

UNIT-II

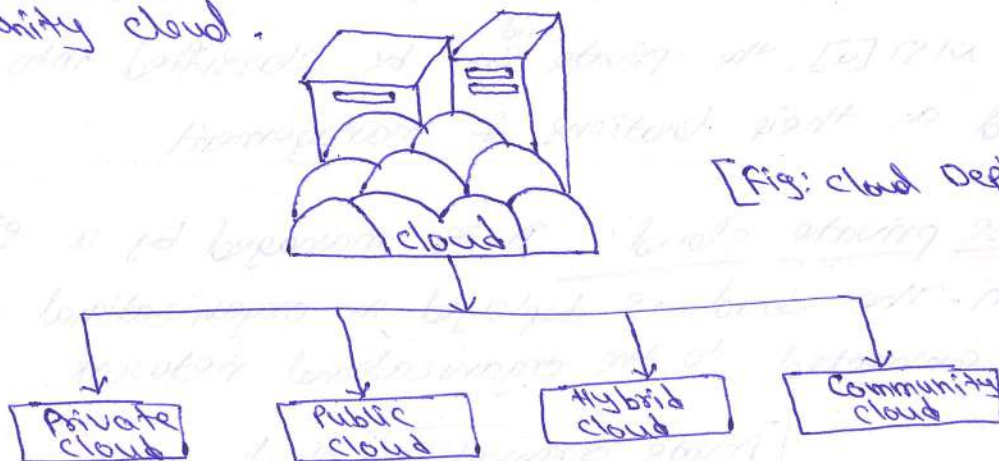
① Cloud Deployment models

Deployment models can be defined as the different ways in which the cloud can be deployed.

→ The deployment refers to setting up and launching cloud resources (like applications, services, or infrastructure) so they can be used by users or systems.

Basically, there are four types of deployment models in the cloud:

- i) Private cloud
- ii) Public cloud
- iii) Hybrid cloud
- iv) Community cloud.



[Fig: cloud Deployment Model]

i) Private cloud: A private cloud is a cloud environment used exclusively by one organization. It can be hosted either on-premises ^{own site} or by a third-party provider, but the infrastructure is dedicated to that organization.

Characteristics:

- Secure: It has dedicated resources to ensure data isolation and protection.
- Central Control: The organization mostly has full control over the cloud as usually the private cloud is managed by the organization itself.
- Weak SLAs: The SLAs may depend on internal teams; not always guaranteed.

Suitability: It refers to the instances where this cloud model can be used. It also signifies the most suitable conditions and environment where this cloud model can be used, such as:

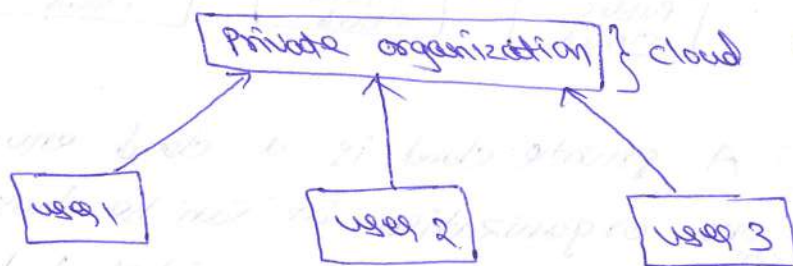
- The organizations or enterprises that require a separate cloud for their personal or official use.
- The organizations that want autonomy & complete control over the cloud.
- The organizations that have a less no. of users.

Not Suitable for:

- organizations that have high user base.
- organizations that have financial constraints
- organizations that do not have prebuilt infrastructure.

According to NIST [4], the private ^{cloud} can be classified into several types based on their location & management.

on-premise private cloud: It is managed by a single organization. The cloud is deployed in organizational premises and is connected to the organizational network.

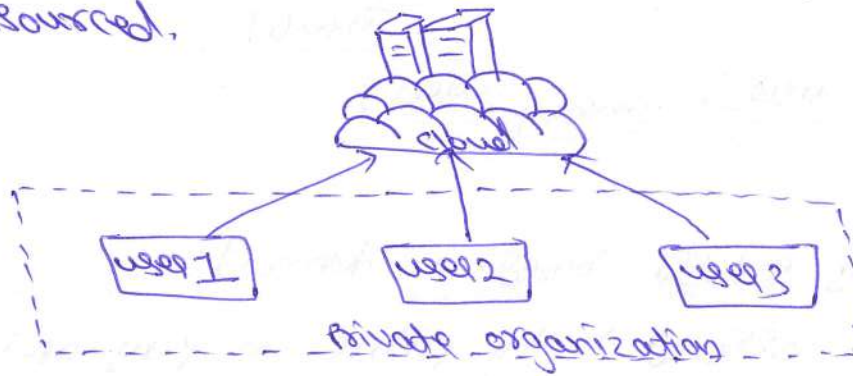


[Fig: on-premise private cloud]

Issues:

- SLA: Terms and agreements need to be strictly followed
- Network: cloud is totally dependent on the n/w that is laid out
- Performance: It controlled by the network management team, and most of this would have good performance as the no. of resources is low
- Location
- Multitenancy
- Main tenancy

Outsourced private cloud: This cloud outsourced to a third party. A third party manages the whole cloud. Everything is same as usual private cloud except that here the cloud is outsourced.



Issues:

- > SLA: The whole cloud is managed by the third party.
- > Network: The cloud's internal n/w managed by third party of the organization.
- > Location: When there is a change of location, the data need to transmit through long distance.
- > Performance: The performance of cloud depends on third party.
- > Maintenance: Maintained by the third-party organization.

