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 :-

⇒ Cloud Computing becomes must have technology in every IT organization because of its prominent benefits over the existing Computing technologies.

⇒ It has the ability to make a better use of distributed hardware and software resources which are combined together to acheive higher throughput at lower cost and can able to solve large scale Computation problems in lesser time.

⇒ Cloud Computing has the aggregation of computing resources (like CPU, memories) networking solutions, storage Management and Virtualization solutions which are available on demand, & delivered economically.

⇒ Today, the use of cloud Computing is massive in size because it has the ability to deliver resources and services of any size at any time without worrying to setup anything at faster rate with economical cost.

A simple definition of cloud Computing involves delivering types of services over the internet.

Cloud delivers,

* Software and Analytics.
* Secure and safe data storage.
* Networking Recources.

The term cloud has been used historically as a metaphor for the internet.

Different cloud-based applications:-

→ Send a file to your friends via the web.
→ Use a mobile app
→ download an Image
→ binge a Netflix show
→ Play an Online Video game.
→ Sharing your Information on oneDrive, Sharepoint, or an email Server.. All these services are stored in the cloud.

What is cloud Computing:-

* It simply states that cloud Computing means storing and accessing the data and programs over the internet rather than the Computer's hard disle.

* The data can be anything such as music, files, images, documents any many more.

Cloud Computing Architecture :-

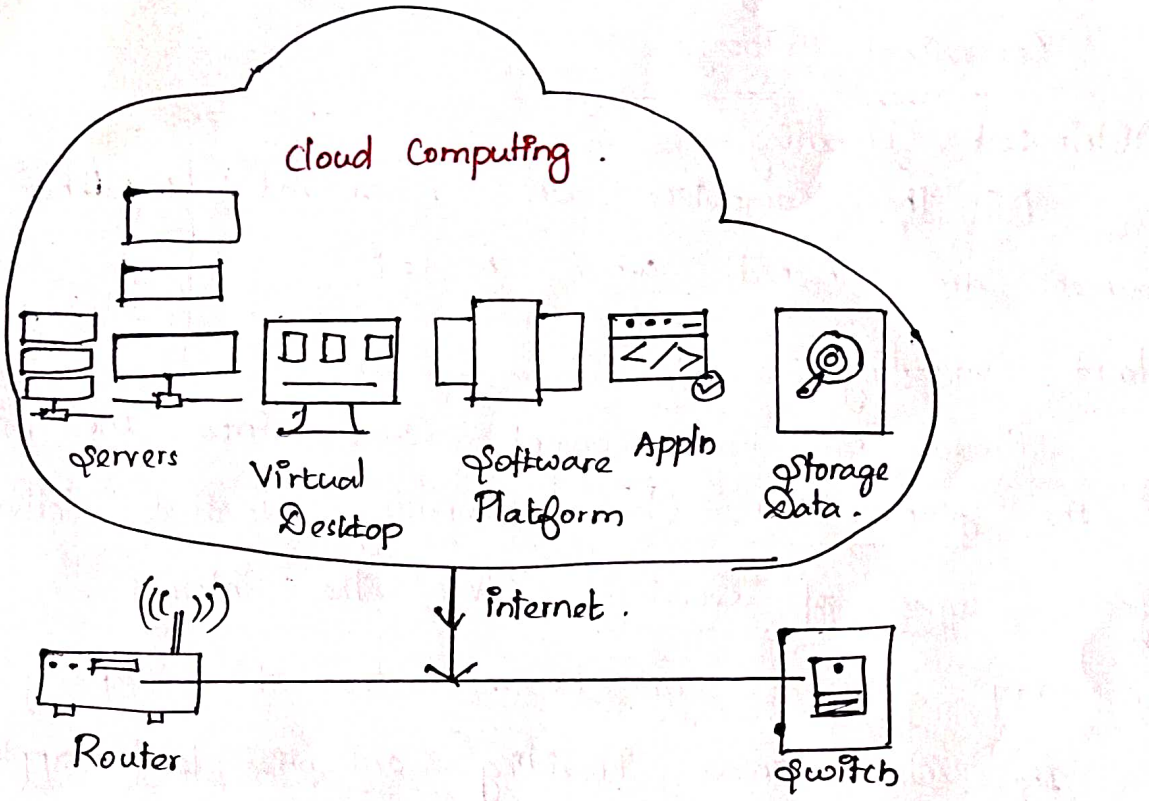
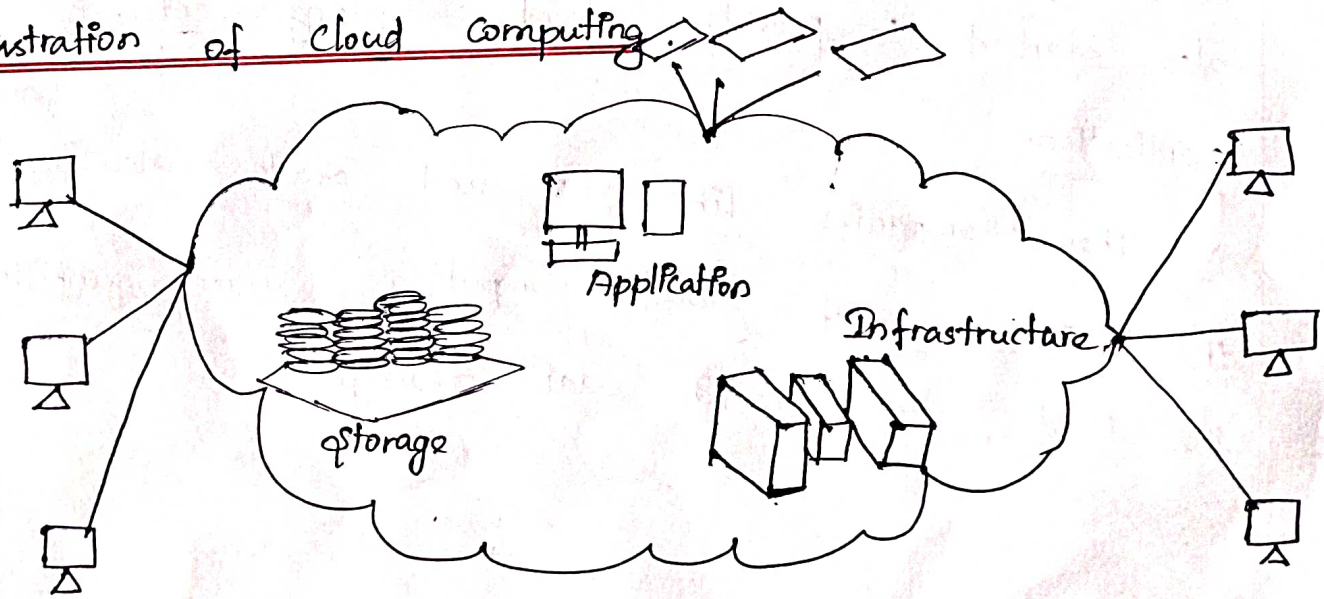


Illustration of Cloud Computing .



History of Cloud Computing.

* Client/Server Computing :-
  Centralized Storage.

* Distributed Computing :-
  All the computers are networked together and resources are shared when needed.

* Cloud Computing :
  Cloud Computing Concept Came into the picture in the year 1950. Cloud Computing involves delivering different types of services over the internet.

* In 1999, Salesforce.com became the 1st Company to enter the cloud arena, Providing enterprise-level applications to end users through the internet.

* Then in 2002, Amazon Came up with Amazon Web Services providing services like Computation, storage, and even human intelligence.

* In 2009, Google Apps and Microsoft's Windows Azure also started to provide cloud computing enterprise applications.

* Other Companies like HP and Oracle also joined the stream of cloud Computing, for fulfilling the need for greater data storage.

Types of cloud computing :-

1. Infrastructure as a service (Iaas)
2. Platform as a service (Paas)
3. Software as a service (Saas)

Cloud Computing Service Models :-

Software as a service (Saas)
Applications, Management, and User Interface provided over the network.

Platform as a service (Paas)
Application development framework, operating systems and deployment framework.

Infrastructure as a service (Iaas)
Virtual Computing, Storage & network Resources that can be provisioned on demand.

Saas → End User Utilization
Mail, Sales force, Drop box,

Paas → Software Development.
Google, force.com.

Iaas → System Architecture.
amazon Web Services.

Infrastructure - as - a - service (Iaas) :- resource.

⇒ Iaas provides the users capability to (provision Computing and storage resources)

⇒ These resources are provided to the users

as (virtual machines instances and virtual storage)

⇒ Virtual Resources provisioned by the users are billed based on ~~per~~ a pay-per-use paradigm.

Platform-as-a-Service (Paas)

⇒ Paas provides the users the capability to develop and deploy application in the cloud using the development tools, application programming interfaces, software Libraries and services provided by the cloud service provider.

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

⇒ Google apps is one of the most famous Platform-as-a-service providers.

⇒ This is the idea that someone can provide the hardware (as in Iaas) plus a certain amount of application software upon which you can build your application.

[⇒ It is an application development and deployment platform delivered as a service to developers over the web]

Software-as-a-Service (Saas) → It is the broadest market.

[Saas provides the users a complete software application] or the User Interface to the application itself.

The ⇒ Cloud service provider manages the underlying cloud infrastructure including servers, network, operating systems, storage, and application software, and the user is unaware of the underlying architecture of the cloud.

⇒ Applications are provided to the user through a thin client interface (Eg. browser).

⇒ Saas applications are platform independent and can be accessed from various client devices such as work stations, Laptop, Tablets and Smartphones, running different operating systems.

Deployment Models :-

Public Cloud.

⇒ In the public cloud deployment model,

⇒ cloud services are available to the general public or a large group of companies.

⇒ cloud resources are shared among different users (individuals, large organizations, small and medium enterprises and governments)

Private cloud :-

In the private cloud deployment model,

⇒ cloud infrastructure is operated for exclusive for single organization.

⇒ Private cloud services are dedicated for a

single organization].

⇒ Private clouds are best suited for applications where security is very important and organizations that want to have very tight control over their data.

Hybrid Cloud:-

⇒ [It combines the services of multiple cloud (Private or public cloud)]

⇒ It is suited for organizations that want to take advantage of secured applications and data hosting on a private cloud.

⇒ And at the same time benefit from cost savings by hosting shared applications and data in public clouds

Community Cloud:-

⇒ The cloud services are shared by several organizations that have the same policy and Compliance Considerations.

⇒ Community cloud best suited for organizations that want access to the same application and data, and want the cloud costs to be shared with the larger group.

The World of Business:-

Businesses can employ cloud Computing in different ways. Some users maintain all apps and data on the cloud,

While others use a hybrid Model, Keeping Certain apps and data on private servers and others on the cloud.

Google cloud
Amazon Web Services (AWS)
Microsoft Azure
IBM cloud
Alibaba cloud.

Definition of cloud:-

"Cloud Computing is a model for enabling ubiquitos, convenient, On-demand network access to a shared pool of configurable Computing resources (Eg. networks, servers, storage, applications & services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

This cloud model is composed of five essential characteristics, three service models and four deployment models.

Five Essential characteristics:-

1. On demand Self-Service.
2. Broad Network Access.
3. Resource Pooling.
4. Rapid Elasticity or expansion.
5. Measured Service.

Three "Service Models"

1. Software as a Service - SaaS
2. Platform as a Service - Paas.
3. Infrastructure as a Service - Iaas.

Deployment Models :-

1. Private Cloud.
2. Community cloud.
3. Public cloud.
4. 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.

Short Definition :-

A computing paradigm with virtual network of remote servers allowing users to store, process and access data, providing On-demand Computational services with features like elasticity, scalability, security and redundancy in cloud computing.

Top 10 Cloud Service Providers :-

* Microsoft Azure.
* Amazon Web Services (AWS)
* Google cloud.

* Alibaba cloud
* IBM cloud
* orade
* Salesforce.
* SAP
* Rackspace cloud.
* VMWare.

Evolution of cloud computing:-

Hardware Evolution:-

- First generation computers.
- Second generation computers.
- Third generation computers.
- Fourth Generation Computers.
- Fifth Generation Computers.

Internet Software Evolution:-

⇒ Establishing a common Protocol for the internet.
⇒ Evolution of IPv6
⇒ Finding a common Method to communicate Using the Internet Protocol.
⇒ Building a common Interface to the internet.
⇒ The appearance of Cloud formations - from Computeo to a Grid of Many.

Server Virtualization:-

⇒ Parallel Processing.
⇒ Vector Processing.

Symmetric Multiprocessing Systems.

Massively Parallel Processing Systems.

Cloud Computing :-

⇒ Cloud Computing Leverages dynamic resources to deliver a large number of services to end users.

⇒ It is a high Throughput Computing (HTC) paradigm.

⇒ It enables users to share access to resources from anywhere at any time.

Hardware Evolution :-

⇒ In 1930, Binary Arithmetic was developed.

Computer processing technology, terminology and programming Languages.

⇒ In 1939, Electronic Computer was developed.

Computations were performed using Vacuum tube technology.

⇒ In 1941, Konrad Zuse's Z3 was developed.

Support both floating-point and binary arithmetic.

Different Generations of Hardware Evolution :-

* First Generation Computers.
* Second Generation Computers.
* Third Generation Computers.
* Fourth Generation Computers.

1. First Generation Computer:-

Time period : 1942 to 1955

Technology : Vacuum tubes

Size : Very Large System

Processing : Very slow

Examples :-

1. ENIAC (Electronic Numerical Integrator and Computer)
2. EDVAC (Electronic Discrete Variable Automatic Computer.

Advantages :-

* It made use of Vacuum tubes which was the advanced technology at that time.
* computations was performed in milliseconds.

Disadvantages :-

* Very big in size, Weight was about 30 tons,
* Very costly.
* Requires more power Consumption.
* Large amount heat was generated.

2. Second Generation Computer :-

Time period :- 1956 to 1965

Technology :- Transistors

Size : Smaller

Processing :- Faster

Examples :- Honeywell 400, IBM 7094.

Advantages

* Less heat than first Generation.
* Assembly language and punch cards were used for input.
* Low cost than first Generation computers.
* Computations were performed in microseconds.
* Better portability as Compared to first Generation.

Disadvantages :-

* A cooling system was required.
* Constant maintenance was required.
* Only used for specific Purposes.

Third Generation Computers :-

Time period :- 1966 to 1975

Technology :- Iss (Integrated Circuits)

Size : Small as Compared to the second Generation Computers.

Processing :- Faster than 2nd Generation Computers.

Examples :-

* PDP_8 (Program Data Processor)
* PDP-11

Advantages :-

* Cheaper Compared to the second Generation Computers.
* They were fast and reliable.
* IC not only reduce the size of the computer but it also improves the performance of the computer.
* Computations were performed in nanoseconds.

Disadvantages:-

* IC chips are difficult to maintain.
* The highly sophisticated technology required for the manufacturing of IC chips.
* Air Conditioning is required.

Fourth Generation Computer:-

Time period:- 1975 to Till Date

Technology:- Microprocessor

Size Small as Compared to third Generation Computer.

Processing:- Faster than Third Generation Computers.

Examples:- IBM 4341, DEC 10.

Advantages:- * Fastest in Computation, Size get reduced, * Heat generated is small, Less Maintenance is required.

Disadvantages:-

* The Microprocessor design and fabrication are very complex.
* Air Conditioning is required in many Cases.
* Advance technology is required to make the ICs.

Internet Software Evolution:-

Internet Protocol:-

It is the standard communication protocol used by every computer on the internet.

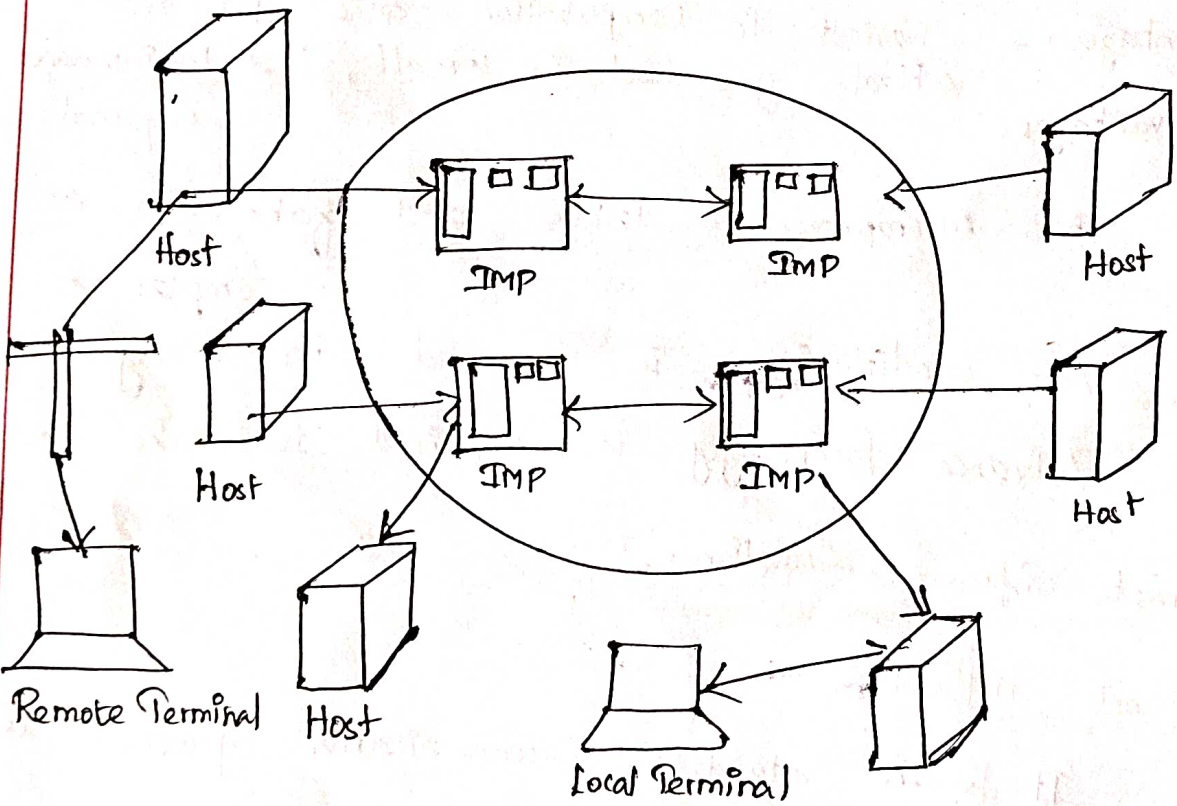
Internet was significally developed by three individuals.

* Vannevar Bush - MEMIX (1930)
* Norbert Wiener.
* Marshall McLuhan.

* Licklider was founder for the creation of the ARPANET (Advanced Research Projects Agency Networks)
* Clark deployed a minicomputer called an Interface Message Processor (IMP) at each site.

NCP Network Control Program (NCP) - first Networking Protocol that was used on the ARPANET.

IMP Architecture.



IMP - Interface Message Processor:-

* First Packet Switching Router.
* First Generation Gateway.
* Used to connect users networks to ARPANET.

⇒ TCP V3 and IP V3 ,

⇒ TCP V4 and IPV4

Today, IPV4 is the standard Protocol.

Evolution of IPV6 :-

⇒ IPV4 was never designed to scale to global levels.

⇒ To increase the available address space, it had to process large data packets (ie.: more bits of data)

↳ Leads to Longer IP address caused Pbms for existing hardware & software.

To overcome these problems, Internet Engineering Task Force (IETF) developed IPV6, which was released in January 1995.

IPV6 → It is sometimes called the Next Generation Internet Protocol (IPNG) or TCP/IP V6.

Finding Common Method to Communicate Using the Internet Protocol :-

In 1960s → The word hypertext was created by Ted Nelson

In 1962 → Engelbart's first project was Augment, He developed the mouse, Graphical User Interface (GUI), and the first working hypertext system, named NLS (ON-Line System)

In 1980s → Web was developed, in Europe by Tim Berners-Lee and Robert Cailliau

ARPANET:-

Advanced Research Project Agency:-

⇒ Foundations for the Networking World.

⇒ First Networking Protocol used by ARPANET was NLP - Network Control Protocol.

Establishing a Common Protocol for the Internet.

NCP essentially provided a transport Layer consisting of the ⇒ ARPANET Host-to-Host Protocol (AHHP) & Initial Connection Protocol ICP

AHHP – Unidirectional data transfer.

⇒ It specified how to transmit a unidirectional, flow controlled data stream between two hosts.

ICP:- Bidirectional data transfer.

⇒ It specified how to establish a bidirectional pair of data streams between a pair of connected host processes.

Application Protocols:-

* File transfer Protocol (FTP) used for file transfer,
* Simple Mail Transfer Protocol (SMTP), used for sending email.
* In 1983, ARPANET changed its Network protocols to TCP/IP.

four versions of TCP/IP.

TcpV1

TCP V2

Building a Common Interface to the Internet:-

⇒ In the fall of 1990, Berners-Lee developed the first web browser.

feature : an integrated editor that could create hypertext documents.

⇒ In 1991, Enhanced the ~~Server~~ and Notice the download Web Server and browser.

Then, Berners-Lee → enhanced the Server and browser by adding support for the FTP Protocol.

* He also added a Telnet Server.
* Mosaic browser.
  - It was the first widely popular web browser available to the general public.
* Then www was developed.

In 1994.-
Netscape released first version of its browser, Mozilla 0.96b.

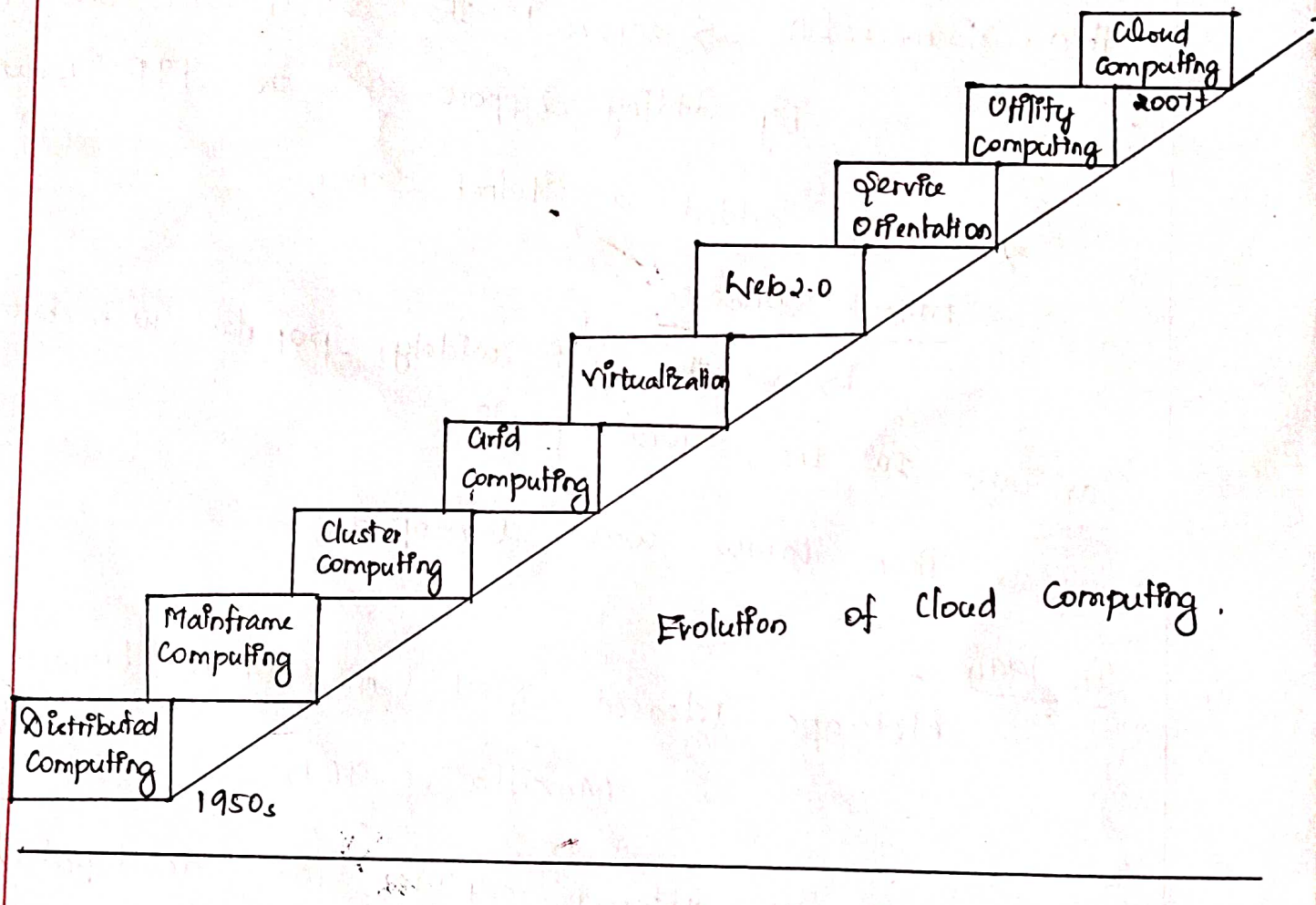
In 1995 :- Microsoft Internet Explorer was developed that ~~spor~~ support graphical Web browser.

Mozilla Firefox.
Released in November 2004, became very popular almost immediately.

The Appearance of cloud formations - from one computer to a cloud of many.

Evolution of cloud services :-

Cloud computing is often referred to as "successor of grid computing"



Evolution of Cloud Computing.

Server Virtualization :-

⇒ Virtualization is method of running multiple independent virtual operating systems on a physical Computer.

⇒ It maximizes the return on investment for the computer.

⇒ It is a way of reducing the majority of hardware acquisition and maintenance Costs..

