Clients are provided services through service interfaces.
  - *Scalable and Elastic:* Services can be altered to affect capacity and performance on demand.
  - *Shared Services:* Resources are pooled to create greater efficiencies.
  - *Metered Usage:* Services are billed on a usage basis.

Load Balancing and Virtualization

- Cloud computing provides virtualized network access to a service. No matter where you access the service, you are directed to the available resources.
- *Cloud Load Balancing is the process of distributing workloads and computing resources in a cloud computing environment.*
- It allows enterprises to manage application or workload demands by allocating resources among multiple computers, networks or servers.
- Load Balancing can be implemented in hardware or in software.

- Load balancing is an optimization technique; It can be used to:
  - Increase Utilization
  - Reduce Response Time
  - Avoid System Overload
  - Increase Throughput (Capacity)
  - Lower Latency (Delay before processing request)

- The following network resources can be load balanced:
  - Network interfaces and services (such as DNS, FTP, and HTTP).
  - Connections through intelligent switches.
  - Processing through computer system assignment.
  - Storage resources.
  - Access to application instances.

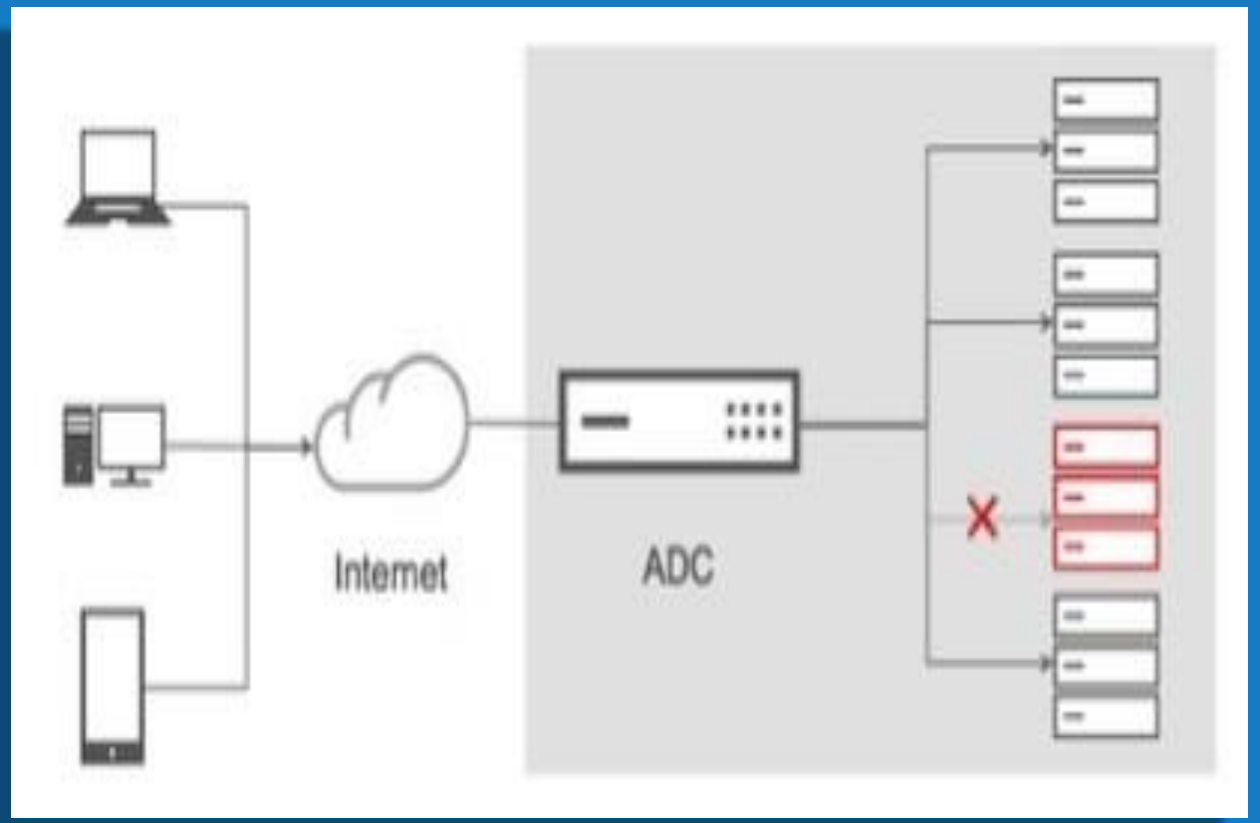
- Load balancing make a system reliable through managed redirection. It uses various algorithms for traffic routing.
- When a request from a client or service requester arrives, the load balancer uses a scheduling algorithm to assign where the request is sent.
- Typical scheduling algorithms used are Round Robin, Weighted Round Robin, Fastest Response Time, Least Connections and Weighted Least Connections.
- It also creates a session so that subsequent traffic related to that session is routed to the same resource.

Advanced Load Balancing

- Workload managers are more sophisticated load balancers.
- In order to assign tasks to each resource, they determine the current utilization of the resources in their pool, the response time, the work queue length, connection latency and capacity, and other factors.
- An ***Application Delivery Controller*** (ADC) is a combination of load balancer and application server. It is placed between a firewall/router and a data center providing the web services.
- ADCs are also referred to as a *content switch*, *multilayer switch* or *web switch*.

- An ADC is assigned a virtual IP address (VIP) that it maps to a pool of servers.
- An ADC is considered to be an advanced version of a load balancer and lowers the workload of the Web servers.
- Services provided by an ADC include data compression, content caching, security, server health monitoring, Secure Sockets Layer[SSL] offload and advanced routing based on current conditions.
- An ADC is considered to be an application accelerator, which focus on two areas of technology:
  - Network optimization
  - Application/framework optimization.

- An architectural layer containing ADCs is described as an Application Delivery Network (ADN).
- The purpose of an ADN is to distribute content to resources based on application specific criteria.
- ADN provide a caching mechanism to reduce traffic, traffic prioritization and optimization, and other techniques.