Advantages:

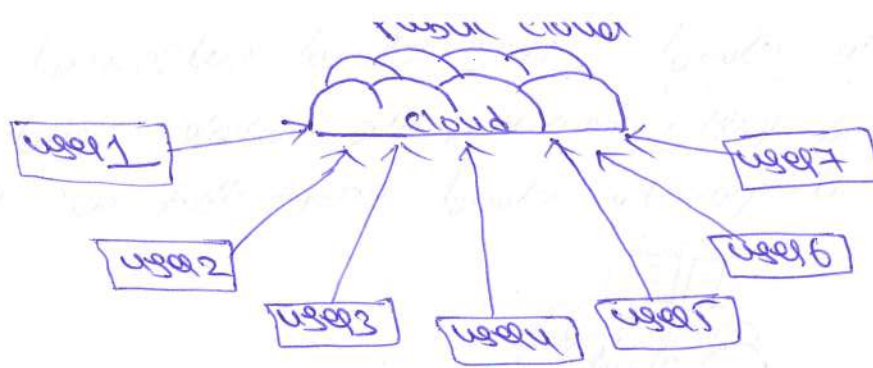
- > The cloud is small in size & easy to maintain.
- > It provides a high level of security & privacy to the users.
- > It is controlled by the organization.

Disadvantages:

- > For the private cloud, budget is a constraint (limitation).
- > The private clouds have loose SLAs.

Public cloud: The public cloud is the cloud infrastructure that is provisioned for open use by the general public.

It may be owned, managed & operated by a business, academic or government organization or by a third-party cloud service provider.



[Fig: public cloud]

Characteristics:

- Highly Scalable: Easily scalable based on demand.
- Affordable: It is offered to the public on pay-as-you-go basis; hence, the user has to pay only for what they are using (usually on a per-hour basis).
- less secure: Because it is offered by a third party and have full control over the cloud.
- Highly available: Because any-body from any part of the world can access the public cloud with proper permission.

Suitability:

The public cloud can be used whenever:

- There is no physical infrastructure available
- An organization has financial constraints
- The requirement for resources is varying (not fixed)

not suitable for:

- Security is very important
- organization expects autonomy (act independently)
- Third party reliability is not preferred.

Issues:

SLA, Performance, Multitenancy, location, cloud management, network, Maintenance.

Advantages:

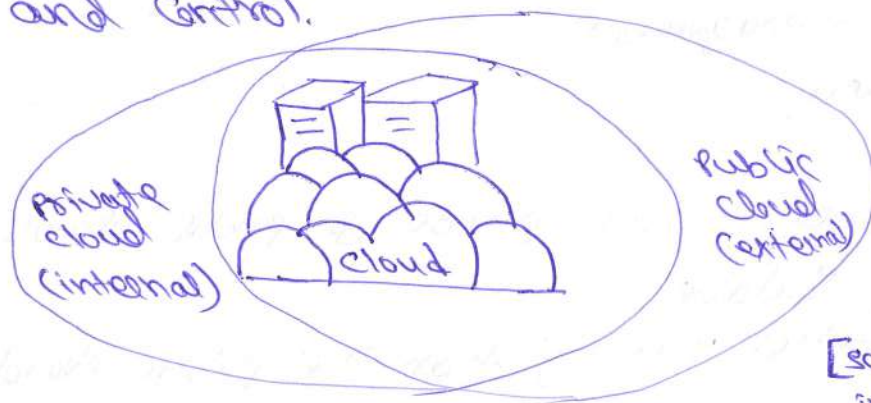
- There is no need for maintaining the cloud
- Strict SLAs are followed
- No limit for no. of users.
- Highly scalable (without breaking or slowing down)

Disadvantage:

- Privacy & organizational autonomy are not possible
- Security is an issue

ii) Hybrid cloud

A hybrid cloud is computing environment that combines private cloud (on-premises or dedicated infrastructure) with public cloud service, allowing data & applications to move b/w the two environments as needed for better flexibility, scalability and control.



[scalable: Able to handle increasing amount of work or capable of growth without performance loss]

Characteristics:

- Scalable: The hybrid cloud is combination of one or more deployment models. usually, the private with cloud gives hybrid cloud.
- partially secure: The private cloud is considered to be secured, but as the hybrid cloud also uses the public cloud, there is risk of security breach.
- Stringent SLAs: stringent means (strict and must be followed carefully). As the hybrid cloud involved a public cloud invention, the SLAs are stringent & might as per public cloud service provider.

Suitability:

Suitable for:

- organizations that want the private cloud environment with the scalability of the public cloud.
- organizations that require more security than the public cloud.

Not Suitable for:

- organizations that consider security as a prime objective.
- organizations that will not be able to handle hybrid cloud management.

Issues:

SLA
Network
Performance
Multitenancy, cloud maintenance.
Location, cloud management
Security & privacy,

Advantages:

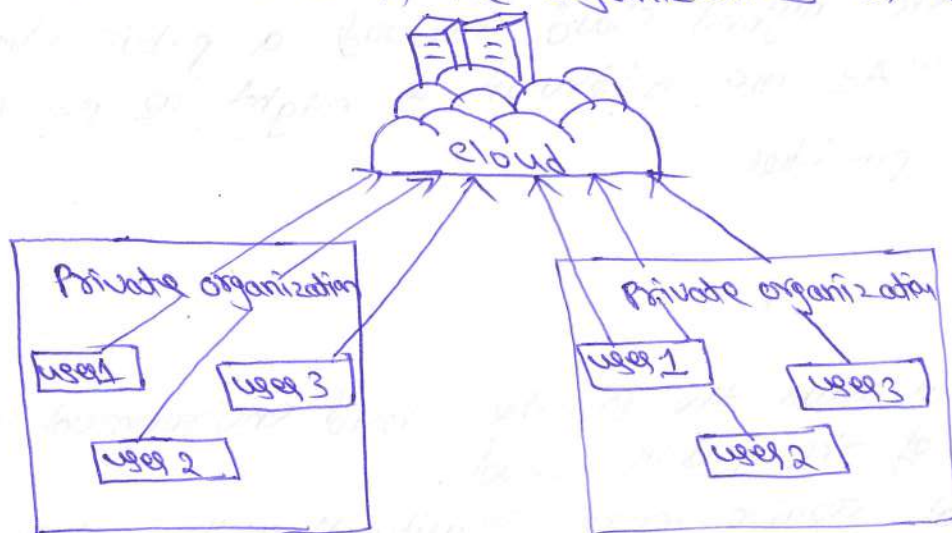
- It gives the power of both private & public cloud.
- It is highly scalable
- It provides better security than the public cloud.