↳ Result It produce Money saving for a Company.

Server Virtualization :-

* Parallel Processing
* Vector Processing.
* Symmetric Multiprocessing Systems.
* Massively Parallel Processing Systems.

Parallel Processing :-

⇒ Parallel processing is performed by the simultaneous execution of program instructions that have been allocated across multiple processors.

objective :- Running a program in less time.

The next advancedment in parallel processing - Multiprogramming

⇒ In Multiprogramming system, Multiple programs submitted by users but each allocated to use the processor for a short time.

↳ This approach is known as "Round-Robin (RR scheduling) scheduling"

RR Schedule :-

⇒ All executable processes are held in a circular queue. The time slice is defined based on the number of executable processes that are in the queue.

Vector Processing :-

⇒ Vector processing was developed to increase processing performance by operating in a multitasking manner.

⇒ Matrix operations were added to computers to allow a single instruction to manipulate two arrays of numbers performing arithmetic operations.

⇒ This was suitable for certain types of applications that contain vector data or matrix data.

Symmetric Multiprocessing Systems :-

⇒ Symmetric Multiprocessing Systems (SMP) was developed to address the problem of resource management in master/slave models.

⇒ In SMP systems, each processor is equally capable and responsible for managing the workflow as it passes through the system.

⇒ The primary goal is to achieve sequential consistency.

Massively Parallel Processing Systems :-

In Massive parallel processing, a computer system

with many independent arithmetic units or entire microprocessors, which run in parallel.

⇒ All the processing elements are interconnected to act as one very large computer.

4. Underlying Principles of parallel and Distributed Computing:-

4.1 Eras of Computing

4.2 Parallel or Distributed Computing.

4.3. Elements of Parallel Computing

4.3.1 What is Parallel processing?.

4.3.2. Hardware Architectures for parallel processing

4.3.3. Approaches to parallel ~~processing~~ programming

4.3.4. Levels of Parallelism.

4.3.5. Laws of Caution.

4.4. Elements of Distributed Computing.

4.4.1. General Concepts and Definitions.

4.4.2. Components of a Distributed Systems.

4.4.3. Architectural styles for Distributed Computing.

4.4.4. Models for Inter-process Communication.

4.5. Technologies for Distributed Computing.

4.5.1. Remote Procedure Call

4.5.2 Distributed Object Framework.

4.5.3. Service Oriented Computing.

1) Eras of Computing :-

⇒ Two fundamentals and dominant models of computing are sequential and parallel.

Four Key elements of Computing developed during three eras.

- ⇒ Architecture.
- ⇒ Compilers
- ⇒ Applications.
- ⇒ Problem solving Environment.

2) Parallel and Distributed Computing :-

Parallel computing :-

⇒ It implies a tightly coupled system,

⇒ Computation is divided among several processors sharing the same memory.

⇒ Parallel systems are featured with multiple processors sharing the same physical memory and that were considered a single computer.

Distributed Computing System :-

Refers to a Loosely coupled systems.

	Parallel Computing	Distributed Computing.
1.	Many operations are performed simultaneously	System Components are Located at different locations.
2.	Single Computer is required	Uses Multiple Computers.
3.	Multiple processors perform multiple operations.	Multiple Computers perform multiple operations.
4.	It may have shared or distributed memory.	It have only distributed memory.
5.	Processors Communicate with each other through bus.	Computer Communicate with each other through message passing.
6.	Improves the System performance.	Improves system scalability, fault tolerance and resource sharing Capabilities.

4.3 Elements of Parallel Computing :-

4.3.1. What is Parallel Processing :-

⇒ Processing of multiple tasks simultaneously on multiple processors is called Parallel processing.

⇒ The parallel program consists of multiple active processes (tasks) simultaneously solving a given problem.

⇒ A given task is divided into multiple subtasks using a divide-and-conquer technique, and each subtask is processed on a different Central processing Unit (CPU)

⇒ Programming on multiprocessor system using the divide and conquer technique is called Parallel

programming.

Advantages:-

⇒ Multiple processors provides higher Computing Power.

⇒ Higher Performance than a single processor system.

4.3.2 Hardware Architecture for parallel processing.

⇒ The core elements of parallel processing are CPUs.

⇒ Based on the number of instructions and data streams, that can be processed simultaneously.

Computing systems are classified into four categories.

Four Categories:-

⇒ Single-Instruction, Single Data (SISD) Systems.

⇒ Single-Instruction, Multiple Data (SIMD) Systems.

⇒ Multiple-Instruction, Single Data (MISD) Systems.

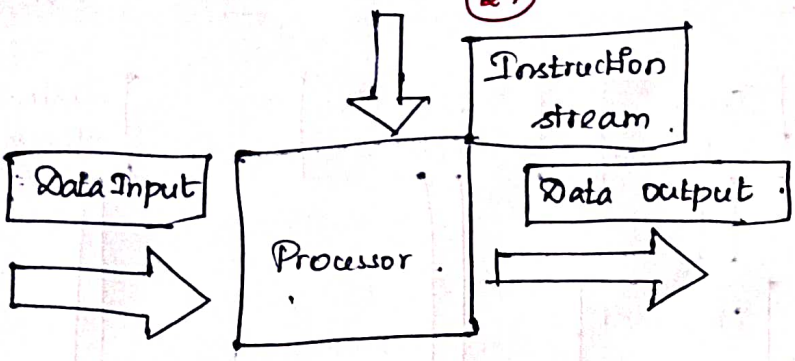
⇒ Multiple-Instruction, Multiple Data (MIMD) Systems.

* Shared Memory MIMD Machines.
* Distributed Memory MIMD machines.

Single-Instruction, Single Data (SISD) Systems.

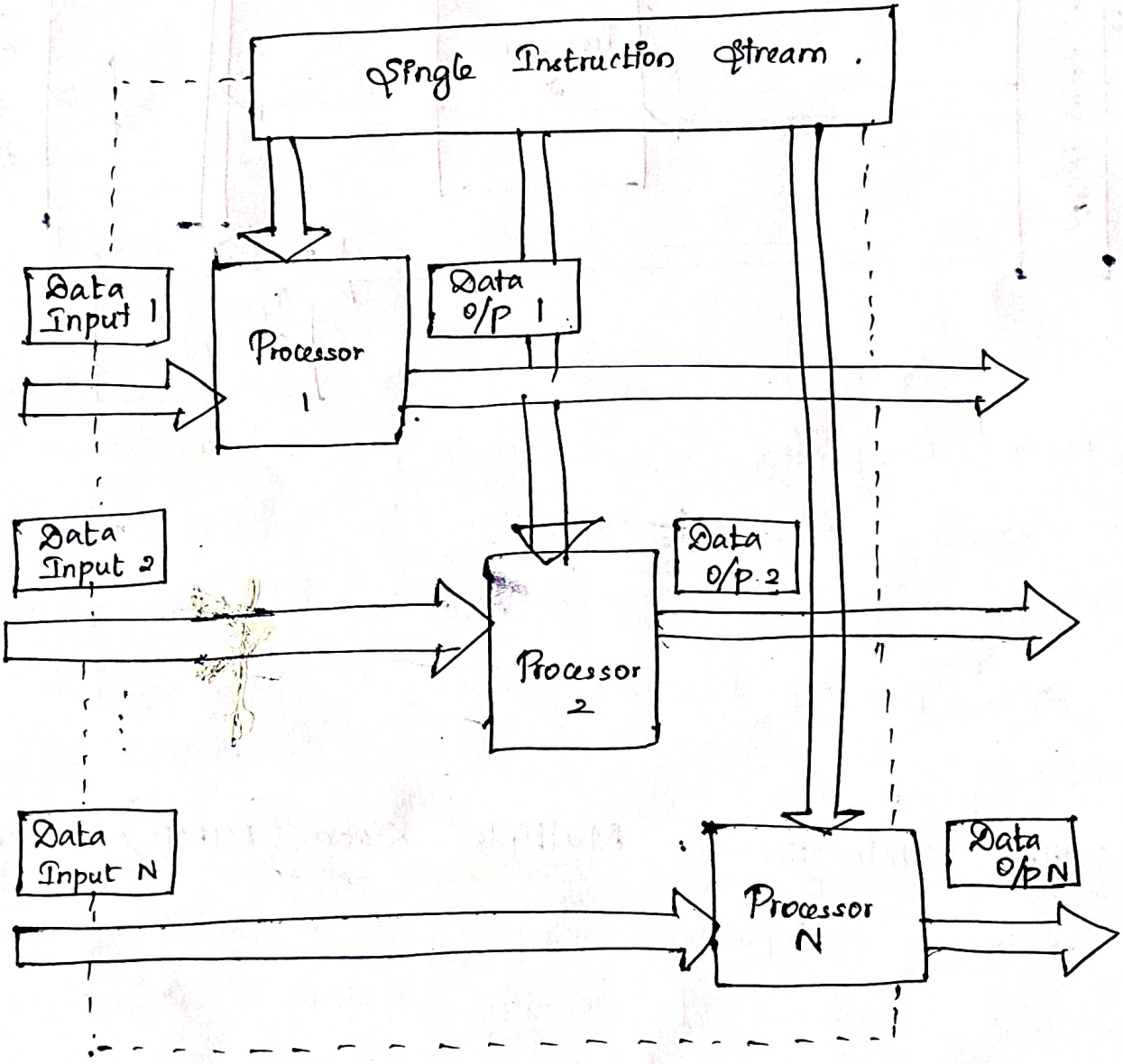
⇒ SISD Computing system is a uniprocessor machine capable of executing a single instruction, which operates on a single data stream. → Machine Instructions are processed sequentially.

Eg) IBM PC, Macintosh, & Workstations.



Single Instruction, Single Data (SISD) Systems.

Single Instruction, Multiple Data (SIMD) Systems.

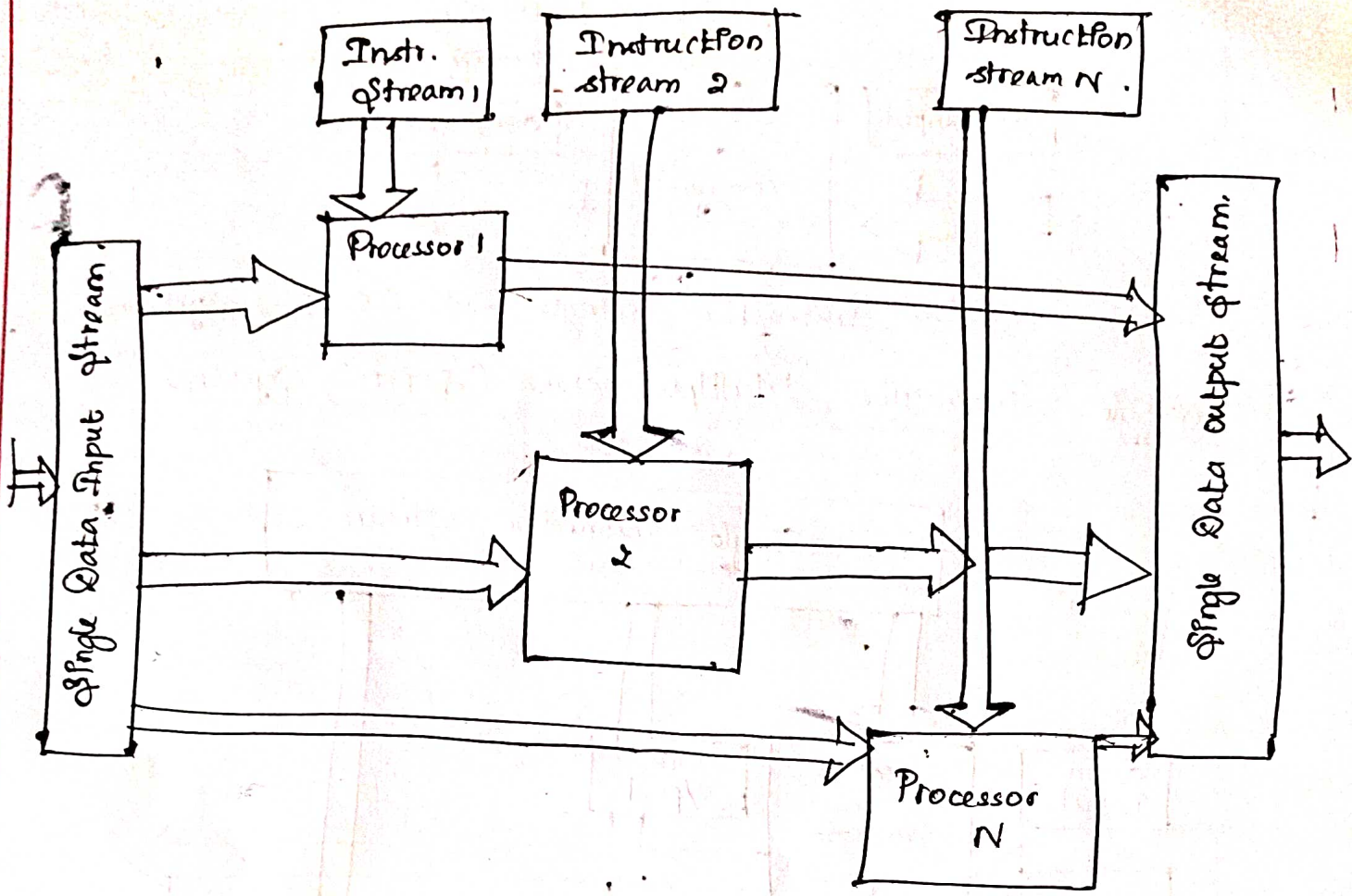


⇒ SIMD Computing System is a Multiprocessor machine.

⇒ Capable of executing same instruction on all the CPUs but operating on different data streams.

Eg) Cray's Vector Processing Machine.

Multiple Instruction, Single Data (MISD) Systems :-



⇒ MISD Computing is a multi processor machine capable of executing different instructions on different PEs all of them operating on the same data set.

⇒ MISD model not useful in most of the applications.

Multiple Instruction, Multiple Data (MIMD) System:-

MIMD Computing System is a multi processor machine capable of executing multiple instructions on multiple on multiple data set.

MIMD Machines are broadly categorized into

⇒ shared Memory MIMD

⇒ Distributed Memory MIMD.

Shared Memory MIMD Machines:-

⇒ All the PEs are connected to a single global memory and they have access it. System based on this model are also called tightly coupled multiprocessor systems.

⇒ The communication between PEs in this model takes place through the shared memory.

Eg) IBM SMP (Symmetric multi Processing)

Distributed Memory MIMD Machines:-

⇒ All PEs have a local memory. Systems based on this model are also called Loosely coupled multi processor system.

The communication between PEs in this model takes place through the interconnection network, inter process communication channel. (IPC)

2.3.3 Approaches to Parallel Programming:-

Variety of Parallel Programming Approaches.

⇒ Data Parallelism.

⇒ Process Parallelism.

⇒ Farmer and Worker Model.

Data Level Parallelism:-

The divide and conquer technique is used to

split data into multiple data set, and each data set is processed on different PEs using the same instruction.

Process Parallelism :-

⇒ A given operation has multiple activities that can be processed on multiple processors.

Farmer & Worker Model :-

⇒ Job distribution Approach is used.

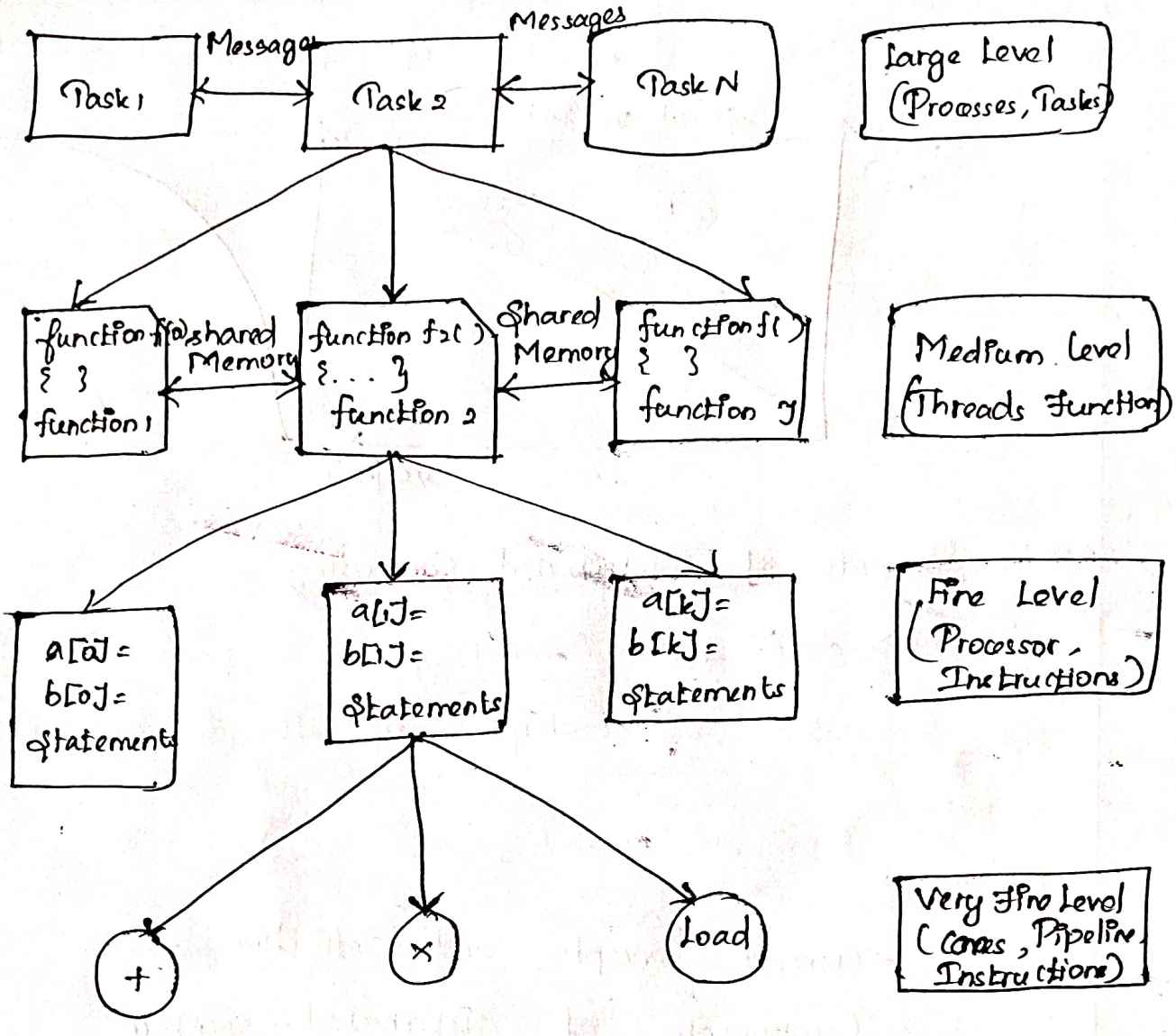
⇒ One processor is configured as Master and all other remaining PEs are designed as Slaves.

Master assigns the jobs to the Slave, PEs

These approaches can be ~~used~~ utilized in different levels of Parallelism :.

4.3.4. Levels of Parallelism :-

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



4.3.5. Laws of Caution.

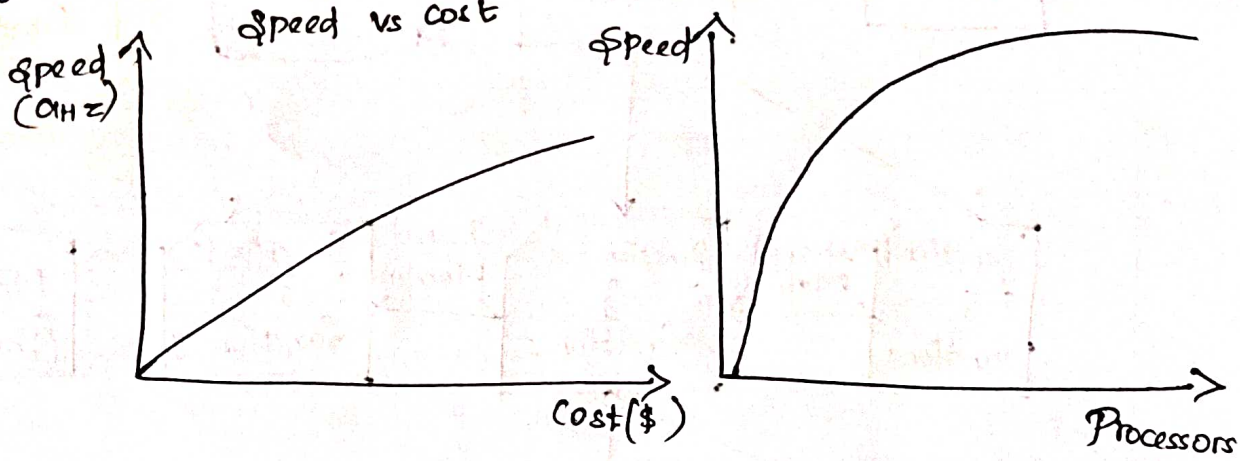
⇒ It can help us to understand how much benefit an application or a software system can gain from parallelism.

⇒ Parallelism is used to perform multiple activities together so that the system can increase its throughput or its speed.

Here are two important guidelines :-

1. Speed of Computation is propotional to the square root of system cost. They never increase linearly.

Speed of Parallel Computer increases as the logarithm of the number of processors (ie $y = k \log(N)$)



4.4. Elements of Distributed Computing :-

⇒ Distributed Computing is performed by the execution of multiple operations by multiple computers.

4 Concepts :-

⇒ General Concepts and Definitions.
⇒ Components of distributed System.
⇒ Architectural styles of distributed Computing.
⇒ Models for inter process communication.

General Concepts and definition :-

A distributed system is a collection of independent Computers that appear to its users as a single coherent system.

Components of Distributed System :-

Layer 1 : (bottom layer)
Layer 2 : Operating System.
Layer 3 : Middleware.
Layer 4 (at top layer) : Applications.

4.4.3. Architectural styles for distributed Computing :-

The architectural styles are classified into two major classes.

1. Software Architectural styles.
   - Relates to the Logical organization of the software.
2. System Architectural styles :-
   - Relates to the physical organization of distributed systems.

4.4.3.1 Software Architectural styles :-

⇒ Based on the arrangement of software components

⇒ They provide an intuitive view of the whole system.

Software Architectural style Categories :-

Category	Most Common Architectural styles.
Data Centered	Repository Blackboard.
Data Flow	Pipe and Filter Batch sequential.
Virtual Machine	Rule Based Interpreter.
Call and Return	Main Program and subroutine call/ top-down systems. Layered System.
Independent Components	Communicating Processes. Event systems.

1. Data Centered Architectures :-

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

Goal: Integrity of data.

Repository	Blackboard
It is characterized by two main components. * Central data structure * Collection of Independent component.	It is characterized by three main components. → Knowledge sources. → Blackboard. → Control.

2. Data Flow Architectures :-

Data flow styles explicitly incorporate the pattern of data flow.

Styles :-

Batch Sequential	Pipe and Filter.
Coarse grained	File grained.
High Latency	Reduced Latency due to the incremental processing of input.
External access to Input	Localized input.
No Concurrency	Concurrency possible.
Non interactive	Interactivity awkward but possible.

3. Virtual Machine Architecture :-

⇒ It contains an abstract execution environment (generally referred as a virtual Machine).

⇒ Popular examples within this Category are Rule Based System, Interpreters, and Command Language processors.

Rule Based	Interpreter :-
Programs are expressed in the form of rules or predicates that hold true	Go through program one by one to interpret to convert into another form.

4. Call and Return Architecture :-

⇒ This identifies all systems that are organized into components mostly connected together by method calls. There are three Categories.

* Top down style :- dereloped with imperative Programming.
* Object Oriented style : Object Programming Models.
* Layered style :- Provides the implementation in different levels of abstraction of the system.

5. System Architectural Styles :-

System architectural styles cover the physical organization of Components and processes over a distributed infrastructure.