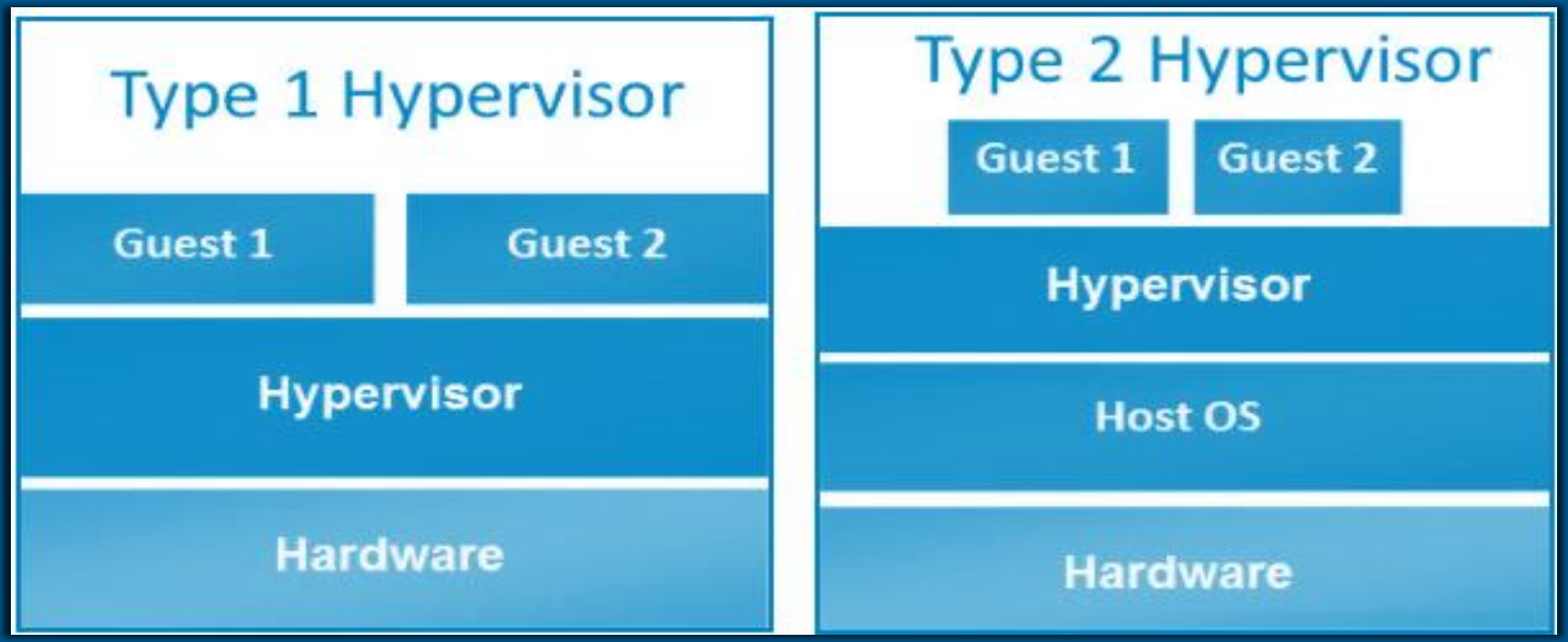


Hypervisors

- Virtual systems are created out of physical systems. A portions of computer system resources can be set aside to create a virtual machine.
- Virtual machines has all the attributes and characteristics of a physical system; but it is a software, that imitates a physical machine.
- A system/hardware virtual machine has its own memory address space, processor resource allocation, and device I/O using its own virtual device drivers.

- Virtual machines that are designed to run only a single application or process are referred to as *Process virtual machines*.
- Virtual machines can run multiple machine instances, each with their own OS.
- *A **Hypervisor** or **Virtual Machine Monitor (VMM)** is a computer software or a firmware or a hardware that creates and runs virtual machines.*
- *It is a low-level program that allows multiple operating systems to run concurrently on a single host computer.*
- Hypervisors use a thin layer of code in software or firmware to allocate resources in real-time.

- A computer on which a hypervisor runs one or more virtual machines is called a host machine, and each virtual machine is called a guest machine.
- Hypervisor/Virtual Machine Types:
  - Type 1
  - Type 2



Type 1 Hypervisor

- Type 1 hypervisors run directly on the system hardware. They are often referred to as a "native" or "bare metal" or "embedded" hypervisors.
- Type 1 VMs have no host operating system because they are installed on a bare system.
- Because they run directly on the hardware, Type 1 hypervisors support hardware virtualization.
- They provide higher performance, availability, and security.
- An OS running on a Type 1 VM is a full virtualization because, it is a complete simulation of the hardware that it is running on.
- Eg: Oracle VM,VMware ESX and ESXi