Disadvantages:

- security features are not good in public cloud
- Managing a hybrid cloud is complex.
- It has stringent SLAs.

Community cloud:

A community cloud is a cloud infrastructure shared by multiple organizations that have a common goals, concerns or requirements (like security, Compliance, or policy) and it may be managed by one or more of the organizations or a third party.



Characteristic:

- collaborative and distributive maintenance: It is wholly collaborative and usually no single party has full control over the whole cloud. This is usually distributive, and hence, better cooperation gives better result

> Partially Secure: It refers to the property of the Community cloud where few organizations share the cloud, so there is a possibility that the data can be leaked from one organization to another.

> Cost effective: This cloud is cost effective as the whole cloud is being shared by several organizations of a community. Usually, not only cost but every other sharable responsibilities are also shared or divided among the groups.

Suitability:

It is suitable for:

- who want to establish a private cloud but have financial constraint
- Do not want to complete maintenance responsibility of the cloud.
- Establish the cloud in order to collaborate with other clouds

not suitable for:

- prefer autonomy and control over the cloud.
- Does not want to collaborate with other organizations.

> On-Premise Community cloud: It consists of the cloud deployed within the premises and is maintained by the organizations themselves.

Issues:

- SLA
- Network
- Performance
- Multi-tenancy, cloud management
- Location, cloud maintenance.
- Security & privacy

Outsourced Community cloud:

In the Outsourced Community cloud, the cloud is outsourced to a third party. The third party is responsible for maintenance and management of the cloud. (7)

Issues:

> SLA: The SLA is b/w the group of organizations and the service provider. The SLA here is stringent as it involves a third party.

> Network: The issues related to the network are same as the on-site community cloud.

> Performance

> Security & privacy

> Laws and conflicts.

> Cloud management

> Cloud maintenance

➔

Advantages:

> It allows establishing a low-cost private cloud.

> It allows collaborative work on the cloud.

> It allows sharing of responsibilities among the organization.

> It has better security than the public cloud.

Disadvantages:

> Autonomy of an organization is lost.

> Security features are not as good as the private cloud.

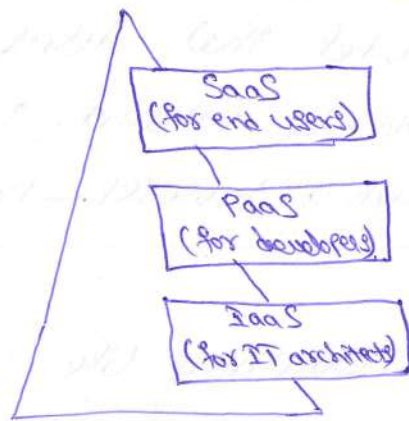
> It is not suitable if there is no collaboration.

2) cloud service models

cloud computing provides computing resources, development platforms, and applications as a service to the end users. The IT industries started subscribing the cloud services instead of buying the products.

→ cloud computing is a model that enables the users to access the shared pool of resources such as compute, network, storage, database, and application as an on-demand service without need to buy or own it.

The NIST definition of the three basic service models:



[Fig: Basic cloud Service Models]

IaaS: A cloud service model that "provides virtualized hardware resources" (like servers, storage, and networks) over the internet.

→ It will be provided by Amazon web services (AWS), Microsoft Azure, Google cloud platform (GCP).

→ It will be used by system administrators, network engineers, IT departments, developers who need control over computing resources.

Ex: Amazon EC2

Microsoft Azure virtual machines

Google Compute Engine.

PaaS: A cloud service model that offers a "ready-to-use" platform for developers to build, test, and deploy applications.

→ It is provided by cloud vendors like Google, Microsoft, IBM, Heroku, etc.

→ It is used by developers, software engineers, application designers who want to focus on coding and app logic.

Ex: Google App Engine

Microsoft Azure App Service

Heroku

Red Hat OpenShift

SaaS: A cloud service model that delivers "fully functional software applications" over the internet. Means users simply use the software through a browser - no installation or maintenance needed.

→ It is provided by SaaS companies like Google, Microsoft, Salesforce, Zoom, etc.

→ It is used by end-users, businesses, students, professionals, anyone who needs the application.