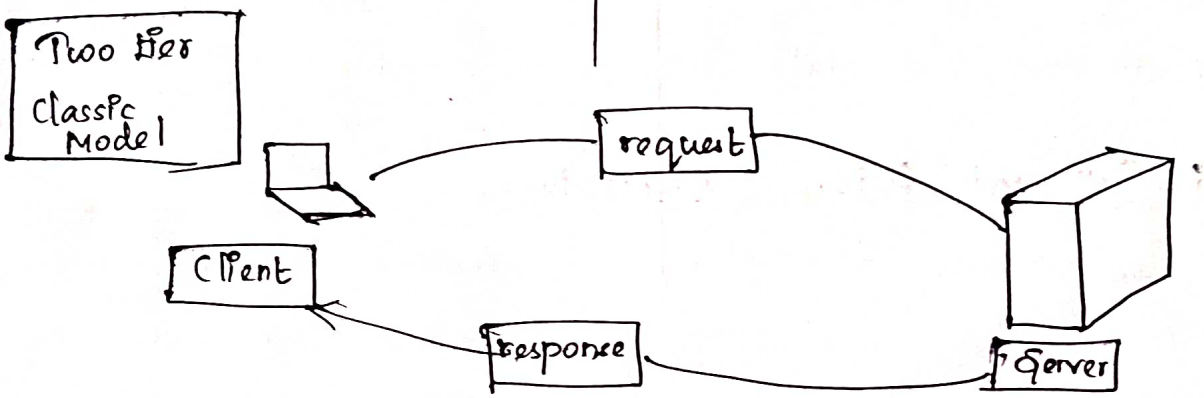
Two fundamental Reference style :-

→ Client / Server

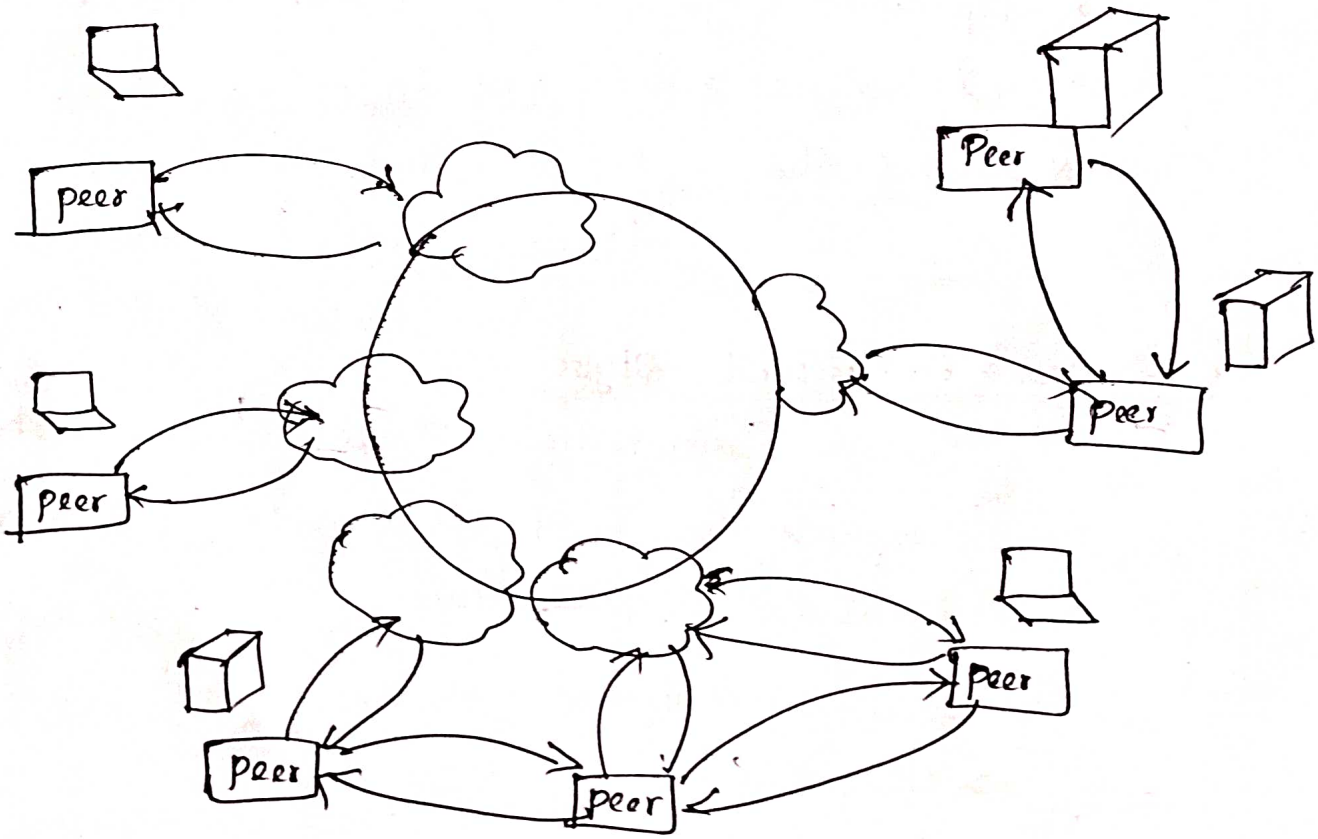
→ Peer to Peer.

Client / Server	Peer to Peer
1. Information and the services can be centralized	1. Information and the services cannot be centralized.
2. The server must be designed to efficiently serve requests coming from different clients.	2. Each peer acts as a server when it processes requests from other peers and as a client when it issues requests to other peers.

Client / Server Architecture Styles:-



Peer - to - Peer Architectural Styles:-



4.4.4. Models for Inter process Communication :-

⇒ IPC is a fundamental aspect of distributed systems design and implementation.

⇒ IPC is used to either exchange data and information or coordinate the activity of processes.

IPC Models :-

1. Message Passing Model. → Point to Point message Model.
   → Public and Subscrib " "
   → Request & reply " "
2. Remote Procedure Call (RPC)
3. Distributed objects.
4. Distributed agents and active objects.
5. Web service.

4.5 Technologies for Distributed Computing :-

4.5.1. Remote Procedure Call
4.5.2. Distributed Object framework.
4.5.3. Service Orented Computing.
4.5.4. Web Services :

4.5.1 Remote Procedure Call :-

⇒ RPC is a fundamental abstraction enabling the execution procedures on clients request.

⇒ The called procedure and calling procedure may be on the same system or they may be on different systems.

⇒ The important aspect of RPC is marshalling and unmarshalling.

4.5.2. Distributed object Framework:

Extend object-oriented programming systems by allowing objects to be distributed across a heterogeneous network.

Examples of Distributed object Frameworks.

⇒ Common Object Request Broker Architecture (CORBA):

⇒ Distributed Component Object Model (DCOM/COM+)

⇒ Java Remote Method Invocation (RMI)

⇒ .NET Remoting.

4.5.3. Service Oriented Architecture (SOA)

⇒ SOA is an architectural style supporting service orientation. It organizes a software system into a collection of interacting services.

⇒ SOA encompasses a set of design principles that structure system development and provide means for integrating components into a coherent and decentralized systems.

There are two major roles within SOA.

* Service Provider.
* Service Consumer.

4.5.4. Web Services:-

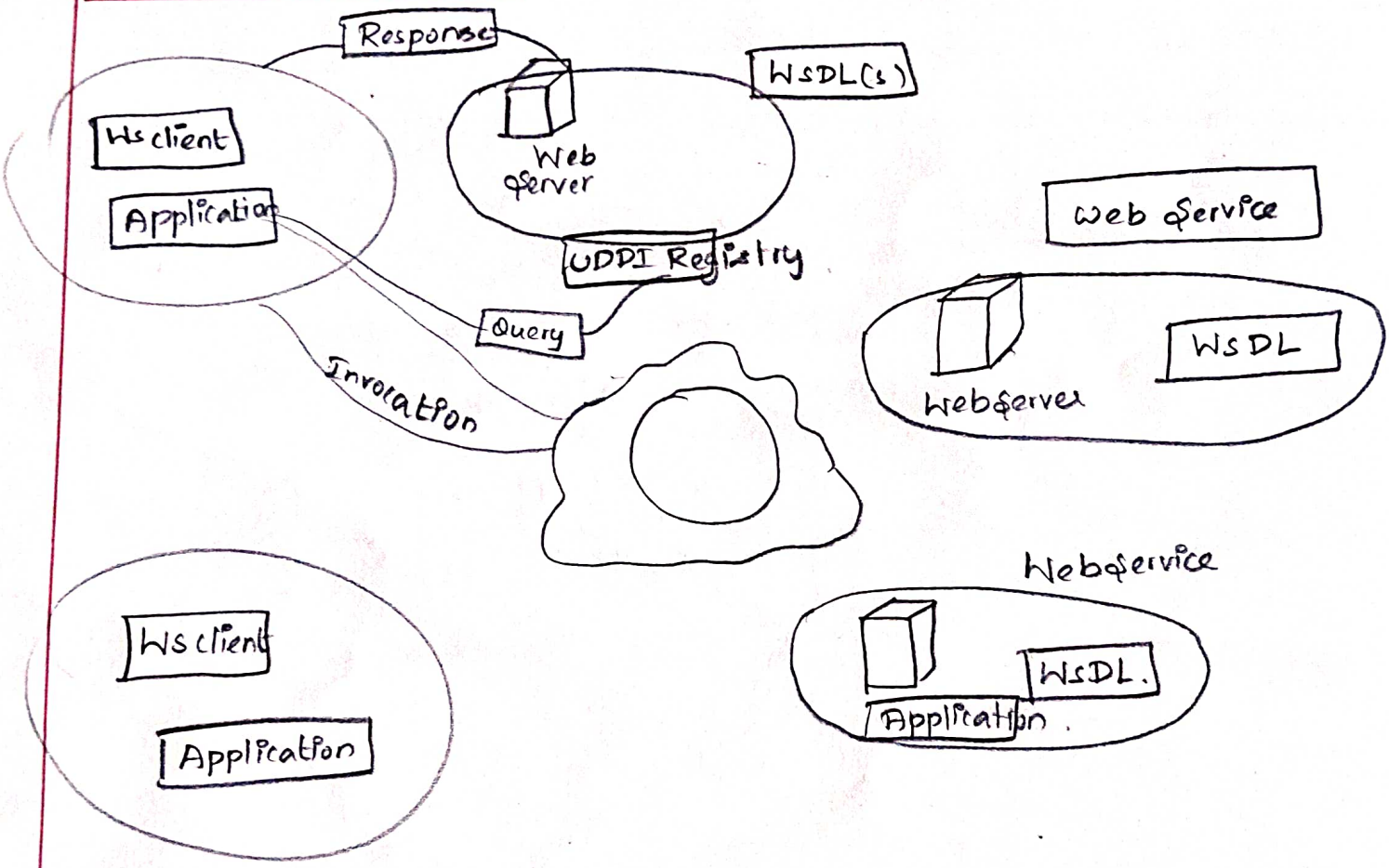
Web services are the prominent technology for implementing SOA systems and applications.

Aspect of Web Services :-

1. They allow for interoperability across different platform and programming languages.

2. They are based on well-known and vendor independent standards such as HTTP, SOAP, WSDL.

Webservice Interaction :-



Cloud characteristics :

Characteristics of Cloud Computing as per NIST :-

→ National Institute of standards and Technology (NIST)

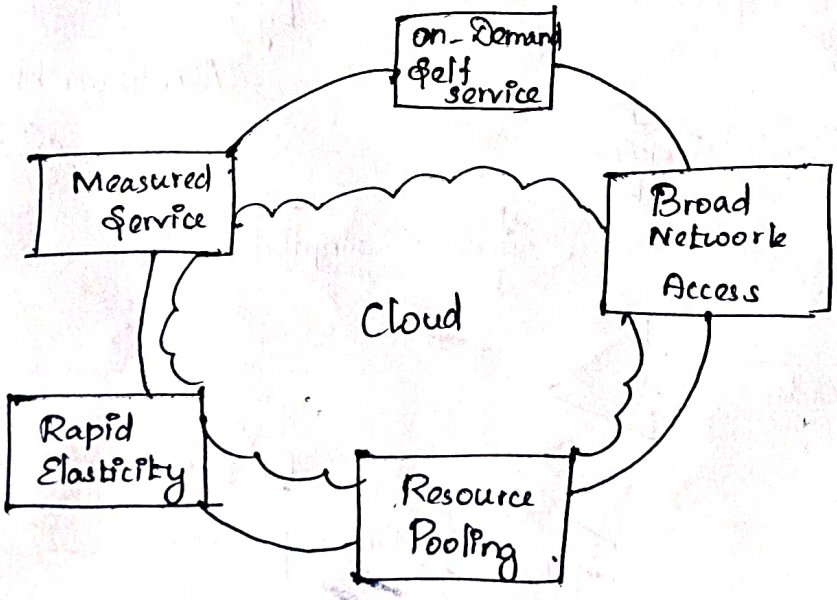
⇒ It is an agency in US.

⇒ NIST is responsible for defining standards in science and Technology.

The computer security Division of NIST has provided a formal Definition and characteristics of Cloud Computing

NIST five Essential Characteristics of Cloud Computing :-

1. On demand self-service.
2. Broad Network access.
3. Resource Pooling.
4. Rapid Elasticity.
5. Measured service



Essential Characteristics of Cloud Computing defined by NIST.

1. On demand - Self-Service :-

The definition from the NIST is,

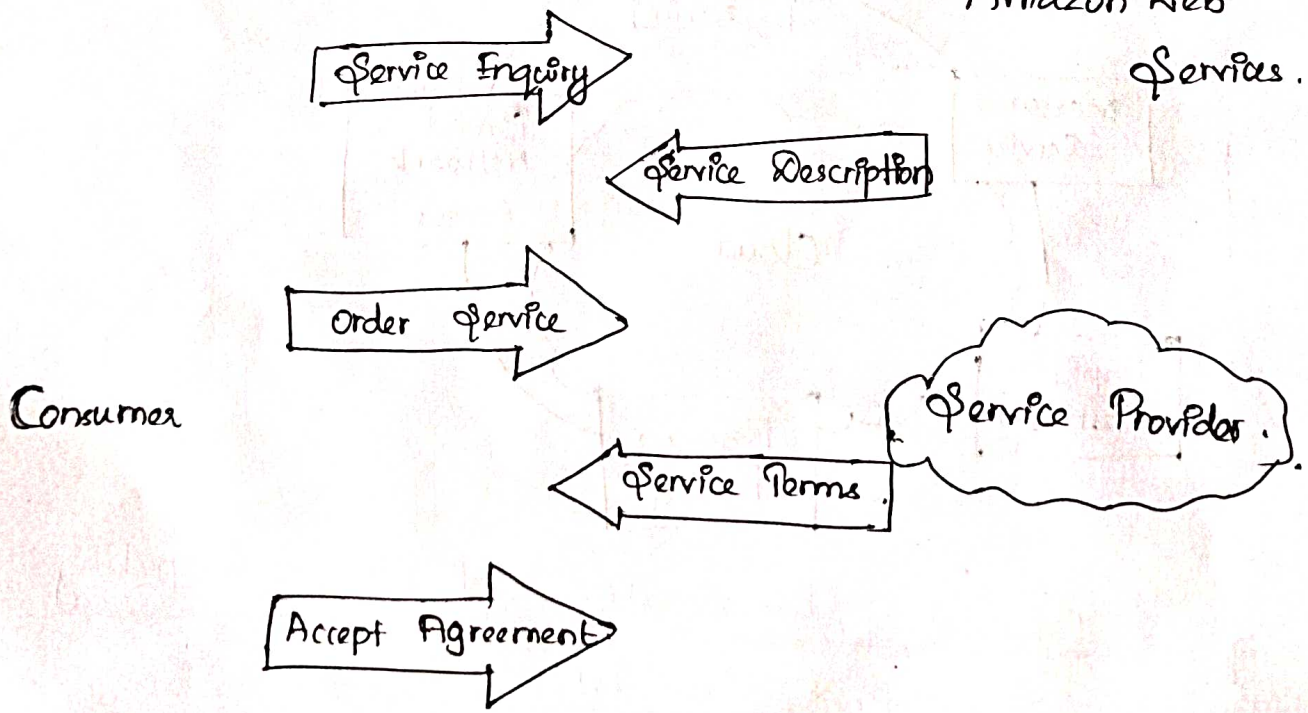
"A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider".

⇒ Cloud Computing provides resources on demand, ie) When the consumer wants it. This is made possible by Self-Service and automation.

⇒ Self Service means the consumer performs all the actions needed to acquire the service himself without need of others instead of going through an IT department. help

⇒ The consumer request is then automatically processed by the cloud infrastructure, without human intervention on the provider's side.

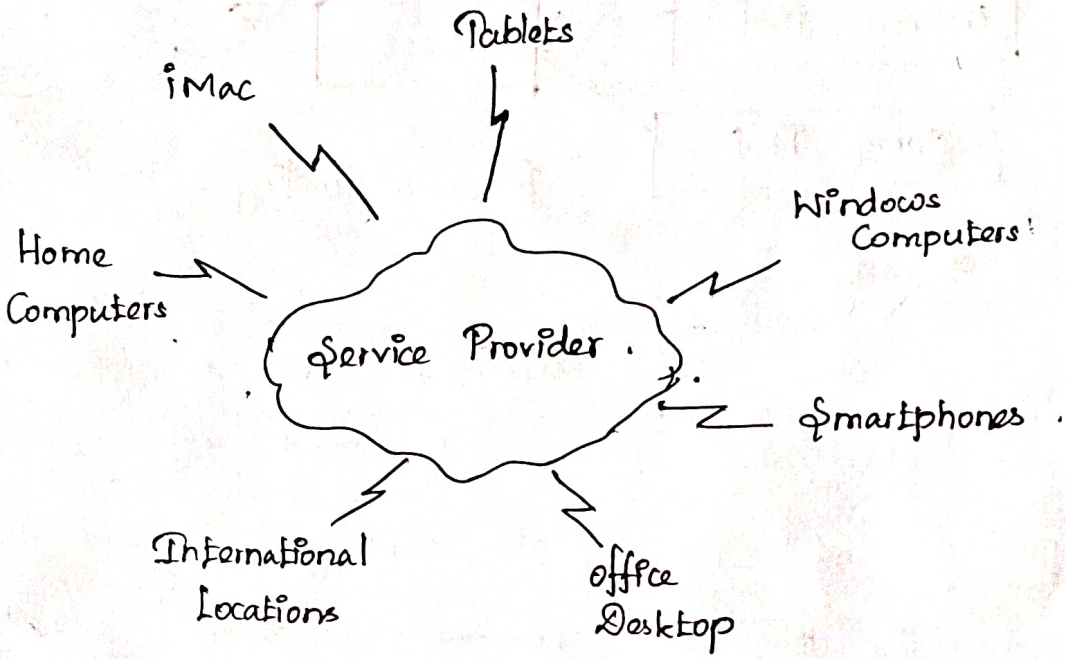
Eg) Amazon Web Services.



On demand Self Service,

2. Broad Network Access :-

=> Cloud Capabilities are available over the network and accessed through standard mechanism that promote use by heterogeneous thin or thick platforms eg) mobile phones, tablets, laptops, workstations.



3. Resource Pooling :-

=> The cloud service provider's computing resources are pooled together to serve multiple Customers using a Multi-tenant Model, with different physical and virtual resources.

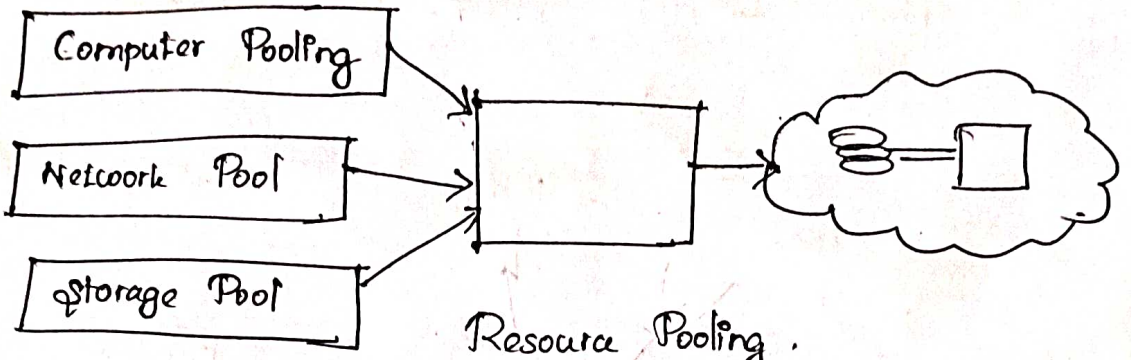
=> These resources are dynamically assigned ~~arranged~~ and reassigned as per Consumer demand.

=> The examples of resources include storage processing, Memory and Network Bandwidth.

These resources are provided in the

3. Location independence Manner

⇒ Where the customer has no control or knowledge over the exact location of the provided resources.

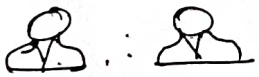


Resource Pooling :

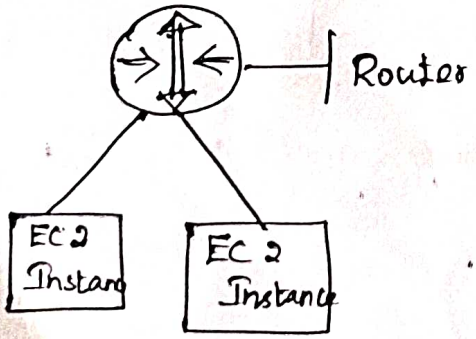
4. Rapid Elasticity :-

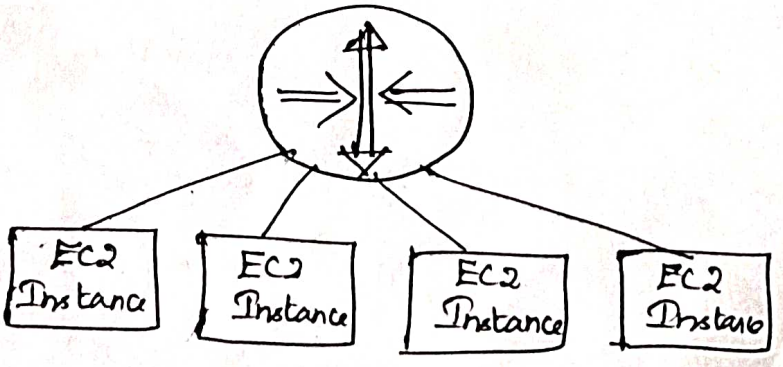
In cloud, the different resource capabilities can be elastically provisioned and released automatically as per demand. To scale rapidly outward and inward the elasticity required. To the customers, the capabilities are available for provisioning appears to be unlimited and can be seized in any measure at any time.

Low demand period.



High demand period.





5. Measured Service :-

⇒ Cloud systems automatically control and optimize the resource use by Consumers.

⇒ They are Controlled by leveraging the metering capability at some level of abstraction appropriate to the type of service. (Eg) Storage, Processing, Bandwidth, Active User Accounts).

⇒ The cloud system provides a mechanism for Measuring the Usage of resources for monitoring, Controlling, and billing purposes. They are reported for providing transparency for both providers and Consumers of the Utilized service.

Apart from that there are some other characteristics of cloud computing are given below.

a) Cloud Computing mostly uses open source REST based API's (Application Programmer Interfaces) builded on Web services that are Universally available and allow users to access the cloud services through Web browser easily and efficiently.

b) Most of the cloud services are Location independent which are provisioned at any time, from anywhere and on any devices through Internet.

c) It provides the agility to improve the reuse of cloud's Resources.

d) It provides the end User computing where users have

their own control on the resources used by them. opposed to the control of Centralized IT servic.

e) It provides Multi-tenancy environment for sharing a large pool of resources to the users with additive features like Reliability, scalability, elasticity, security etc.

Elasticity In cloud :- Ability to quickly scale in/out service.

⇒The cloud ~~compri~~ computing comprises one of the important characteristics called "Elasticity".

Elasticity.

⇒It is very important for mission critical or business critical applications where any compromise in the performance may leads to huge business loss.

⇒So elasticity comes into picture where additional resources are automatically provisioned for such applications to meet the performance requirements and demands.

Elasticity Working way :-

It works when number of user access increases

⇒Applications are automatically provisioned the extra computing, storage and network resources like CPU, Memory, storage or bandwidth.

When a smaller number of user access,

⇒It will automatically decrease those as per requirement.

The Elasticity in cloud is a feature is associated with Scale-out Solutions (horizontal scaling)

⇒ Which allows for resources to be dynamically added or removed when needed.

⇒ It is generally associated with public cloud resources which is commonly featured in as per-use or pay-as-you-go services.

Elasticity :-

⇒ It has the ability to grow or shrink infrastructure resources (like Compute, Storage or network) dynamically as needed to adapt to workload changes in the application in an automatic manner.

⇒ It makes make maximum resource utilization which result in savings in infrastructure cost overall.

⇒ Depends on the environment elasticity is applied on resources in the infrastructure that is not limited to hardware, Software, Connectivity, Oos and other policies.

Eg) Elasticity is mostly used in IT organizations where during the peak hours when all the employees are working on cloud. (between 9 Am to 9 Pm) ⇒

⇒ the resources are scaled into the highest mark. While during non-peak hours.

⇒ When limited employees are working (between 9 PM to 9 A.M)

⇒ The resources are scaled out to lowest mark.

Where a discrete bill is generated for Low usage and high usage which saves the huge cost.

Benefits / Pros of Elastic Cloud Computing :-

1. Cost Efficiency :-

⇒ Cloud is available at much cheaper rates than traditional approaches.

⇒ By using cloud solution Companies can save licensing fees as well as eliminate overhead charges such as the (cost of data storage, software Updates, management.)

2. Convenience and Continuous availability :-

⇒ Cloud makes easier access of shared documents and files with view and modify choice.

⇒ Moreover it guaranteed continuous availability of resources and In case of system failure.

3. Backup and Recovery :-

⇒ The cloud providers offers reliable and flexible backup / recovery solutions.

4. Cloud is environmentally friendly :-

⇒ (The cloud is more efficient) than the typical IT infrastructure and (it takes fewer resources to compute) thus saving energy.

5. Scalability and Performance :-

Scalability is a built in feature for cloud deployments. Cloud instances are deployed automatically only when needed and as a result enhance performance with excellent speed of Computations.

6. Increased Storage Capacity :-

The cloud can accommodate and store much more data compared to a personal computer ⇒ offers almost unlimited storage capacity.

Disadvantages/Cons of Elastic Cloud Computing :-

1. Security and Privacy in the cloud :-

⇒ Security is the biggest concern in cloud computing. (Companies essentially hide their private data and information over cloud as remote based cloud infrastructure is used.)

⇒ (The cloud service provider to manage, protect and retain data confidential.)

2. Limited Control.

⇒ Since the applications are running remotely companies. So that have a limited control over the function and execution of the hardware & software.

3. Dependency and Vendor Lock-In :-

⇒ One of the major drawback of cloud computing is the implicit depency on the provider. It is also called "Vendor_Lock-in".

(ie In cloud computing Vendor is very important, because difficult to migrate data from old provider to new).

4. Increased Vulnerability :-

⇒ Cloud based solutions are exposed on the public internet therefore are more vulnerable target for malicious users and hackers.

Eg) Major cloud Providers (AWS, SalesForce, Azure) are great examples of cloud Elasticity.