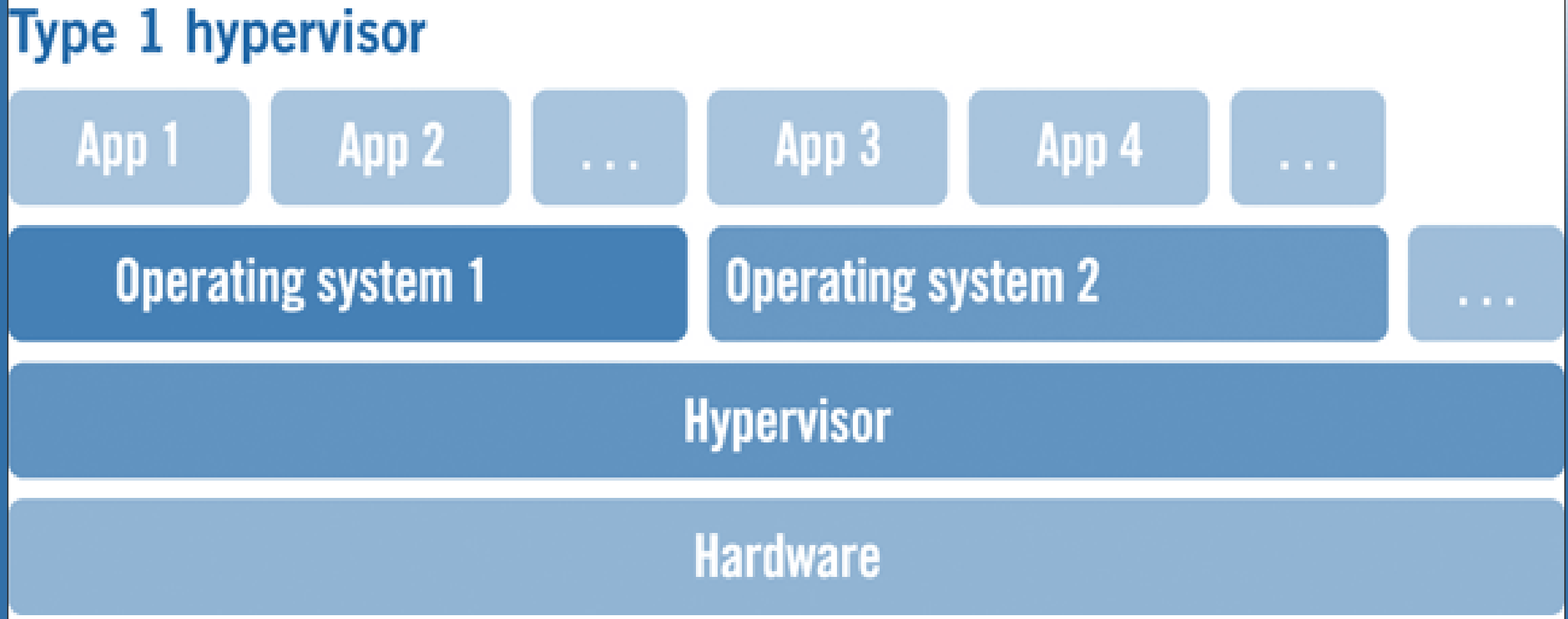


Figure 2. A Type 1 or bare-metal hypervisor sits directly on the host hardware.

Type 2 Hypervisor

- It run on a host operating system.
- Because they run as an application on top of an operating system, Type 2 hypervisors perform software virtualization.
- A software interface is created which replicates the device to which the system normally interact.