Ex: Gmail

Google Doc

Microsoft 365

Salesforce CRM

Dropbox

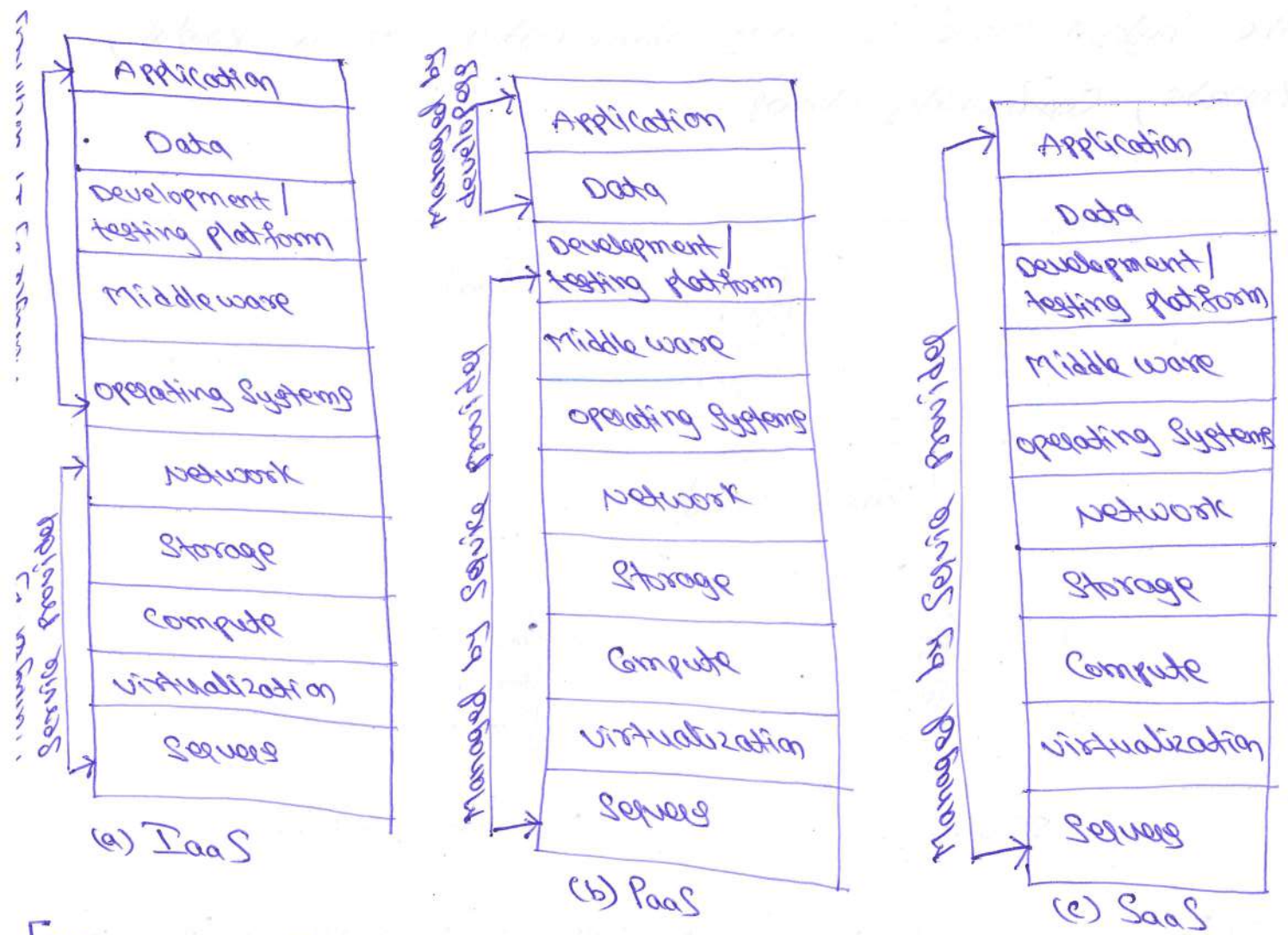
Zoom.

Note: In SaaS, the end users are responsible for maintaining the deployment platform, the SaaS providers are responsible for maintaining the underlying hardware, the PaaS of end users are responsible for managing the application that they have developed.

In IaaS, the end users are responsible for maintaining the development platform and the application running on the top of the underlying infrastructure. The IaaS providers are responsible for maintaining the underlying hardware.

→ In PaaS, the end users are responsible for managing the application that they have developed. The providers are responsible for underlying infrastructure will be maintained.

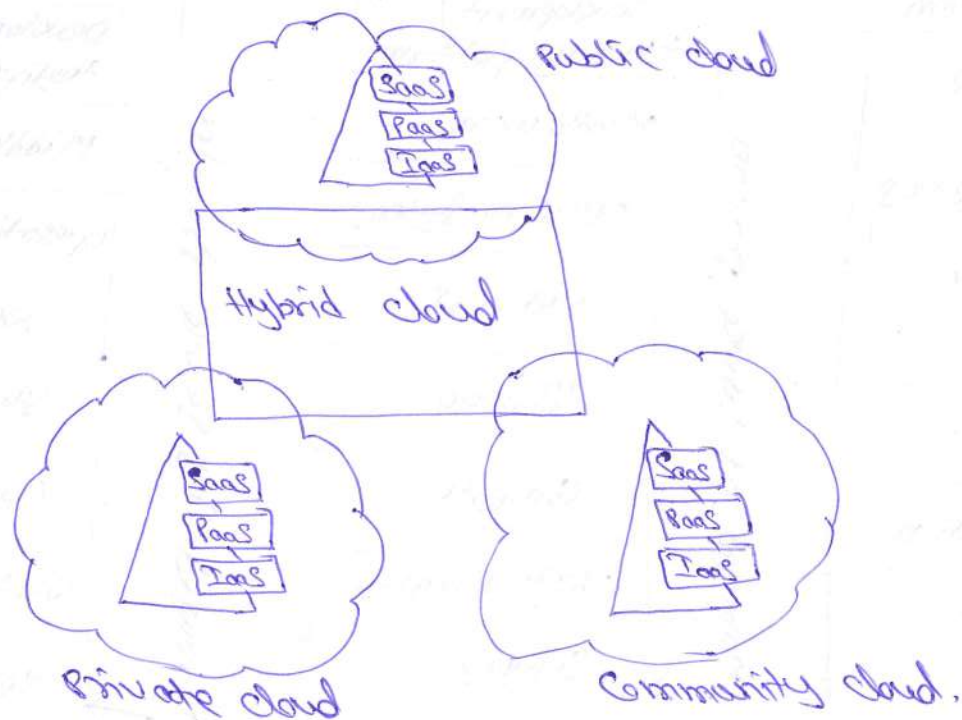
→ In SaaS, the end user is free from maintaining the infrastructure, development platform, and application that they are using. All the maintenance will be carried out by the SaaS providers.



