Unit - I INTRODUCTION

Introduction to cloud computing - Definition of cloud - Evolution of cloud computing - Underlying Principles of Parallel and Distributed computing - Cloud Characteristics - Elasticity in cloud - On demand Provisioning.

Introduction to cloud computing:

→ A simple definition of cloud computing involves delivering different types of services over the Internet.

→ Cloud delivers,

(i) Software and analytics

(ii) Secure and safe data storage

(iii) networking resources.

Different cloud-based Applications:

(i) Send a file to your friends via the web

(ii) use a mobile app

(iii) download an image

(iv) binge a Netflix show

(v) play an online video game

(vi) Storing your information on OneDrive, Sharepoint or an email server.

What is cloud computing?

→ Cloud computing means storing and accessing the data and programs over the internet rather than the computer's hard disk.

Types of Cloud Computing Services:

(i) Infrastructure As-A-Service (IAAS)

(ii) Platform As-A-Service (PAAS)

(iii) Software As-A-Service (SAAS)

Service Models:

(i) Iaas

(ii) Paas

(iii) Saas

Infrastructure -As-A- Service (IAAS):

→ Involves a method for delivering everything from operating systems to servers and storage through IP based connectivity.

→ Popular examples of Iaas system include IBM Cloud and Microsoft Azure.

Platform - As-A-Service:

→ It is a set of software and development tools hosted on the providers servers.

→ Google App is the one of the most famous Platform-as-a Service Providers.

Software -As-A- Service:

→ Software-as-a Service is the broadest market.

→ The software interacts with the user through a user Interface.

The World of Business:

(i) Google Cloud

(ii) Amazon Web Services (AWS)

(iii) Microsoft Azure

(iv) IBM Cloud

(v) Alibaba Cloud.

Definition of Cloud:

NIST Definition of Cloud computing:

→ Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Characteristics of Cloud computing:

(i) On Demand Self-service (ii) Broad network Access

(iii) Resource Pooling (iv) Rapid elasticity or expansion

(v) measured service.

Service models:

(i) Saas (ii) Paas (iii) Iaas

Deployment models:

(i) Private Cloud (ii) Community cloud

(iii) Public Cloud (iv) hybrid cloud.

Other Definition of Cloud computing:

→ The practice of using a network of remote servers hosted on the Internet to store, manage and process data rather than a local server or a personal computer. This is an Internet definition of Cloud Computing.

Cloud Service Providers:

(i) Microsoft Azure (ii) Amazon Web Services (AWS)

(iii) Google cloud (iv) Alibaba Cloud

(v) IBM Cloud (vi) Oracle

(vii) Salesforce (viii) SAP

(ix) Rackspace Cloud (x) VMware

Evolution of Cloud Computing :

Hardware Evolution:

(a) First Generation Computers

(b) Second Generation Computers

(c) Third Generation Computers

(d) Fourth Generation Computers

(e) Fifth Generation Computers.

Internet Software Evolution:

(a) Establishing a Common Protocol for the Internet.

(b) Evolution of IPV6

(c) Finding a common method to Communicate using the Internet Protocol.

(d) Building a Common Interface to the Internet

(e) Appearance of Cloud Formation

Server Virtualization:

(a) Parallel Processing

(b) Vector Processing

(c) Symmetric Multiprocessing System.

(d) Massively Parallel Processing System

Internet Software Evolution:

Internet Protocol:

→ It is the Standard Communication Protocol used by every Computer on the Internet.

→ The Conceptual foundation for Creation of the Internet was significantly developed by three Individuals:

(a) ARPANET (Advanced Research Project Agency Network)

(b) IMP (Interface Message Processor)

(c) NCP (Network Control Program).

IMP Interface Message Processor:

(i) First packet switching router

(ii) First generation gateway

(iii) Used to Connect users networks to ARPANET

(iv) ARPANET - Advanced Research Project Agency.

Establishing a common Protocol for the Internet:

→ NCP essentially provided a transport layer consisting of the ARPANET Host-to host protocol (AHHP) and the Initial Connection protocol (ICP)

Transport Layer Protocols:

(i) AHHP - ARPANET Host-Host Protocol - Unidirectional Data transfer

(ii) ICP - Initial Connection Protocol - Bidirectional Data transfer.

Application Protocols:

(i) File Transfer Protocols (FTP) used for file transfers.

(ii) Simple Mail Transfer Protocol (SMTP) used for sending email.

Finding Common Method To Communicate using the Internet Protocol:

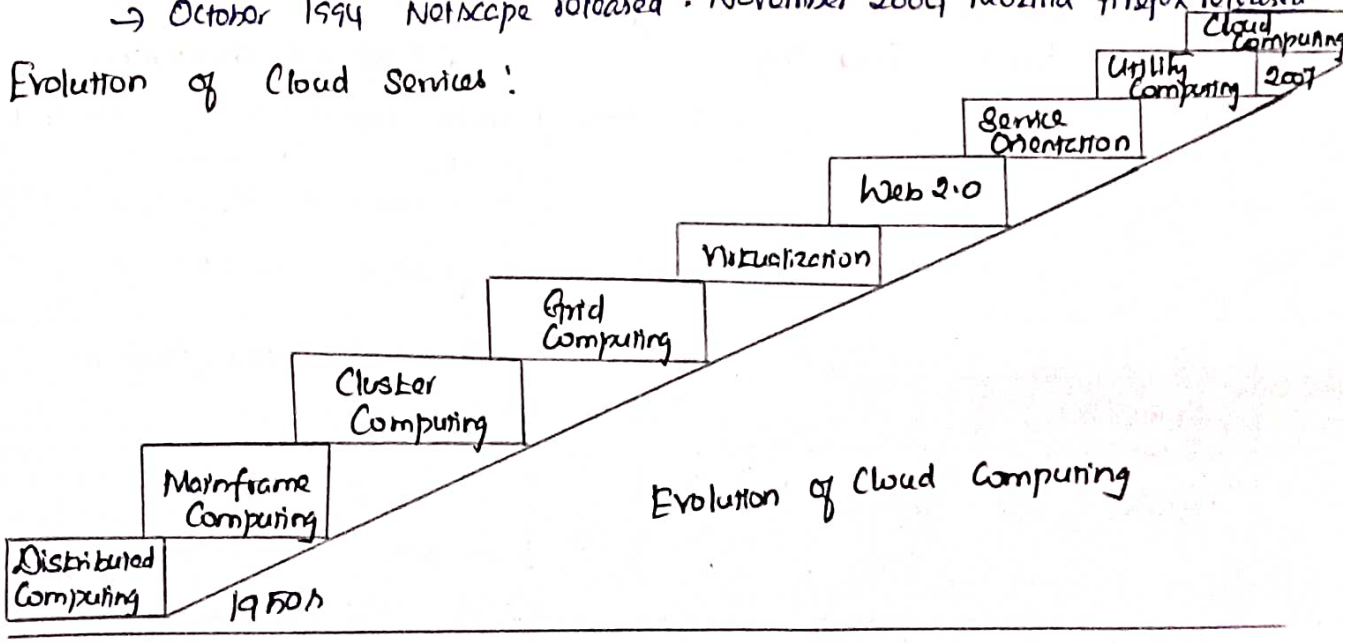
→ In 1960, word hypertext was created by Ted Nelson.

→ In 1962 - Engelbart's first project

→ NLS was designed to cross reference research papers for sharing among geographically distributed researchers.

→ October 1994 Netscape released. November 2004 Mozilla Firefox released.

Evolution of Cloud Services:



Evolution of Cloud Computing

Underlying Principles of Parallel and Distributed Computing

Eras of Computing:

→ Two fundamental and dominant models of computing are sequential and parallel.

→ Four key elements

(i) Architecture

(ii) Compilers

(iii) Applications

(iv) Problem Solving environments.

Parallel Vs Distributed Computing:

Parallel Computing:

→ Implies a tightly coupled system.

→ parallel systems are featured with multiple processors sharing the same physical memory and that are considered a single computer.

Distributed Computing:

→ refers to loosely coupled system.

→ Wider range of systems and applications.

(i) Computing Grids

(ii) Internet Computing System.

Parallel Vs Distributed Computing:

Parallel Computing	Distributed computing.
Many operations are performed simultaneously	Some components are located at different locations.
Single Computer is required	uses multiple computers
Shared or distributed memory	only distributed memory.
Multiple processor perform multiple operations.	Multiple computers perform multiple operations.
Improves the System Performance.	Improves system scalability, fault tolerance and resource sharing capabilities.

What is Parallel Processing?

→ Processing of multiple tasks simultaneously on multiple processors is called parallel processing.

→ The parallel program consists of multiple active process solving a given problem.

Advantages:

(i) Multiple processors provides higher computing power.

(ii) Higher performance than a single processor system.

Hardware Architectures for Parallel Processing:

→ The core elements of parallel processing are CPUs

→ Four categories

(i) Single Instruction, Single Data (SISD) systems

(ii) Single Instruction, Multiple Data (SIMD) systems.

(iii) Multiple Instruction, Single Data (MISD) Systems.

(iv) Multiple Instruction, Multiple Data (MIMD) Systems.

Approaches to Parallel Processing:

(i) Data Parallelism

(ii) Process Parallelism

(iii) Farmer and Worker Model.

Data level Parallelism:

→ divide and conquer technique is used.

→ to processing on SIND models.

Process Parallelism:

→ multiple activities that can be processed on multiple processors.

Farmer and Worker Model:

→ Job distribution approach is used.

→ one processor is configured as master and all other remaining PEs are designated as slaves.

Levels of Parallelism:

Size	Code Item	Parallelized By
Large	Separate and heavy weight process	Programmer
Medium	Function or Procedure	Programmer
Fine	Loop or instruction block	Parallelizing Compiler
Very Fine	Instruction	Processor.

Elements of Distributed Computing:

→ the execution of multiple operations by multiple computers.

→ four concepts.

(i) General concepts and definitions

(ii) Components of distributed system

(iii) Architectural Styles of distributed computing

(iv) Models for Inter Process Communication.

General Concepts and definitions:

→ Distributed computing studies the models, architectures and algorithms used for building and managing distributed system.

→ A distributed system is a collection of independent computers that appears to its users as a single coherent system.

Components of Distributed System:

→ Layer 1: bottom Layer – computer and network hardware

→ Layer 2: Operating System – Inter Process Communication, Process scheduling

→ Layer 3: Middleware – distributed Computing.

→ Layer 4: Applications – Specialized Software is deployed to turn a set of networked computers into a distributed system.

Architectural Styles for Distributed Computing:

→ understanding the organization of the software systems in distributed computing.

→ two classes.

(i) Software Architectural Styles.

(ii) System Architectural Styles.

Software Architectural Styles:

→ logical arrangement of software components.

→ an intuitive view of the whole system.

Architectural Style categories:

(i) Data Centered Architecture

(ii) Data Flow Architecture

(iii) Virtual machine Architecture

(iv) Call and Return Architecture

Data centered Architecture:

→ These architectures identify the data as the fundamental element of the software system.

→ Goal: Integrity of Data.

Data flow Architecture:

→ data flow styles explicitly incorporate the pattern of data flow.

→ determined by the data flow from component to component.

→ 2 styles

1) Batch Sequential:

→ characterized by an ordered sequence of separate programs executing one after the another.

→ very popular in the mainframe era of computing and still finds applications today.

2) Pipe and filter Style:

→ Expresses the activity of a software system as sequence of data transformations.

→ Each component of the processing chain is called filter.

Virtual Machine Architectures:

→ It contains an abstract execution environment.

→ popular examples are,

(a) Rule based systems

(b) Interpreters

(c) Common language processors.

Call and Return Architectures:

→ Identifies all systems that are organized into components mostly connected together by method calls.

→ three categories,

(a) Top down style: → imperative programming

(b) Object oriented style → Object Programming Models

(c) Layered style: → abstraction of system.

System Architectural Styles:

→ Cover the physical organization of components and processes over a distributed infrastructure.

→ Two fundamental styles

(a) Client / Server

(b) Peer - to - Peer

Client/Server	Peer-to-Peer
Information and the services can be centralized.	Information and services cannot be centralized.
Server used to serve requests coming from different clients.	Each peer acts as server.
Single Point of failure.	No Single Point of failure.

Methods of Inter Process Communication:

→ IPC is fundamental aspect of distributed system design and implementation.

→ Either exchange data and Information or coordinate the activity of processes.

IPC Models:

(i) Message Passing Model.

(ii) Remote Procedure call (RPC)

(i) Distributed objects.

(ii) Distributed agents and active objects.

(iii) Web Service.

Models of Message Passing:

(i) Point-to-Point Message Model.

(ii) Publish and Subscribe Message Model.

(iii) Request-reply message Model.

Technologies of Distributed Computing:

(i) Remote Procedure call

(ii) Distributed object Frameworks

(iii) Service Oriented Computing

(iv) Web Services.

Examples of Distributed Object Frameworks:

(i) Common Object Request Broker Architecture (CORBA)

(ii) Distributed Component Object Model (DCOM/COM+)

(iii) Java Remote Method Invocation (RMI)

(iv) .NET Remoting.

Service Oriented Architecture (SOA):

→ SOA is an architectural style supporting Service Orientation. It organizes a software system into collection of interacting services

→ two major roles:

(i) Service Provider

(ii) Service Consumer.

Web services:

→ prominent technology for implementing SOA systems and applications.

→ They leverage Internet technologies and standards for building distributed system.

Cloud characteristics:

Characteristics of cloud computing as per NIST:

→ NIST is responsible for defining standards in science and technology

→ NIST provided a formal definition and characteristics of cloud computing.

Characteristics of Cloud Computing in NIST:

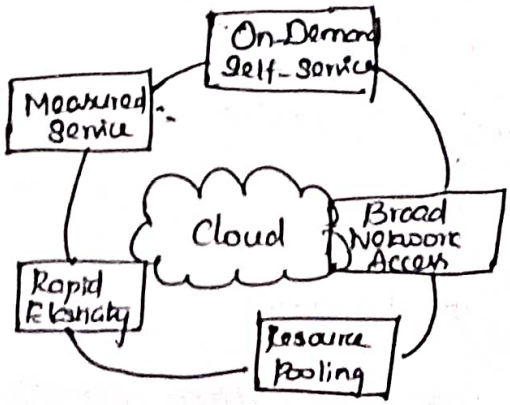
(i) On-demand self service

(ii) Broad network access

(iii) Resource Pooling

(iv) Rapid Elasticity

(v) Measured Service



190 17788 Essential characteristics of cloud computing:

(i) On Demand self service

(ii) Broad Network Access

(iii) Resource Pooling

(iv) Rapid Elasticity

(v) Measured Service

(vi) Multi-Tenacy.

On Demand Self-Service:

→ Cloud computing provides resources on demand i) When the consumer wants it. This is made possible by self service and automation.

Broad Network Access:

→ Cloud capabilities are available over the network and accessed through standard mechanisms that promote use by heterogenous thin or thick client platforms.

→ eg mobile phones, tablets, laptops and workstations.

Resource Pooling:

→ The providers computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Rapid Elasticity:

→ Capabilities can be elastically provisioned and released in some cases automatically to scale rapidly outward and inward commensurate with demand.

Measured Service:

→ Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service.

→ Resource usage can be monitored, controlled and reported, providing transparency for both the provider and consumer of the utilized service.

Multitenancy:

→ The customers are also called tenants, can have different business divisions inside the same company.

→ the customers are often entirely different organizations.

Elasticity in cloud:

What is Elastic Cloud Computing and how it Benefits Business?

→ Small and large business have switched their data to cloud storage.