On-Demand Provisioning :-

The definition from the NIST is, "A consumer can unilaterally provision Computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider."

⇒ Cloud computing provides resources on demand, ie) when the consumer wants it. This is made possible by self-service and automation.

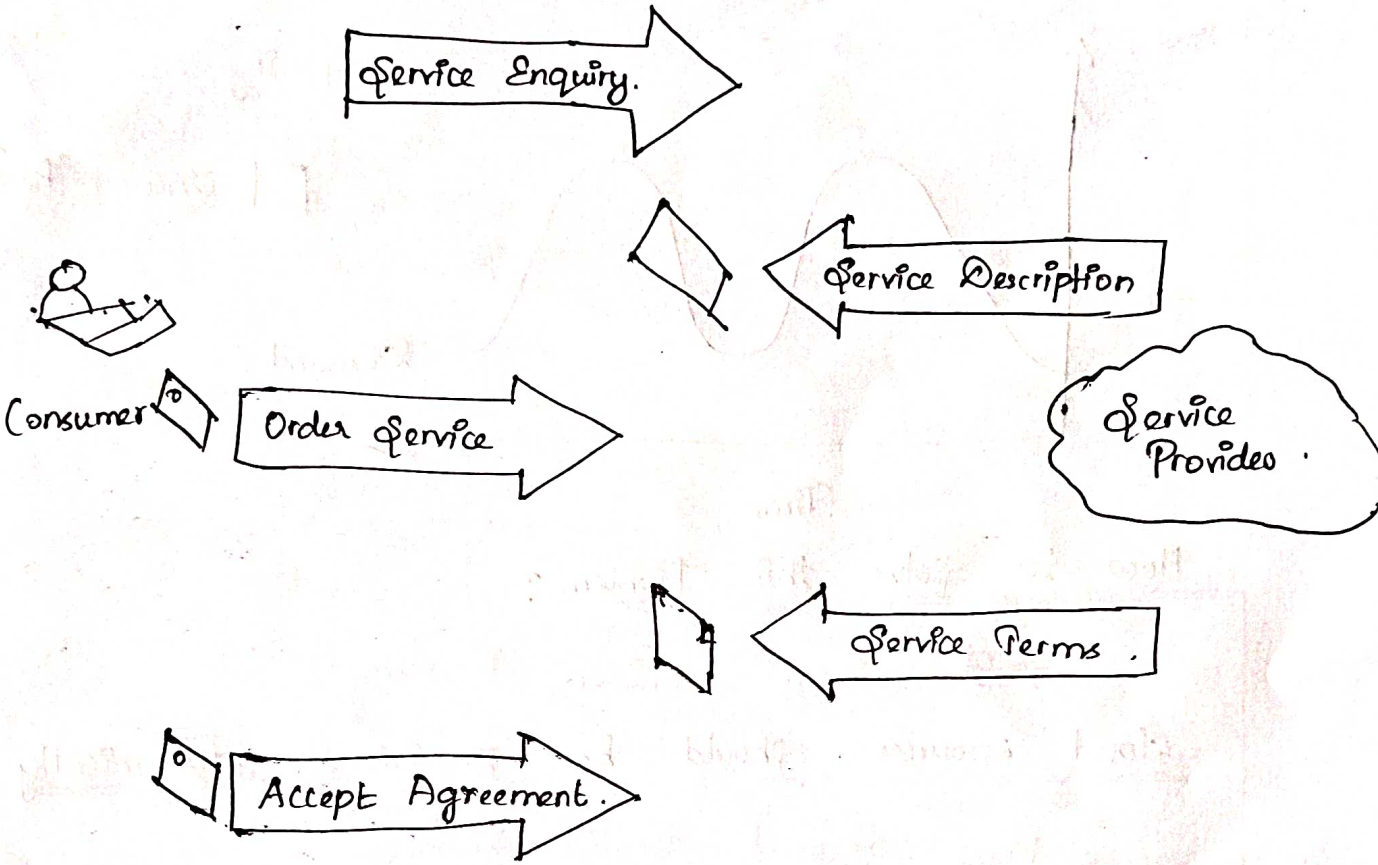
Self service :-

⇒ Means that the Consumer performs all the actions needed to acquire the service himself, instead of going through an IT department.

Automation :-

The Consumer's request is then automatically processed by the cloud infrastructure, without human

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



On-demand self service.

Cloud On-Demand Self Service Examples:-

⇒ Amazon Web services will automatically provision a virtual machine running in the AWS cloud.

Resource Provisioning:-

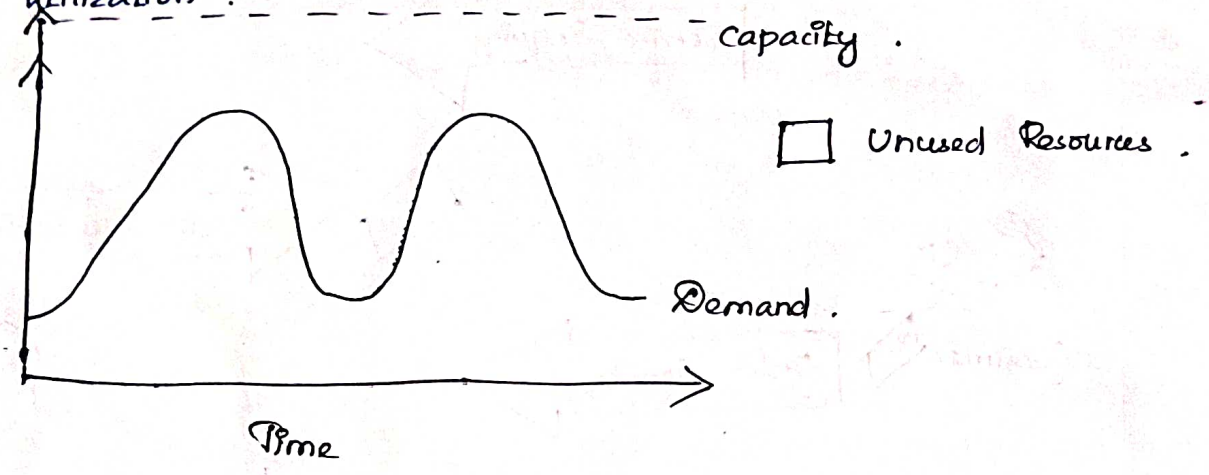
Resource Provisioning is the allocation of a cloud provider's resources and services to a customer.

The growing catalog of cloud services that customers can provision includes Infrastructure as a

as a service, software as a service and platform as a service, in public or private cloud environments.

Why On-demand Provisioning?-

⇒ Overestimate system utilization which result in low utilization.



How to solve this Problem?

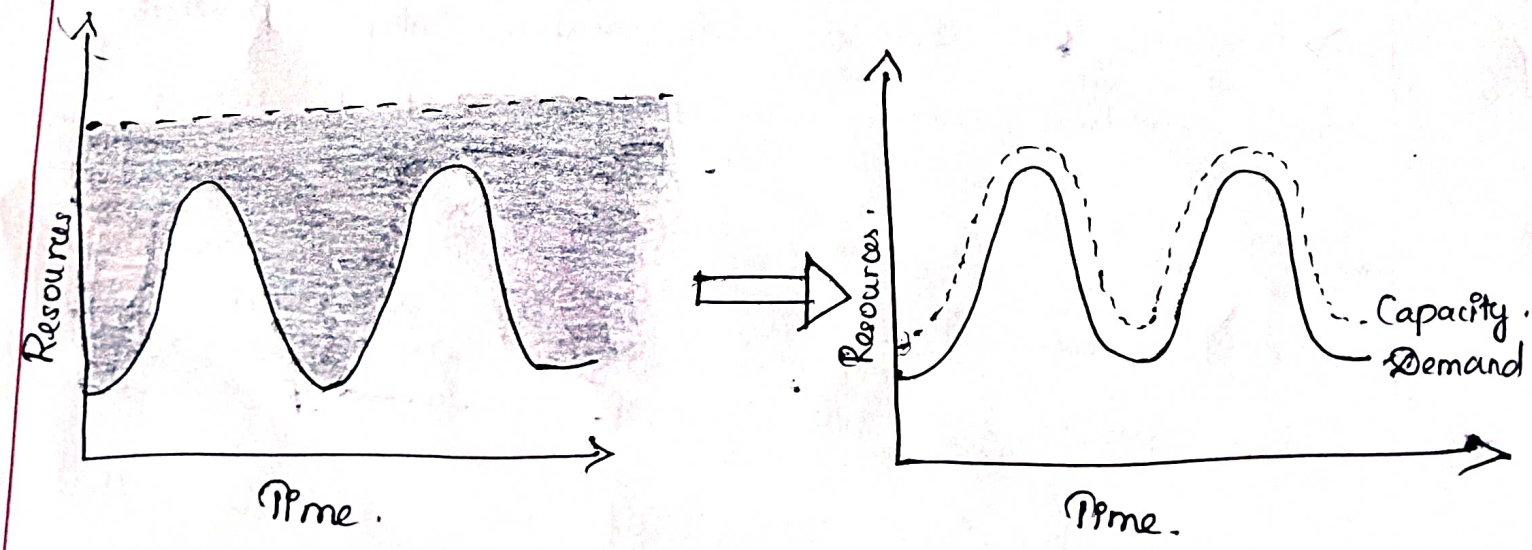
⇒ Dynamically Provision Resources.

Cloud resources should be provisioned dynamically.

⇒ Meet seasonal demand variations.

⇒ Meet demand variations between different industries.

⇒ Meet burst demand for some extraordinary events.



Advantages: -

* Cost effective.
* Efficient Resource Usage.
* Scalability and Performance.
* 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 Architectures: (SOA)

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

SOA Definition:-

"SOA, service oriented architecture, can best 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 heterogeneous applications and technologies".

Aim of SOA:-

To make service interoperability extensible and effective.

Architecture styles (2) of SOA :-

* Loose Coupling
* Published Interfaces.
* Standard Communication Model.

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

"SOA is a form of a distributed systems' Architecture" characterized by :

Logical View :-

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

⇒ The service is defined as the message exchanged between provider and requester.

Message Orientation :-

⇒ The internal structure of SOA such as 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 the service.

Granularity services :-

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

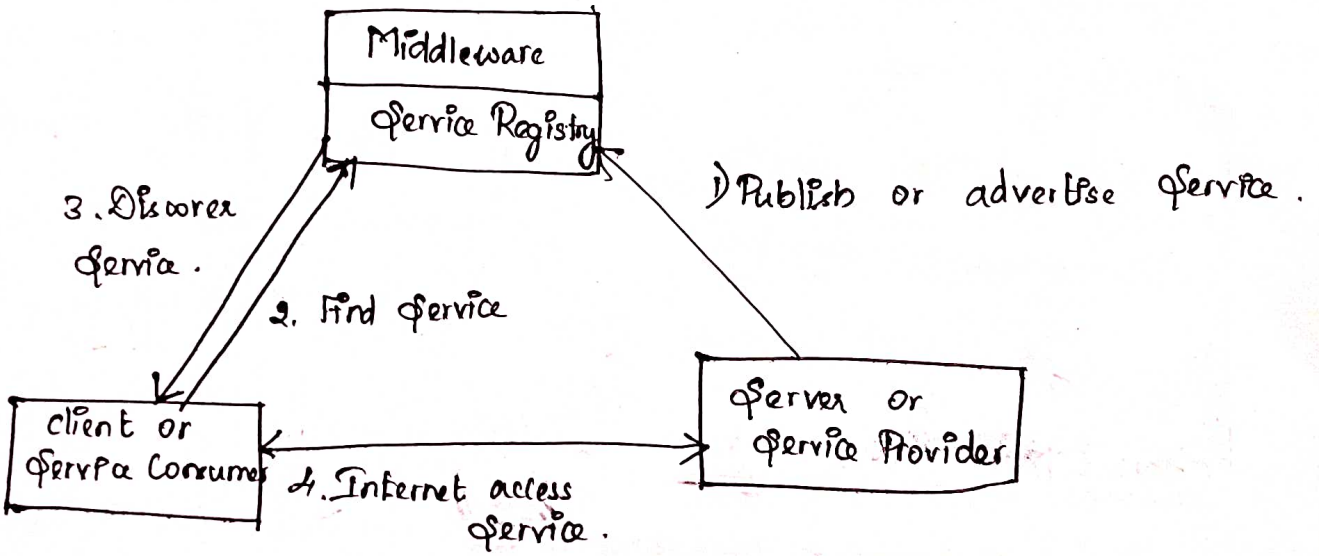
Network Orientation services :-

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.

Architecture of SOA :-



⇒ The SOA provides methods for design, deployment and management of services that are accessible over the network and executable.

Architecture of SOA has three components namely,

⇒ Service Providers, Service Consumers, & Service Registry.

Service Provider is responsible for publishing the services into a registry and provides access to those

using API and interfaces for the consumers.

Service Consumer :- is responsible for invoking and accessing the services published by provider through standard interfaces and API s.

⇒ Whenever the service consumer invokes a servia, initially it has to find it inside service registry using interfaces.

⇒ If it is found in registry, then the discovery details are provisioned to the consumer through which consumer can access the service from service provider.

⇒ The service registry stores the reference of services published by provider and allows consumers to locate and access those using references.

Characteristics of SOA :-

⇒ Provides interoperability between the services.

⇒ Provides Methods for servia encapsulation, service discovery, service Composition, service reusability and service integration.

⇒ Facilitates QoS (Quality of service) through service contract based on service Level Agreement (SLA)

⇒ Provides Loosely coupled services.

⇒ Provides location transparency with better scalability and availability.

Rest and Systems of Systems :-

Two Major Service-Oriented Architecture Styles :-

REST (REpresentational State Transfer.
WS (Web Service)

REST – Representational State Transfer.

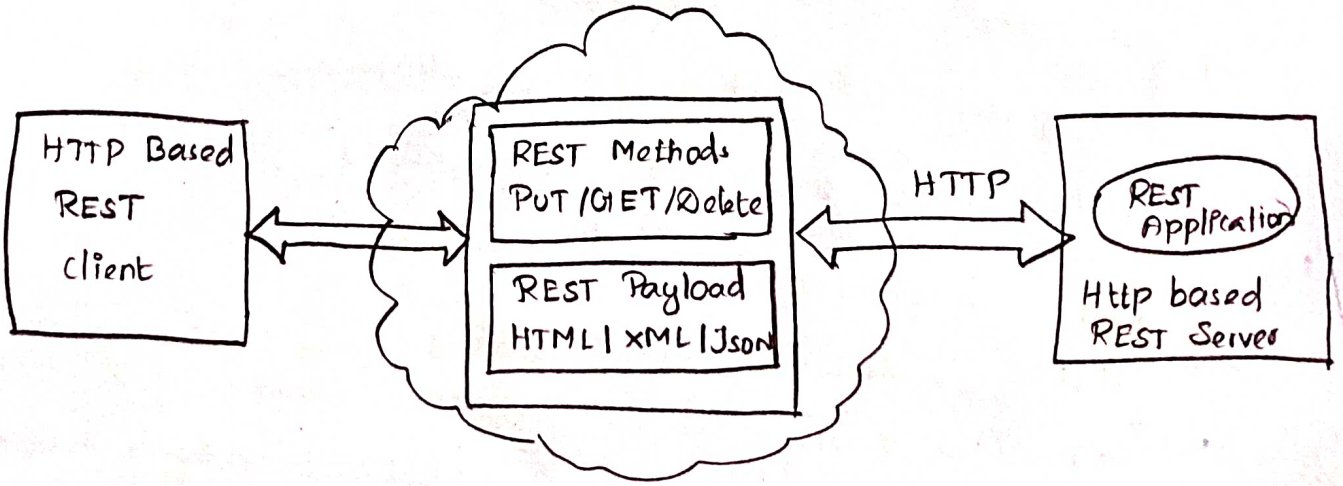
What is REST?

REST is a software Architecture style for distributed systems that defines a set of constraints to be used for creating Web based services.

REST architecture style introduced by Roy Thomas Fielding. REST is useful for designing distributed hypermedia systems, such as the World Wide Web. It is being used by the enterprises such as Google, Amazon, Yahoo!, Facebook and Twitter.

Advantages :-

* Simplicity
* REST services can be easily published and consumed by clients.



REST interaction between User and Server in HTTP

RESTful applications use HTTP requests to post data (create & and/or update)

read data (eg: make queries)

delete data &

Resource Identification through URIs :-

⇒ In Restful Web services, the set of resources are often exposed by the publishers over the internet which are accessed by the clients through interaction mechanism.

⇒ Each resource in a REST has a unique name identified by a Uniform Resource Identifier (URI)

⇒ URI is utilized for giving a global addressing tending to resources which are involved in an interaction between components and facilitates servia discovery.

i) Uniform, Constrained Interface :-

Interaction with RESTful Web services is done via the HTTP standard, Resource are manipulated using a fixed set of four CRUD (Create, read, update delete operations : PUT, GET, POST, DELETE

PUT → Creates a new resource

DELETE → destroy a resource

GET → Retrieves the current state of a resource

POST → Transfer a new state onto a resourc.

Self Descriptive Messages :-

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

⇒ In REST, resources can be accessed in a variety of standard formats (Eg. HTML, XML, plain text, PDF, JPEG, JSON .. etc)

⇒ Meta data about the resource can be used for cache control, transmission error detection, authentication (or) authorization and access control

⇒ Whenever a user types http://www.example.com in the address bar of their Web browser, the browser sends the following request.

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

This method is self-descriptive because it told the server what HTTP method was used.

Stateless Interactions :-

The REST interactions are "stateless"

The server doesnot store any state about the client session on the server side. This restriction is called statelessness.

The client is responsible for sending any state information to the server whenever it is needed.

Advantages:-

1. Increased Scalability:-

Any server can handle any request because there is no session related dependency.

2. Less Complex:-

By removing all server-side state synchronization logic.

3. Improves Visibility:-

Disadvantages:-

Decrease the network performance by increasing the repetitive data (per-interaction overhead)

REST → It is simple, lightweight in nature and can be integrated with HTTP. So the RESTful web services be an alternative to SOAP stack.

Advantages of REST:-

1. Lightweight Infrastructure.
2. Inexpensive
3. Easy to adopt.
4. REST support for caching, Clustering, and Load balancing.

WADL - Web Application Description Language.

WADL is a XML vocabulary to describe RESTful web services, enabling them to be discovered and accessed immediately by potential clients.

Restlet :

It is a lightweight framework that implements REST architectural elements such as resources, representation, connector and media type for any kind of RESTful system, including Web services.

In a Restlet framework both the client and the server are components, components communicate with each other via connectors.

REST Architectural Elements :-

REST Elements	Elements	Example.
Data Elements	Resource	The intended Conceptual target of a hypertext ref.
	Resource Identifier	URL.
	Representation	HTML document, JPEG image, XML etc.
Connectors	client	libwww, libwww-perl.
	server	libwww, Apache API.
Components	Origin Server	Apache httpd, Microsoft IIS
	Gateway	CGI, Reverse Proxy.

A service which is available over Internet is called Web Service API is Application programming Interface that acts as intermediary between two different applications.

RESTful Web Service in Amazon S3 Interface.

A good example of RESTful Web Service application in high performance computing systems is the Amazon Simple Storage Service (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 contain the metadata

⇒ Meta data are stored in containers called "Buckets"

⇒ Each Buckets are identified by a unique key.

Bucket Purposes :-

The organize the Amazon S3 namespace at the highest level.

Identify the account responsible for storage and data transfer charges.

Play role in access control.

Serve as the unit of aggregation for Usage reporting.

Amazon S3 provides three types of Resources :-

a list of User Buckets.

a particular bucket.

a particular bucket S3 object, accessible through

https://s3.amazonaws.com/{name of Bucket}/{name of object}.

These resources are retrieved, created, or manipulated by basic HTTP standard operations.

GET, HEAD, PUT and DELETE.

GET → Can be used to list buckets created by the users.

HEAD → for getting a specific object's metadata.

PUT → Can be Used for Creating a bucket.

DELETE → For removing a particular bucket or object.

REST Request	REST Response
PUT [bucket_name] HTTP/1.0 Date : Wed, 15 Mar 2011 14:45:15 GMT Authorization: AWS [aws-access_key_id]: [header-signature] Host: s3.amazonaws.com.	HTTP/1.1 200 OK x-amz-id-2: VjzdTviOor..... x-amz-request-id : 91ABCC60F9FC49E7 Date: Wed, 15 Mar 2010 14:45:20 GMT Location : /[bucket_name] Content_length: 0 Connection: keep-alive

A request to Amazon S3 contains.

⇒ A request method, a URI, request headers which contains basic information about the request. and sometimes a query string and request body.

The response contains.

⇒ Status Code, Response headers, and sometimes a response body.

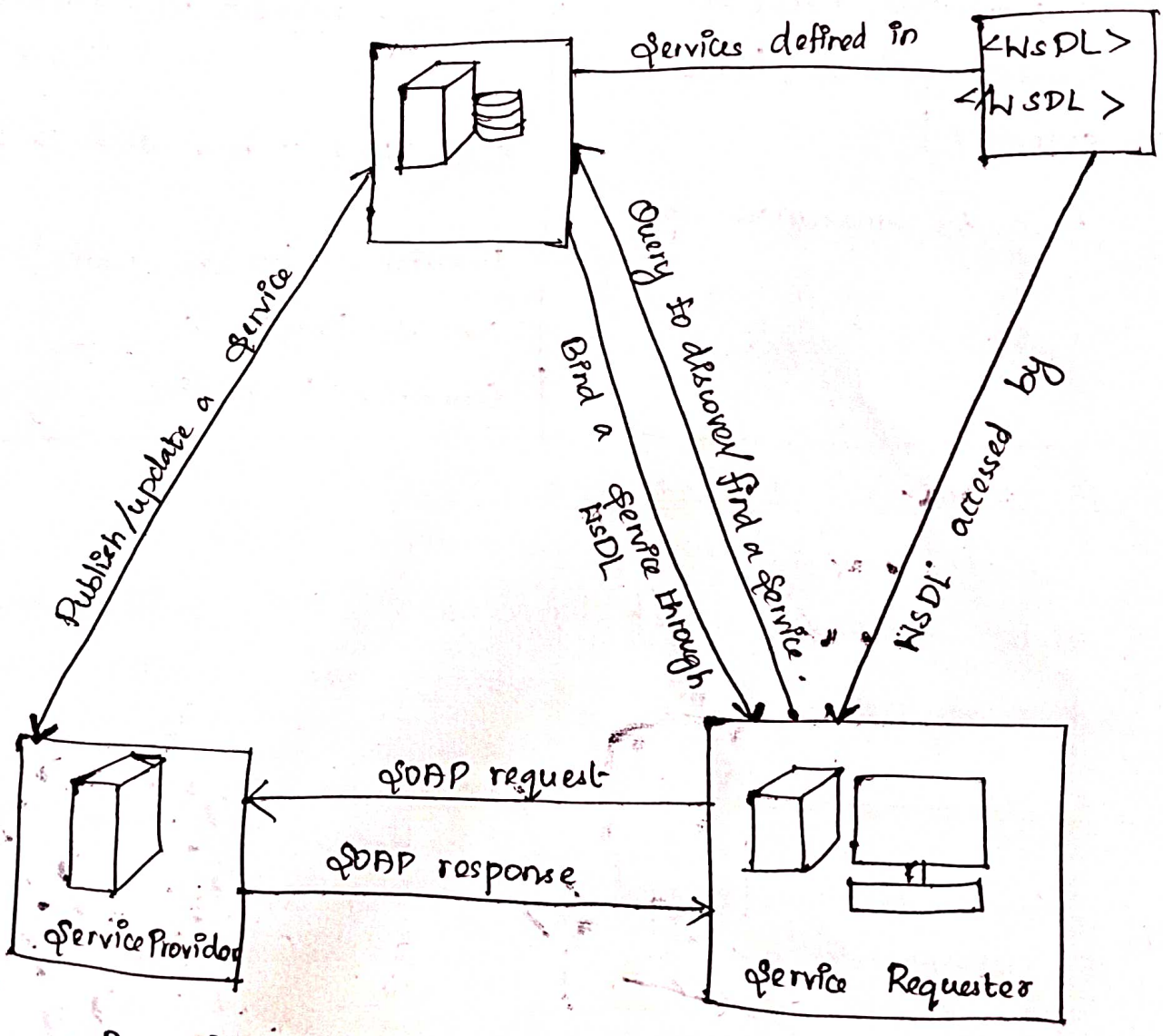
3.

Web Services :-

Web Services :-

The term "Web Service" is often referred to a self-contained, self describing, modular application designed 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.

Functionality of Web Service :-



A simple Web service interaction among provider, user and the UDDI registry.

Web service defined by W3C Working group:-

⇒ Web service is a software system designed to support interoperable machine-to-machine interaction over the network.

Technologies for Web service:-

Simple object Access Protocol. (SOAP)

Web service Description Language. (WSDL

Universal Description, Discovery, and Integration (UDDI)

Simple object Access Protocol :- (SOAP)

⇒ 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.

⇒ It is a platform independent and language independent. By using SOAP you will be able to interact with other programming language applications.

⇒ SOAP message consists of a root element called envelope, which contains a header.

⇒ Header Consists of routing information.

authentication, transaction management, Message parsing instructions, and Quality of service QoS configurations.

⇒ Body Element:
that carries the payload of the message (Message content)

⇒ The content of the payload will be marshalled by the sender SOAP engine and unmarshaled at the receiver side.

⇒ XML schema that describes the structure of the SOAP message.

⇒ SOAP based Web services are also referred to as "Big Web services"

Web Service Description Language (WSDL)

⇒ WSDL is an XML based Language for describing Web services.

⇒ WSDL is a W3C recommendation.