[Fig: users and service provider responsibilities of cloud service models]

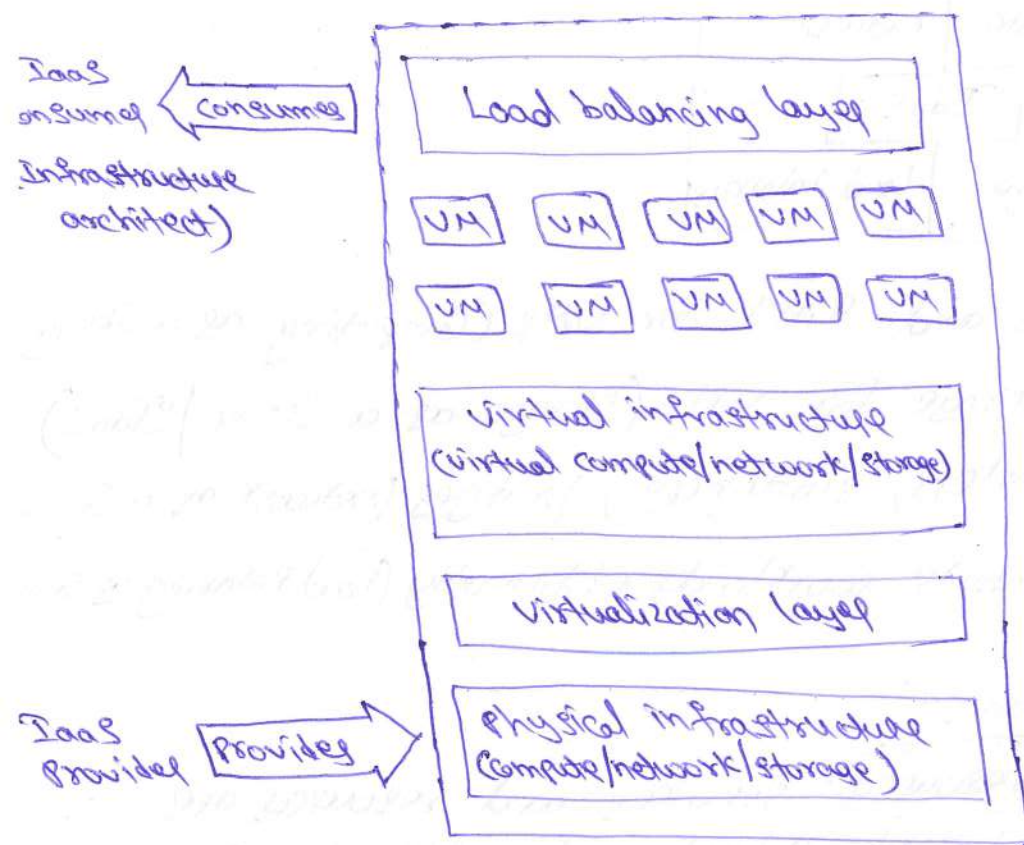
The different service models of cloud computing can be deployed and delivered through any one of the cloud deployment models.

- The NIST defined 4 types of cloud deployment models.
- The public cloud is provided for the general public
 - The private cloud is used by an organization for its multiple business units.
 - The Community cloud is for some group of organization with the same goals.
 - The hybrid cloud is any combination of the public, private, community clouds.



[Deployment and delivery of different cloud service delivery models]

1.1 Infrastructure as a Service:



[Fig: overview of IaaS]

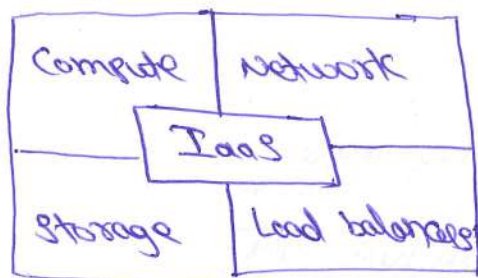
IaaS changes how computing, storage and networking resources are consumed.

- > Before cloud: In traditional data centers, computing power was accessed through physical infrastructure.
- > With IaaS: Computing shifts from physical to virtual infrastructure, delivered over the internet.
- > ^{it uses} virtual infrastructure technology to provide virtual machines (VM), where each VM can have its own computing power, storage and networking.

The figure explains:

- > Physical infrastructure: compute/network/storage
- > virtualization layer that abstracts physical resources.
- > virtual infrastructure in the form of VMs
- > The load balancing layer to distribute workload among VMs

A typical IaaS provider may provide the following services as:



- Compute: virtual CPUs and RAM for VMs (Computing as a Service)
- Storage: Backend storage for VMs (Storage as a Service / SaaS)
- Network: virtual routers, switches, bridges (Network as a Service)
- Load Balancing: distribute workloads efficiently (Load Balancing as Service)

Characteristics of IaaS:

web access to the resources: Services and resources are accessed via the internet through a browser or API, without needing local installation.

centralized management: the infrastructure (servers, storage, networking) is monitored & managed from a central control point.

Elasticity and dynamic scaling: Resources can be quickly increased or decreased based on workload demands.

shared infrastructure: Multiple customers share the same physical hardware, with logical separation for security.

Preconfigured VMs: virtual machines come with pre-installed OS and basic software, ready for use.