- Additional applications can be installed, executed and can be monitored; such as browser, word editor, application softwares etc.
- Eg: Microsoft Hyper V, VMware Workstation 6.0

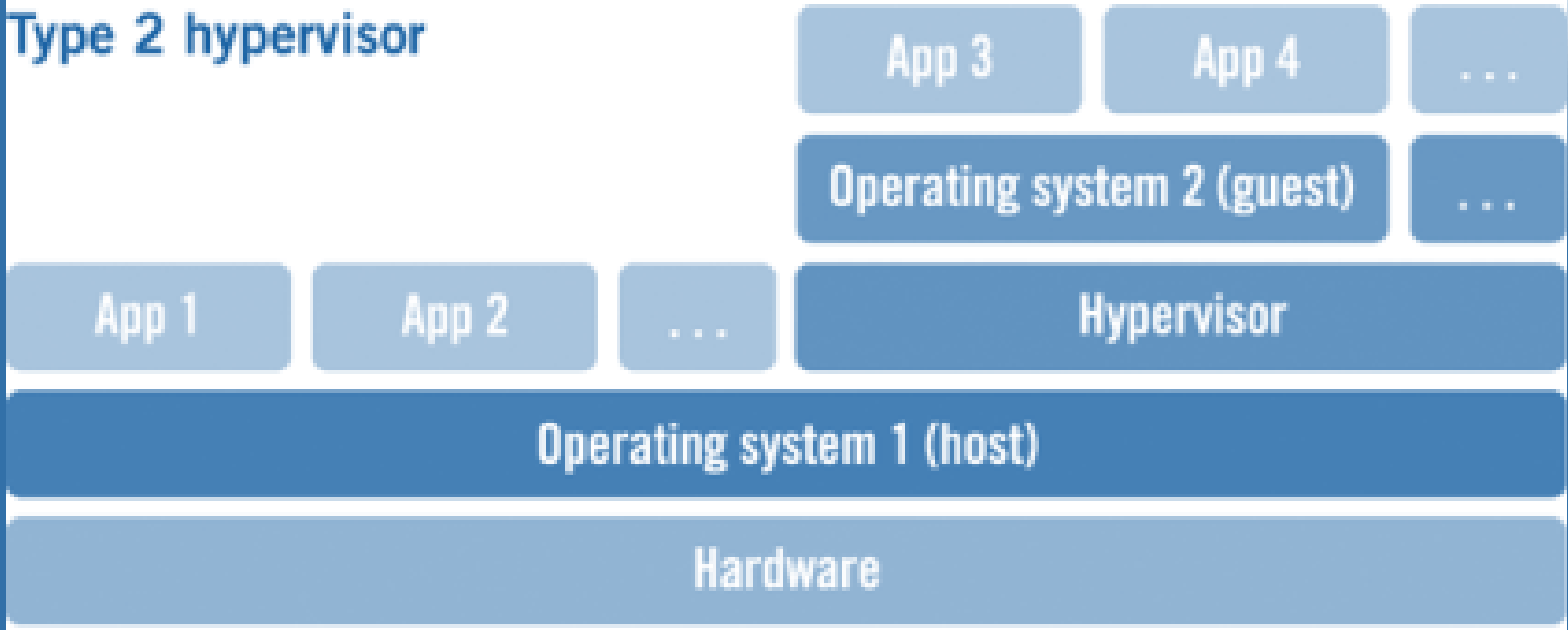
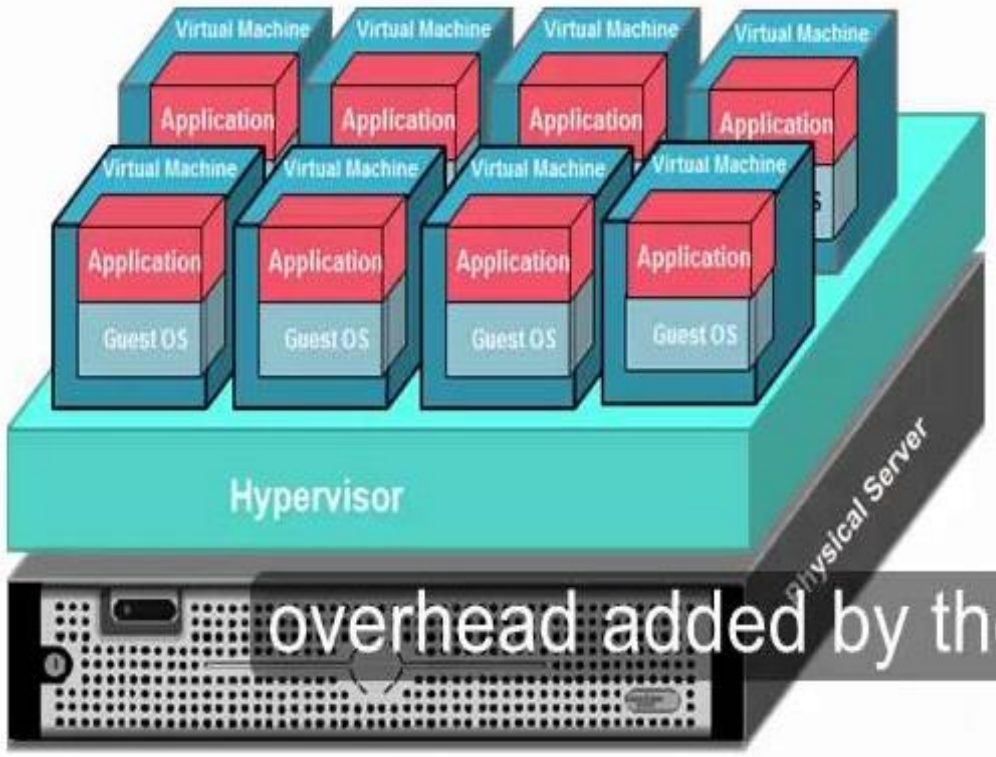
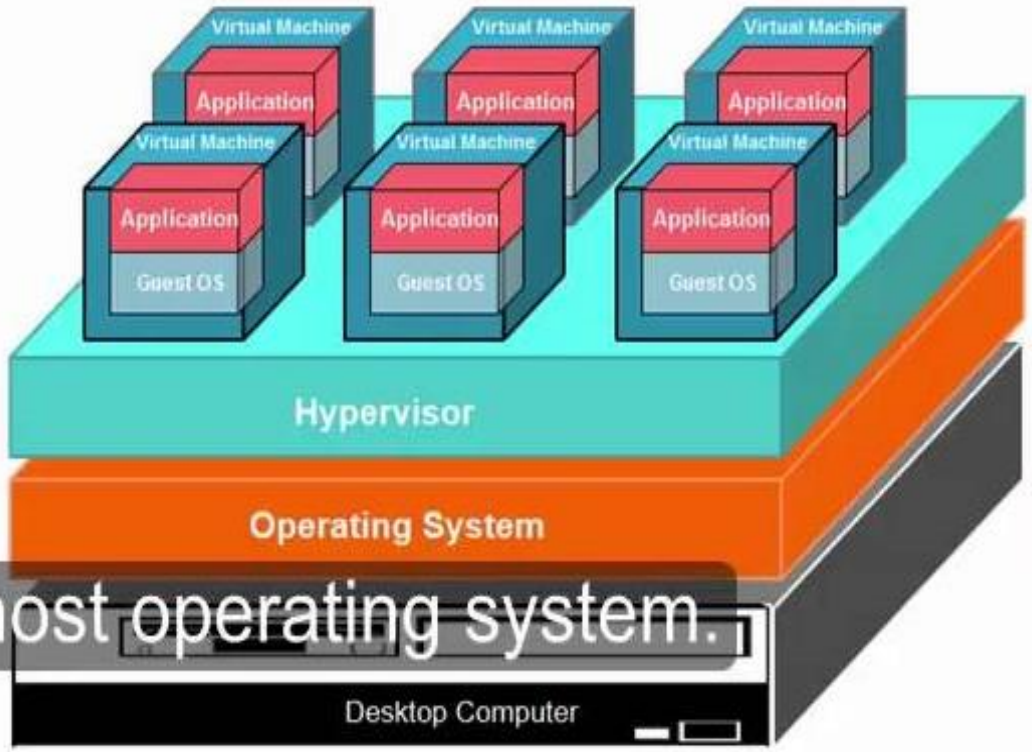