⇒ Web service interfaces are described in a machine executable format using WSDL.

⇒ Interface is a set of operations supported by a Web service.

⇒ It standardize the representation of input and output parameters of its operations as well as the service protocol & binding.

⇒ WSDL enables disparate clients to automatically understand how to interact with a Web service.

Universal Description, Discovery and Integration (UDDI)

⇒ UDDI is a global registry used for publishing the web services by provider and discovering them by consumers.

⇒ The consumer can search the specific web service by its names, identifiers, categories or the specifications implemented by the web service.

⇒ It provides a set of rules for registering and retrieving about a business process and its services.

⇒ It is a directory service where businesses can register and search for web services.

Difference Between Web service Technology and other Technologies such as J2EE, CORBA, and CGI scripting.

⇒ Web service is based on standardized XML, providing a Language-neutral representation of data.

⇒ Using HTTP as the tunneling protocol, web services enables remote communication through firewalls and proxies.

WS-I Protocol Stack:-

specification of SOAP based Web Services.

⇒ A Web service protocol stack is a list of protocols that are utilized to define, find, execute and make Web services collaborate with one another.

⇒ The Web services protocol stack not only covers the specifications of RESTful Web services but also a SOAP-based Web services.

Each layer in a WS protocol stack provides a set of standards and protocols for successful working of Web services.

Bottommost and first Layer in Protocol Stack.

⇒ Which is responsible for transporting a message between applications.

Second Layer in protocol stack is Messaging Layer.

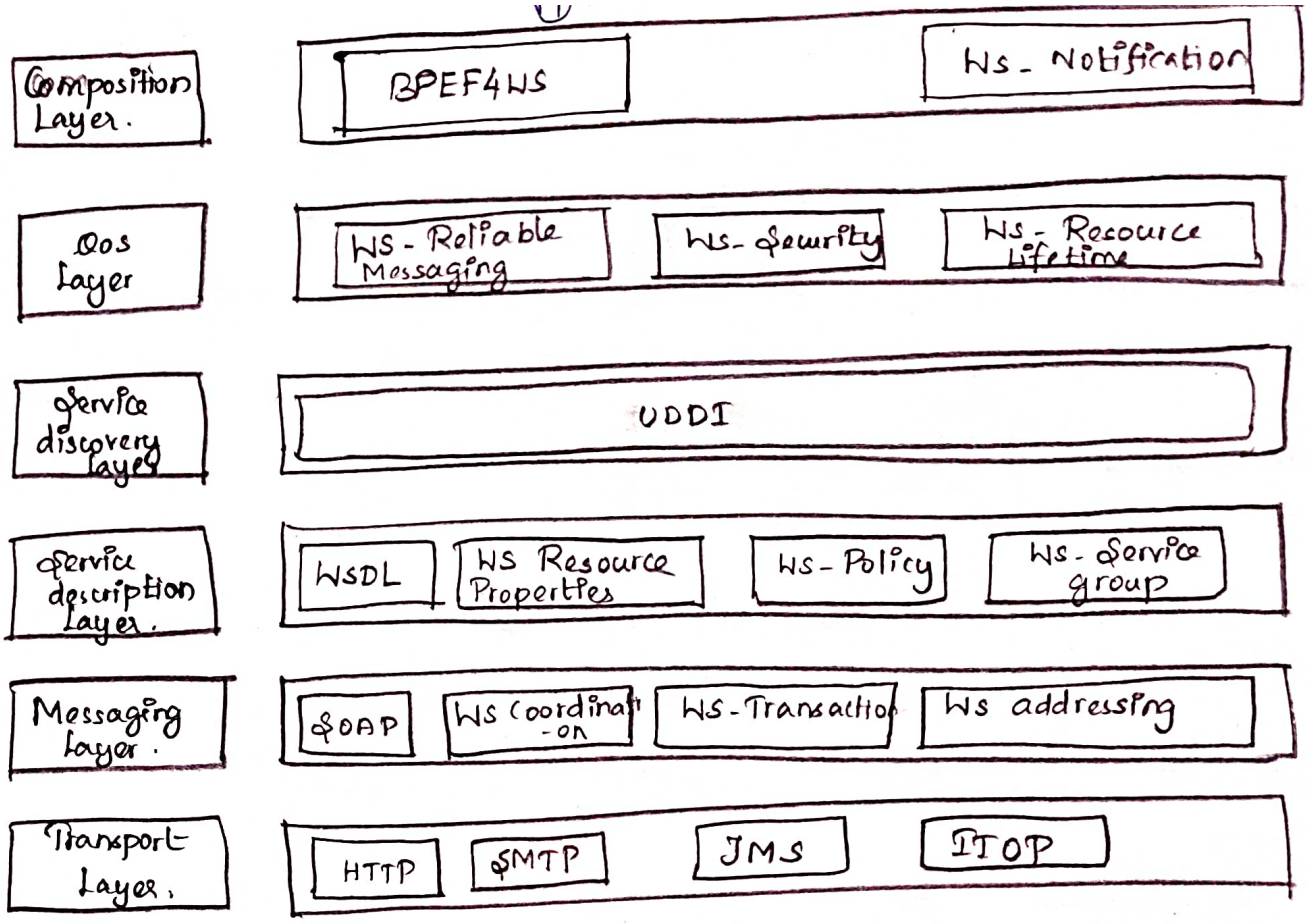
⇒ Which is required for encoding in transit messages in XML.

Third Layer in WS protocol stack is a service Description Layer which is used for describing the public interface to a specific Web service.

The fourth Layer is service Discovery Layer that uses UDDI registry to register aor publish a Web service written by provider and discover by consumer for the invocation.

The fifth Layer is QoS (Quality of Service) Layer.

The sixth Layer is a Composition Layer which is used for composition of business processes.



Web Service Protocol Stack.

WS-* Core SOAP Header Standards.

WS - *

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)

Definition :-

Pub/Sub is an asynchronous communication method in which messages are exchanged between applications without knowing the identify of the sender or recipient.

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

"publish_subscribe" Model links the source and destination through a message bus.

Message / Event Bus :-

⇒ It knowns ~~the~~ 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.

publishes :-

⇒ The producer of the message.

⇒ Responsible for defining the topic of their message.

Subscriber :-

⇒ Receivers ~~the~~ of the Message.

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

Message Filtering :-

In the publish_subscribe model, subscribers typically receive only a subset of the total messages published. The process of selecting messages for reception and processing is called filtering. There are two common forms of filtering :

* Topic_based
* Content_Based

Topic - based pub/sub Model :-

⇒ In a topic_based system, Messages are published to "topics" or named logical channels.

⇒ Subscribers in a topic - based system will reaive

all message published to the topics to which they subscribe.

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

Content - based delivery systems:-

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

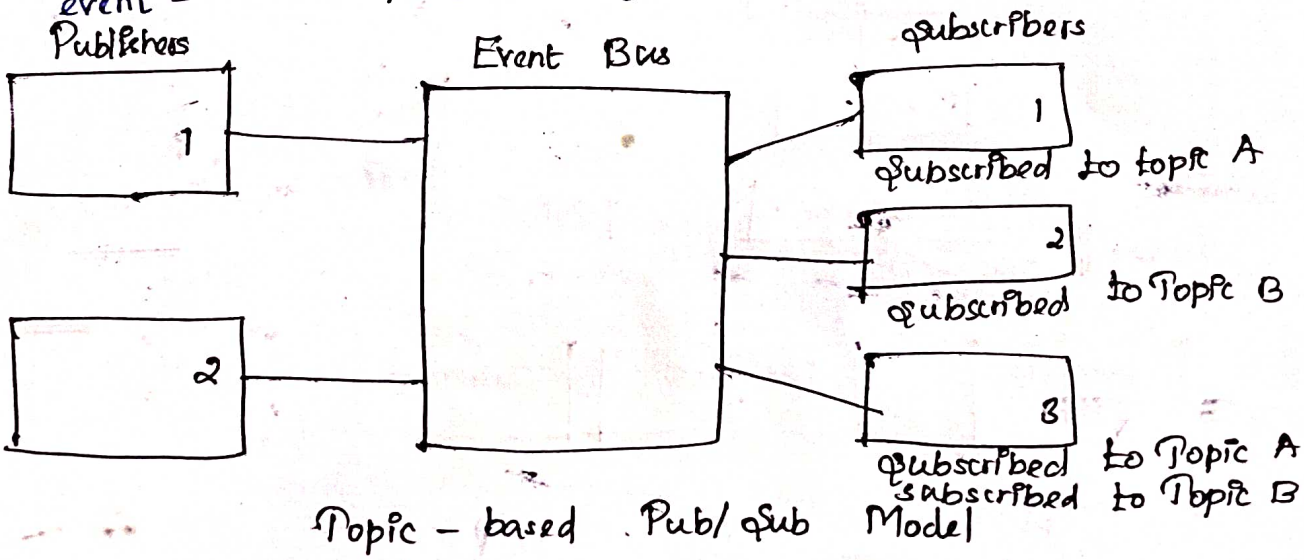
⇒ The subscriber is responsible for classifying the messages.

Relationship:-

⇒ There exists a many - to - many relationship between publishers and subscribers.

Publish - Subscribe Messaging Middleware:-

⇒ Allows straightforward implementation of notification or event - based programming models.



Topic - based Pub/Sub Model

In topic - based Examples:-

The publishers are responsible for defining the

topics of their messages.

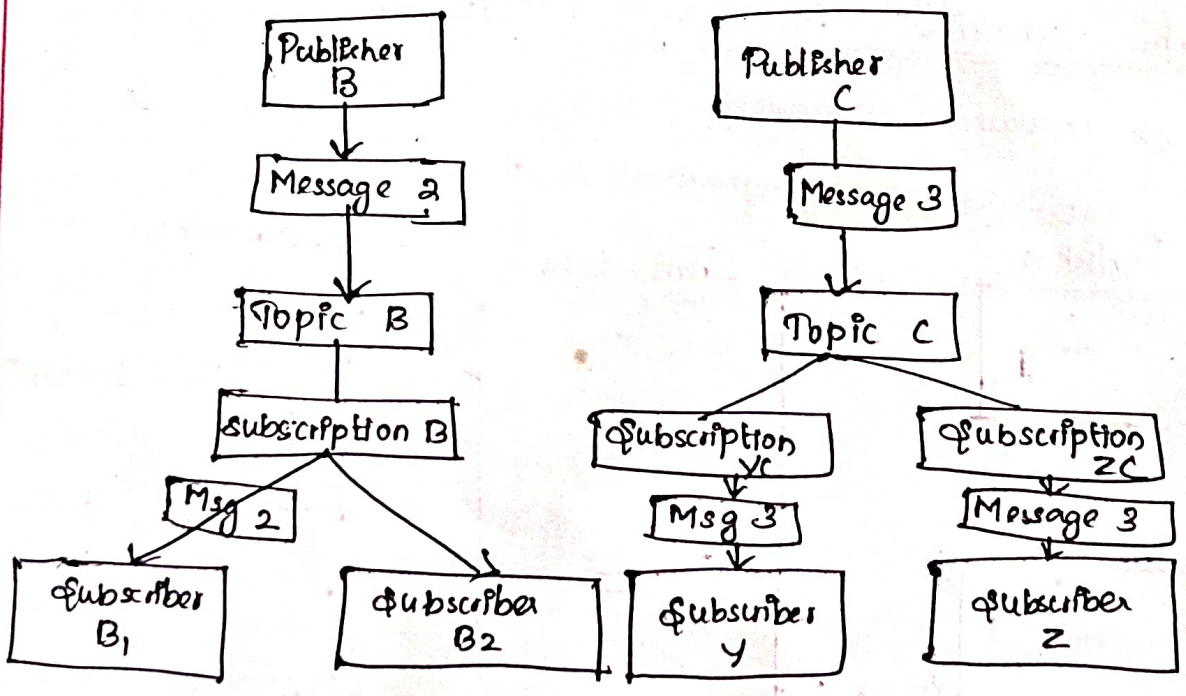
⇒ In the above diagram, any message published with Topic A will be sent to subscriber 1 and subscriber 3.

⇒ Similarly any messages 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 (NEN)

Google cloud Pub/Sub offers both Competing Consumers and publish - subscribe channel semantics, manage through topics (Publish-Subscribe) and subscriptions (Competing Consumers) as illustrated in this diagram.



Point-to-Point

Publish-Subscribe

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 Architectures: (SOA)

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

SOA Deffnition:-

"SOA, service oriented architecture, can best 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 heterogeneous applications and technologies".

Aim of SOA:-

To make service interoperability extensible and effective.

Basics of Virtualization :- ①

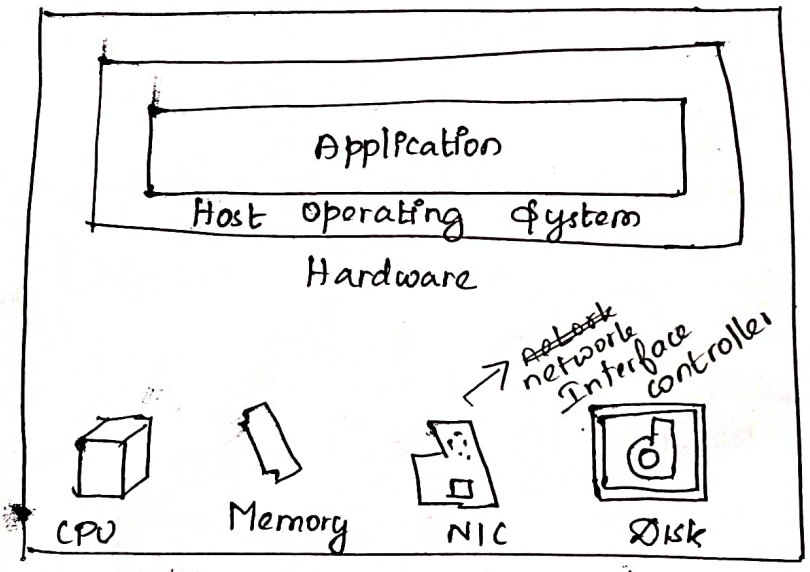
⇒ Virtualization is a computer architecture technology by which multiple Virtual Machine (VMs) are multiplexed in the same Hardware machine. The idea of virtual machine can be dated back to the 1960s.

Eg) : Running Ubuntu on Windows OS.

Host OS → Windows OS.

Guest OS / Virtual Machine (VM) → Ubuntu.

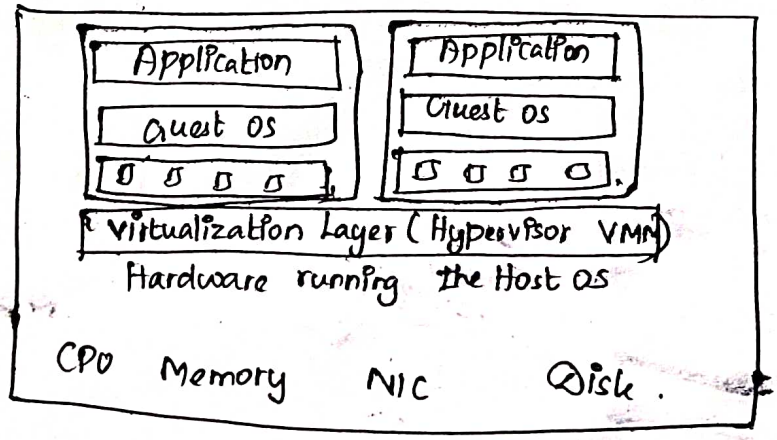
Before Virtualization



→ Network Interface Card
→ Network Adapter
→ LAN adapter

Traditional Computer.

After Virtualization :-



After Virtualization :-

⇒ Different applications of guest operating system can run on the same hardware, independent of the host os.

⇒ Virtualization is done by software / Middleware called a virtualization Layer.

⇒ This virtualization Layer is known as hypervisor or Virtual Machine Monitor (VMM)

Virtualization :-

⇒ It is a process of creating a virtual machine over existing operating system and hardware.

Host Machine :-

⇒ The machine on which the Virtual Machine is created.

Guest Machine :-

⇒ Virtual machine referred as a guest machine.

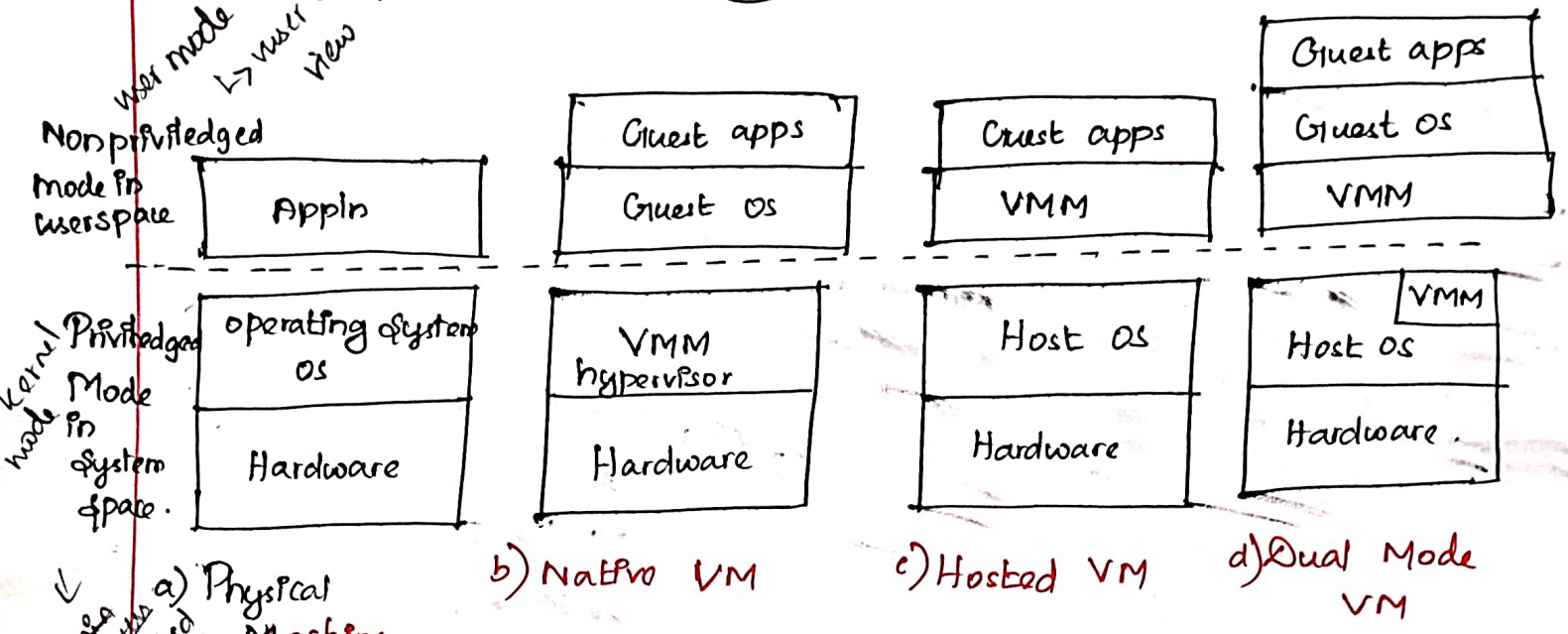
Hypervisor :-

⇒ Hypervisor is a firmware (→ basic machine instructions that allow h/w to functio) or Low-level program that act as a Virtual Machine monitor / Manager.

⇒ It is a middle Layer between the host machine and virtual machine.

Virtualization Models / Architecture:-

(3)



a) Physical Machine

b) Native VM

c) Hosted VM

d) Dual Mode VM

Privileged Mode - Protected access
→ authorized user only accessed
→ Edit also
→ configuration possible

Native VM:-

⇒ A VMM (hypervisor) runs in priviledge mode.

⇒ This 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 need not 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:-

⇒ To enhance resource sharing by many users.

⇒ To improve computer performance in terms of

⇒ resource Utilization

⇒ Application flexibility.

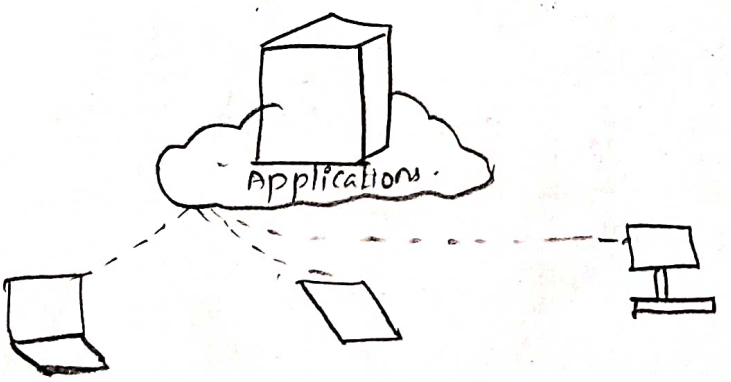
Types of Virtualization :-

1. Application Virtualization.
2. Desktop Virtualization.
3. Hardware Virtualization.
4. Storage Virtualization.

1. Application Virtualization: (App virtualization)

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

⇒ The applications are virtualized and delivered from a server to the end user's device, such as laptops, smartphones, and tablets.



Accessing Remote Application.

2. Desktop Virtualization:

⇒ The virtualization of the desktop is referred to as Virtual Desktop Infrastructure (VDI)

⇒ Desktop operating system (OS), such as window 7, will run as a virtual Machine on another computer.

Advantages :-

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

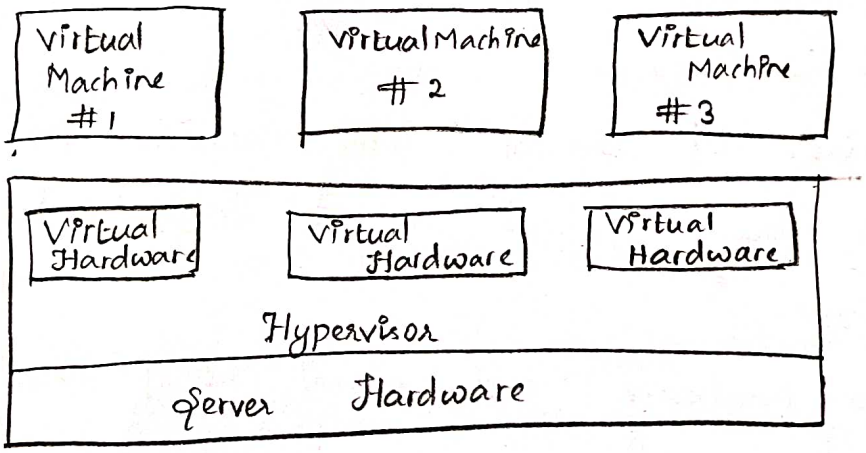
2) Lowers the cost of software licensing and updates.

3. Maintenance and patch management → Process of applying updates 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 platform is also called server virtualization.



Types of Hardware Virtualization :-

1. Full Virtualization :-

⇒ The hardware architecture is completely simulated.

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

Ex) VM_Ware

2. Para Virtualization :-

⇒ Guest OS need modification to run any

any application (6)

Ex) Xen

Network Virtualization :-

⇒ It is the process of combining hardware network resources and software network resources into a single administrative unit.

⇒ The end point of network virtualization is the virtual Network.

Virtual Network :-

Virtual Networks are classified into two parts.

1) External
2) Internal.

External Virtual Network :-

⇒ Consists of several local networks that are administrated by software as a single entity.

Building blocks :-

⇒ Switch hardware and virtual Local area ~~Network~~ Network VLAN software technology.

Eg) Large Corporate Networks and data Centers.

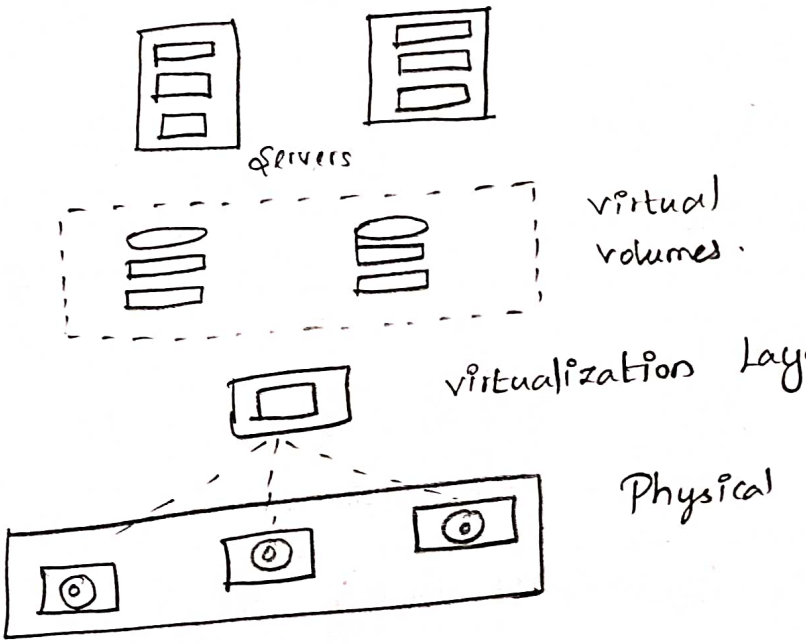
Internal Virtual Network :-

⇒ Consist of host running multiple virtual machines with atleast 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 :- (7)

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

⇒ 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 machines.