Metered services: pay-as-you-go pricing, while billing is based on actual resource usage.

Suitability of IaaS:

IaaS reduces the total cost of ownership (TCO) & increases the return on investment (ROI) for start-up companies that cannot invest more in buying infrastructure.

IaaS can be used in the following situations:

- Unpredictable spikes in usage (spikes → sudden)
- Limited capital investment
- Infrastructure on demand.

Avoid using the IaaS:

- when regulatory compliance does not allow off-premise hosting
- when usage is minimal
- when better performance is required.
- when there is a need for more control on physical infrastructure

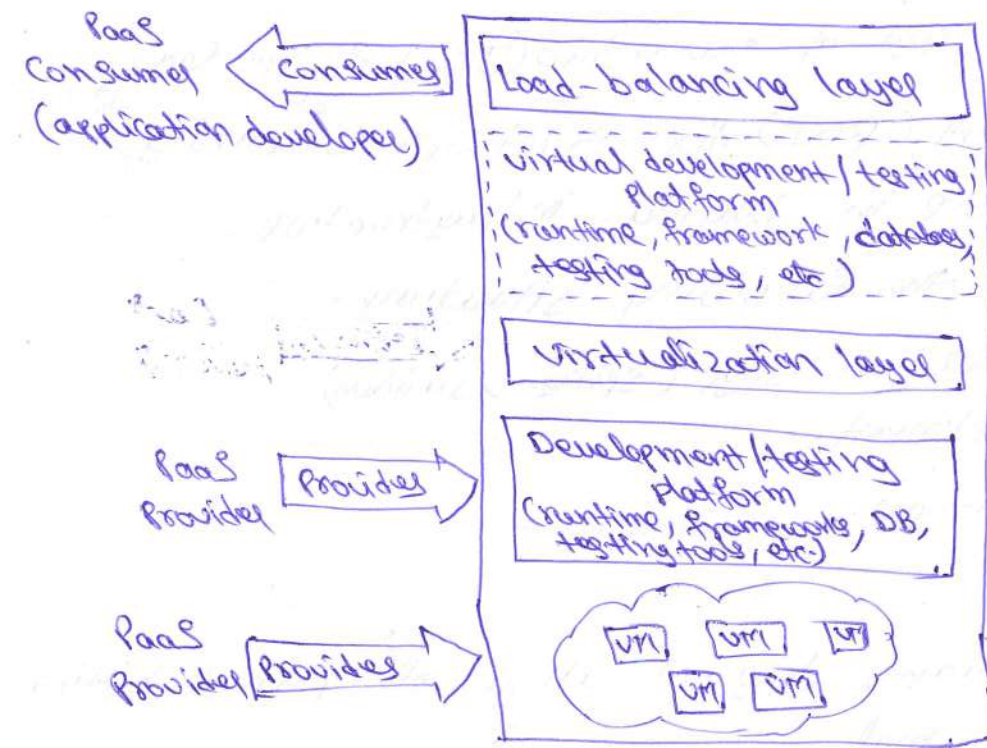
Benefits:

- Pay-as-you-use model: This model eliminates the unnecessary spending on buying hardware.
- Reduced TCO: IaaS reduces the need for buying hardware resources and thus reduces the TCO.
- Elastic resources: The load balancer transfers the additional resource request to the new server and improves application efficiently.

Drawbacks:

- Security Issues
- Interoperability issues
- Performance issues

2 Platform as a Service:



[Fig: overview of PaaS]

PaaS provides a complete development and deployment environment in the cloud.

> It includes tools, frameworks, runtime environments, database and testing tools so that developers can build, test, and deploy applications without managing underlying infrastructure.

> PaaS providers handle the servers, storage, networking, and virtualization, so developers can focus only on writing and running code.

How it works:

PaaS providers: They own and maintain the physical & virtual infrastructure.

Virtualization layer: Converts physical resources into VMs.

Virtual development / testing platform offers:

Runtime environments (Java, Python, .NET, etc)

Frameworks (Spring, Django, Rails, etc)

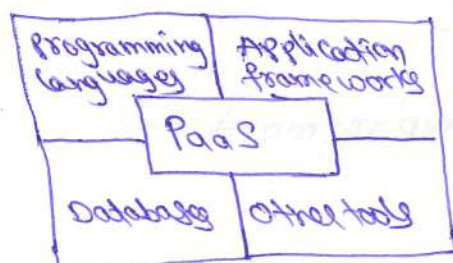
Databases

Testing tools

Load Balancing Layer: Distributed application workload across multiple servers for better performance.

PaaS Consumers: Usually application developers who use the platform to develop and deploy apps.

Services provided by PaaS providers are



Programming languages: PaaS providers offer multiple languages for development (Java, PHP, Python, Perl, Ruby, Scala).

Application frameworks: Ready-made frameworks to speed up development (Django, Spring, Node.js, Rack).

Database services: Since all applications need a database, PaaS often includes them (ClerkDB, Cloudant, MongoDB).

Other tools: Includes build tools, deployment tools, testing tools, and software load balancers. The purpose is to fully support development, testing, and deployment.

Characteristics of PaaS:

All in one: PaaS providers offer a single platform to develop, test, deploy, host, and maintain applications within same IDE.

Web access to the development platform: PaaS allows access via web-based UI, so developers can work from anywhere without installing a local IDE.

Offline Access: Some PaaS providers allow offline development by letting developers sync their local IDE with the cloud platform when internet connectivity is restored.

Suitability of PaaS:

Most of the startup PaaS development companies and independent software vendors widely use PaaS in developing an application.

PaaS technology is getting attention from other traditional software development companies also.

PaaS is suitable option for:

- > Collaborative development
- > Automated testing and deployment
- > Time to market.