Figure 1. A Type 2 hypervisor runs as an application on a host operating system.

Type-1 Hypervisor	Type-2 Hypervisor
Higher performance and scalability because of being bare-metal type	Lower performance as a result of host operating system overhead



Type-1 Hypervisor



Type-2 Hypervisor

overhead added by the host operating system.

Emulation

- In emulation, the virtual machine replicates the hardware, so it can be independent of the underlying system hardware.
- A guest operating system using emulation does not need to be modified in any way.

Advantages:

- Widest hardware compatibility
- More portable applications/images

Disadvantages:

- Worst user experience. Too much overhead of running customized applications, graphics etc
- Worst performance.

Para Virtualization

- The host OS provides a virtual machine interface for the guest OS. The guest access hardware through that host VM.
- Here, an OS running as a guest must be ported/modified to work with the host interface.

Advantages:

- Good image portability

Disadvantages:

- Complex driver architecture
- Compatibility is limited by the vendors

Full Virtualization (Hardware pass-through)

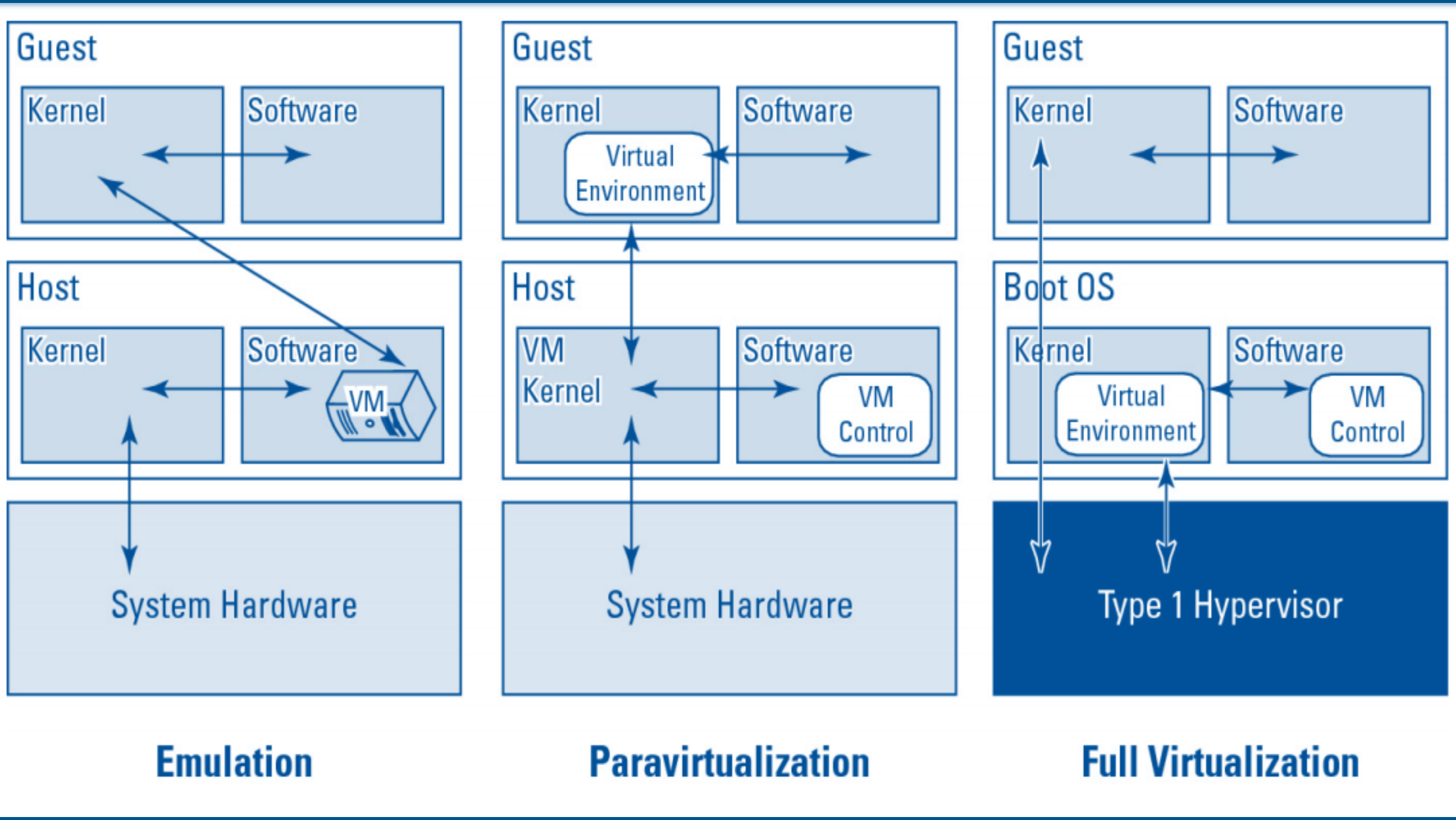
- Here, the VM is installed as a Type 1 Hypervisor directly onto the hardware.
- All OS's communicate directly with the VM hypervisor, so guest OS do not require any modification.
- Guest operating systems in full virtualization systems are generally faster than other virtualization schemes.

Advantages:

- Best user experience
- Native performance

Disadvantages:

- Hardware-specific images


Guest
Kernel
Software
Host
Kernel
Software
VM
System Hardware
Emulation
Guest
Kernel
Virtual Environment
Software
Host
VM Kernel
Software
VM Control
System Hardware
Paravirtualization
Guest
Kernel
Software
Boot OS
Kernel
Virtual Environment
Software
VM Control
Type 1 Hypervisor
Full Virtualization

Machine Imaging

- A mechanism to provide system portability, instantiate applications and deploy systems in the cloud through storing the state of a systems using a system image.
- A system image makes a copy or a clone of the entire computer system inside a single container such as a file.
- The system imaging program is used to make this image and can be used later to restore a system image.

- Some imaging programs allow you to view the files contained in the image and perform partial restores.
  - Eg: AWS's Amazon Machine Image (AMI); a file system image that contains an OS, device drivers, applications and state information of the virtual machine used.

Porting Applications

- Cloud computing applications run on virtual systems and they can be moved as needed to respond to demand.
- Systems (VMs running applications), storage, and network assets can all be virtualized.
- Applications that run in datacenters are captive to the OS and hardware platforms that they run on. For Example, porting an application build on Microsoft Azure platform to AWS or GoogleApps is difficult.

- Also, the applications are tightly coupled with the OS on which they run due to the use of Dynamic Link Libraries (DLL) and registry files. So moving an application from one platform to another is not simple.
- Though, Cloud Developers wanted the ability to port their applications from one cloud vendor to another, major cloud vendors don't have technologies that interoperate with one another.
- Some technologies used for porting applications:
  - Simple Cloud API by by Zend Technologies
  - Virtual Application Appliance by AppZero

Simple Cloud API

- The *Simple API* for Cloud Application Services is an open source initiative by *Zend Technologies* to create a common application program interface that will allow applications to be portable.
- Founding supporters are IBM, Microsoft, Nivanix, Rackspace, and GoGrid.
- Simple Cloud API has common interfaces for:
  - File Storage Services
  - Document Storage Services
  - Simple Queue Services

Virtual Application Appliance

- *Virtual Application Appliance (VAA),* developed by *AppZero* company, gives the ability to run an application from whatever platform you want.
- It is created as an architectural layer between the Windows or the UNIX OS and the applications. The virtualization layer serves as the mediator for file I/O, memory I/O, and application calls and response to DLLs.
- The running application in AppZero changes none of the registry entries or any of the files on the Windows Server.
- VAA creates a container which encapsulates the application and all its dependencies within a set of files; it is an application image for a specific OS.

- Dependencies includes DLL, service settings, configuration files, registry entries, and machine and network settings.
- This container forms an installable server-side application stack that can be run after installation, but has no impact on the underlying OS.
- VAAs are created using the AppZero Creator wizard, managed with the AppZero Admin tool, and may be installed using the AppZero Director, which creates a VAA runtime application.
- AppZero Dissolve removes the VAA virtualization layer from the encapsulated application and installs that application directly into the operating system.

Capacity Planning

- Capacity Planning
- Defining Baseline and Metrics
  - ✓ Baseline Measurements | System Metrics | Load Testing
  - ✓ Resource Ceiling | Server and Instance Types
- Network Capacity
- Scaling

Capacity Planning

- Capacity planning examines the available systems, measures their performance, and determines patterns in usage that enables the planner to predict demand.
- A system uses processor, memory, storage, and network capacity to satisfy cloud computing demands.
- Each of these resources has a utilization rate, and these resources reaches a ceiling that limits performance when demand increases.

- Resources are provisioned and allocated to meet demand. *The goal of capacity planning is to accommodate the workload.*
- *Capacity planning* measures the maximum amount of work that can be done using the current technology and then adds resources to do more work as needed.
- It is the goal of a capacity planner to identify the critical resource that has resource ceiling and add more resources to move the bottleneck to higher levels of demand.

Capacity planning steps:

1. Determine the characteristics of the present system.
2. Measure the workload for the different resources in the system: CPU, RAM, disk, network, and so forth.
3. Load the system until it is overloaded, determine when it breaks, and specify what is required to maintain acceptable performance.
4. Predict the future based on historical trends and other factors.
5. Deploy or tear down resources to meet your predictions.
6. Iterate Steps 1 through 5 repeatedly.