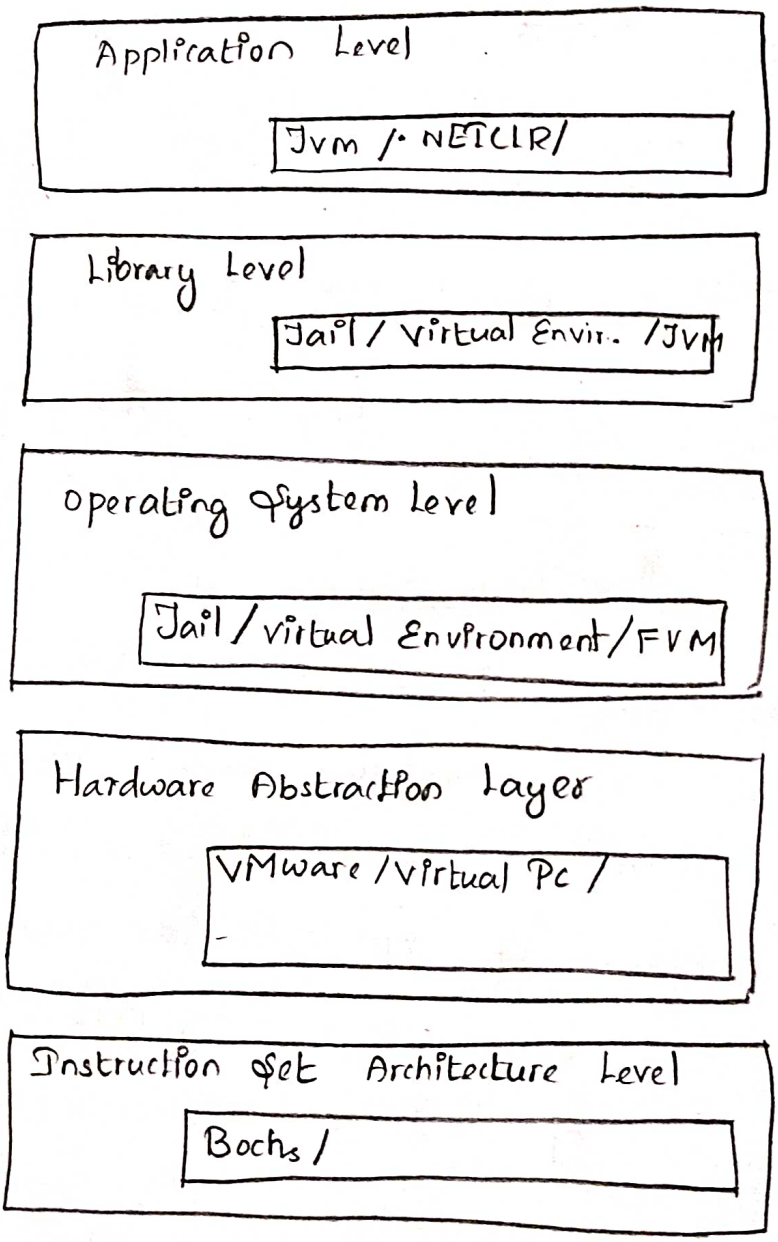
⇒ The purpose of a VM is to enhance resource sharing by many users and improves computer performance in terms of resource utilization and application flexibility.

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 users.

Common virtualization layers include the instruction set architecture (ISA) Level, Hardware Level, operating system, 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 machines.

⇒ One source instruction may require tens or hundreds of native target instructions to perfom 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 right on top of the bare hardware.

⇒ On the one hand, this approach generates a virtual Hardware environment for a VM.

⇒ On the other hand, the process manages the underlying hardware through virtualization.

⇒ The idea is to virtualize a computer resources, such as its processors, 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 single physical server and the os instance to utilize the hardware and software in data centers.

⇒ The containers behave like Real servers.

Library Support Level : (10)

⇒ 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 two good examples of this class of VM.

Relative Merits of Different Approaches ...

	Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation.
1.	ISA	X	XXXXX	XXX	XXX
2.	Hardware level virtualization.	XXXXX	XXX	XXXXX	XXXX
3.	OS level virtualization.	XXXXX	XX	XXX	XX
4.	Runtime library support	XXX	XX	XX	XX
5.	User application level.	XX	XX	XXXXX	XXXXX

"Implementation complexity" implies the cost to implement that particular virtualization level.

"Application Isolation" refers to the effort required to

isolate resources committed to different VMs.

Five 'X' implies → best cases.

One 'X' implies → Worst Case.

VMM Design Requirements and Providers:

Hardware-Level virtualization inserts a layer between real Hardware and traditional Operating System.

This layer is commonly called the virtual Machine monitor (VMM) and it manages the hardware resources of a computing system.

There are 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.

⇒ 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 :-

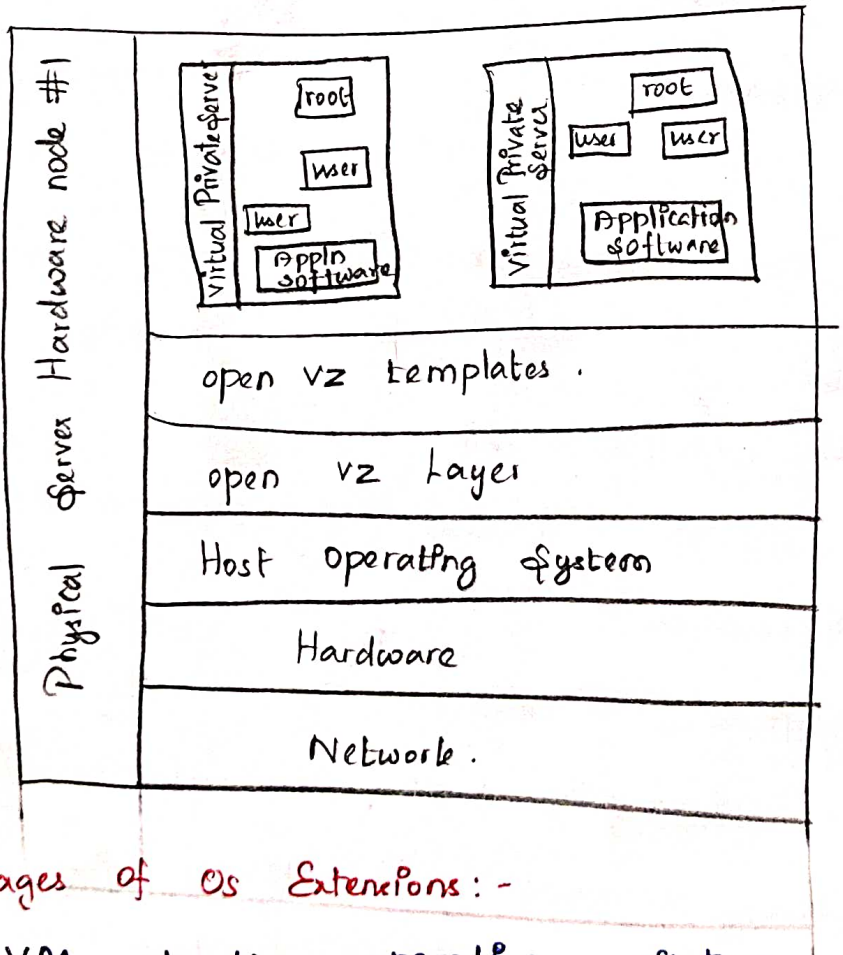
⇒ Cloud computing is transforming the computing Landscape by shifting the hardware and staffing cost of managing computational center to to third parties, just like banks.

Why OS Level Virtualization :-

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

→ Because it overcomes issue such as slow performance and low density.

⇒ OS level virtualization inserts a virtualization layer inside an operating system to partition a machine's physical resources. It enables multiple isolated VMs within a single operating system kernel. This kind of VM is often called a virtual Execution Environment (VE). Virtual Private system (VPS) or simply Containers.



Advantages of OS Extensions :-

(1) VM at the operating system server has minimal startup or shutdown cost, low resume

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.

Disadvantages of OS Extensions :-

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.

⇒ Linux kernel offers an abstraction layer to allow software process to coork 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 ABI 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 sets directly between the physical hardware. 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

~~Unit~~ hardware resources for the guest domain.

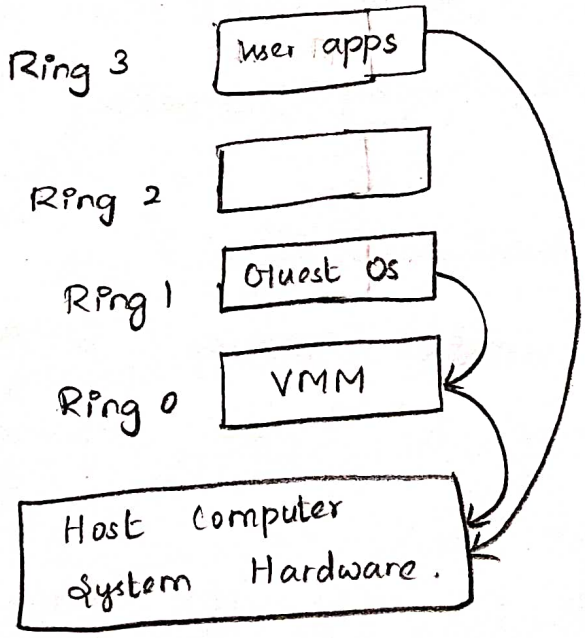
Binary Translation with full virtualization :-

⇒ 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 VMMs.



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 increase the cost of memory usage.

Para Virtualization with Compiler Support :-

⇒ Para virtualization needs to modify the guest operating system. A para virtualized VM provides special API 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 compiler 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 uses 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 deesnot 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, Processor such as the x86 employ a special ~~memory~~ running mode and instructions known as hardware assisted virtualization.

Hardware support for virtualization:-

⇒ Modern operating system and processors permit multiple processes to run simultaneously.

⇒ The VM ware Work station assumes the host based virtualization.

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) Behavior sensitive Instructions.

⇒ Privileged Instructions execute in a previleged 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 previleged 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.

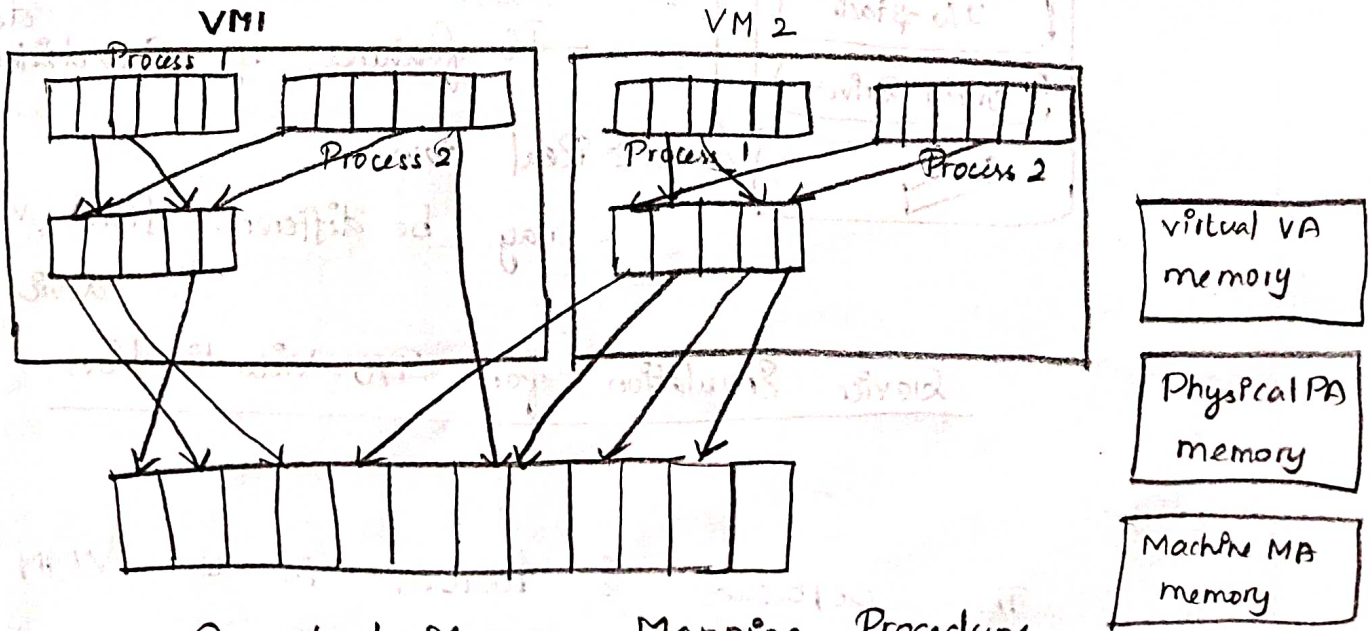
⇒ In a virtual Execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.

VMM respectively :

⇒ Virtual memory to physical memory and physical memory to machine memory.

⇒ MMU virtualization should be supported, which is transparent to guest OS.

⇒ But the guest OS cannot directly access the access actual machine memory.



Two level Memory Mapping Procedure.

Every page table of the guest OSes has a separate

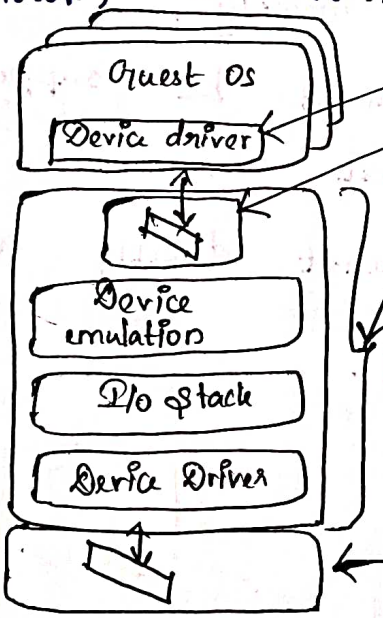
page table is called the shadow page table.

I/O Virtualization :-

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

⇒ There are three ways to implement I/O virtualization : full device emulation, para virtualization, direct I/O.

⇒ Full device emulation is the first approach for I/O virtualization. Generally this approach emulates well-known, real-world devices.



* Guest device driver.
* Virtual device.
* Virtualization Layer
  - emulates the virtual device
  - remaps guest and real I/O addresses
  - multiplexes and drives the physical device.
  - I/O features eg) Cow disks.
* Real device
  - may be different from virtual device.

Device Emulation for I/O Virtualization.

The software is located in the VMM and act as a virtual device. The I/O access request of

the guest OS are trapped (x) in the VMM which interact with the I/O devices.

⇒ The para-virtualization method of I/O virtualization is typically used in Xen.

⇒ It is also known as the split driver model consisting of a front end driver and a backend driver.

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 donot 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.

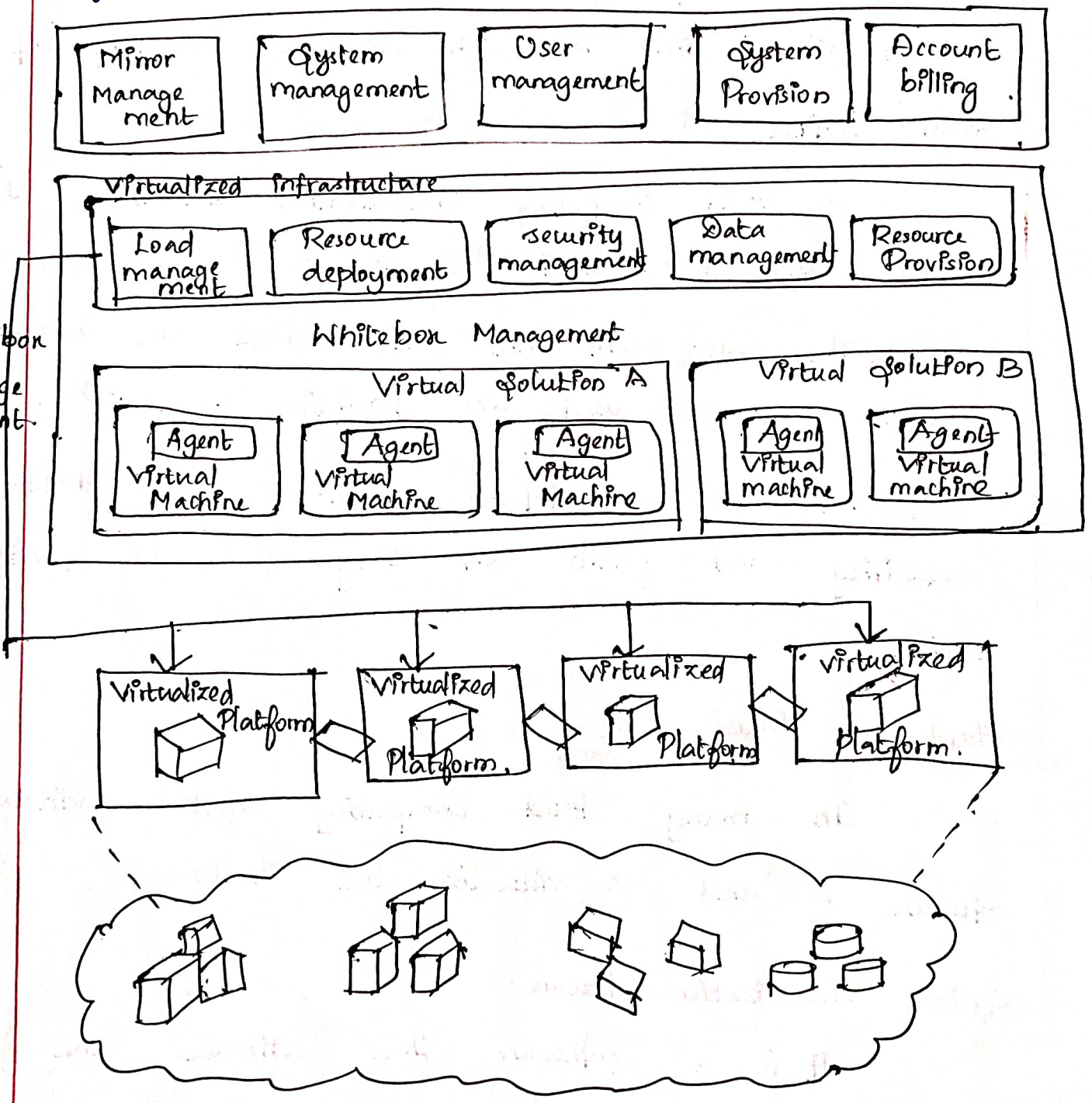
System Virtualization Software:-

It is a software that stimulus the execution of hardware, and runs even unmodified.

operating systems.

⇒ Cloud computing systems are virtualization software as the running environment for legacy software such as old operating systems and unusal 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.

(i) AWS
(ii) Microsoft Azure.
(iii) GIAE.

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

Microsoft ⇒ Provides programming Level virtualization for users to build their applications.

GIAE ⇒ Provides limitted 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 strengthem the synergy of green Computing.

Virtualization for Iaas:-

(i) VM technology has increased in ubiquity.

(ii) This enabled users to create customized environments for cloud Computing.

Benefits of using VMs :-

(i) system administrators consolidate workload.

(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 back up agents.

(iv) Long time to restart.

2. Recovery Using Virtual Machine :-

(i) to recover one VM by another VM.

(ii) To recover a VM platform, the installation and configuration times for the OS and back up agents are eliminated.

Advantages :-

(i) Shorter disaster recovery time.

(ii) Simple and Inexperience.

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.

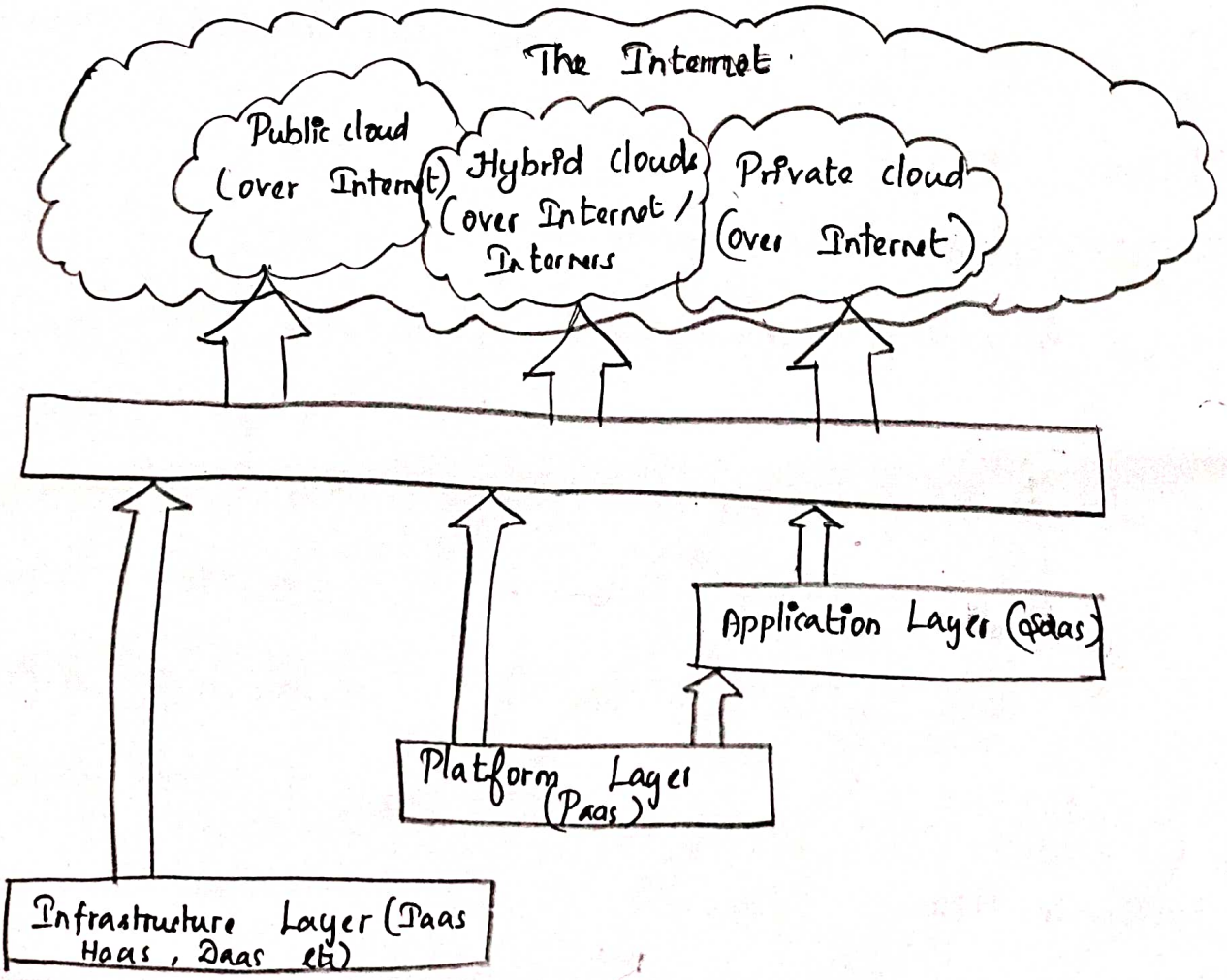
Layered Cloud Architecture Design:-

The cloud architecture is developed with three layers

(i) Infrastructure

(ii) Platform

(iii) Application.



⇒ Three development Layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud.

⇒ The public, private and hybrid cloud services are delivered to users through networking support over the internet and intranet.

Infrastructure Layer:-

⇒ Serves as the foundation for building the platform Layer.

⇒ is built with virtualized Compute, storage and network resources.

⇒ Support Iaas service.

⇒ Internally virtualization provides automated provisioning of resources and optimize the infrastructure management.

Platform Layer:-

⇒ Support Paas service.

⇒ is a foundation for implementation the application Layer.

⇒ It is for general purpose and repeated usage of the collection of software resources.

⇒ It provides scalability, dependency and security protection to users.

Application Layer:-

⇒ Supports Saas applications.

It is formed with a collection of all needed software modules for saas applications.

Concepts in Layered Cloud Architecture:

(i) Market oriented cloud Architecture

(ii) QoS factor.

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

mechanism:-

⇒ Client can benefit from the potential cost reduction of providers.

⇒ Lead to a more competitive market.

Lower process.

Entities of Market - Oriented Cloud Architecture:-

* Users or brokers
* Request Examiner.
* Pricing Mechanism.
* VM Monitor
* Accounting.
* Dispatcher.

* Service request monitor
* Virtual Machine

QoS Factors :-

Different QoS factors are

* Time
* Cost
* Reliability
* Security

The QoS Requirements can't be static and might from time to time on demand.

Types of cloud computing :-

1. Infrastructure as a service (Iaas)
2. Platform as a service (Paas)
3. Software as a service (Saas)

Cloud Computing Service Models :-

Software as a service (Saas)
Applications, Management, and User Interface provided over the network.

Platform as a service (Paas)
Application development frameworks, operating systems and deployment framework.

Infrastructure as a service (Iaas)
Virtual Computing, Storage & network Resources that can be provisioned on demand.

Saas → End User Utilization
Mail, Sales force, Dropbox,

Paas → Software Development.
Google, force.com.

Iaas → System Architecture.
amazon Web Services.

Infrastructure - as - a - service (Iaas) :- resource.

⇒ Iaas provides the users capability to provision Computing and Storage resources.

⇒ These resources are provided to the users

as (virtual machines instances) and virtual storage.

⇒ Virtual Resources provisioned by the users are billed based on ~~per~~ a pay-per-use paradigm.

Platform-as-a-Service (Paas)

⇒ Paas provides the users the capability to develop and deploy application in the cloud using the development tools, application programming interfaces, software libraries and services provided by the cloud service provider.

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

⇒ Google apps is one of the most famous Platform-as-a service providers.

⇒ This is the idea that someone can provide the hardware (as in Iaas) plus a certain amount of application software upon which you can build your application.

[⇒ It is an application development and deployment platform delivered as a service to developers over the web]

Software-as-a-Service (Saas) → It is the broadest market.

[Saas provides the users a complete software application] or the User Interface to the application itself.

The ⇒ Cloud service provider manages the underlying cloud infrastructure including servers, network, operating systems, storage, and application software, and the user is unaware of the underlying architecture of the cloud.