→ Cloud computing or cloud is defined as using various services such as software development platforms, servers, storage, over the Internet.

Elastic Cloud Computing mean?

→ Elastic Computing is nothing but a concept in Cloud Computing in which computing resources can be scaled up and down easily by the cloud service Provider.

Elastic Cloud computing or fully automated scalability.

→ removes manual labor for increasing or decreasing resources as everything is controlled by triggers by the system monitoring tools.

Scalability and Elasticity:

Scalability:

→ refers to the ability of system to accomodate larger loads just by adding resources either making hardware stronger or adding additional nodes.

Elasticity:

→ refers to the ability to fit the resources needed to cope with loads.

→ load increase you scale up by adding more resources.

Benefits / Pros of Elastic cloud Computing:

(i) Cost Efficiency.

(ii) Convenience and Continuous availability

(iii) Backup Recovery

(iv) Cloud is environmentally friendly.

(v) Scalability and Performance.

(vi) Increased Storage capacity.

Disadvantages / Cons of Elastic cloud Computing:

(i) Security and Privacy in the cloud.

(ii) Limited Control

(iii) Dependency and Vendor Lock in.

(iv) Increased Vulnerability.

On-demand Provisioning:

→ Cloud Computing provides resource on demand when the consumer wants it. This mode is possible by self service and automation.

→ Computer Services such as Email, Application Network, or Server Service can be provided without requiring interaction with each service provider.

Cloud On-demand Self Service Example:

(i) AWS EC2 Walkthrough.

(ii) Amazon Web Services.

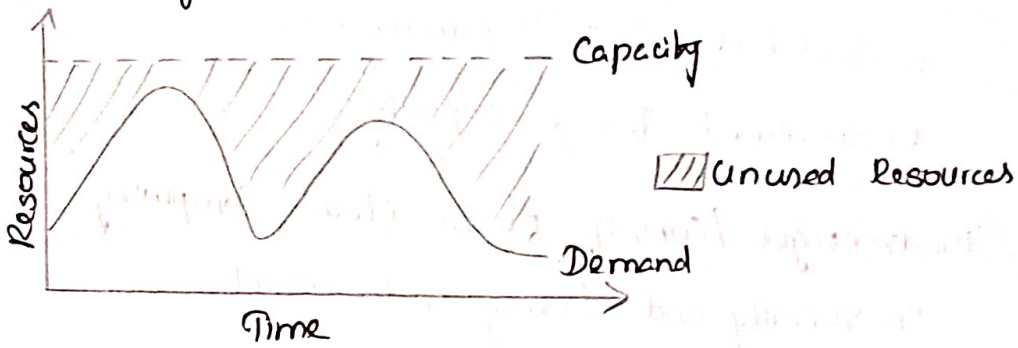
Resource Provisioning:

→ allocation of Cloud provider's resources and Services to a customer.

→ The growing catalog of Cloud Services that Consumers can provisions includes Infrastructure as a Service, Software as a service and Platform as a Service in public or privet Cloud environments.

Why On-demand Provisioning?

→ Over estimate system utilization which result in low utilization.



How to Solve this problem?

→ Dynamically provision resources.

→ Meet demand variations between different industries.

→ Meet seasonal demand variations

→ Meet burst demand for some extraordinary events.

Advantages:

(a) Cost effective

(b) Efficient resource usage.

(c) Scalability and Performance

(d) Convenience and Continuous availability.

UNIT II CLOUD ENABLING TECHNOLOGIES

Service Oriented Architecture - REST and Systems of Systems - Web Services - Publish- Subscribe Model - Basics of Virtualization - Types of Virtualization - Implementation levels of Virtualization - Virtualization Structures - Tools and Mechanisms - Virtualization of CPU - Memory - I/O Devices - Virtualization support and Disaster Recovery.

Service Oriented Architecture:

→ SOA is about how to design a software system that make use of services of new or legacy applications through their published or discoverable interfaces. These applications are distributed over the networks.

Definition of SOA:

Service Oriented Architecture can be defined as services that provide a platform by which disparate systems can communicate with each other. These services are essentially groups of software components that help a company to carry out important business processes. SOA implementation makes Interoperability between heterogenous applications and technologies.

Architecture Styles of SOA:

(i) Loose coupling

(ii) Published Interfaces

(iii) Standard Communication Model.

Properties of SOA:

World Wide Web Consortium (W3C) defines SOA with the following properties.

Logical view:

→ The SOA is an abstracted, logical view of actual programs, databases, business processes.

Message Orientation:

→ The implementation language, process structure, and database structure are abstracted in SOA.

→ SOA services communicate among each other through messages to perform tasks.

Description Orientation:

→ A service is described by machine-executable meta-data. It includes only those details that are important for the use of service.

Granularity Services:

→ SOA use a small number of operations with relatively large and complex messages.

Network Orientation Services:

→ SOA services can be used over a network.

Platform neutral messages:

→ SOA messages are sent in a platform neutral standardized format delivered through the XML interfaces.

SOA styles:

→ Two major styles.

(a) REST (REpresentational State Transfer)

(b) WS (Web Services).

REST and Systems of Systems.

REST:

→ Representational State transfer.

→ REST is a software architecture style for distributed systems.

→ REST is useful for designing distributed hypermedia systems, such as the word wide web. It is being used by the enterprise such as Google, Amazon, Yahoo, Facebook and Twitter.

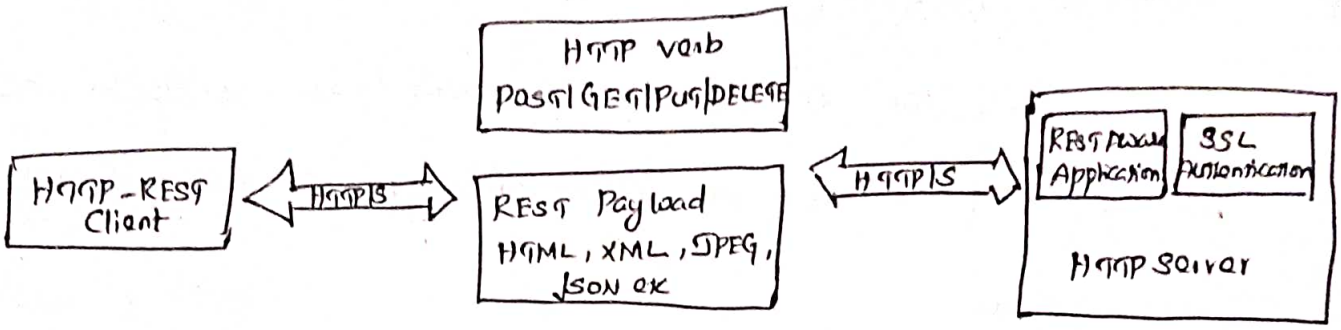
Advantages:

(a) Simplicity

(b) REST Services can be easily published and consumed by clients.

REST Architecture:

→ REST architectural style introduced by Roy Thomas Fielding.



⇒ REST applications use HTTP requests to post data (create and/or update) read data (make queries) and delete data.

Resource Identification through URIs:

(i) Uniform Constrained Interface.

(ii) Self-descriptive message.

Uniform Constrained Interface:

→ Interaction with REST web services is done via the HTTP standard, resources are manipulated using a fixed set of four CRUD (Create, read, update, delete) operations - PUT, GET, POST and Delete.

⇒ Put - Creates a new resource.

⇒ Delete - destroys a resource.

⇒ GET - retrieves the current state of a resource.

⇒ POST - transfers a new state on to a resource.

Self - Descriptive Message:

→ Each client request and server response is a self-descriptive message. That means each message contains all the information necessary to complete the task.

→ In REST resources can be accessed in a variety of standard formats (HTML, XML etc)

→ Metadata about the resource can be used for cache control, transmission error detection authentication or authorization and access control.

→ When a user types http://www.example.com in the address bar of their web server, the browser sends the following HTTP request.

```
GET / HTTP 1.1
Host : www.example.com.
```

This message is self descriptive because it told the server what HTTP method was used and the protocol that was used (HTTP 1.1)

Stateless Interactions:

→ REST Interactions are stateless.

→ The server does not store any state about the client session on the server side. This restriction is called statelessness.

Advantage:

(i) Increases scalability - Any server can handle any request because there is no session related dependency.

(ii) Less complex - by removing all server side - state synchronization logic.

(iii) Improves visibility.

(iv) Disadvantage:

→ Decrease the network performance by increasing the repetitive data.

Advantages of REST:

(i) Light weight Infrastructure

(ii) Inexpensive

(iii) Easy to adopt

(iv) REST support for caching, clustering and load balancing.

Restlet:

→ It is a light weight framework that implements REST architectural elements such as resources, representation, connector and media type for any kind of RESTful system, including web services. Components communicate with each other via connectors.

REST Web Service in Amazon S3 Interface:

→ Amazon S3 is a data storage for Internet Applications

→ It provides simple web services to store and retrieve data from anywhere at any time via the web.

Objects:

→ Objects are the fundamental entities in Amazon S3

→ objects are the data names that contains meta data.

→ Meta data are stored in containers called buckets and, each buckets are identified by a unique key.

Buckets purposes:

→ Organize the Amazon S3 namespace at the highest level.

→ Identify the account responsible for storage and data transfer charges play a role in access control.

→ Serves as the unit of aggregation for usage reporting

→ Amazon S3 provides three type of resources.

(i) A list of user buckets

(ii) a particular bucket

(iii) a particular S3 object.

Standard Operations:

GET - used to list of buckets created by the user.

PUT - used for creating a bucket.

DELETE - for removing a particular bucket or object.

HEAD - getting a specific object's meta data.

Services and Web services:

→ Web Service Interaction.

→ Technologies for web services.

(i) Simple Object Access Protocol (SOAP)

(ii) Web Services Description Language (WSDL)

(iii) Universal Description, Discovery and Integration (UDDI)

→ WS-I Protocol Stack

→ WS - Core SOAP Header Standards.

Definition of Web Service:

→ The term web service is often referred to as a self-contained, self-describing, modular application designed to be used and accessible by other software applications across the web.

→ Once a web service is deployed other applications and other web services can discover and invoke the deployed service.

Web service defined by W3C Working Group:

→ Web service is a software system designed to support interoperable machine to machine interaction over a network.

Technologies for Web Services:

SOAP: (Simple Object Access Protocol):

→ It is a XML based protocol for accessing web services.

→ SOAP is a W3C recommendation for communication between two applications.

→ It is an extension and an evolved version of XML-RPC.

→ XML schema that describes the structure of SOAP message.

→ SOAP based web services are also referred to as "big web services"

WSDL: (Web Services Description language).

→ WSDL is an XML based language for describing web services.

→ WSDL is a W3C recommendation.

→ WSDL enables disparate clients to automatically understand how to interact with a web service.

UDDI: (Universal Description, Discovery and Integration):

→ UDDI provides a global registry for advertising and discovery of web services by searching for names, identifiers, categories & the specification implemented by the web service.

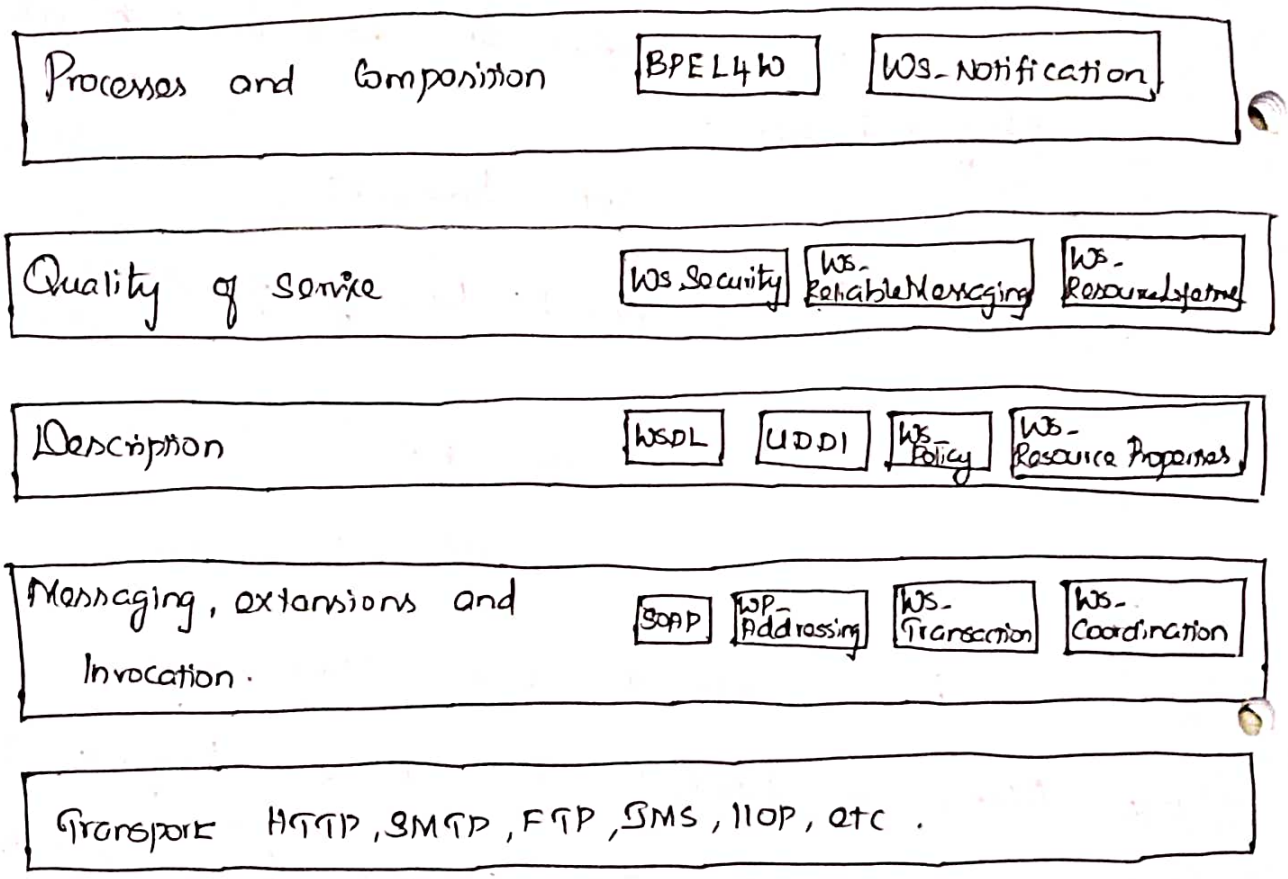
→ It is a directory service where business can register and search for web services.

WS-I Protocol Stack:

→ define non functional requirements.

→ guarantee a certain level of quality in message Communication and reliability.

→ define transactional policies such as WS-Security, WS-Agreement, WS-ReliableMessaging, WS-transaction and WS-Coordination.



⇒ SOAP messages are encoded using XML, all self-described data be sent as ASCII strings. The description contains start and end tags which often constitute half or more of the message's bytes.

BPEL4WS (Business Process Execution Language for Web Services)

→ BPEL4WS is an XML based language - built on top of web service specifications.

→ business process is a large gained stateful service.

→ BPEL enables organizations to automate their business.

Web Services:

→ Open Grid services Architecture (OGSA) is an extensions of web services.

→ Globus Toolkit 4 (GT4) and GT5, pure standard web services.

SOAP Request - Response for creating an S3 Bucket:

→ Amazon S3 as a cloud based persistent storage service is accessible through both a SOAP and REST interface.

→ However REST is the preferred mechanism for communicating with S3.

WS - Core SOAP Header Standards:

→ WS is a prefix used to indicate specifications associated with web services and there exist many WS standards including WS- Addressing, WS- Discovery, WS - Federation, WS - Policy, WS Security and WS - Trust.