Defining Baseline and Metrics

- In business, the current system capacity or workload should be determine as a measurable quantity over time.
- Many developers create cloud-based applications and Web sites based on a LAMP solution stack.
- LAMP stands for:
  - *Linux: the operating system*
  - *Apache HTTP Server: the Web server.*
  - *MySQL: the database server*
  - *PHP [Hypertext Preprocessor]: the scripting language*
- These four technologies are open source products.

1. Baseline Measurements

- Two important overall workload metrics in this LAMP system:
  - *Page views or hits* on the Web site, as measured in hits per second.
  - *Transactions completed* on the database server, as measured by transactions per second or perhaps by queries per second.
- The total workload might be served by a single server instance in the cloud, a number of virtual server instances, or some combination of physical and virtual servers.

- Workload characteristics are determined by:
  - WT : the total workload for the system per unit time.
  - WAVG : the average workload over multiple units of time.
  - WMAX : the highest amount of work recorded by the system.
  - WTOT : the total amount of work done by the system.

2. System Metrics

- Capacity planning must measure system-level statistics, determining what each system is capable of, and how resources of a system affect system-level performance.
- A machine instance (physical or virtual) is primarily defined by four essential resources:
  - CPU
  - Memory (RAM)
  - Disk
  - Network connectivity

- In Linux/UNIX, sar command display the level of CPU activity. In Windows, the Task Manager serves this purpose.
- Linux performance measurement tool RRDTool (Round Robin Database tool) capture time-dependent performance data from resources such as a CPU load, network utilization (bandwidth), and so on and store the data in a circular buffer. It is commonly used in performance analysis work.
- Some LAMP Performance Monitoring Tools are:
  - Alertra: Web site monitoring service
  - Collectd: System statistics collection daemon

3. Load Testing

- The aim of Load Testing is to check what happens to a system when the load increases.
- It is also referred to as performance testing, reliability testing, stress testing, and volume testing.
- Upon reaching the maximum load, cloud can create virtual clone of the system to perform tasks.
- Examples of load generation tools: HP LodeRunner, IBM Rational Performance Tester, JMeter
- Load balancers serves more requests to more powerful systems and fewer requests to less powerful systems.

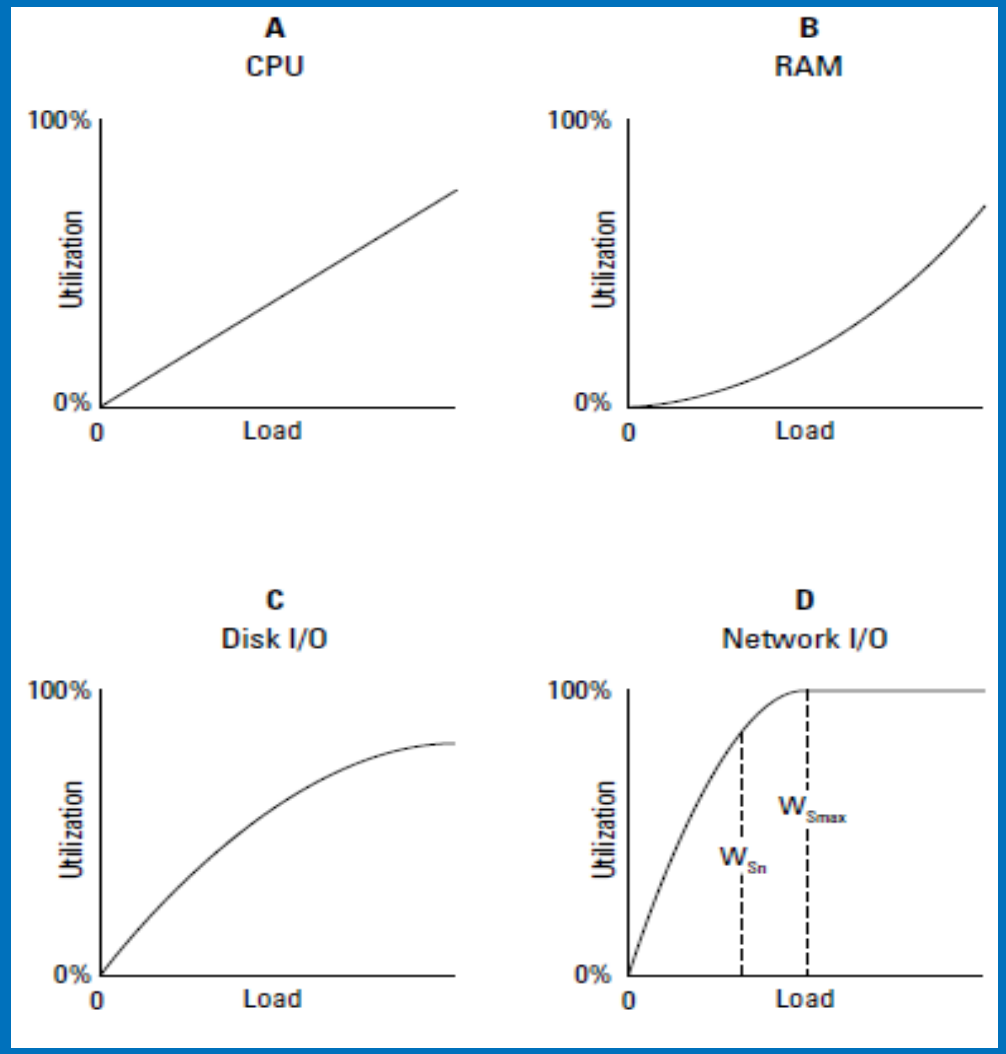
- Load testing seeks to answer the following questions:
  - What is the maximum load that my current system can support?
  - Which resource(s) represents the bottleneck in the current system that limits the system's performance?
  - Can I alter the configuration of my server in order to increase capacity?
  - How does a server's performance relate to other servers that might have different characteristics?