⇒ Applications are provided to the user through a thin client interface (Eg. browser).

⇒ Saas applications are platform independent and can be accessed from various client devices such as work stations, laptop, tablets and smartphones, running different operating systems.

Deployment Models :-

Public Cloud.

⇒ In the public cloud deployment model,

⇒ cloud services are available to the general public or a large group of companies.

⇒ cloud resources are shared among different users (individuals, large organizations, small and medium enterprises and governments)

Private Cloud :-

In the private cloud deployment model,

⇒ Cloud infrastructure is operated for exclusive for single organization.

⇒ Private cloud services are dedicated for a

single organization.

⇒ Private clouds are best suited for applications where security is very important and organizations that want to have very tight control over their data.

Hybrid Cloud:-

⇒ It combines the services of multiple cloud (Private or public cloud)

⇒ It is suited for organizations that want to take advantage of secured applications and data hosting on a private cloud.

⇒ And at the same time benefit from cost savings by hosting shared applications and data in public clouds

Community cloud:-

⇒ The cloud services are shared by several organizations that have the same policy and Compliance considerations.

⇒ Community cloud best suited for organizations that want access to the same application and data, and want the cloud costs to be shared with the larger group.

The World of Business:-

Businesses can employ cloud Computing in different ways. Some users maintain all apps and data

Public, Private and Hybrid clouds :- (9)

The cloud enables anyone with an internet connection to access IT resources on demand.

These resources may be,

(i) Compute resources.
(ii) Storage resources.
(iii) Networking Resources.

Four cloud Development Models.

⇒ Public cloud
⇒ Private cloud
⇒ Hybrid cloud
⇒ Community cloud.

Public cloud :-

⇒ is built over the internet.
⇒ It can be accessed by any user who has paid for service.

Example of public cloud..

(i) Google App Engine (GAE)
(ii) Amazon Web Services (AWS)
(iii) Microsoft Azure
(iv) BM Blue cloud.
(v) Salesforce.com Force.com.

Private cloud :.

is owned and managed by a client.

Its access is limited to owning clients and their partners.

On-site private cloud.

Out-sourced Private cloud.

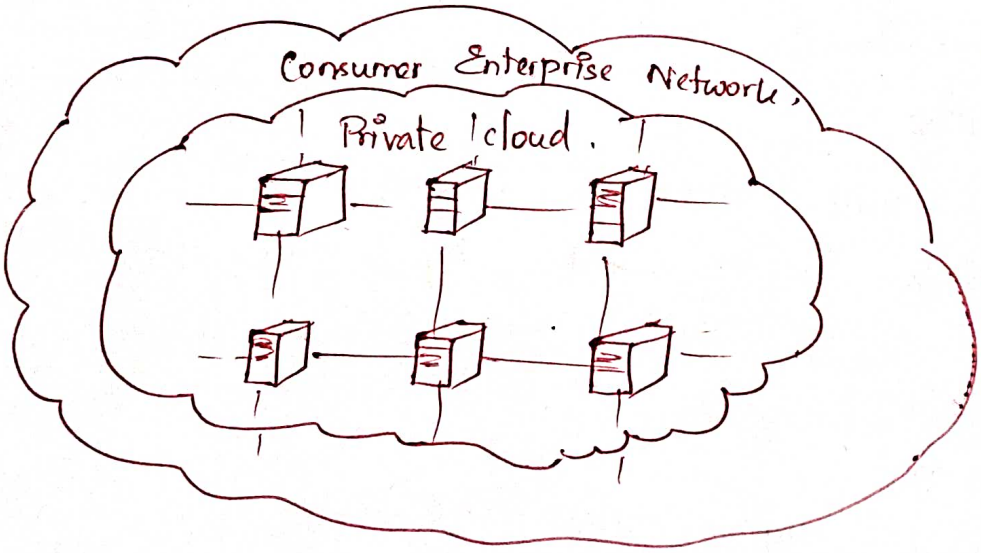
On source Private clouds:-

Owned by a third party and you get a part of their stack isolated for your use and need.

On site ~~pub~~ private cloud:-

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

This means you own the hardware like server, router, switches, storage devices and IT infrastructure.



NIST Cloud Computing Reference Architecture.

⇒ NIST stands for National Institute of standards and Technology.

⇒ NIST Composed for major six workgroups specific to cloud computing.

⇒ Cloud computing target business usee ases coork group.

⇒ Cloud computing Reference Architecture and Taxonomy work group.

⇒ Cloud computing standards roadmap coork group

⇒ Cloud computing SAJACC (standards Acceleration to Jumpstart Adoption of Cloud Computing) coork group.

⇒ Cloud Computing Security coorkgroup.

Objectives of NIST Cloud computing Reference Architecture.

⇒ Illustrate and understand the various level of services

⇒ To provide technical referenee.

⇒ Categorize and Compare services of Cloud Computing

⇒ Analysis of security, Interoperability and Portability

The conceptual referenee architeeture involves five actors entity participates in Cloud Computing.

Cloud Consumer:-

A person or an org uses a service from cloud Providers.

Cloud Provider:-

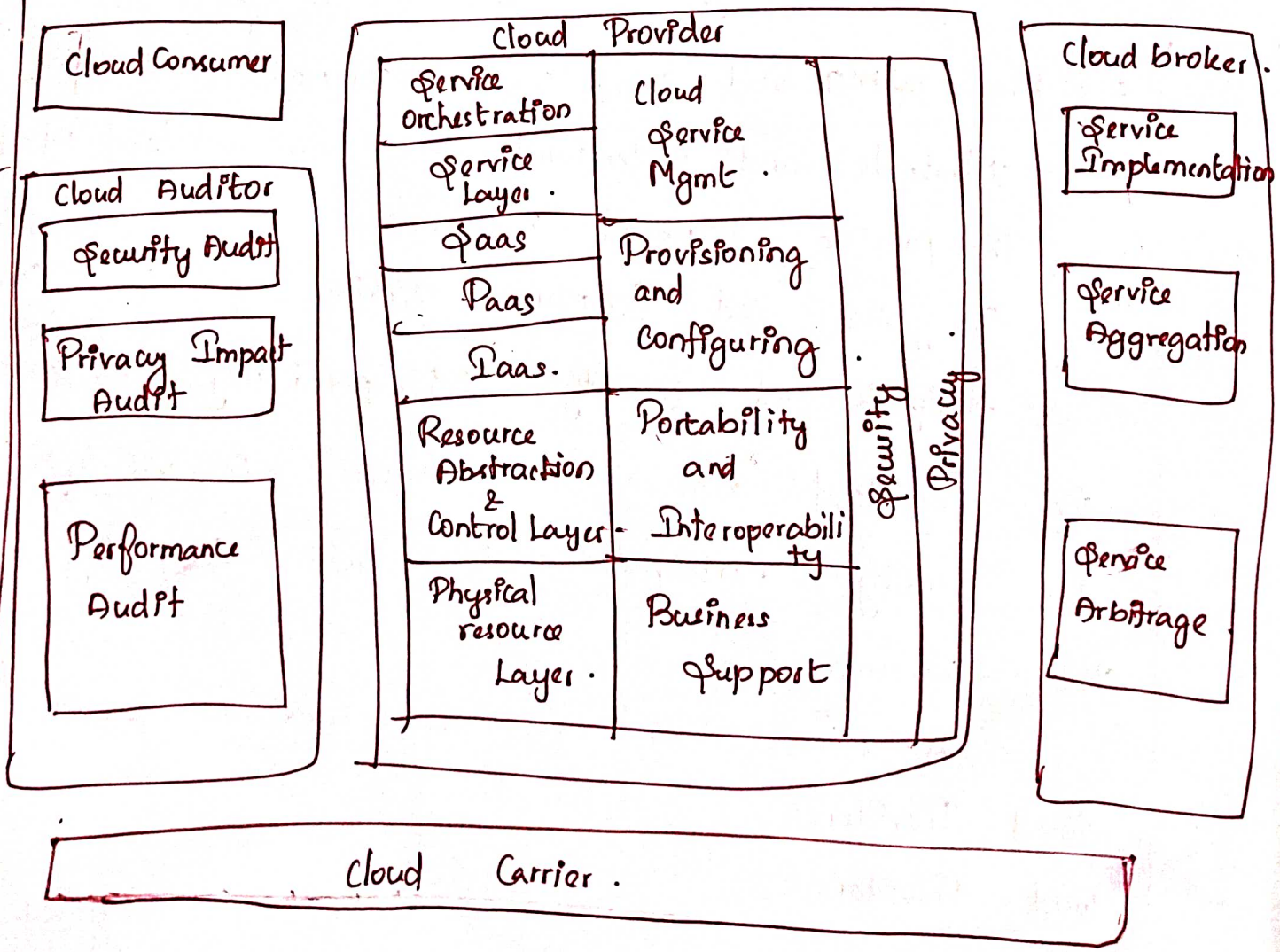
A person, organization or entity responsible for making a service.

Cloud auditor :-

A party that conduct independent assessment of cloud services, information system operation, performance and security of cloud Implementation.

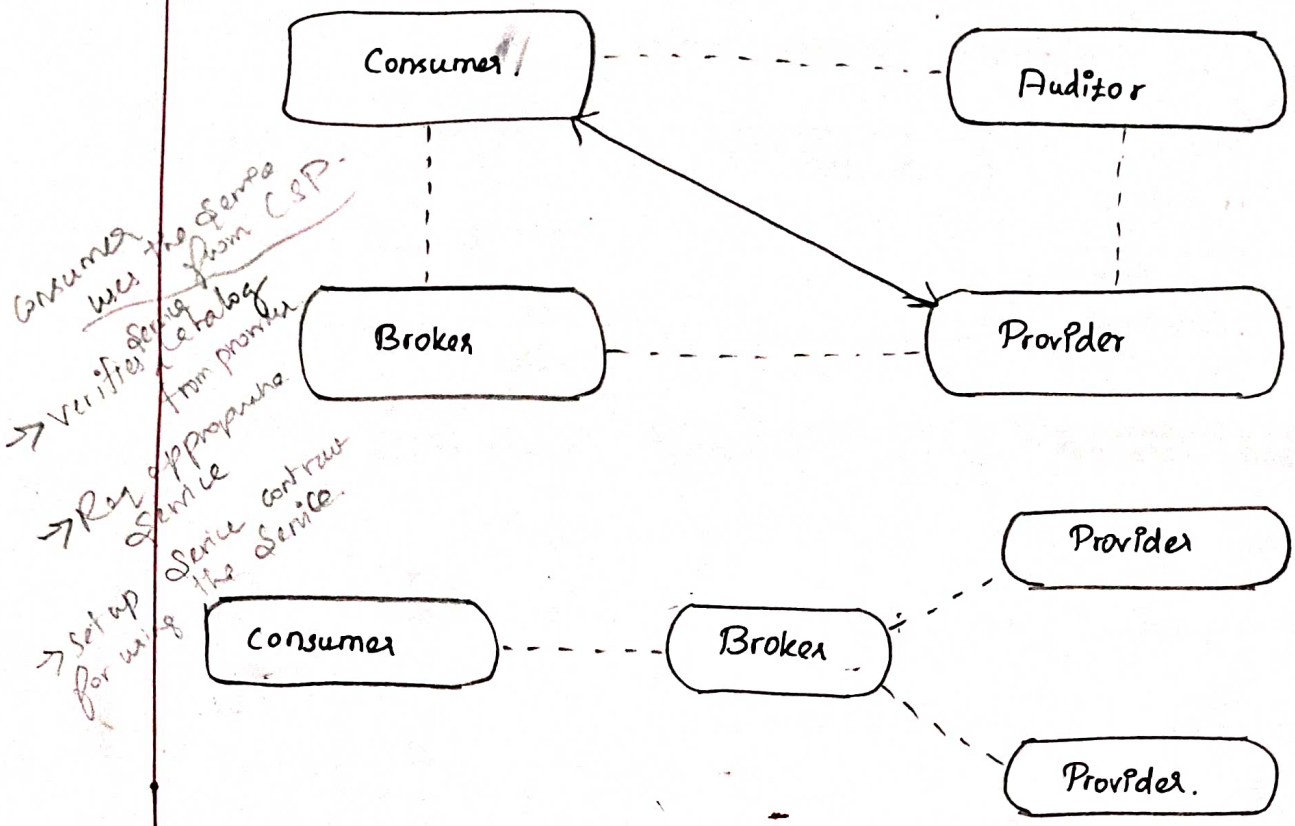
Cloud broker :- An entity that manages the performance and delivery of cloud services and negotiate relationship between cloud providers and consumer.

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



NIST Cloud Computing Reference Architecture.

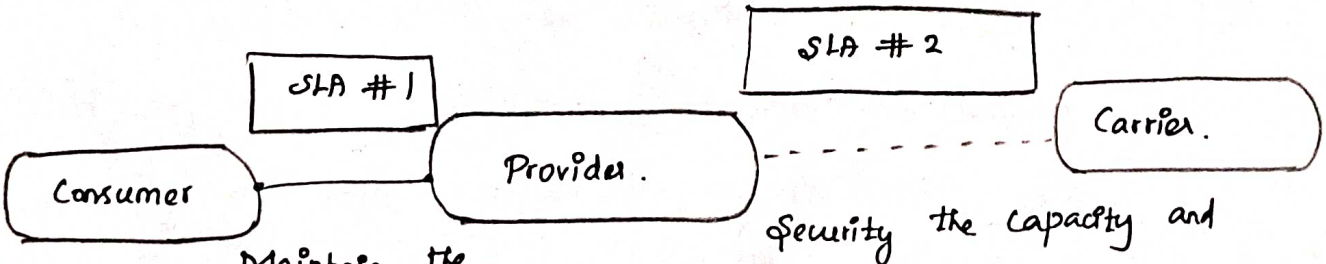
Interaction between Actors: - (13)



Service From Cloud Broker.

Cloud Consumer may request service from a cloud broker instead of contacting service provider directly. In this case, cloud broker can create a new service by combining multiple services.

Illustrates the Usage of different kind of service Level Agreement (SLA) between Consumer, Provider and Carrier.



Maintain the consistent level of service.

Security the capacity and Functionality.

Multiple SLA between Actor.

shows the scenario where the cloud auditor conducts independent assessment of operation and security of the cloud service Implementation :-

auditor will audit or examine services for its security controls, privacy performance

Cloud Consumer is a principal stack holder for the cloud Computing service and requires service Level Agreements to specify the performance requirements fulfilled by cloud Providers.

=> The service Level agreement covers quality of service and security aspects.

There are three kinds of cloud Consumers.

* SaaS Consumers.
* PaaS Consumers
* IaaS Consumers.

Types :-

Service Aggregation :-

→ Provides data integration.

→ cloud broker combines and integrate multiple service into one to more new services.

Service Intermediation :-

→ Enhance a given service by importing same service specific capacity and providing value added service to cloud Consumers.

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.

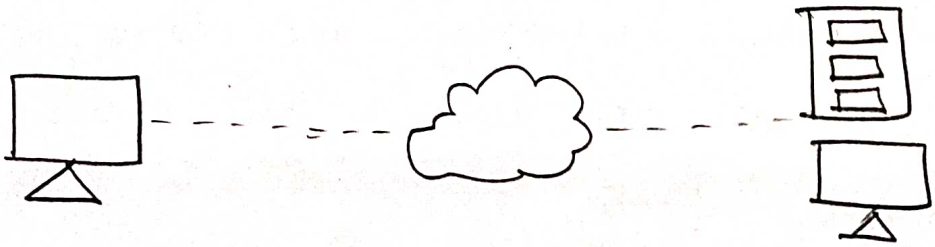
Cloud storage systems utilize dozens or hundreds of data servers. Because servers require maintenance or repair, it is necessary to store the saved data on multiple machines providing reduncy :

(1) storage_as_a servia :-

→ The term storage as a service, or saas.

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

→ The end Users does not have to pay for Infrastructure.



storage as a service.

→ They simply pay how much they transfer and save on the providers servers.

storage services :-

→ Backup, replication and disaster recovery.

→ Advantage of saas is cost savings.

Eg for Saas.

⇒ Google Docs.

⇒ Allows users to upload, documents, spreadsheets, and presentation to google's data servers.

~~Web email messages on their own servers~~.

Web email providers like Gmail, Hotmail, and Yahoo!.

⇒ Mail store email messages on their own servers.

⇒ Users can access their email from computers and other devices connected to the internet.

Flickr and Picasa :-

⇒ Host millions of digital photographs.

⇒ Users can Create their own online photo albums.

You tube :-

⇒ Hosts millions of User uploaded video files.

Hostmonster and GoDaddy :-

⇒ store files and data for many client web sites.

Facebook and Myspace :-

⇒ are social networking and allow members to post pictures and other Content. That content is stored on the Company's servers.

MediaMax and Strongspace :-

⇒ Offer storage space for any kind of digital data.

Techniques to Secure Data :-

Encryption :-

A complex algorithm is used to encode

information. To decode the encrypted files, a user needs the encryption key.

Authentication Processes:

⇒ This requires a user to create a name and Password.

Authorization Practices:-

⇒ List the authorized people to access information stored on the cloud system.

other concern is reliability:-

⇒ If a cloud system storage is unreliable, it becomes a liability. No one wants to save data on unstable system.

Advantages of Cloud storage:-

⇒ Attractive solutions for organizations.

Load balancing:-

⇒ Balance server loads.

Data Movement:-

⇒ Move data among various dataCenters.

⇒ Ensuring that information is stored close for faster data access.

Disaster Recovery and Data Protection:-

⇒ Allows the Users to protect the data in case there's a disaster.

Amazon S3 is the best known storage solution.

Cloud Storage Providers:

⇒ There are hundreds of cloud storage providers

⇒ Top cloud storage Providers.

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

⇒ Google offers cloud storage solution called GDrive

⇒ IBM offers number of cloud storage called Blue cloud.

Amazon S3: -

⇒ The well known cloud storage service is Amazon's simple storage service (S3), which is launched in 2006.

⇒ Make web scale Computing easier for developers.

⇒ Provides a simple web service interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web.

⇒ It gives any developer access to the same highly scalable data storage infrastructure that Amazon uses to run its own global network of web sites.

⇒ The service aim to maximize benefits of scale and to pass those benefits on to developers.

Amazon S3 functionality :. (19)

→ Write, read, and delete objects containing from 1 byte to 5 gigabytes of data each.

⇒ Unlimited number of objects that can be stored.

⇒ Each object is stored and retrieved via a unique developer assigned key.

⇒ objects can be made private or public

⇒ Use REST and SOAP Interfaces.

Unit IV Resource Management and Security in cloud

Inter cloud Resource Management - Resoura Provisioning and Resoura Provisioning Methods - Global Exchange of cloud Resources - Security Overview - Cloud Security challenges - Software as a Servia 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 provisioning or supplying cloud resources.

⇒ Provides supply cloud services by signing SLAs with end users.

⇒ Understanding provisioning of resources will lead to broken SLAs and penalties.

Difficulties :-

Unpredictability of Consumer demand.

Software and hardware failure.

Heterogeneity of Services.

(iv) Power management.

(v) Conflicts in signed SLAs.

Virtualized cluster of services.

(i) Efficient Installation of virtual Machines.

(ii) Live VM Migration.

(iii) Fast recovery from failures.

Example of VMM:-

⇒ Amazon EC2 Users Xen.

⇒ Xen VMM is used in IBM's blue cloud.

⇒ EC2 Platform.

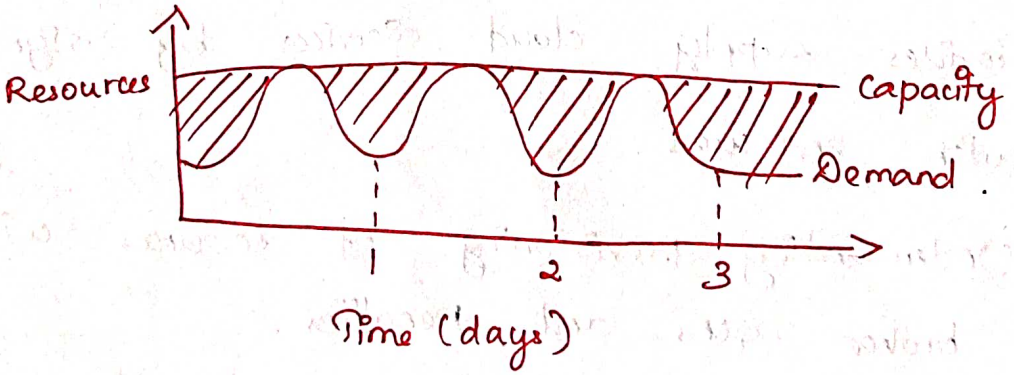
⇒ Predefined virtual Machine Templates.

Resource Provisioning Methods:-

Three cases of static cloud resource provisioning policies.

a) overprovisioning with the peak Load:-

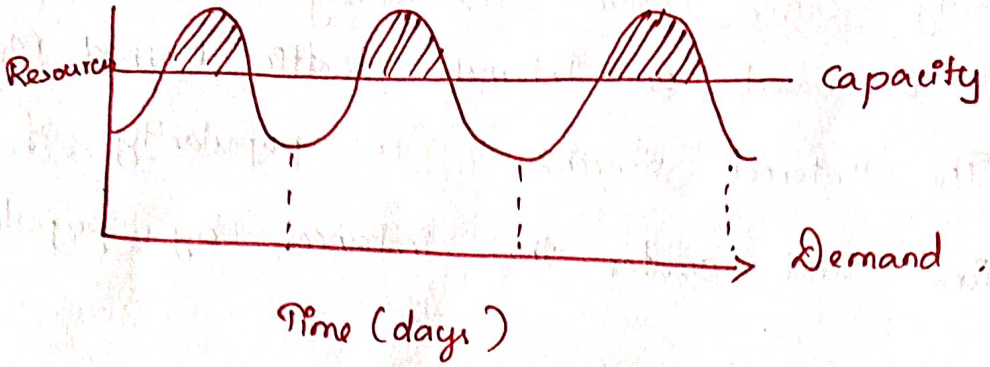
→ Causes heavy resource waste (shaded area)



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

→ Loss for both user and provider.

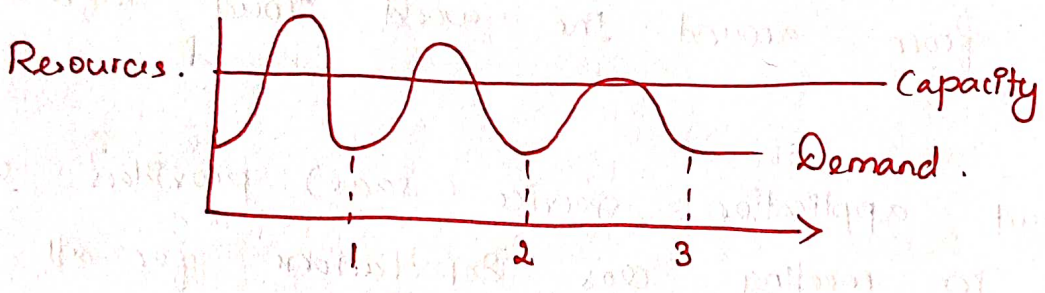
→ Paid demand by the users.



c) Constant Provisioning of Resources with fixed capacity.

⇒ user demands are declined.

⇒ Resources in resource waste.



Resource Provisioning Methods :-

Three Methods.

1) Demand Driven Resource Provisioning :-

⇒ Provides static Resources.

⇒ Used in grid computing for many years

⇒ Adds / removes computing Instances / Resources based on the current Utilization level of the allocated Resources.

2) Event Driven Method :-

⇒ It is based on predicted workload by time.

⇒ This scheme add or removes machine instances.

3. Popularity Driven Resource Provisioning :-

⇒ It is based on Internet traffic Method Monitored.

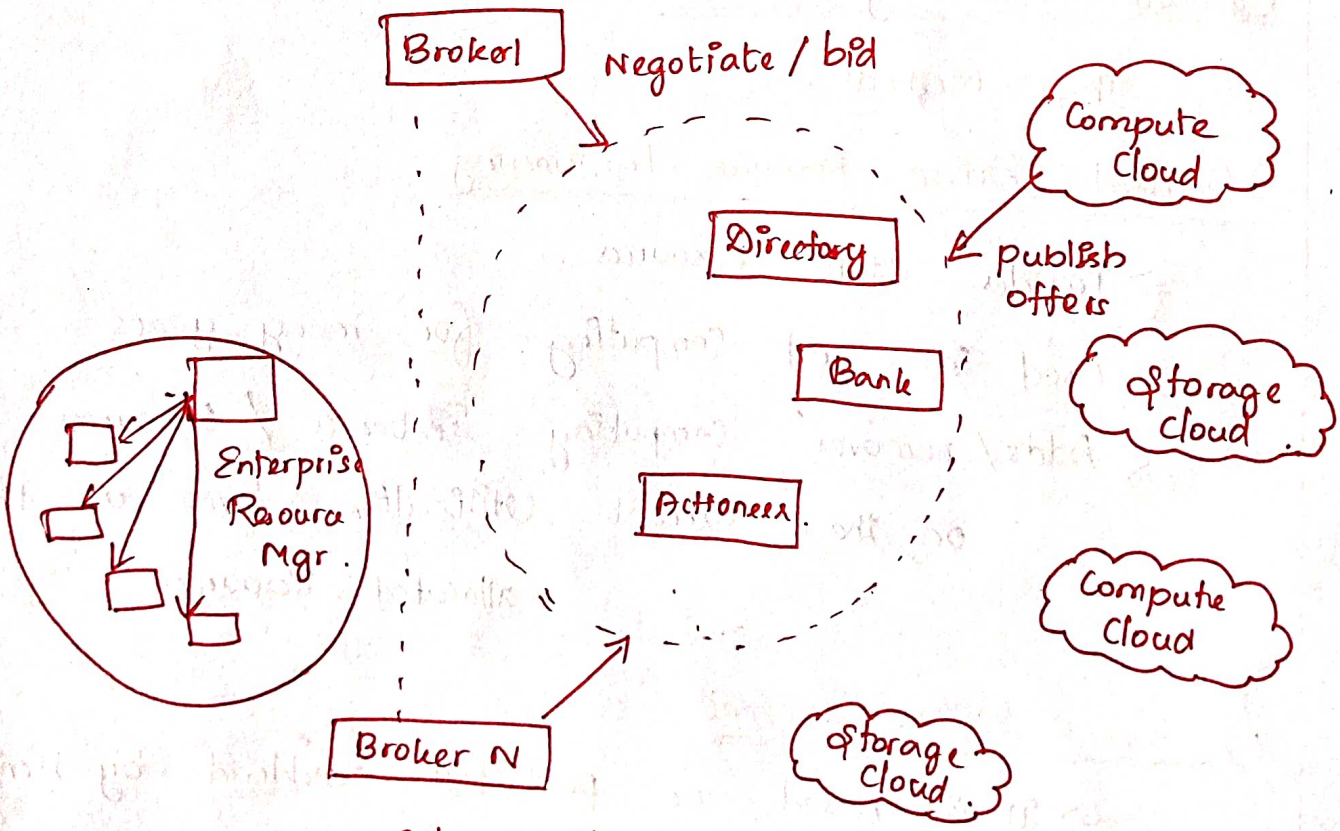
⇒ The Internet searches for popularity of certain applications and creates the instances by popularity demand.