Publish Subscribe Model : (Pub/Sub)

→ pub/sub is an asynchronous communication method in which messages are exchanged between applications without knowing the identity of the sender or recipient.

→ In pub/sub model any message published to a topic is immediately received by all of the subscribers to the topic.

→ publish - subscribe model links the source and destination through a message bus.

Message / Event Bus:

→ It knows what topic each subscriber is subscribed to, the event bus will filter messages based on topic and send the messages to subscribers that are subscribed to the topic of the message.

publisher:

→ The producer of the message

→ responsible for defining the topic of their messages.

Subscriber:

→ Receives the message.

→ They will specify the topics for which they wish to receive associated messages.

Message filtering:

→ The process of selecting messages for reception and processing is called filtering. There are two common forms of filtering,

(a) Topic based filtering

(b) Content based filtering.

Topic-based publsub model

→ In topic based system messages are published to topics or named logical channels.

→ The publisher is responsible for defining the topics to which subscribers can subscribe.

Content Based Delivery systems.

→ In a content based system messages are only delivered to a subscriber, if the constraint/content of those messages matches constraints defined by the subscriber.

→ The subscriber is responsible for classifying the messages

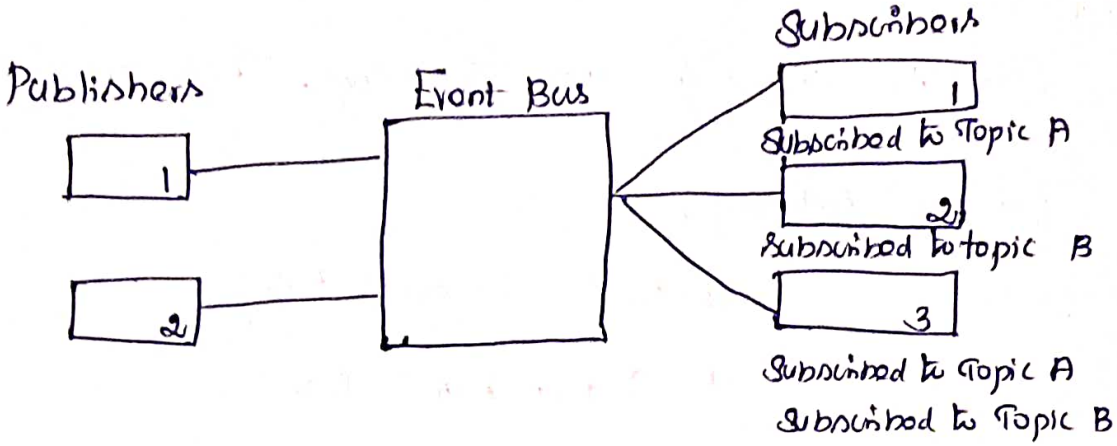
Relationship

→ There exists many to many relationship between publishers and subscribers.

Publish-subscribe messaging middleware.

→ It allows straight forward implementation of notification or event-based programming models.

→ The messages could be labeled by desired notify topic and contain content elaborating the notification



⇒ The publishers are responsible for defining the topics of their messages.

⇒ In above diagram, any message published with Topic A will be sent to subscriber 1 and Subscriber 3. Similarly any message published with topic B will be sent to Subscriber 2 and Subscriber 3. If a publisher were to send a message about Topic A, but wrongfully define it as topic B. It would be sent to the subscribers of Topic B only.

Example:

→ Google Cloud pub/sub (New)

→ Google cloud pub/sub offers both Competing consumers and Publish subscribe channel semantics, managed through topics (Publish-Subscribe) and subscriptions (Competing Consumers).

Basics of Virtualization:

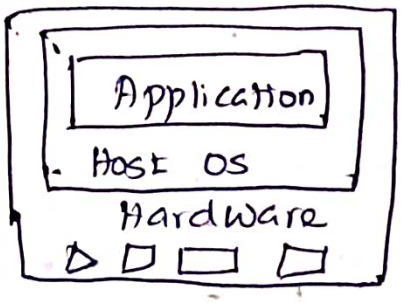
→ Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in same hardware machine. The idea of VMs can be dated back to the 1960s.

→ Example: Running Ubuntu on Windows OS

Host OS → Windows OS

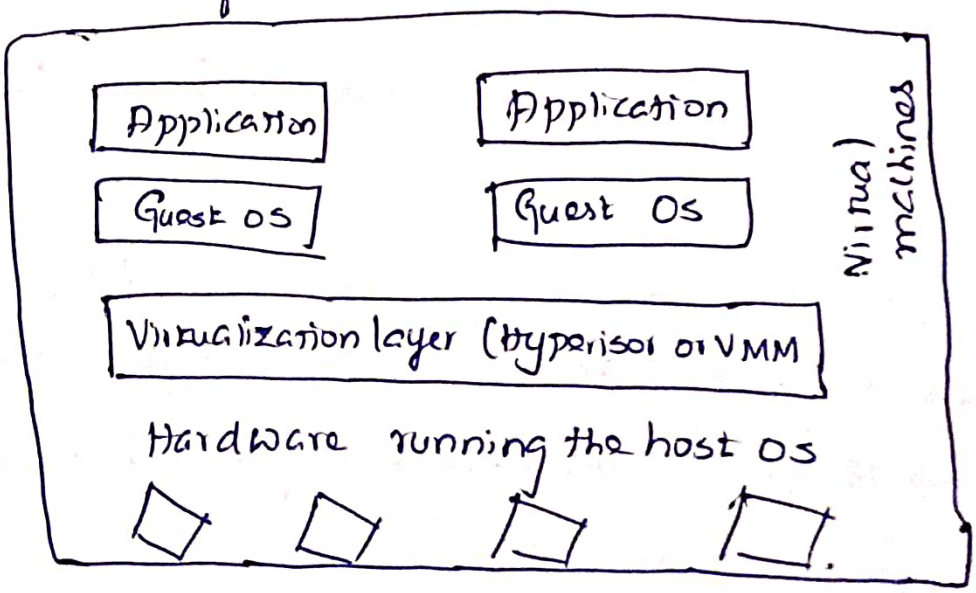
Guest OS / Virtual Machine (VM) → Ubuntu.

Before Virtualization:



After Virtualization:

→ different applications of guest operating system can run on the same hardware, independent of host OS.

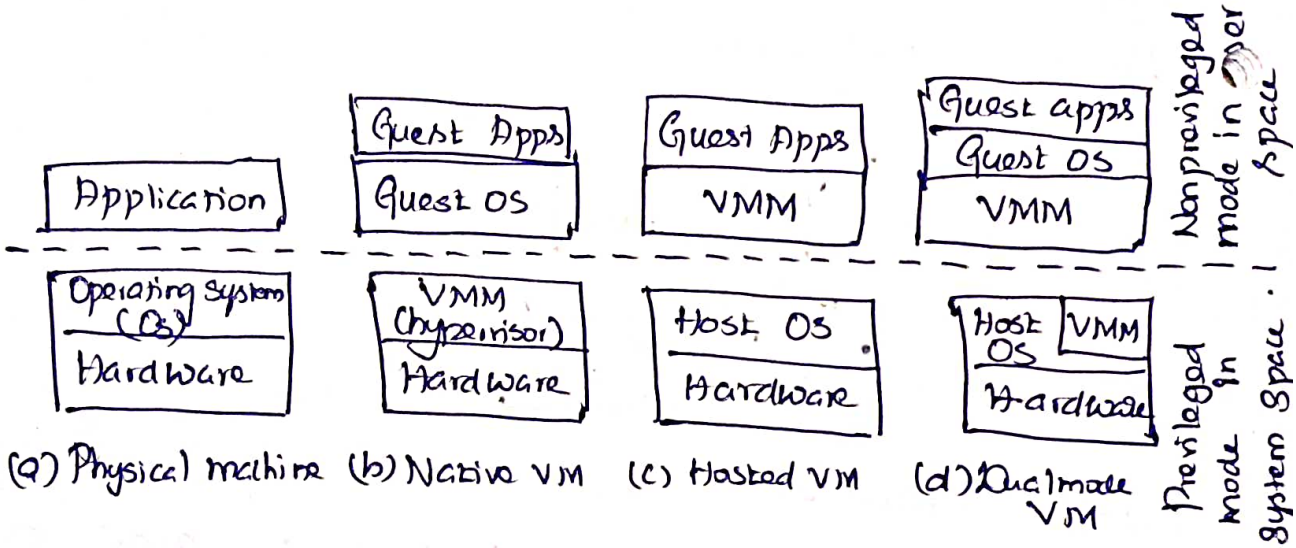


→ Virtualization is done by software / middleware called a virtualization layer.

→ The virtualization layer is known as hypervisor or virtual machine monitor (VMM)

Virtualization Models / Architecture:

→ Virtualization is the process of creating virtual machine over existing operating system and hardware.



Host Machine:

→ The machine on which the virtual machine is created.

Guest machine:

→ Virtual machine is referred to as Guest machine

Hypervisor:

→ is a firmware or low level program that acts as a virtual machine Manager / Monitor.

→ It is a middleware layer between the host machine and virtual machine.

Native VM:

→ A VMM (hypervisor) runs in privileged mode.

→ The hypervisor approach is also called bare-metal VM, because the hypervisor handles the bare hardware (cpu, memory, and I/O) directly.

Host VM:

→ VMM runs in non privileged mode.

→ The host OS needs not to be modified.

Dual Mode VM:

→ Part of the VMM runs at the user level and another part runs at the supervisor level.

→ The host OS may have to be modified.

Purpose:

(1) To enhance resource sharing by many users.

(2) To improve computer performance in terms of
  ↳ resource utilization
  ↳ application flexibility.

Types of Virtualization:

(1) Application Virtualization.

(2) Desktop Virtualization

(3) Hardware Virtualization

(4) Network Virtualization.

(5) Storage Virtualization.

Application Virtualization:

→ also called app virtualization.

→ It allows user to access and use an application from a separate computer than the one on which the application is installed.

→ The application are virtualized and delivered from a server to the end users device, such as laptops, smart phones and tablets.

Desktop Virtualization:

→ The virtualization of the desktop is referred to as virtual Desktop Infrastructure (VDI).

→ Desktop operating system (os) such as Windows 7 will run as a virtual machine on another computer.

Advantages:

(i) Can work from anywhere without the need to bring their work computer.

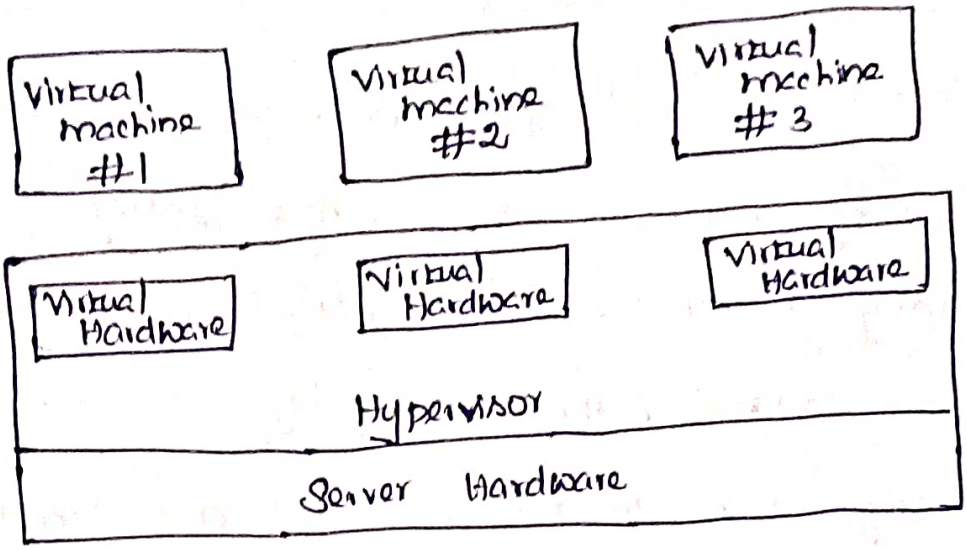
(ii) lowers the cost of software licensing and updates.

(iii) Maintenance and patch management are simple.

Hardware / Server Virtualization:

→ A hypervisor is loaded directly on the hardware system. Hypervisor acts as an intermediary between the server hardware and virtual machines.

→ Hardware virtualization when done for server platforms, is also called server virtualization.



Types of Hardware Virtualization:

i) Full Virtualization:

→ the hardware architecture is completely simulated.

→ Guest OS doesn't need any modification to run any applications.

→ Ex. VM-vare.

ii) Para Virtualization:

→ Guest OS need modification to run any applications.

→ Ex. Xen.

Network Virtualization:

→ is the process of combining hardware network resources and software network resources into a single administrative unit

→ The end product of network Virtualization is the virtual network.

→ Virtual networks are classified into two parts

(i) External

(ii) Internal

External Virtual Networks:

→ Consist of several local networks that are administered by software as a single entity.

→ Building blocks: switch hardware and virtual local area network (VLAN) software technology

→ eg large Corporate networks and data centers.

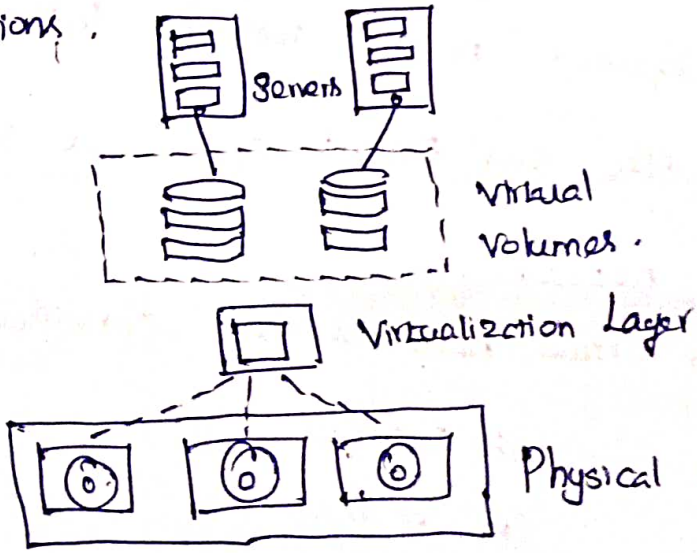
Internal Virtual Network:

→ Consists of host running multiple virtual machines with at least one physical NIC. Those network interfaces cards or virtual NICs. These VMs can communicate with each other through a virtual network on a single host.

Storage Virtualization:

→ the process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device.

→ Storage virtualization is also implemented by using software applications.

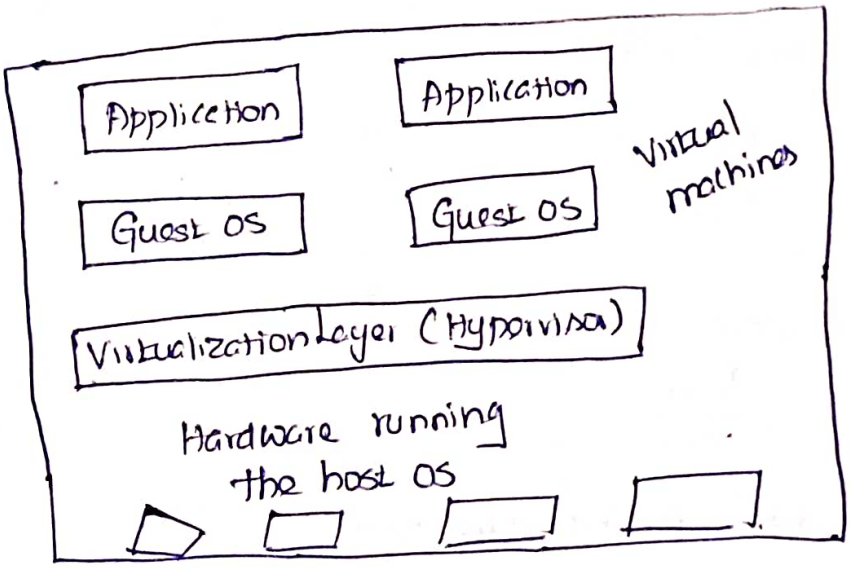


Implementation Levels of Virtualization:

→ Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.