4. Resource Ceiling

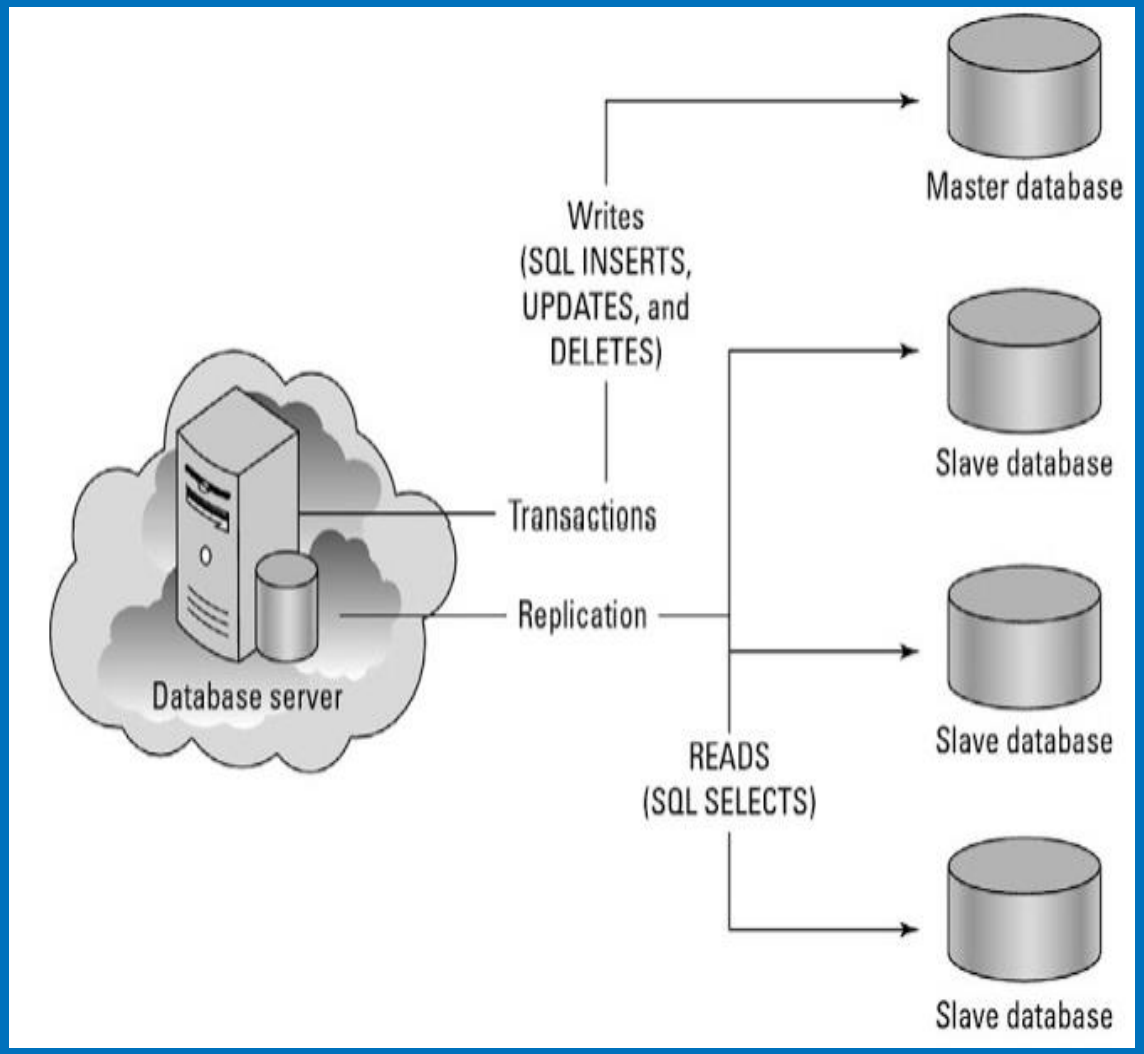
- Among several components (like the CPU, RAM, Network I/O and Disk I/O) of a particular server, if any component reaches its maximum utilization while functioning, this factor is the current system's *resource ceiling*.
- Since a particular component reaches its maximum utilization, more utilization of other resources might not be possible even though the system is not fully loaded.

Some common issues and solutions:

- Network I/O is often a bottleneck in Web servers.
  - *So, Websites prefer to scale out using many low-powered servers instead of scaling up with fewer but more powerful servers.*
- Database servers usually exhibit resource ceilings.
  - *Solution is to replicate master MySQL database and create a number of slave MySQL databases. All READ operations are performed on the slave MySQL databases, and WRITE operations on the master MySQL database.*



Network I/O Resource Ceiling



Database Servers Resource Ceiling

5. Server and Instance Types

- Capacity planning makes the growth and shrinkage of resources predictable.
- In cloud computing, you can automatically or manually increase the capacity, on demand, quickly and efficiently.
- It can be made possible by standardizing and characterizing the hardwares and platforms.
- Assign servers with standardized roles and populate those servers with identical services.
- A server with the same set of software, system configuration, and hardware should perform similarly if given the same role in an infrastructure.

- Capacity planning compare the capability of different systems and choose the solution that is right sized and provides the service with the best operational parameters at the lowest cost.
- Server instances of various sizes are created to performs different kinds of tasks.
- For example; In an *Amazon Machine Instance (AMI):*
  - A *Small Instance* has 1.7 GB memory, 1 EC2 Compute Unit, 160GB instance storage, 32-bitplatform and Moderate I/O Performance.
  - A *High-CPU Extra Large Instance* has 7 GB of memory, 20 EC2 Compute Units, 1,690 GB of instance storage, 64-bit platform and High I/O Performance.

Network Capacity

- There are three aspects to assessing network capacity:
  - Network traffic to and from the network interface at the server, be it a physical or virtual interface or server(Server side network)
  - Network traffic from the cloud to the network interface (Measurement of WAN traffic)
  - Network traffic through your ISP to your local network interface (your computer)

- To measure network traffic at a server's network interface, a network monitor is used, which is a form of packet analyzer.
- Eg: Microsoft includes a utility called the Microsoft Network Monitor as part of its server utilities.
- The statistics function of tools like Wireshark, Dsniff etc provides a measurement of network capacity as expressed by throughput.

Scaling

- Scalability is a key feature provided in cloud computing.
- This is achieved with the help of providing adequate infrastructure by increasing or decreasing the resources.
- You can either scale vertically (scale up) or scale horizontally (scale out), and each method is broadly suitable for different types of applications.

Vertical Scaling

- In this method, we add resources to a system to make it more powerful. Results in single powerful supercomputer.
  - Eg: Replacing a dual-processor machine instance equivalence with a quad-processor machine instance equivalence
- Vertical scaling allows you to use a virtual system to run more virtual machines (operating system instance), run more daemons on the same machine instance, or take advantage of more RAM (memory) and faster compute times.