Not suitable for:

- > Frequent application migration
- > Customization at the infrastructure level.
- > Integration with on-premise application.

Advantages:

- > Quick development and deployment
- > Reduces TCO.
- > Different teams can work together
- > Ease to use.
- > Less maintenance overhead.

Disadvantage:

- > Vendor lock-in
- > Security issue.
- > Less flexibility
- > Depends on Internet connection.

2.3 Software as a Service:

- SaaS is a cloud model where software is hosted by a Service provider and delivered to customers over the internet.
- Unlike the traditional software model (where we buy a license and install it locally), SaaS runs online and is accessed through a web browser or lightweight client.
 - The software runs on the provider's servers.
 - Users simply log in through the internet. No installation, updates or maintenance required on their device.

SaaS may provide business services, social n/w, document management and mail services.



Business Services: SaaS providers offer ERP (Enterprise Resource planning), CRM (customer Relationship Management), billing, sales tracking, HR management, etc.

Social networks: Social networking platforms (Facebook, LinkedIn) use SaaS to handle massive and fluctuating user loads.

Document Management: cloud services to create, manage, store, and track electronic documents (Google Doc, Microsoft 365)

Mail Services: Email platforms like Gmail, outlook.com which cloud makes it easier to handle unpredictable growth in users and email volume.

Characteristics of SaaS

One to many: Single instance of the application can be shared by multiple tenants or customers.

web access: SaaS is delivered over the internet through a web browser, no need for installation on device.

Centralized management: All updates, upgrades and maintenance are handled by the SaaS provider from a central location.

Multidevice support: SaaS can be accessed from various device like desktops, laptops, tablets, smartphones, thin clients.

Sustainability of SaaS

- on demand software
- software for start-up companies
- software compatible with multiple devices.
- s/w with varying loads.

Not suitable for:

- Real-time application
- Better on-premise application.

Advantages:

- No client side installation
- Cost savings
- Less maintenance
- Easy to access
- Disaster recovery.
- Multitenancy.

Disadvantages:

- Security
- Connectivity requirements
- Loss of control.

3) Technological Drivers for cloud Computing

It is to identify the various technological drivers of CC paradigm. The CC enables service providers to offer various resources such as IaaS, PaaS and SaaS to the requesting users on a pay-as-you-go model.

- The cloud service consumers (CSCs) are benefited from reduced cost and high quality of service (QoS).
- Enabling technologies are the success of cloud computing depends on technologies like "Service oriented Architecture (SOA)", virtualization, multicore processors, storage, networking, web 2.0/3.0, programming models, OS, pervasive computing, application environments.

SOA is need for
→ loose coupling
→ Reusability
→ Scalability & flexibility
→ interoperability
→ cost efficiency

3.1 SOA and cloud

SOA is a flexible set of design principles and standards used for systems development and integration.

- The SOA is an architectural style (a design approach) for building software applications.
- A properly implemented SOA-based systems organizes software into independent, reusable and loosely coupled services that can be used by the service consumers within various business domains.
- But the cloud computing is a service delivery model in which shared services and resources used by users across the internet.
- * → SOA is not cloud, but cloud often uses SOA principles to provide flexible, reusable services.
- SOA is used for availing services such as financial services, health-care services, manufacturing services and HR services.

3.2 SOA and SOC

SOA → Architectural model (the "design" of services).

SOC → Paradigm (the "use" of SOA for developing distributed applications).

The service-oriented computing (SOC) is the computing paradigm that uses SOA to build interoperable distributed applications.
↳ (work across diff environments)

→ It focuses on the rapid and low-cost development of applications across heterogeneous platforms.

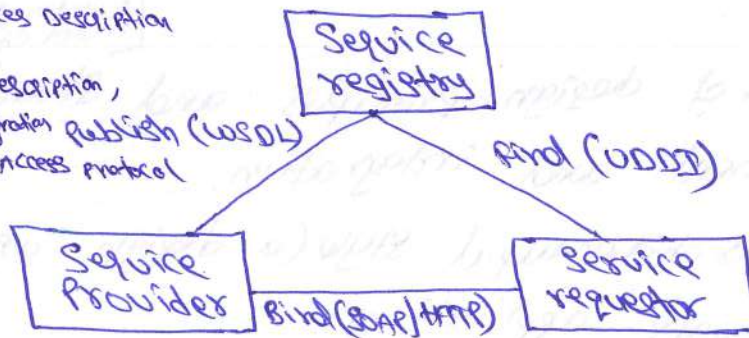
→ SOC makes it possible to create cooperating services that are loosely coupled, and these services can be used to create dynamic business process & agile applications.
↳ (means changing to another without breaking the whole system)

→ SOC uses the services architectural model of SOA as:

WSDL → web services description language

UDDI → universal description, discovery & integration publish (WSDL)

SOAP → simple object access protocol



[Fig: Services architectural model of SOA]

Service provider → creates and publishes service

Service Registry → stores service descriptions (WSDL) so they can be discovered.

Service Requestor → finds and uses the required service.

3.3 Benefits of SOA

SOA enables cloud computing to offer on-demand, reusable, loosely coupled services, extending beyond enterprise boundaries.

1) Reuse of services: Service can be reused, it reduces development and maintenance cost.

2) Agility: Quick adoption of changes via standards like web services.

3) Monitoring: It helps monitor and optimize performance of services.

4) Extended Reach: Enterprises can collaborate & share processes more effectively.

3.4 Technologies used by SOA

- 1) web service: It enable platform, independent access to the services over the internet.
- 2) Simple object Access protocol (SOAP): The protocol is used for communication between web services.
- 3) RPC: Remote procedure call (RPC) allows running procedures on remote computers without knowing network details.
- 4) RMI-IIOP: This denotes Java remote method invocation (RMI) interface over the internet inter-ORB protocol (IIOP). In this Java's interface for distributed computing across platform using "Common object Request Broker Architecture (CORBA)".
- 5) REST: Representational state transfer (REST) is a stateless architecture that runs over HTTP. It used for interacting b/w clients and services.