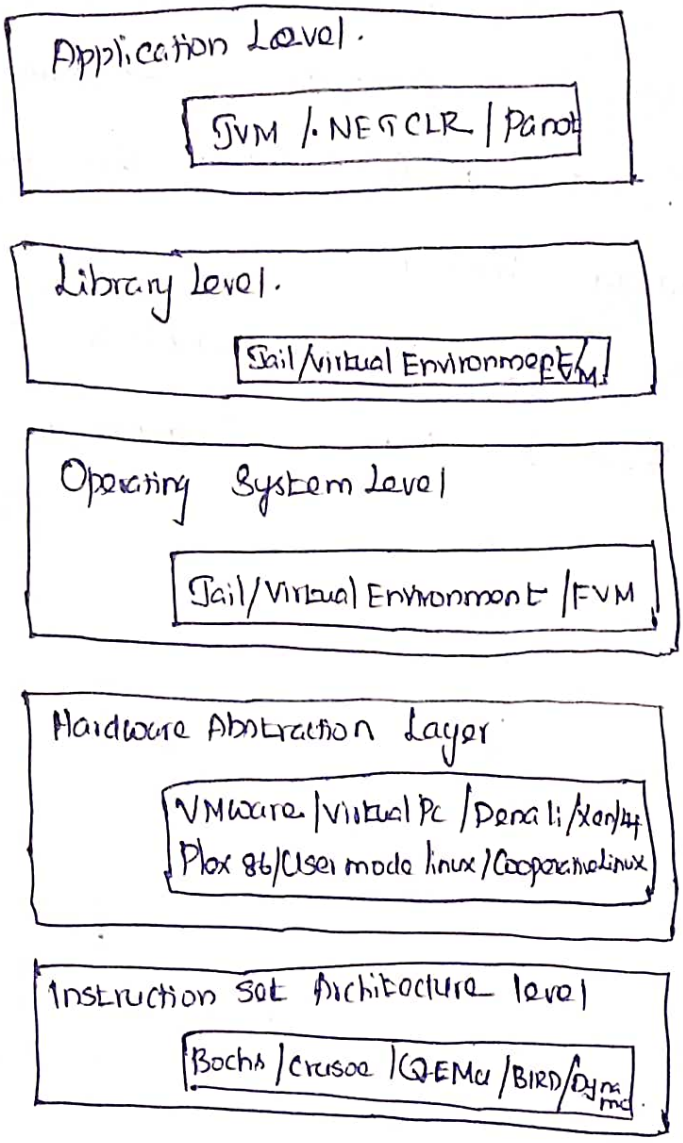
→ The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.

Levels of Virtualization Implementation:



→ The main function of software layer for virtualization is to virtualize the physical hardware of the host machine into virtual resources to be used by VMs.

→ Common virtualization layers include the instruction set architecture (ISA) level, hardware level, Operating System level, library support level, and application level.



Instruction Set Architecture Level:

→ At the ISA level, virtualization is performed by emulating a given ISA by the ISA of host machine.

→ One source instruction may require tens or hundreds of native target instructions to perform its function

→ Instruction set emulation requires binary translation and optimization.

→ A virtual set architecture (VISA) thus requires adding a processor specific software translation layer to the compiler.

Hardware Abstraction Level:

→ Hardware level virtualization is performed on right top of the bare hardware

→ The idea is to virtualize a computers resources, such as its processor, memory and I/o devices.

Operating system Level:

→ This refers to an abstraction layer between traditional OS and user applications.

→ OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers.

→ The containers behave like real servers.

Library Support level:

→ Virtualization with library interfaces is possible by controlling the communication link between application and the rest of a system through API hooks.

→ The software tool WINE has implemented this approach to support windows applications on top of UNIX hosts.

User Application Level:

→ Virtualization at the application level virtualizes an applications as a VM.

→ The Microsoft .NET CLR and Java Virtual Machine (JVM) are too good examples of this class of VM.

Relative Merits of Different Approaches:

→ The column headings correspond to four technical merits. "Higher Performance" and "Application Flexibility" are self explanatory.

→ Implementation Complexity implies the cost to implemen that particular virtualization level.

→ "Application Isolation" refers to the effort required to isolate resources committed to different VMs.

VMM Design Requirements and Providers:

⇒ Three requirements for a VMM

(i) a VMM should provide an environment for programs which is essentially identical to the original machine.

(ii) programs run in this environment should show, at worst only minor decreases in speed.

(iii) a VMM should be in complete control of system resource

⇒ Two Possible exceptions,

(i) differences caused by the availability of system resources

(ii) differences caused by timing dependencies.

Virtualization support at the OS level:

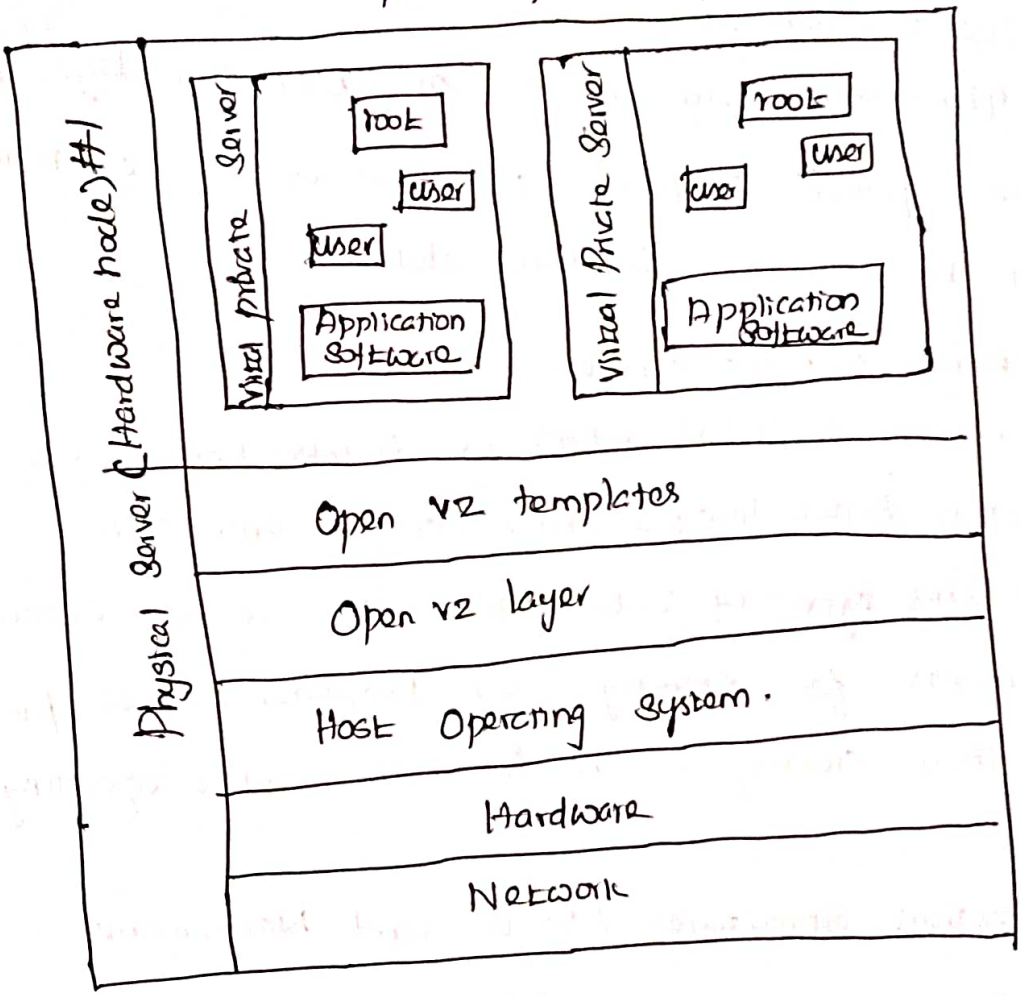
Cloud computing is transforming the computing landscape by shifting the hardware and staffing costs of manging a computational center to third parties, just like banks.

Why OS level Virtualization:

→ In cloud computing environment perhaps thousands of VMs need to be initialized simultaneously.

→ OS-level virtualization provides a feasible solution for these hardware level virtualization issues.

→ From the users point of view, VEs look like real servers.



Advantages of OS Extensions:

(i) VMs at the operating system server has minimal startup or shutdown costs, low resource requirements and high scalability

(ii) for an OS level VM, it is possible for a VM and its host environment to synchronize state changes when necessary.

Disadvantage of OS Extensions:

(1) all the VMs at operating system level on a single container must have same kind of guest operating system Virtualization on Linux or Windows Platforms.

→ Virtualization support on the Windows based platform is still in the research stage.

→ The linux kernel offers an abstraction 'layer' to allow software process to work with and operate on resources without knowing the hardware details.

Middleware Support for Virtualization:

→ Library level virtualization is also known as a user-level Application Binary Interface (ABI) or API emulation.

→ This type of virtualization can create execution environments for running alien programs on a platform rather than creating a VM to run the entire operating system.

Virtualization Structures / Tools and Mechanisms:

→ Before virtualization the operating system manages the hardware

→ After virtualization, a virtualization layer is inserted between the hardware and the operating system.

→ The virtualization layer is responsible for converting portions of the real hardware into virtual hardware.

Hypervisor and XEN Architecture:

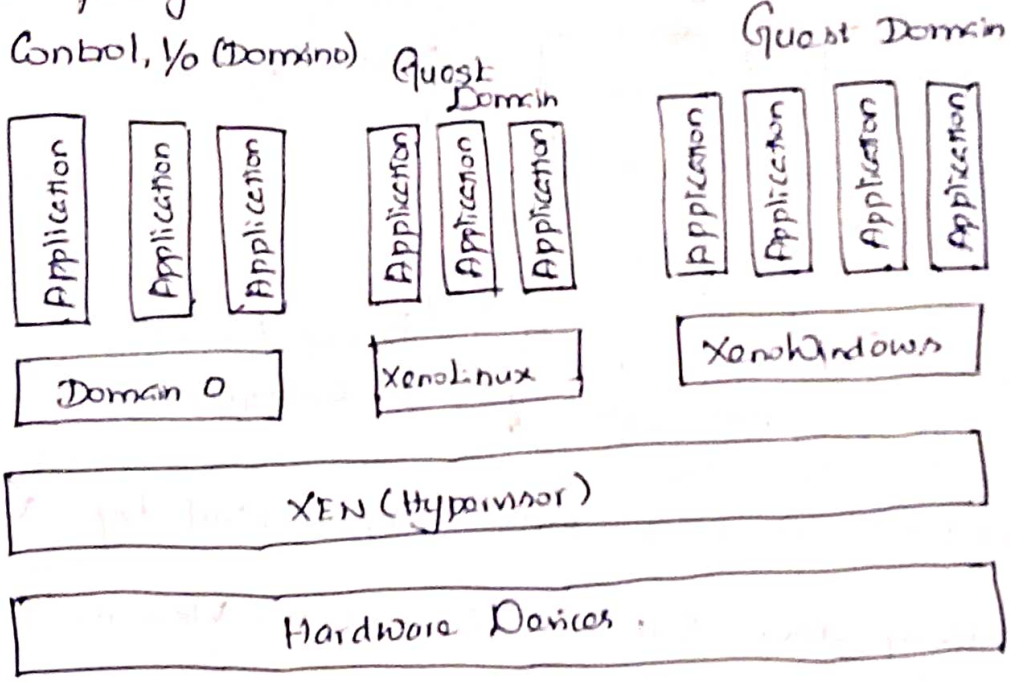
→ The hypervisor supports hardware level virtualization on bare metal devices like CPU, memory, disk & network interfaces.

→ The hypervisor software sits directly between the physical hardware and its OS. This layer is referred to as VMM or hypervisor

→ The hypervisor provides hypercalls for guest OS and applications.

The XEN Architecture:

→ The XEN hypervisor implements all the mechanisms leaving the policy to be handled by Domain 0.



→ Domain 0 is designed to access hardware directly and manage devices. Therefore one of the responsibilities of Domain 0 is to allocate and map hardware resources for the guest domains.

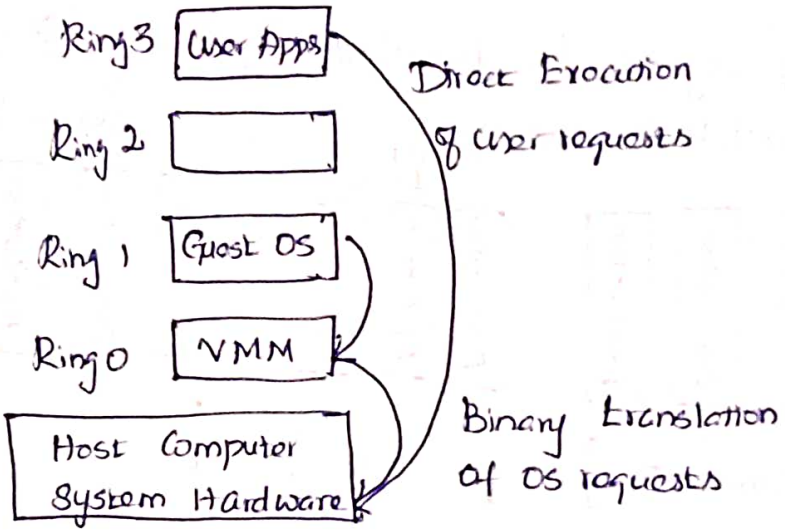
Binary Translation with Full Virtualization:

→ Binary translation to trap and to virtualize the execution of certain sensitive, non virtualizable instructions.

→ Both the hypervisor and VMM approaches are considered full virtualization.

→ Non critical instructions do not control hardware or threaten security of the system but the critical instruction do.

Binary Translation of Guest OS Requests using a VMM



→ This approach was implemented by VMware and many other software companies. VMware puts the VMM at Ring 0 and the guest OS at ring 1

→ Binary translation employs a code cache to store translated hot instructions to improve performance, but it increases the cost of memory usage.

Para Virtualization with Compiler Support:

→ Para virtualization needs to modify the guest operating systems. A para virtualized VM provides special APIs requiring substantial OS modifications in user applications

→ Performance degradation is a critical issue of a virtualized system.

→ The guest operating systems are para virtualized.

→ They are assisted by an intelligent compilor to replace the non virtualizable OS instructions by hypercalls.

Para Virtualization Architecture:

→ When the x86 processor is virtualized, a virtualization layer is inserted between the hardware and the OS.

→ According to the x86 ring definition, the virtualization layer should also be installed at Ring 0.

→ First its compatibility and portability may be in doubt, because it must support the unmodified OS as well.

→ Second the cost of maintaining para virtualized OSes is high because they may require deep OS kernel modifications.

KVM (Kernel-Based VM):

→ This is a Linux para virtualization system - a part of the Linux version 2.6.20 kernel.

→ Memory management and scheduling activities are carried out by the existing Linux kernel.

→ The KVM does not rest, which makes it simpler than the hypervisor that controls the entire machine.