Horizontal scaling

- It adds capacity to a system by adding more individual nodes.
  - Eg: In a dual-processor machine instance, you add more dual-processor machines instances.
- Scaling out indefinitely leads you to an architecture with a large number of servers, which is the model that many cloud and grid computer networks use.
- It allows you to run distributed applications more efficiently and is effective in using hardware more efficiently because it is both easier to pool resources and to partition them.

Exploring Platform as a Service

- Exploring Platform as a Service
- Application Development
- Using PaaS Application Frameworks
  - *Drupal* | *Squarespace* | *Eccentex*
  - *LongJump* | *WaveMaker* | *Wolf Frameworks*

Exploring Platform as a Service

- The PaaS model provides the tools and environment that are needed to create applications that can run in a SaaS model.
- Applications developed in PaaS systems can be composite business applications, data portals, or mashups with data derived from multiple sources.
- Application frameworks are powerful tool for creating cloud computing applications.

- PaaS systems provides a toolkit to work with and also a VM to run your software. Using these, the developers design the software and UI that serves the needs.
  - Eg: Windows Azure Platform, Drupal, Squarespace, Wolf. These tools are very well developed and require almost no coding.

- The services provided by PAAS model are:
  - *Application Development.*
  - *Collaboration:* Allows multiple individuals to work on the same projects.
  - *Data management:* Tools for accessing and using data.
  - *Instrumentation, performance, and testing:* Tools for measuring applications and optimizing their performance.
  - *Storage:* Data can be stored in either the PaaS vendor's service or accessed from a third-party storage service.
  - *Transaction Management.*

Application Development

- Common application types developed using PaaS tools are:
  - Composite business applications.
  - Data portals.
  - Mashups of multiple data sources.
- Major Application Development Tools are: Google AppEngine, Microsoft Windows Azure Platform, Eccentex AppBase, LongJump, and Wolf
- The vendor provides a full software development stack for the programmer to use; the developer need not go outside of the service to create his application.

- All PaaS application development must have lifecycle management. As an application ages, it must be upgraded, migrated, grown, and eventually phased out or ported.
- Most PaaS vendors offers integrated systems with lifecycle development platforms.
- An integrated lifecycle platform includes the following:
  - The virtual machine and operating system (often offered by an IaaS).
  - Data design and storage.
  - A development environment with defined Application Programming Interfaces.
  - Middleware.
  - Testing and optimization tools.
  - Additional tools and services.

Drupal

- A full-strength developer tool which is modular and exposes its functionality through a set of published APIs.
- The contrib modules can be added to Drupal to replace other modules, enhance capabilities, or provide entirely new features.
- Third-party modules include messaging systems, visual editors, a content construction kit (CCK) for database schema extension, views, and panels.

Squarespace

- A next-generation website builder and deployment tool.
- Presents itself as a blogging tool, social media integration tool, photo gallery, form builder and data collector, item list manager, traffic and site management and analysis tool.

Eccentex

- Uses its AppBase architecture to create Cloudware applications.
- AppBase includes a set of different tools for building following applications: Business Objects Build, Presentation Builder, Business Process Designer, Dashboard Designer, Report Builder, Security Roles Management

LongJump

- Its development environment is based on Java and uses REST/SOAP APIs.
- LongJump's PaaS is based on standard Java/JavaScript, SOAP, and REST.
- It creates browser-based Web applications that are database-enabled.
- LongJump comes with an Object Model Viewer, forms, reports, layout tools, dashboards, and site management tools.
- Access control is based on role-based and rule-based access, and it allows for data-sharing between teams and between tenants.

WaveMaker

- An application development environment for creating Java-based Web and cloud Ajax applications.
- The software is open-source and offered under the Apache license.
- WaveMaker is a drag-and-drop environment that runs inside a browser.

Wolf Frameworks

- Wolf Frameworks is an application development framework which is open, standards-based, and portable.
- It is a PaaS vendor, who offers a platform, where you can build a SaaS solution that is open and cross-platform.

- Wolf Frameworks is based on: AJAX, asynchronous Java, XML and .NET Framework
- Using Wolf, applications can be built without writing technical code.
- It also allows application data to be written to the client's database server, and data can be imported or exported from a variety of data formats.
- Wolf supports forms, search, business logic and rules, charts, reports, dashboards, and both custom and external Web pages.
- After you create entities and assign their properties, you create business rules with a rules designer.
- You can automate tasks via business rules.

3 MARKS QUESTIONS

Explain the role of Load Balancing in cloud computing.
Virtualization feature and need for Load balancing-1.5 Marks
Types of Load balancing-1.5 Marks
#Page-95

Explain Advanced Load Balancing.
Definition-1 Marks
Explanation-1 Marks

Explain Emulation, Para-Virtualization and Full-Virtualization.
Emulation-1 Marks
Para-Virtualization-1 Marks
Full-Virtualization-1 Marks
#Page-101

Elaborate on the technologies used for Porting Cloud Applications.
Simple Cloud API-1.5 Marks
AppZero Virtual Application Appliance-1.5 Marks
#Page-109

Compare Load Testing and Resource Ceiling.
Load Testing-1.5 Mark
Resource Ceiling-1.5 Mark
#Page-121

Define Scaling and its classification.
Definition of scaling-1 Marks
Horizontal scaling-1 Marks
Vertical scaling-1 Marks
#Page-131

Explain the services provided by PAAS model.
Six methods-0.5 marks each

6 MARKS QUESTIONS

Explain Hypervisor and its categories with the help of diagrams.
Definition of Virtual machine and use of Hypervisor-2 Marks
Type 1 Hypervisor with diagram-2 Marks
Type 2 Hypervisor with diagram-2 Marks
#Page-100

Explain the methods used for porting application in cloud.
Porting definition-1 Mark
The Simple Cloud API-2.5 Marks
AppZero Virtual Application Appliance-2.5 Marks

Define capacity planning. Explain the role of capacity planning in Cloud Computing.
or
Explain the parameters and methods used to implement Capacity Planning.
Definition and need of capacity planning.-1 Mark
Baseline Measurements, System Metrics, Load Testing, Resource Ceiling, Server and Instance Types.-5 Mark

Elaborate on the commonly used Application Development frameworks.
Drupal, Squarespace, Eccentex, LongJump, WaveMaker, Wolf Frameworks -1 Marks each