⇒ 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 customer



Global Cloud Exchange.

⇒ 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 : (Ex)

⇒ Act as a market maker.

⇒ Bringing together service Producers and Consumers.

⇒ CEx allows participants to locate providers and consumers with fitting others.

SLAs :-

⇒ Specifies the details of 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 ~~login~~ learn from the managed service provider (MSP) model.

⇒ Ensure that their customers applications and data if they hop to retain their customer base & competitiveness.

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

⇒ A cloud solution provider must ensure that consumer 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 :-

(i) Security Governance :-
(ii) Virtual Machine Security.
(iii) IAM
(iv) 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 Coverance.

⇒ 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 Cente security teams.

⇒ Firewalls, Intrusion Detection, and Prevention, Integrity Monitoring, and log Inspection.

⇒ Traditional line of defense to the virtual Machine.

⇒ To facilate 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 :-

⇒ VM provides consistent control and management throughout the cloud.

⇒ Economics of scale, deployment and cost savings for both service provider and the enterprise.

Identity Access Management: (IAM)

⇒ It is a critical function for every organization.

⇒ A fundamental expectation of SaaS customers, is that the principle of Least previledge is granted to their data

⇒ Cloud services and services on demand & changing the identity management landscape.

The current Models are challenging :-

(i) Trust Assumptions.
(ii) Privacy Implications.
(iii) 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 & 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

⇒ SOAP, HTTP, and XML.

⇒ SAML relies on HTTP.

SAML Core:-

⇒ General syntax and semantics of SAML assertions

⇒ Protocol used to request and transit those assertion 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 like by an email address.

It might be a printer.

SAML Assertions:-

⇒ Transferred from identity providers and service providers

⇒ Authentication statements, Attribute statements, Authorization Question statements.

(i) Authentication standards.

(ii) An attribute statements.

(iii) An authorization or a assertion statement.

Three types of SAML queries.

⇒ Authentication query.

⇒ Attribute query.

⇒ 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 onto 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 :-

=> End point authentication and data confidentiality.

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 :-

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

HDFS :-

⇒ It 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 of 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 systems are,

(i) Block Replication.
(ii) Replica Replacement.
(iii) Heartbeat 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 Name Node to get the location of file blocks.

⇒ Namenode returns the address of a set of Data Node containing Replica Information for the requested file.

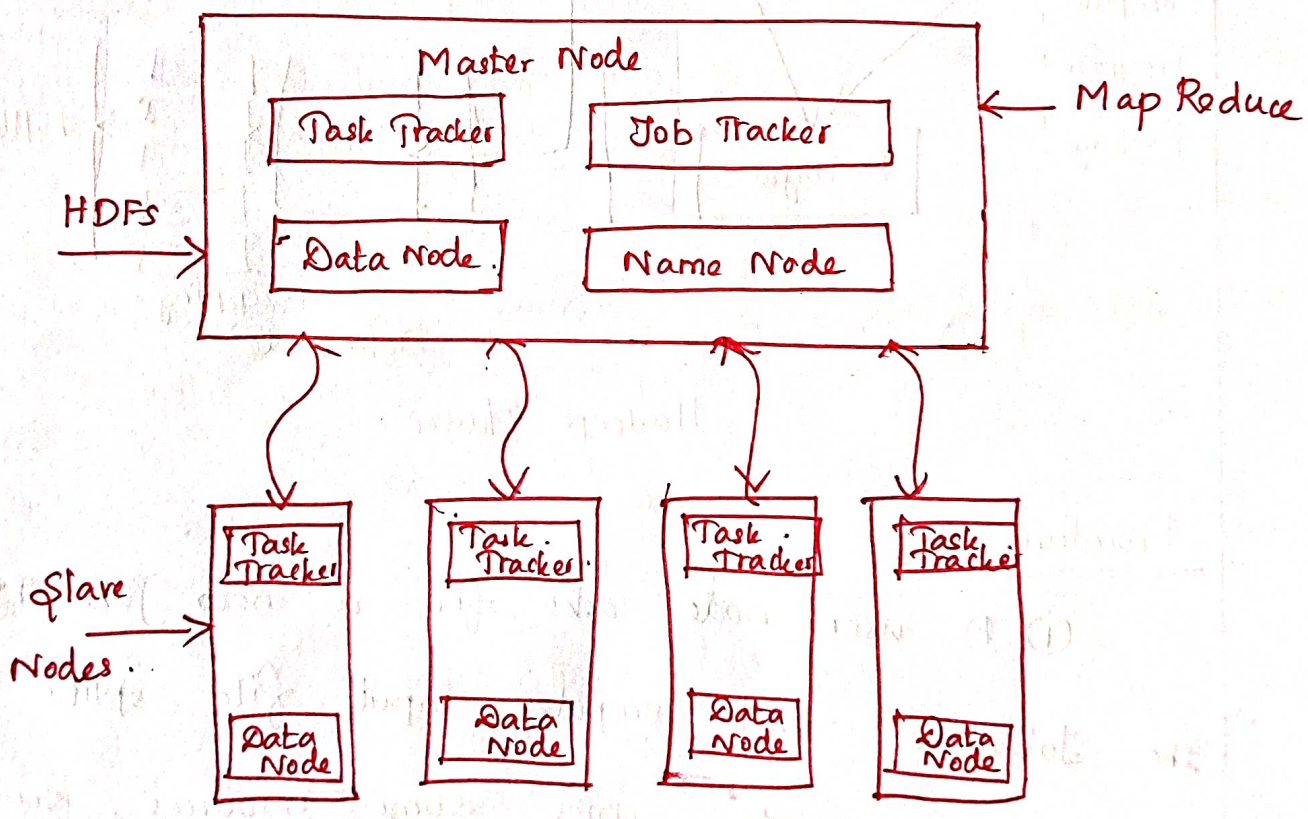
Write :- A user sends a create request to the Namenode to create a new file in file system Namespace.

⇒ The streames stores the block in the first allocated Data Node.

MapReduce :-

⇒ The top 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 :-



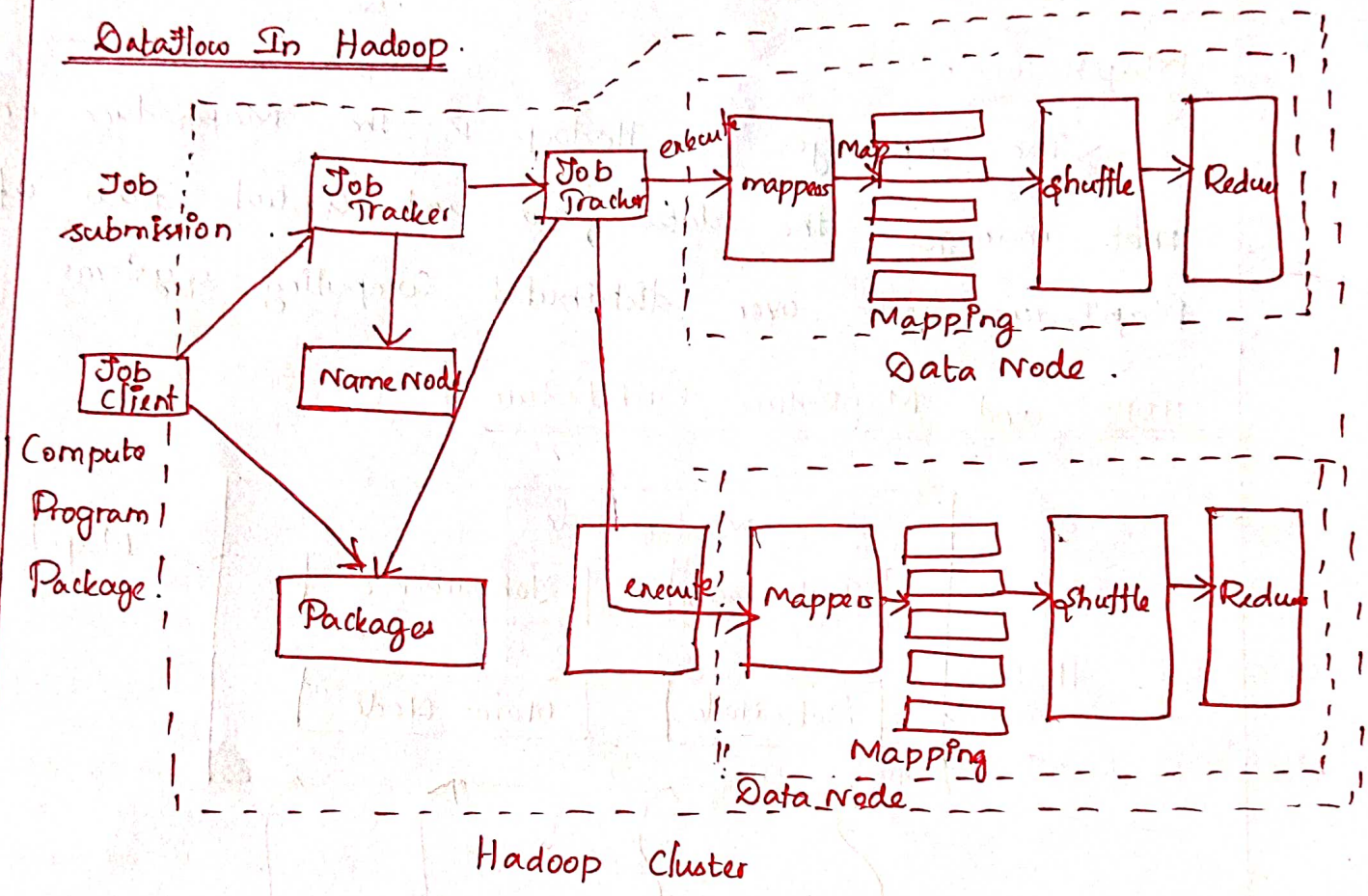
MapReduce engine also has a master / slave architecture consisting of a single JobTracker as the master and a number of Task Trackers as the slaves.

Running a Job in Hadoop :-

Three Components :

(i) User node .
(ii) Job Tracker
(iii) Task Tracker .

Dataflow In Hadoop .



Procedure :-

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

(ii) The user node copies some resources , such as Jobs JAR file , configuration file and computed input splits , to the JobTracker file system .

⇒ The user node submit the Job to the JobTracker by calling the submit Job() function .

Task Assignment :-

⇒ Job Tracker creates one map task for each computed input split by the user mode and assigns the map tasks to the execution slots of the Task Trackers

Task Execution :-

⇒ The control flow to execute a task starts inside the TaskTracker by copying the Job JAR file to its file system.

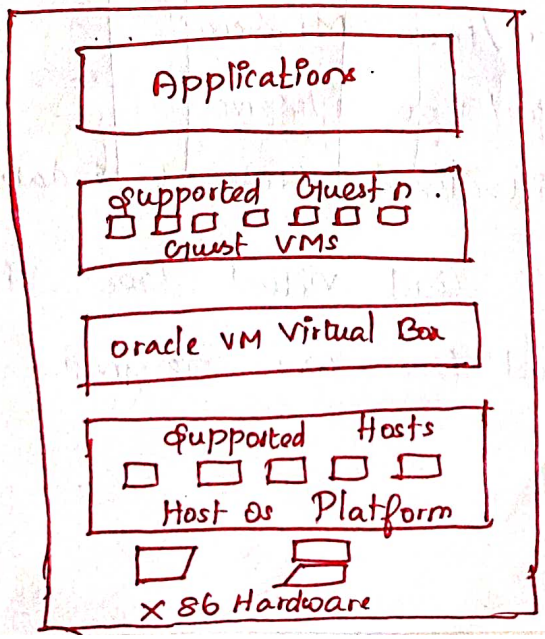
Virtual Box :-

⇒ Oracle VM virtual Box 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.



Virtual Box supported for Windows, mac os, Linux, Solaris, & Opeo 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 Virtual Box Disk - stored as ".vdi"

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

VHD – Windows Virtual PC and Hyper V – ".vhd" filename extensions.

Virtual Box Network Cards :-

For an Ethernet Adapter, Virtual Box virtualizes these Network Interface Cards.

(i) AMD PCnet PCI II.
(ii) AMD PCnet – Fast III
(iii) Intel Pro/1000 MT Desktop.
(iv) Intel Pro/1000 MT Server.
(v) Intel Pro/1000 T-Server.
(vi) Para Virtualized Network Adapter.

⇒ For a Sound Card Virtual Box Virtualize Intel HD Aud
⇒ Oracle VM Virtual Box was designed to be modular and flexible.

Google App Engine :- (18)

⇒ Google platform is based on its search Engine expertise.

⇒ Google App Engine (GAE) which offers a pass platform supporting various cloud and web applications.

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

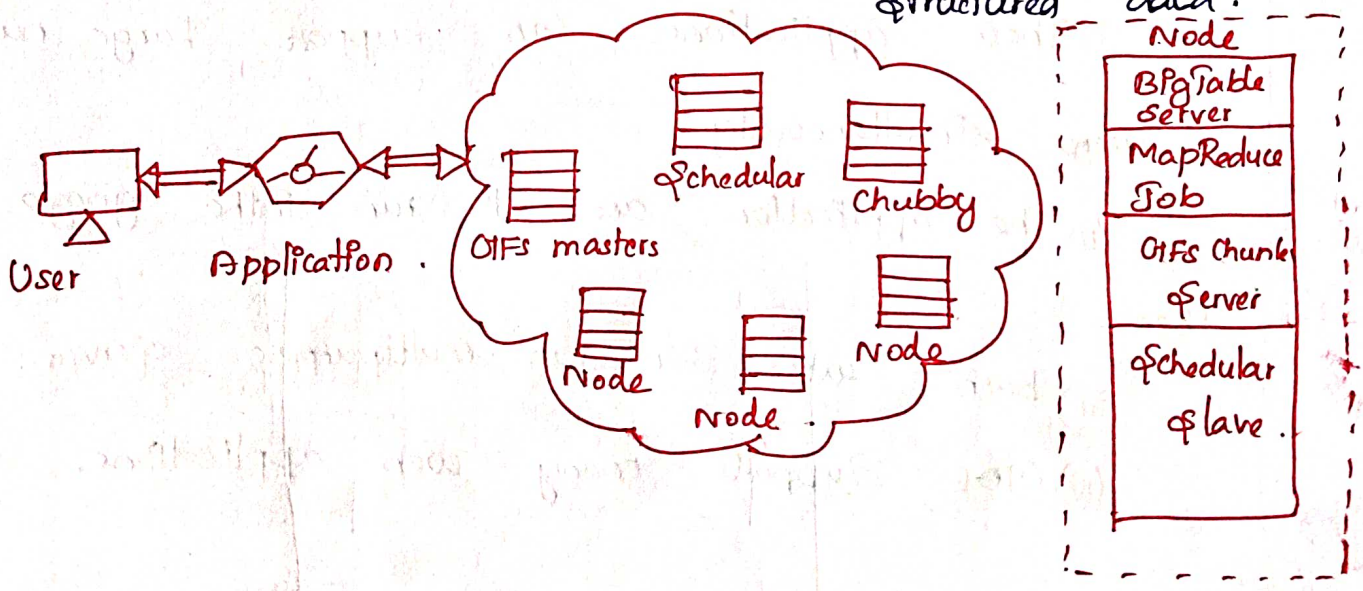
GAE Architecture :-

⇒ GFS is used for storing large amount 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

General Software Components

Functional Modules of GAE :-

⇒ Five major Components,

(i) The data store secures data Management operations

(ii) It supports two development Languages Python and Java.

(iii) Software Development kit (SDK) is used for Local application development

(iv) The administration console is used for easy management of user application development cycles.

(v) The GAE Web service infrastructure provides special interfaces it guarantee flexible use and mangement of storage and network resources of GAE.

GAE Applications :-

(i) These applications can support Large number of users simultaneously.

(ii) The application are all run inthe google data Users.

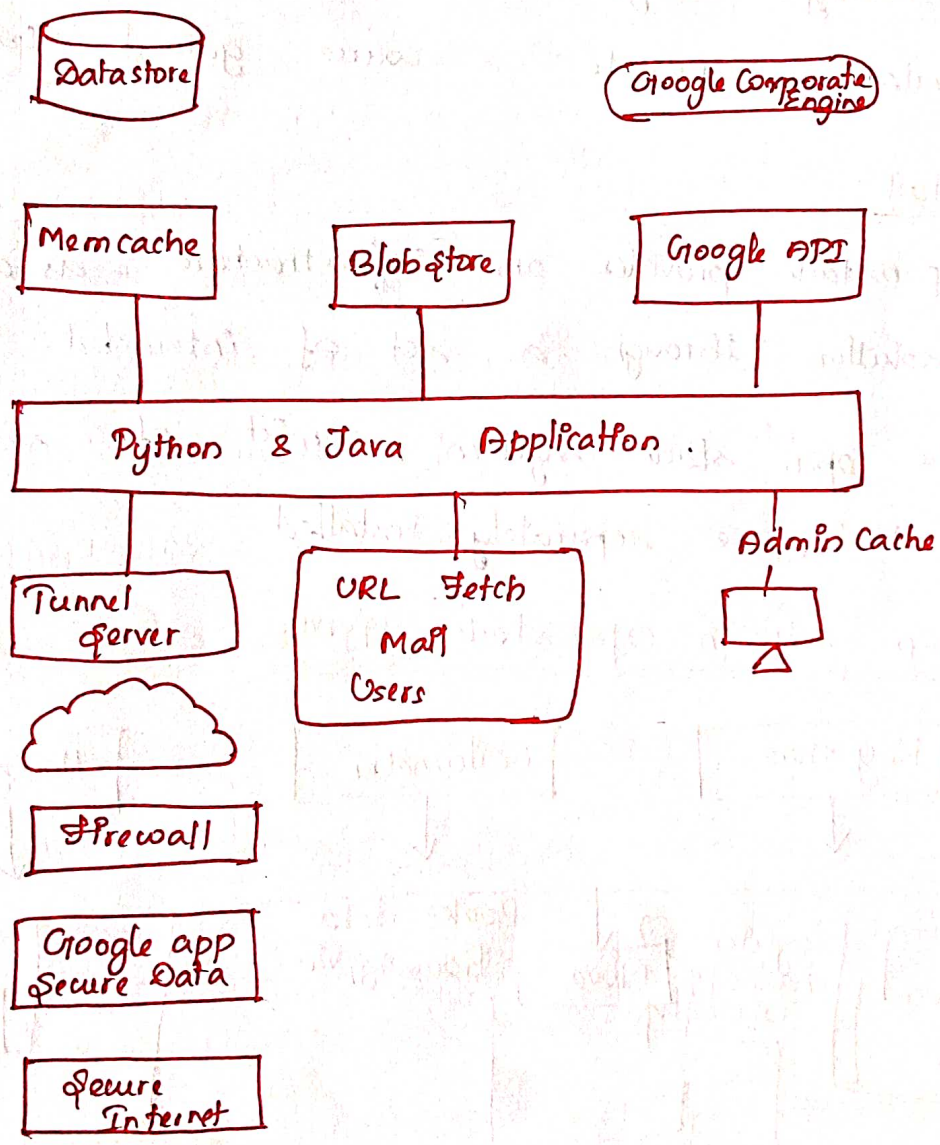
(iii) Each clusters can run multipurpose servers.

(iv) 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. (11)

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

(iii) The applications may need to examine data changes over time.

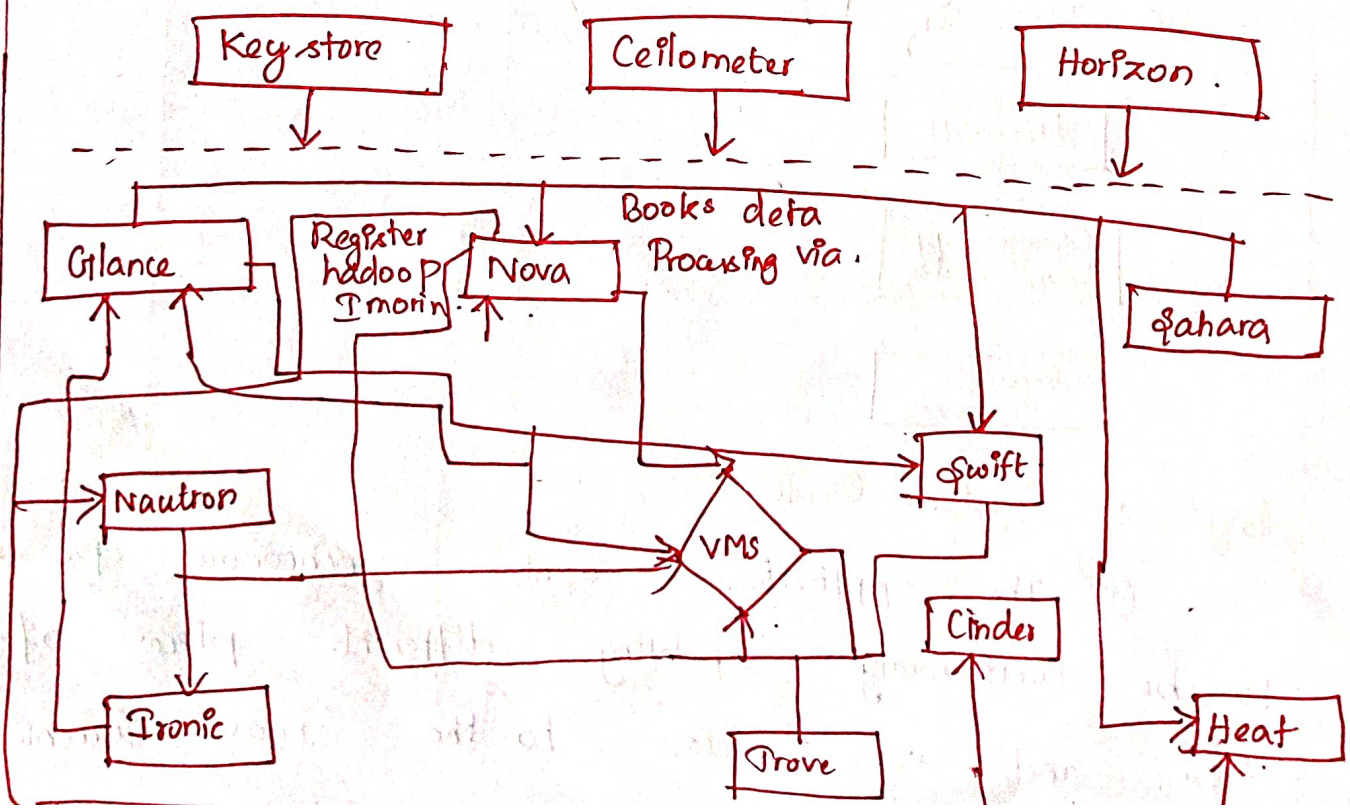
Chubby, Google's Distributed Lock service Chubby is intended to private a course gained Locking service.

Open stack :-

⇒ Openstack provides on infrastructure as a service (Iaas) solution through a set of interacted services.

⇒ The open stack system consist of several keys services that are separately installed.

Relationship between Openstack services :-



Federation in the cloud :- (22)

⇒ It supports microsoft cardspace and Novels Digital Mc.

⇒ Google, Apps, AOL, IBM, LFve Journal and all Incorporated this protocol into their cloud based solutions in the last few years.

⇒ Polling is how most of us check our emails.

⇒ It is flexible and designed to be extended.

⇒ An Amazon EC2 - backed server can run Jetty and cometd 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: -

⇒ occur when a server accepts a connection from a peer network server without verifying its identity using DNS lookups or certificate checking

⇒ Lack of verification of authentication may led 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 present 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 Applications :-

⇒ S2S Federation is a good start toward building a real time communication cloud.

⇒ Finding the entities is a process called discovery.

⇒ XMPP use service discovery to find the more mentioned 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 communication is precursor to building a seamless cloud that can interact with people, devices, information feeds, documents, application interfaces, and other entities.

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

(ii) The process of server to server federation for the purpose of inter domain communication has played a largerole in the success of XMPP.

(iii) Three mechanisms have provedided a stable, secure foundation for growth of the XMPP network and similar real time technologies.