→ KVM is a hardware assisted para virtualization tool which improves performance and supports modified guest OSes.

Virtualization of CPU, Memory and I/O Devices:

→ To support virtualization, processors such as the x86, employ a special running mode and instructions known as hardware assisted virtualization.

Hardware Support for Virtualization:

→ Modern operating systems and processors permit multiple processes to run simultaneously.

→ The VMWare workstation assumes the host based virtualization.

→ Xen is a hypervisor for use in IA-32, x86-64 Itanium and Power PC 970 hosts.

CPU Virtualization:

→ The critical instructions are divided into three categories,

(i) Privileged Instructions

(ii) Control Sensitive Instructions

(iii) Behaviour Sensitive Instructions.

⇒ Privileged Instructions execute in a privileged mode and will be trapped if executed outside this mode.

⇒ Control Sensitive Instructions attempt to change the configuration of resources used.

⇒ Behaviour Sensitive Instructions have different behaviours depending on the configuration of resources, including load and store operations over the virtual memory.

Hardware - Assisted CPU Virtualization:

→ This technique attempts to simplify virtualization because full or para virtualization is complicated.

→ All the privileged and sensitive instructions are trapped in the hypervisor automatically.

→ This technique removes the difficulty of implementing binary translation of full virtualization.

Memory Virtualization:

→ Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems.

→ In a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.

I/O Virtualization:

→ I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.

→ Three ways to implement I/O virtualization

(i) Full Device emulation

(ii) Para virtualization

(iii) Direct I/O.

Virtualization Support and Disaster Recovery:

→ Cloud Computing Infrastructure make use of system virtualization. The VMs are the containers of cloud services.

→ In cloud computing virtualization virtualizes the resources and fundamental infrastructure.

→ The user will not care about the computing resources that are used for providing the services.

→ Application developers do not care about scalability and fault and they focus on service logic.

Hardware Virtualization:

→ In many cloud computing systems virtualization software is used to virtualize the hardware.

System Virtualization Software:

→ It is a software that stimulus the execution of hardware and runs even unmodified operating systems.

→ Cloud computing systems use virtualization software as the running environment for legacy software such as old operating systems and unusual applications.

→ Virtualization software is also used as the platform for developing new cloud applications that enable developers to use any operating systems and programming environments they like.

Virtualization Support in Public Clouds:

→ Three public clouds in the context of virtualization support.

(a) AWS
(b) Microsoft Azure.
(c) GAE.

AWS ⇒ Provides extreme flexibility (VMs) for users to execute their own applications.

Microsoft ⇒ provides programming level virtualization for users to build their applications.

GAE ⇒ provides limited application-level virtualization for users to build applications only based on services.

Storage Virtualization for Green Data Centers:

→ Virtualization concept reduced the power consumption in physical computing systems.

→ Virtualization and server consolidation conserve power in all green data centers and benefits of storage virtualization strengthen the synergy of green computing.

Virtualization for IaaS:

(a) VM technology has increased in ubiquity.

(b) This enabled users to create customized environments for cloud computing.

Benefits of Using VMs:

(i) System administrators Consolidate workloads

(ii) to run legacy code

(iii) to improve security

(iv) can apply performance isolation.

VM Cloning for Disaster Recovery:

→ 2 types

1) Recovery Using Physical Machine:

→ to recover one physical machine by another

(i) hardware configuration

(ii) Installing and configuring the OS

(iii) Installing the backup agents

(iv) Long time to restart.

2) Recovery Using Virtual Machine:

→ to recover one VM by another VM

→ to recover a VM platform, the installation and configuration times for the OS and backup agents are eliminated.

Advantages:

(i) Shorter disaster recovery time

(ii) Simple and In-expensive.

UNIT III CLOUD ARCHITECTURE, SERVICES AND STORAGE

Layered Cloud Architecture Design - NIST cloud computing Reference Architecture - public, private and Hybrid clouds - Iaas - Paas - Saas - Architectural Design Challenges - Cloud Storage - Storage-as-a-Service - Advantages of Cloud Storage - Cloud Storage Providers - S3.

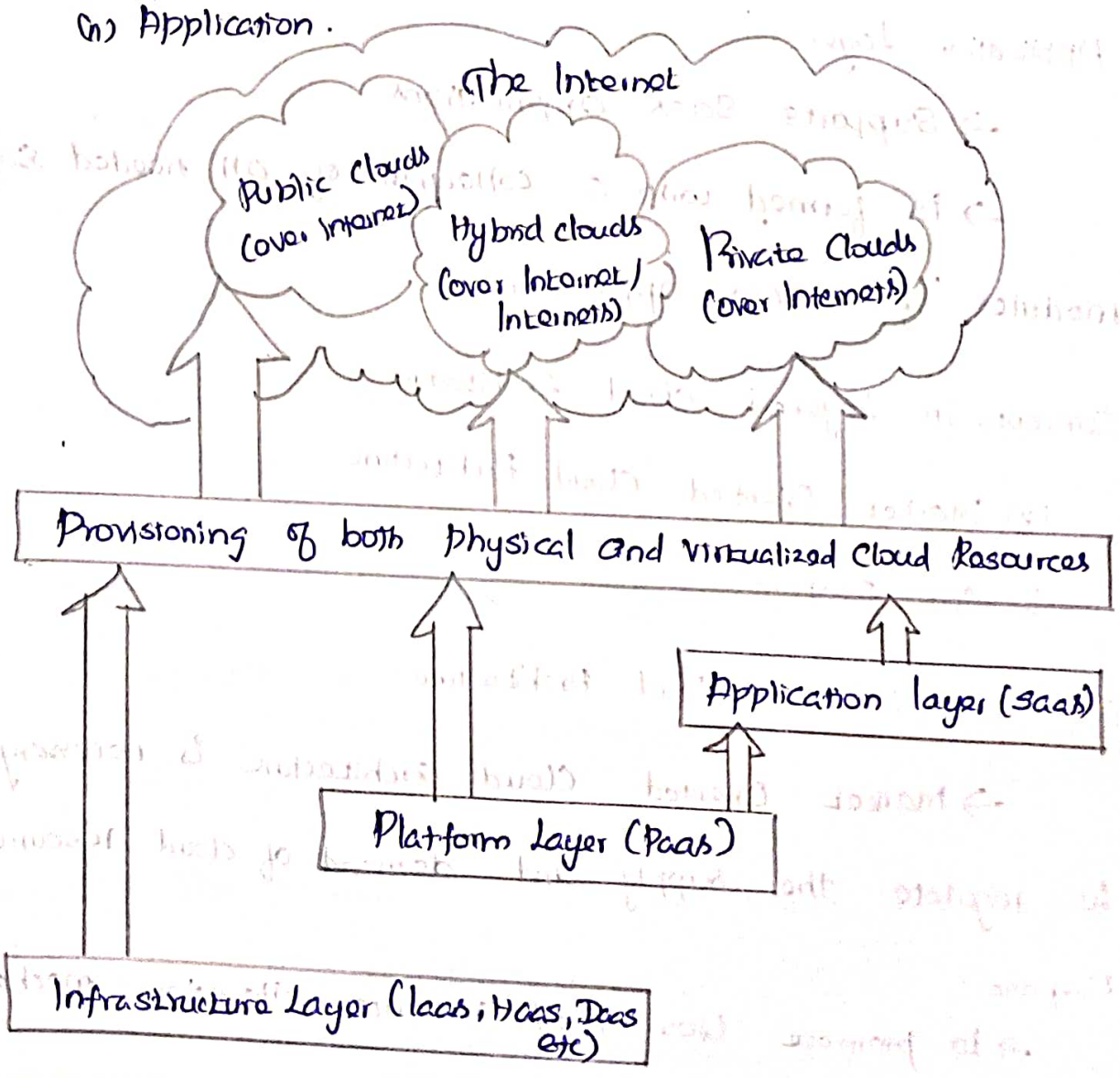
Layered Cloud Architecture Design:

→ The Cloud architecture is developed with three layers.

(1) Infra Structure

(2) Platform

(3) Application.



Infrastructure Layer:

→ Serves as the foundation for building the platform layer.

→ Is built with virtualized compute, storage and network resources.

→ Support Iaas services.

Platform Layer:

→ Supports Paas Services.

→ Is a foundation for implementing the application layer.

→ It serves as a system middleware between the infrastructure and application layers of the cloud.

Application Layer:

→ Supports Saas applications.

→ Is formed with a collection of all needed software modules for saas applications.

Concepts in Layered cloud Architecture:

(1) Market Oriented Cloud Architecture.

(2) Qos Factors.

1) Market-Oriented Cloud Architecture:

→ Market Oriented Cloud architecture is necessary to regulate the supply and demand of cloud resources.

Purpose:

→ to promote Qos based resource allocation mechanisms.

→ Clients can benefit from the potential cost reduction of providers.

→ lead to a more competitive market.

→ lower prices.

Entities of Market-Oriented cloud Architecture:

(1) Users or brokers

(2) Request examiner

(3) Pricing Mechanism

(4) VM Monitor

(5) Accounting

(6) Dispatcher

(7) Service Request Monitor

(8) Virtual machine.

Qos Factors:

→ Different Qos factors are

(1) Time

(2) Cost

(3) Reliability

(4) Security.

→ The Qos requirements can't be static and might from time to time on demand.

NIST Cloud Computing Reference Architecture:

→ NIST stands for National Institute of Standards and Technology

Workgroups

→ Six major workgroups.

(i) Cloud computing target business use cases work group.

(ii) Cloud computing Reference architecture and Taxonomy Workgroup.

(iii) Cloud computing Standards roadmap work group.

(iv) Cloud computing SAJACC.

(v) Cloud computing security Work group.

Objectives of NIST:

(i) Illustrate and understand the various level of services.

(ii) To provide technical reference.

(iii) Categorize and compare services of Cloud computing.

(iv) Analysis of security, interoperability and portability.

Conceptual Reference Model:

⇒ Cloud Consumer: A person or an organization uses a services from Cloud providers.

⇒ Cloud Provider: A person, organization or entity responsible for making a service

⇒ Cloud Auditor: A party that conduct independent assesment of cloud services, information system operation, performence and security of cloud implementation.

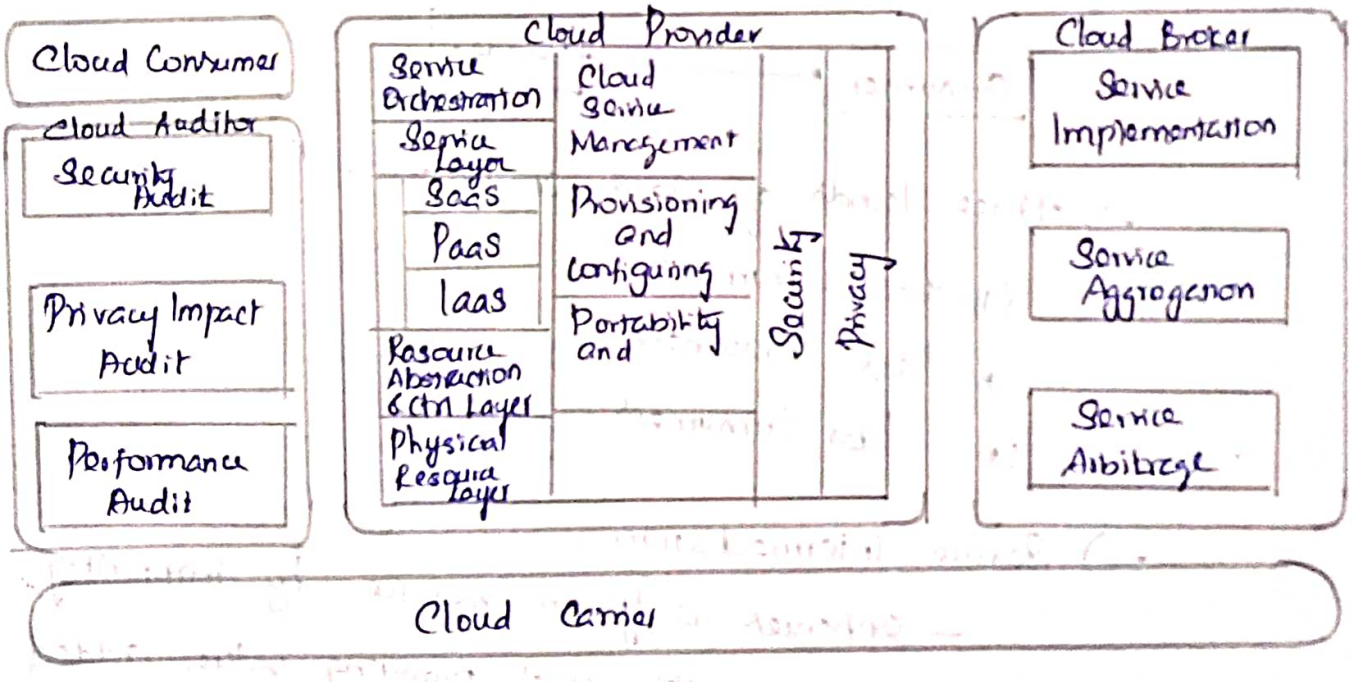
⇒ Cloud Broker: - An entity that manages the performance and delivery of Cloud Services.

⇒ Cloud Carrier: - An intermediary that provides connectivity and transport of Cloud services from cloud providers to Consumers.

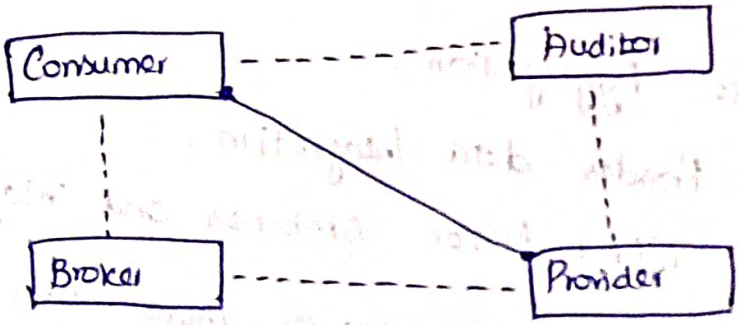
Illustration of the common interaction exist in between Cloud Consumer and provider.

⇒ Broker - used to provide service to consumer.

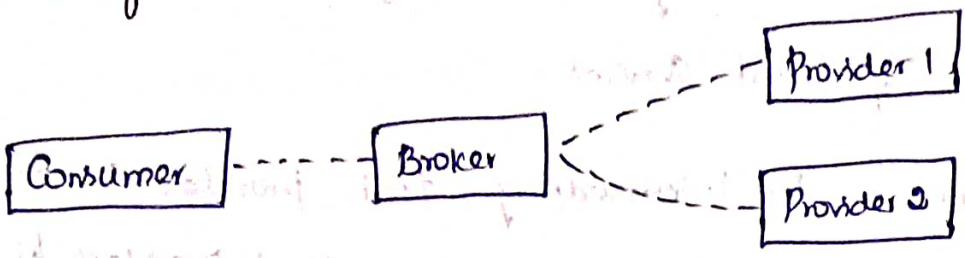
⇒ Auditor - Collects the audit the information.



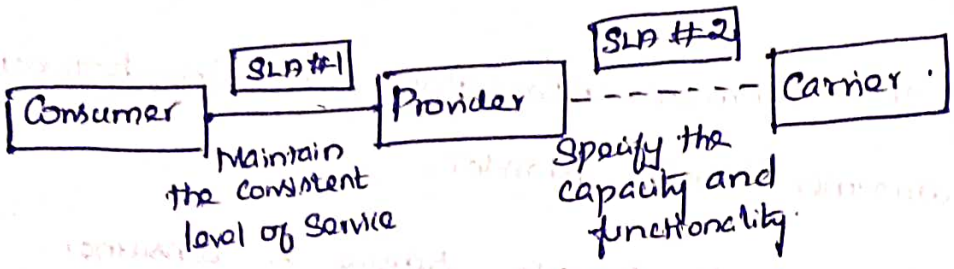
Interaction between Actors.



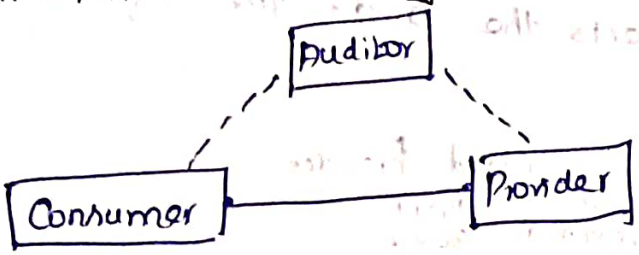
Service from Cloud Broker:



Multiple SLA Between Actors:



Independant Assessments by cloud Auditor:



⇒ three kinds of cloud consumers.

(i) Saas Consumers.

(ii) Paas Consumers

(iii) Iaas Consumers.

⇒ Service Intermediation.

- enhances a given service by improving some specific capacity and providing value added services to cloud consumers.

⇒ Service Aggregation.

- Provides data Integration.
- Cloud broker combines and integrate multiple service in to one or more new services

Public, Private and Hybrid clouds:

→ The cloud enables anyone with an Internet Connection to access IT resources on demand.

⇒ Three resources may be,

(1) Compute resources

(2) Storage resources

(3) Networking Resources

⇒ four cloud Deployment models.

(1) Public cloud

(2) Private cloud

(3) Hybrid cloud

(4) Community cloud.

Public cloud:

→ Is built over the Internet

→ can be accessed by any user who has paid for service

⇒ Example of public clouds

(1) Google App Engine (GAE)

(2) Amazon Web Services (AWS)

(3) Microsoft Azure

(4) IBM Blue cloud.

(5) Salesforce.com force.com.

Private cloud:

→ Is owned and managed by a client.

→ Its access is limited to owning clients and their partners.

(i) On-site private clouds

(ii) Outsourced private clouds.

On site Private Clouds:

→ Data center is physically located in your premises and your software / services will run on it.

Out sourced private clouds:

→ owned by a third party and you get a part of their stack isolated to your use and need.

Hybrid Clouds:

→ is built with both public and private clouds.

→ Make use of the local infrastructure of private cloud and computing capacity of public cloud.

Community Cloud:

→ is accessible by a group of several organizations to share the information

→ It is owned, managed, operated by one or more organizations in the community a third party or a combination of them.

Cloud Service Models:

→ The services provided over the cloud can be generally categorized into three different service models.

(i) Infrastructure as a Service (IaaS)

(ii) Platform as a Service (PaaS)

(iii) Software as a service (SaaS)

Infrastructure as a Service (IaaS)

- It provides a rented cloud architecture.
- This model allows users to use virtualized IT resources for computing, storage and networking.

→ Examples are,
Amazon Ec2, GoGrid, Flexiscale, Amera.

→ This IaaS model encompasses,

(i) Storage as a service

(ii) Compute instances as a service

(iii) Communication as a service.

Examples of IaaS:

(i) Amazon Ec2 clusters and S3 storage to multiple users

(ii) VPC allows the user to isolate provisioned AWS processors, memory and storage from interence by other users

(iii) Autoscaling enables users to automatically scale their VM instance capacity up or down.

Platform-as-a Service (PaaS):

→ Paas includes operating system and runtime library suppo

→ Paas is an integrated computer system.

→ The user does not manage the underlying cloud Infrastructure.

Example:

⇒ Google App Engine for Paas Applications.

→ To develop applications using GAE, a local development environment must be provided.

→ All the functions and application logic can be implemented locally.

(1) Logging on to the GAE System

(2) Signup for an account or use you gmail account name

(3) Download GAE SDK

(4) Python Getting Started.

(5) Java Getting Started Guide

(6) Quata page for free service

(7) Billing Page.

Software as a Service (Saas)

→ The saas model provides software applications as a service.

→ This refers to browser initiated application software used by thousands of cloud customers.

→ These applications can be developed using the services and tools offered by PaaS

Example:

(a) To discover new drugs through DNA Sequence Analysis.

(b) New York Times has applied Amazon's EC2 and S3 Services

(c) Ping Pong, an ecommerce company to perform B2B transactions

---

Architectural Design challenges:

→ Six Open challenges.

Challenge 1: Service Availability and Data Lock-in Problem.

(a) Data lock in problem in Cloud computing

(b) Single to multiple cloud service Providers.

(c) Distributed denial of service (DDOS) attacks.

(d) Standardized APIs.

Challenge 2: Data Privacy and Security Concerns.

(a) Current cloud offerings are essentially public rather than private networks.

(b) Technologies for Security

→ encrypted Storage

→ virtual LANs

→ network middle boxes. (eg firewalls.

(h) Network attacks in cloud includes:

→ Traditional network attacks

↳ buffer overflows, spyware, malware, rootkits, Trojan horse, and worms.

→ Passive attacks

↳ steal sensitive data or passwords.

→ Active attacks.

↳ manipulate kernel data structure which damage the cloud servers.

Challenge 3: Unpredictable Performance and Bottlenecks.

(i) Multiple VMs can share CPUs and main memory

(ii) Data transfer bottlenecks must be removed.

(iii) Weak servers should be removed.

Challenge 4: Distributed storage and Widespread Software Bugs

(i) Data centers – provide scalability, data durability

(ii) Data consistency checking

(iii) Large scale distributed bugs cannot be reproduced.

(iv) Debugging can be done in 2 ways.

→ Using VMs in Cloud computing

→ Using simulators.

Challenge 5: Cloud Scalability, Interoperability, and Standardization.

(i) Computation is depending on virtualization level.

(ii) Open virtualization Format – open secure, portable, efficient and extensible format.

(iv) Cloud Standardization - virtual appliances to run on any virtual platform.

Challenge 6: Software Licensing, and Reputation Sharing.

(v) relied on open source software.

(vi) pay for use and bulk use licensing schemes to widen the business coverage.

Cloud Storage:

→ Storing the data with a cloud service provider.

→ The end user can access the data stored on the cloud using an Internet link.

→ A subscriber copies file to the server over the internet

→ Cloud Storage systems utilize dozens or hundreds of data servers.

Services of Cloud Storage:

(i) Storage-as-a Service

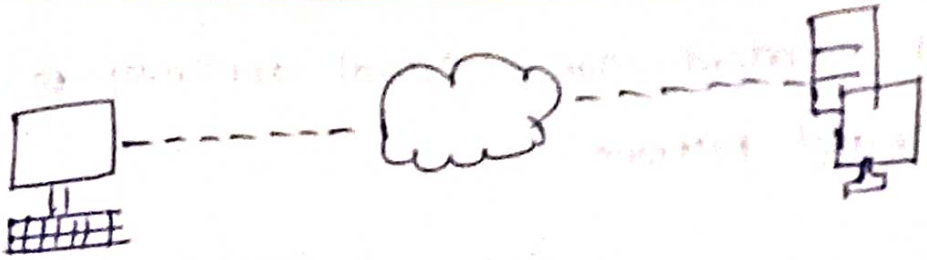
(ii) Advantages of Cloud Storage

(iii) Cloud Storage Providers

(a) S_3.

Storage-as-a Service: (SaaS)

→ A third party provider rents space on their storage to end users.



→ The end user does not have to pay for infrastructure.

→ Storage services - backup, replication and disaster recovery.

Examples of SaaS :

(1) Google Docs

(2) Web email providers like Gmail, Hotmail and Yahoo.

(3) Flickr and Picasa.

(4) YouTube

(5) Hostmonster and GoDaddy.

(6) Facebook and Myspace

(7) MediaMax and Strongspace.

Techniques to Secure Data:

(1) Encryption

(2) Authentication Processes

(3) Authorization Practices.

(4) Reliability.

Advantages of Cloud Storage:

(1) Attractive solution for organizations.

(2) Load balancing

(3) Data movement.

(4) Disaster Recovery and Data Protection.

(5) Amazon S3 is the best known storage solution.

Cloud storage Providers:

(1) Amazon and Nirvanix are the current industry top dogs but many others are in the field, including some well known names.

(2) Google offers cloud storage solution called GDrive.

(3) IBM offers number of cloud storage called Blue Cloud.

Amazon S3:

→ Well known cloud storage service is Amazon's Simple Storage Service (S3) which is launched in 2006.

→ make web scale computing easier for developers.

Amazon S3 Functionality:

(1) Write read and delete objects.

(2) Unlimited number of objects can be stored

(3) Each object is stored and retrieved via Unique key.

(4) Objects can be made private or public

(5) Uses REST and SOAP interfaces.

Design Requirements of Amazon S3:

(1) Scalable

(2) Reliable

(3) Fast

(4) Inexpensive

(5) Simple.

Design Principles of Amazon S3:

(1) Decentralization

(2) Autonomy

(3) Local Responsibility

(4) Controlled Concurrency.

(5) Failure Toleration

(6) Controlled Parallelism.

(7) Symmetry

(8) Simplicity

(9) S3 object and bucket => 5GB in size upto 2kb of metadata

(10) REST or SOAP Interface

(11) Object Removal

(12) Accessing the Buckets

(13) Authorization of request

(14) Amazon AWS Authentication Tools

(15) HTTP Log Information.

Unit IV RESOURCE MANAGEMENT AND SECURITY IN CLOUD

Inter Cloud Resource Management - Resource Provisioning and Resource Provisioning methods - Global exchange of cloud Resources - Security Overview - Cloud Security challenges - Software as a Service Security - Security Governance - Virtual Machine Security - IAM - Security Standards.

Inter Cloud Resource Management:

(i) Resource Provisioning

(ii) Resource Provisioning Methods

(iii) Global Exchange of Cloud Resources.

Resource Provisioning:

Provisioning of Compute Resources (VMs)

→ action of providing or supplying cloud resources.

→ providers supply cloud services by signing SLAs with end users

→ underprovisioning of resources will lead to broken SLAs and penalties.

Difficulties:

(i) Unpredictability of consumer demand

(ii) Software and hardware failures

(iii) Heterogeneity of Services

(iv) Power Management

(v) Conflicts in Signed SLAs

⇒ Virtualized Cluster of Servers:

(a) Efficient Installation of VMs

(b) live VM Migration

(c) Fast recovery from failures.

Example of VMM:

(a) Amazon EC2 uses Xen

(b) Xen VMM is used in IBM's Blue Cloud

(c) EC2 platform

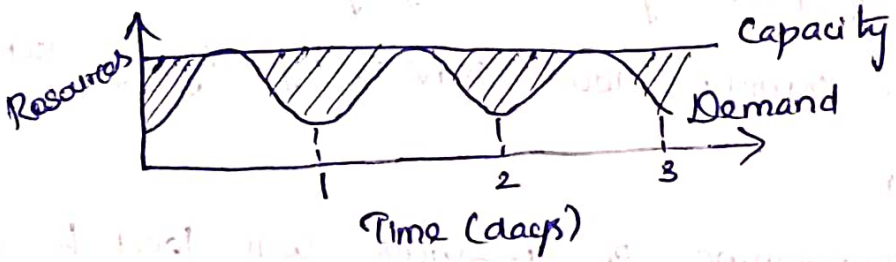
(d) Predefined VM Templates.

Resource Provisioning Methods:

→ Three cases of static cloud resource provisioning policies

a) Over provisioning with the peak load.

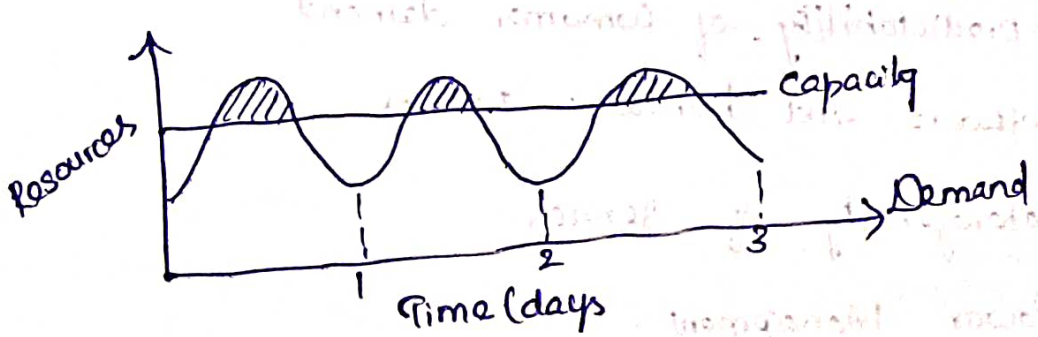
→ causes heavy resource waste (shaded area)



b) Underprovisioning (along the capacity line) of resources.

→ Loss for both user and provider

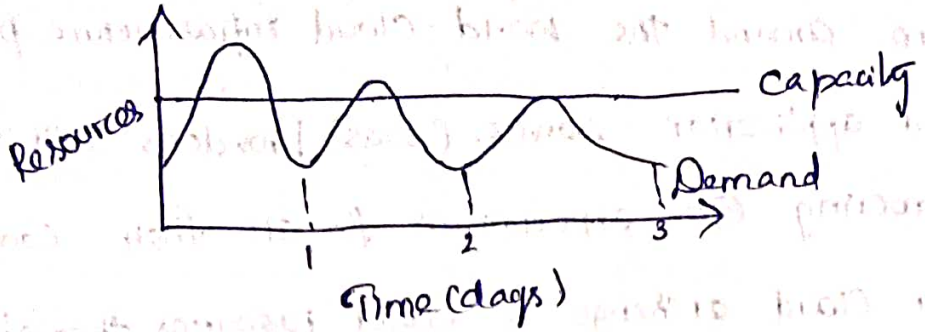
→ paid demand by the users



C) Constant Provisioning of resources with fixed capacity.

→ user demands are declined

→ results in resource waste



Resource Provisioning Methods:

→ Three methods.

1) Demand Driven Resource Provisioning

→ provides static resources

→ used in grid computing for many users

→ adds / removes computing instances / Resources based on the current utilization level of the allocated resources.

2) Event Driven method:

→ is based on predicted workload by time.

→ This scheme adds or removes machine instances based on a specific time event

3) Popularity Driven Resource Provisioning

→ is based on Internet traffic monitored.

→ the Internet searches for popularity of certain applications and creates the instances by popularity demand.

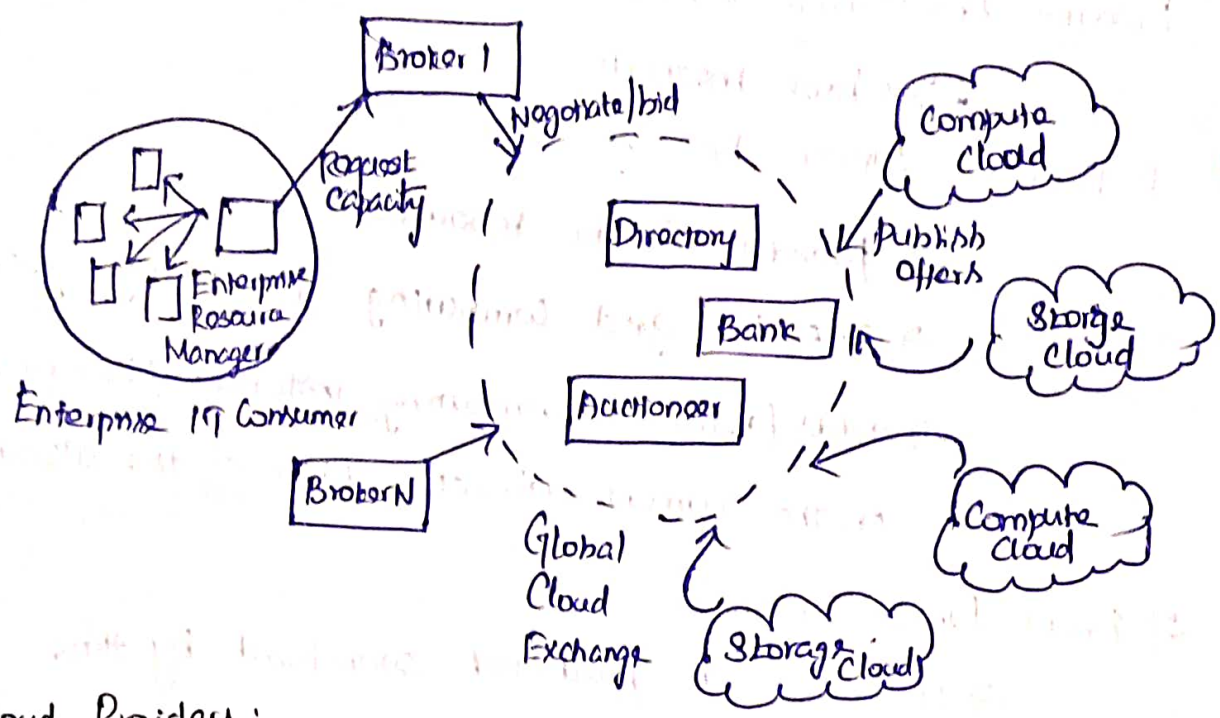
→ This predicts increased traffic with popularity.

Global Exchange of Cloud Resources:

→ to support a large number of application service consumers from around the world cloud infrastructure providers.

→ Cloud application service (SaaS) providers will have difficulty in meeting QoS expectations for all their consumers.

→ Inter cloud exchange of cloud resources through brokering.



Cloud Providers:

→ able to dynamically expand or resize their provisioning capability based on sudden spikes in workload demands.

Service Providers:

→ operate as part of a market-driven resource leasing federation

→ application service providers such as salesforce.com.

Cloud Exchange (CEx)

→ acts as a market maker.

→ bringing together service producers and consumers.

→ CEx allows participants to locate providers and consumers with fitting offers

SLA:

→ Specifies the details of the service to be provided.

→ Based on metrics agreed upon by all parties.

→ Includes incentives for meeting expectations and penalties for violating the expectations.

Security Overview:

Cloud Service Providers:

→ must learn from the managed service provider (MSP) model.

→ ensure that their customer's applications and data are secure if they hope to retain their customer base & competitiveness

→ Moving critical applications and sensitive data to public and shared cloud environment is of great concern for those corporations that are moving beyond their data centers network perimeter defense.

→ A cloud solution provider must ensure that consumers will continue to have same security and privacy controls over their application and services.

→ Solution provider give evidence to customers that their organization and customers are secure and they can meet their service level agreements and that they can prove compliance to auditors.

Software-as-a Security:

→ four methods

(a) Security Governance

(b) Virtual Machine Security

(c) IAM

(d) Security Standards.

Security Governance:

→ A security steering committee should be developed.

→ A charter for the security team is typically one of the first deliverables from the steering committee.

→ Lack of a formalized strategy can lead to an unsustainable operating model and security level.

→ Lack of attention to security governance.

→ Lack of proper governance and management of duties can also result in potential security risks.

Virtual Machine Security:

→ In cloud environment physical servers are consolidated to multiple virtual machine instances on virtualized servers.

→ Data center Security teams

→ Firewalls, Instruction Detection and Prevention, Integrity Monitoring, and log inspection.

→ Traditional line of defense to the virtual machine

→ To facilitate the centralized management of a server firewall policy.

→ Bidirectional stateful firewall.

→ Integrity monitoring and log inspection software must be applied at the virtual machine level.

Advantages of Virtual Machine Security:

(1) VM provides consistent control and management throughout the cloud.

(2) Economies of scale, deployment and cost savings for both service provider and the enterprise.

Identity Access Management (IAM)

→ is a critical function for every organization

→ a fundamental expectation of SaaS customers is that the principle of least privilege is granted to their data

→ Cloud services and services on demand is changing the identity management landscape.

→ The current models are challenging

(1) Trust assumptions

(2) Privacy implications

(3) Authentication and Authorization.

→ Balancing act for SaaS Providers

→ Another issue will be finding the right balance between usability and security.

Security Standards:

→ define the processes, procedures, and practices, necessary for implementing a security program.

→ to ensure a secure environment that provides privacy and security of confidential information.

→ a set of key principles intented to protect this type of trusted environment.

→ Messaging Standards

→ Protocols used - SAML, OAuth, OpenID, SSL/TLS

Security Assertion Markup Language (SAML)

→ XML based standard for communicating authentication, authorization and attribute information.

→ OASIS Security Services Technical committee

→ SAML is built on a number of existing standards, namely SOAP, HTTP, and XML

SAML relies on HTTP

SAML Core:

→ general syntax and semantics of SAML assertions.

→ protocol used to request and transmit those assertions from one system entity to another

SAML Binding:

→ determines how SAML requests and responses map to standard messaging protocols.

→ An important binding is the SAML SOAP Binding

SAML Standardizes:

(i) User authentication, entitlements and attribute

(ii) The relying party

(iii) A Subject

↳ entity in a particular domain

↳ a person identified by an email address

↳ It might be a printer.

SAML Assertions:

→ transferred from identity providers to service providers

→ authentication statements, attribute statements and authorization decision statements.

(i) Authentication standards

(ii) An attribute statement

(iii) An authorization decision statement.

⇒ Three types of SAML queries,

(i) Authentication query

(ii) Attribute query.

(iii) the Authorization decision query.

Open Authentication (OAuth):

→ OAuth Core 1.0

→ for exchanging a username and password for a token

→ to provide tools to protect the token.

OAuth:

→ no privacy at all

→ Implemented in a secure manner.

→ Secrets just like passwords must be protected.

OpenID:

→ an open decentralized standard for user authentication and access control.

→ allows users to log on to many services using the same digital identity.

SSL/TLS:

→ Transport layer Security and Secure Socket layer

SSL:

→ are cryptographically secure protocols.

→ provide security and data integrity over TCP/IP.

TLS:

→ endpoint authentication and data confidentiality.

→ three phases

(A) Peer negotiation for algorithm support

(B) Key exchange and authentication

(C) Symmetric cipher encryption and message authentication.

UNIT V CLOUD TECHNOLOGIES AND ADVANCEMENTS

Hadoop – MapReduce – Virtual Box – Google App Engine – Programming Environment for Google App Engine – Open Stack – Federation in the Cloud – Four levels of Federation – Federated Services and Applications – Future of Federation.

Hadoop:

→ Hadoop is an open source implementation of MapReduce coded and released in Java by Apache.

→ The Hadoop core is divided into two fundamental layers.

(i) MapReduce Engine

(ii) HDFS.

⇒ MapReduce Engine – is the computation engine running on top of HDFS as its data storage manager.

⇒ HDFS: – is distributed file system inspired by GFS that organizes files and stores their data on a distributed computing system.

HDFS Architecture:

→ HDFS has a master/slave architecture containing a single NameNode as the master and a number of DataNodes as workers.

→ The mapping of blocks to Data Nodes is determined by the NameNode.

→ The NameNode also manages the file systems meta data and Namespace.

HDFS Features:

→ Distributed file systems have special requirements. such as,

(i) Performance

(ii) Scalability

(iii) Concurrency Control

(iv) fault tolerance

(v) Security requirements to operate efficiently.

Hadoop - Requirements:

→ reliable requirements of file system are,

(i) Block Replication

(ii) Replica Replacement

(iii) Heart beat and Block report messages

Advantages:

(i) Size of individual block increases.

(ii) Fast Streaming reads of data.

HDFS Operation:

(i) Write (ii) Read.

Read: → a user sends an "open" request to the NameNode to get the location of file blocks.

Read : NameNode returns the address of a list of DataNodes containing replica information for the requested file.

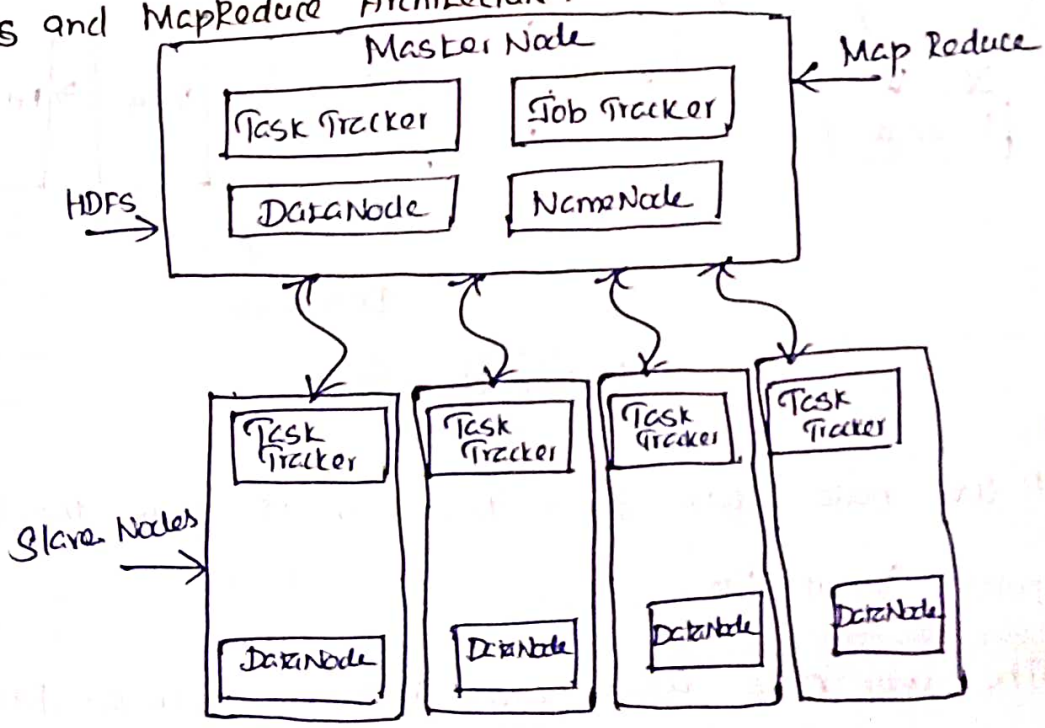
Write : - a user sends a create request to the NameNode to create a new file in file system Namespace.

- The Streames stores the block in the first allocated DataNode.

MapReduce :

→ The topmost layer of Hadoop is the MapReduce engine that manages the data flow and control flow of MapReduce jobs over distributed computing systems.

HDFS and MapReduce Architecture :



→ MapReduce engine also has a master/slave architecture consisting of a single JobTracker as the master and a number of Task Trackers as the slaves.

Running a Job in Hadoop:

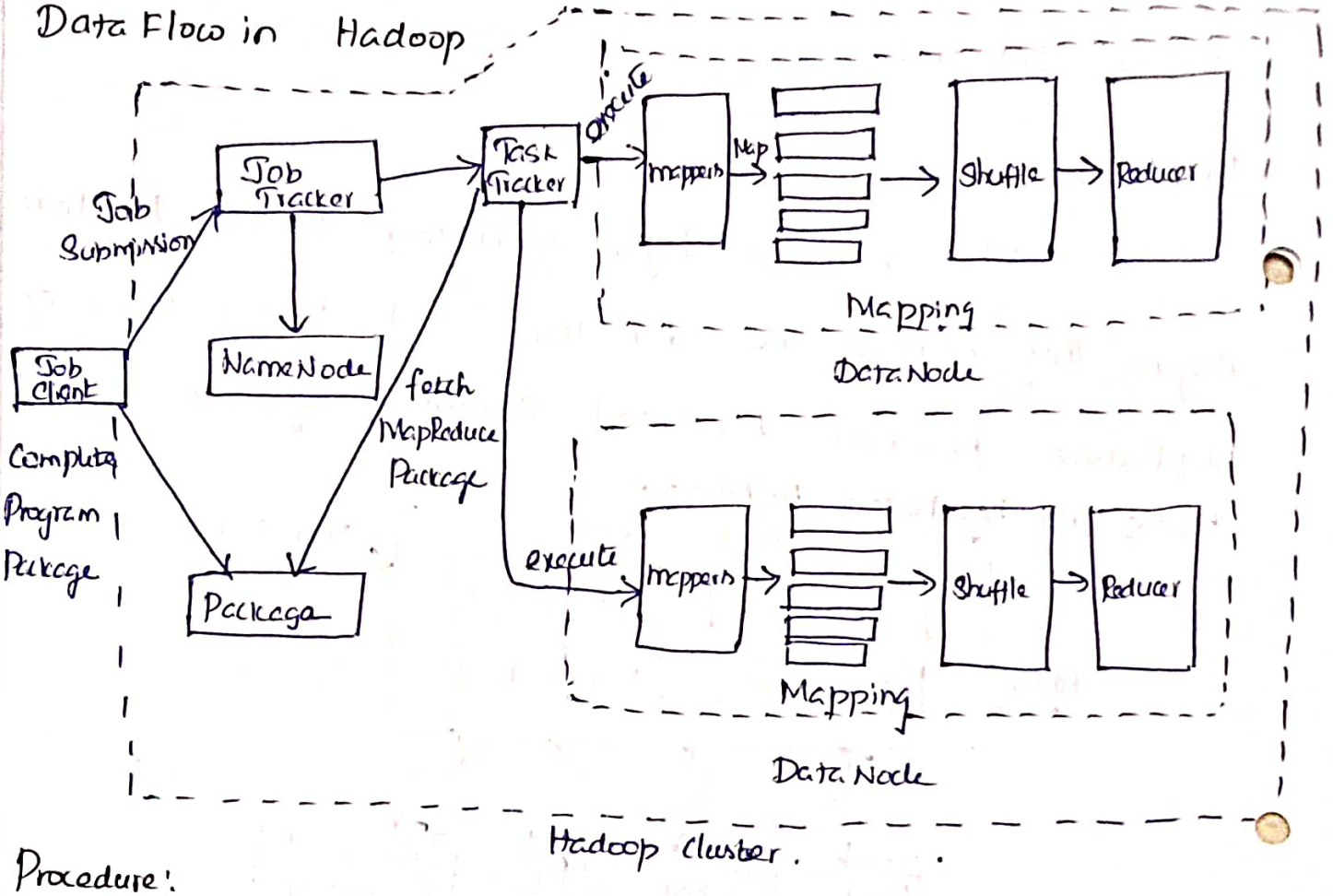
→ Three Components.

(1) User Node

(2) JobTracker

(3) Task Trackers.

Data Flow in Hadoop



Procedure:

(1) A user node asks for a new job ID from the JobTracker and computes input file splits.

(2) The user node copies some resources, such as jobs JAR file, Configuration file, and computed input splits, to the JobTrackers file system.

(3) The user node submits the job to the JobTracker by calling the Submit Job() - function.

(g) Task Assignment - JobTracker creates one map task for each computed input split by the user node and assigns the map tasks to the execution slots of the Task Trackers.

(h) Task Execution - The control flow to execute a task starts inside the Task Tracker by copying the Job JAR File to its file system.

Virtual Box:

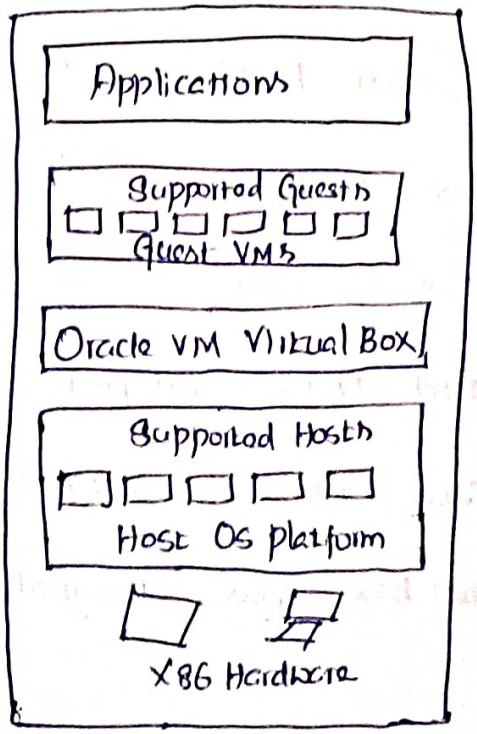
→ Oracle VM VirtualBox is a cross platform virtualization application.

→ It installs on the existing Intel or AMD based computers.

→ It extends the capabilities of existing computer so that it can run multiple OSes, inside multiple virtual machines, at the same time.

→ The user can install and run as many virtual machines.

Architecture of Virtual Box:



→ Virtual Box supported in Windows, macOS, Linux, Solaris and Open Solaris.

→ Guest VMs can also directly communicate with each other if configured to do so.

→ Virtual Box supports both Intel VT-x and AMD-V hardware assisted virtualization.

⇒ hard disk in one of three disk image formats,

VDI – Virtual Box Specific VirtualBox Disk – Stored as ".vdi"

VMDK – VMware products, ".vmdk" filename extensions.

VHD – Windows Virtual Pc and HyperV – ".vhd" filename extension

Virtual Box Network Cards.

→ For an Ethernet Adapter, Virtual Box Virtuallizes these Network Interface Cards,

(a) AMD PCnet PCI II

(b) AMD PCnet - Fast III

(c) Intel Pro/1000 MT Desktop

(d) Intel Pro/1000 MT Server.

(e) Intel Pro/1000 T-Server.

(f) Paravirtualized Network Adapter.

→ For a sound card Virtual Box virtualizes Intel HD Audio.

→ Oracle VM Virtual Box was designed to be modular and flexible.

Google App Engine:

→ Google platform is based on its search engine expertise.

→ Google's App Engine (GAE) which offers a Paas Platform supporting various Cloud and Web Applications.

→ Google has pioneered cloud development by leveraging the large number of data centers it operates.

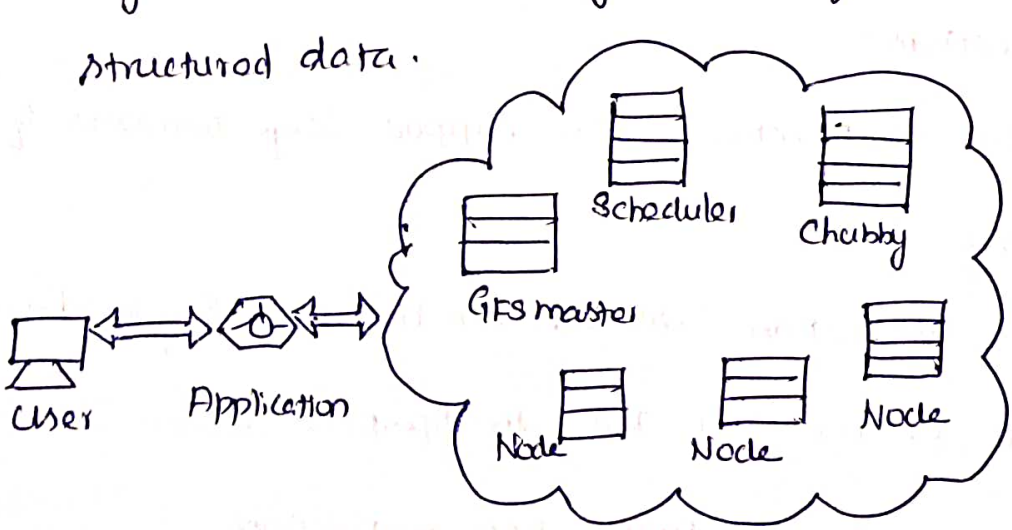
GAE Architecture:

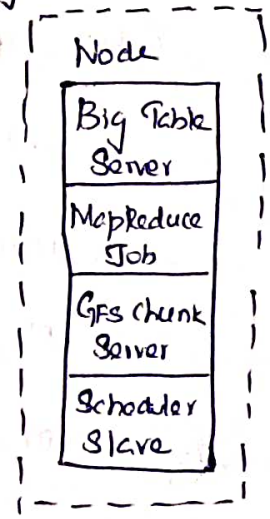
→ GFS is used for storing large amounts of data.

→ MapReduce is for use in application program development.

→ Chubby is used for distributed application lock services.

→ Big table offers a storage service for accessing structured data.





→ Extra services such as chubby for distributed locks can also run in the clusters.

→ GAE can be thought of as the combination of several software components.

Functional Modules of GAE:

→ Five major components

(a) The data store secures data management operations.

(b) It supports two development languages Python and Java.

(c) Software Development kit (SDK) is used for local application development.

(d) The administration Console is used for easy management of user application development cycles.

(e) The GAE web service infrastructure provides special interfaces to guarantee flexible use and management of storage and network resources by GAE.

GAE Applications:

(a) These applications can support large numbers of users simultaneously.

(b) The application are all run in the Google data user.

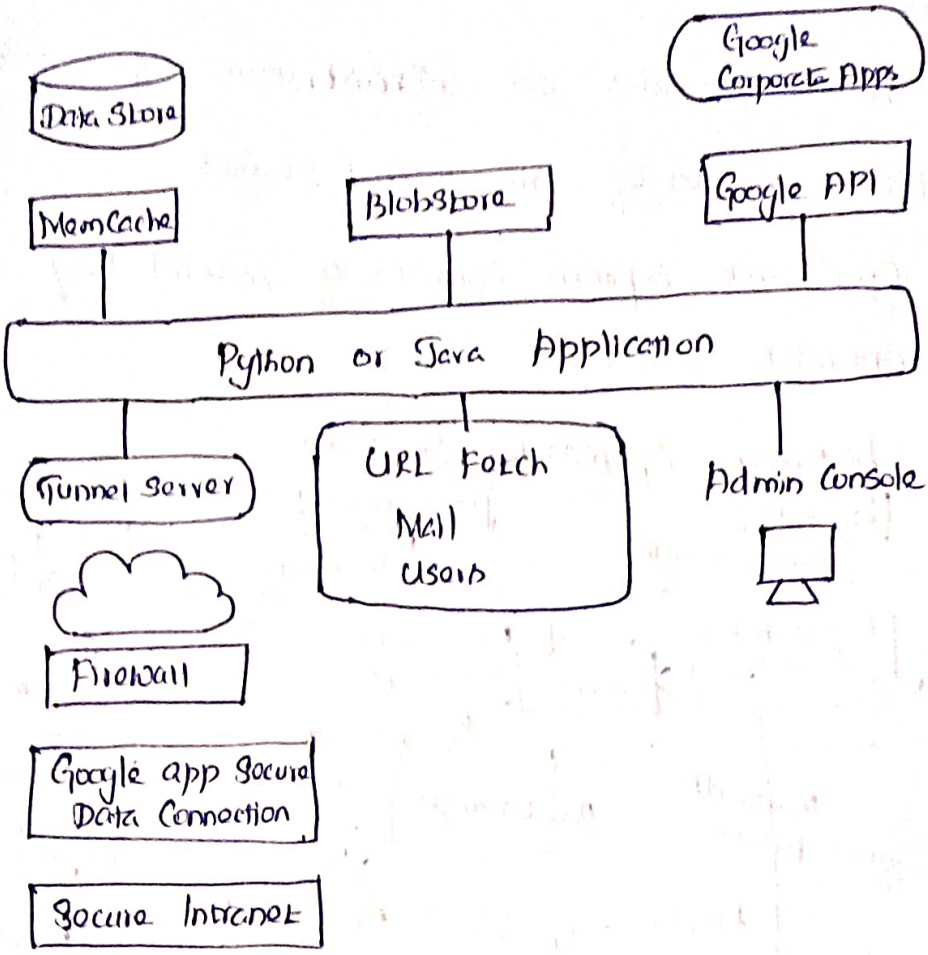
(c) Each cluster can run multipurpose servers.

(d) GAE supports many web applications.

Programming Environment for Google App Engine:

→ Two supported languages - Java and Python.

→ Several web resources (HTTP://code.google.com/appengine) and specific books and articles discuss how to program GAE.



Big Table System Goals:

(1) The applications want asynchronous processes to be continuously updating different pieces of data and want access to the most current data at all times.

(2) The database needs to support very high read/write rates and the scale might be millions of operations per second.

(3) The application may need to examine data changes over time.

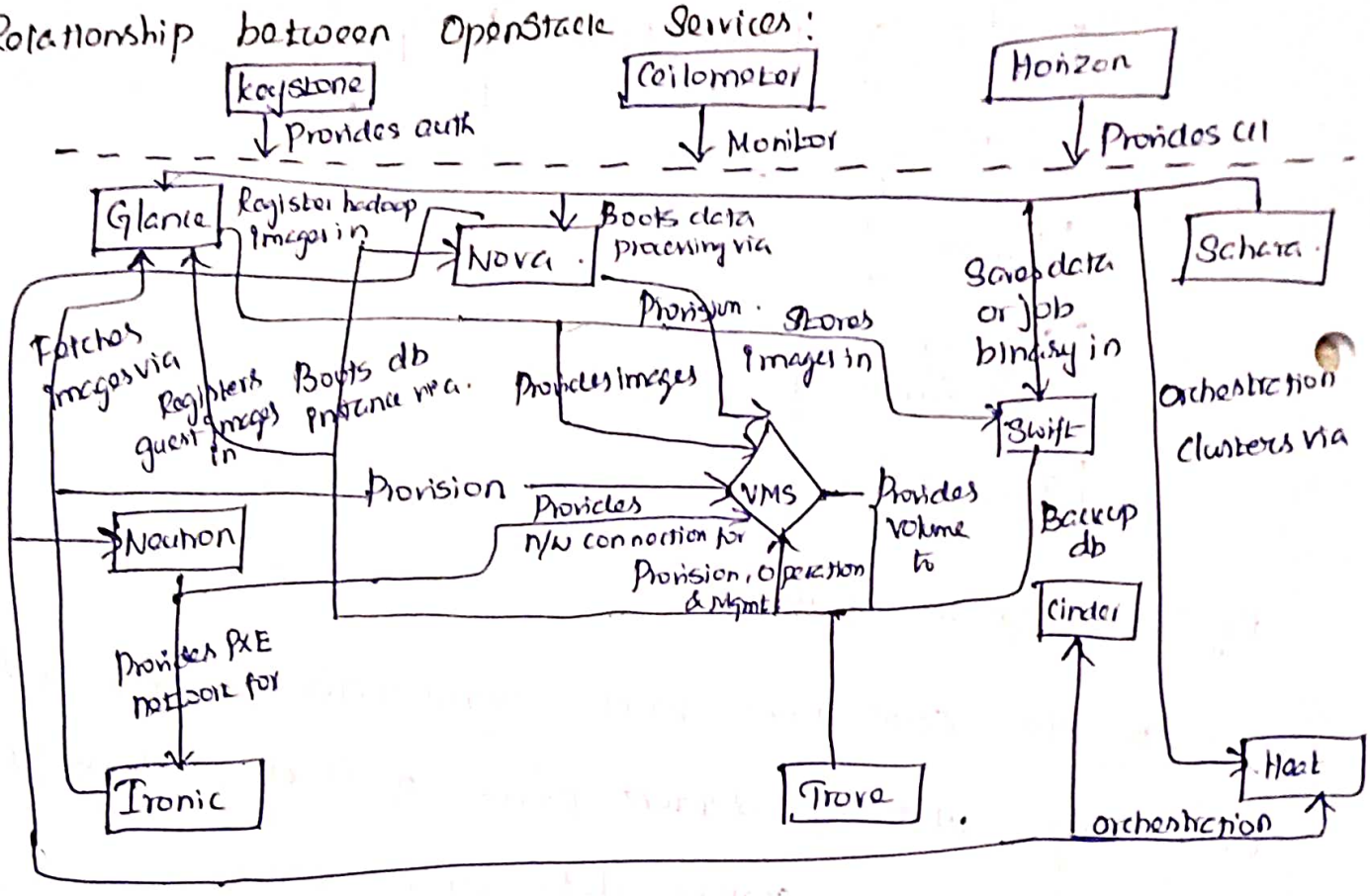
⇒ Chubby, Google's Distributed Lock service. Chubby is intended to provide a coarse gained locking service.

OpenStack:

→ OpenStack provides an Infrastructure as a Service. (IaaS) solution through a set of interrelated services.

→ The OpenStack system consists of several key services that are separately installed.

Relationship between OpenStack Services:



→ For communication between the processes of one service an AMQP message broker is used.

→ The service state is stored in a database

→ The Controller node requires a minimum of two network interfaces.

→ The Compute node runs the hypervisor portion of compute that operates instances.

Federation in the cloud:

→ It supports Microsoft's cardspace and Novell's Digital Me

→ Google, Apple, AOL, IBM, Live Journal and Sive have all incorporated this protocol into their cloud based solutions in the last few years.

→ Polling is how most of us check our email.

→ It is flexible and designed to be extended.

→ An Amazon EC2-backed server can run Jetty and comet from Dojo.

→ Federation differs from peering, which requires a prior agreement between parties before a server-to-server link can be established.

Four levels of Federation:

→ ability for two XMPP servers in different domains to exchange XML stanzas.

Permissive Federation:

→ occurs when a server accepts a connection from a peer network server without verifying its identity using DNS lookups or certificate checking.

→ Lack of verification or authentication may lead to domain spoofing.

Verified Federation:

→ When a server accepts a connection from a peer after the identity of the peer has been verified

→ the connection is not encrypted, and the use of identity verification effectively prevent domain spoofing.

Encrypted Federation:

→ The peer must present a digital certificate

→ The certificate may be self signed, but this prevents using mutual authentication.

Trusted Federation:

→ the use of digital certificates results not only in a channel encryption but also in strong authentication.

→ The use of trusted domain certificates effectively prevents DNS poisoning attacks.

Federated services and Application:

→ S2S Federation is a good start toward building a real time communications cloud.

→ Finding the entities is a process called discovery.

→ XMPP uses servie discovery to find the aforementioned entities.

→ XMPP includes a method for maintaining personal lists of other entities known as roster technology.

→ Most XMPP deployments include custom directories so that internal users of those services can easily find what they are looking for.

Future of Federation:

(i) The implementation of federated communications is a precursor to building a seamless cloud that can interact with people, devices, information feeds, documents, application interfaces and other entities.

(ii) The power of a federated, presence enabled communication infrastructure, it enables software developers and service Providers to build and deploy such applications without asking permission from a large centralized communications operator.

(iii) The process of server-to-server federation for the purpose of inter domain communication has played a large role in the success of XMPP.

(iv) These mechanisms have provided a stable, secure foundation for growth of the XMPP network and similar real time technologies.