3.5 Similarities and Differences b/w SOA and cloud Computing

There are certain common features that SOA and cloud Computing share while being different from one another in certain other areas.

Similarities:

- Both SOA and cloud Computing use a service-based approach to provide solutions and hide technical complexities from users.
- They allow loose coupling, meaning services interact without revealing internal details.
- Common standards, like XML and WSDL, are used to ensure interoperability and transparency.
- They keep the internal details of services hidden, making them easy for users to access and use.

Differences:

- SOA focuses on solving specific business problems by assembling services (vertical orientation).
- cloud computing offers reusable, general services like PaaS, SaaS, IaaS which can be used by multiple applications.
- SOA services are designed for integration in a business application, while cloud services are broadly reusable across different consumers.

3.6 How SOA meets cloud Computing

SOA helps build flexible, reusable, and scalable software services, while cloud computing offers a platform to deploy these services efficiently. Together, they make IT more responsive, cost-effective and adaptable.

SOA as an Enabler: SOA is an enabling technology for cloud computing, providing the structure needed for virtualization, elasticity, and secure resource sharing.

Service-oriented Software: SOA organizes IT resources as independent services (using standards like SOAP, WSDL, UDDI) that can be combined or reused across platforms, vendors, or technologies.

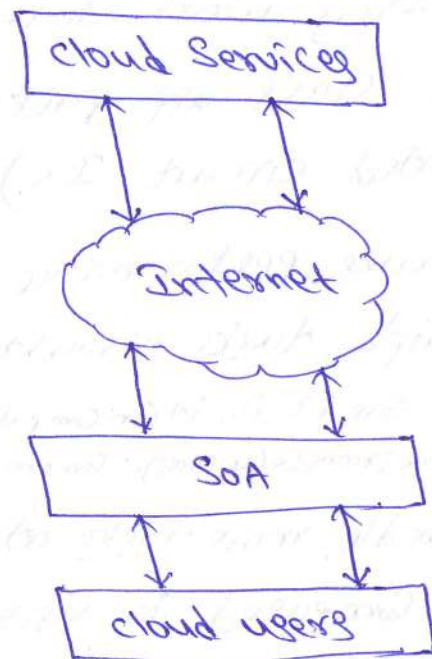
Integration and Flexibility: SOA allows organizations to mix/match resources from different service providers (SPs), supporting flexible and cost-effective deployment of cloud based applications.

Loose coupling & Reusability: SOA builds loosely-coupled applications, whose services can be reused or composed for different business needs, enhancing agility.

"X as a Service" model: Together, SOA & cloud offer various standardized service models (SaaS, PaaS), providing repeatable solution at low cost.

3.7 CCOA

cloud computing open architecture (CCOA) is an architecture for the cloud environment that incorporates the SOA.



[CCOA is a cloud reference architecture that uses SOA principles to design flexible, interoperable, & scalable cloud systems]

[Fig: Convergence of SOA & cloud]

The goals of the CCOA are as follows:

- a) To develop an architecture that is reusable and scalable. That means, in future, the architecture should incorporate any further changes without the need for replacing the entire architecture.
 - b) To develop a uniform platform for the cloud application development. This will allow the cloud users to switch b/w the CSPs without the need of make significant changes in the application.
 - c) To enable the businesses to run efficiently. This goal helps the CSPs to make more money by delivering quality service successfully.
- * The CCOA provides the required general guidelines and instructions for the design and development of scalable, reusable, and interoperable cloud applications incorporating SOA principles into them.

4) Multicore Technology

Multicore technology means having two or more CPUs (cores) inside a single chip (processor).

- Each core can handle its own tasks, but since they are on the same chip, they work together efficiently.
- The chip where multiple cores are packaged is called a die (part of the integrated circuit - IC).
- It's purpose is to improve performance by allowing the system to run multiple tasks simultaneously (parallel processing).

Benefits:

[Ex] Intel Core i3, i5, i7 (in Computers & laptops)
Quad-Core processor (4 cores) → Common in mid-range phones

- 1) Better performance (Can handle more tasks at once)
- 2) Lower power consumption compared to using separate processors
- 3) Efficient parallel processing
- 4) widely used in desktops, laptops, servers, workstations and cloud computing.

4.1 Multicore processors and VM Scalability

- In cloud systems, multiple virtual machines (VM) run on the same physical machine.
- Multicore processors allow many VMs to scale efficiently as long as memory, bus, and bandwidth limits are not exceeded.
- CPU and memory-intensive workloads in the cloud should scale up to the maximum allowed by the memory architecture.

What is core?

A core is the processing unit inside the CPU that reads instructions (from programs), performing calculations & execute tasks.

- The CPU (processor) is the "brain" of the computer. Inside the CPU, each core is like (26) a mini-brain that can work independently.

4.2 Multicore Technology and the Parallelism in cloud

- It means instead of one big processor, multiple smaller ones (cores) are used.
- This allows parallelism - many tasks can run at the same time.
- In cloud computing, parallelism improves:
 - * Speed
 - * Efficiency
 - * Overall performance of multi-tenant environments.
- challenge: To fully benefit, cloud systems must be programmed efficiently to exploit multicore chips.

4.3 Case Study (Intel Atom C2000 Soc)

Intel launched the second-generation Atom C2000 Soc (System on Chip) for microservers.

Key features:

- 64-bit, upto 8 cores
- Based on Silvermont microarchitecture (energy-efficient)
- Low TDP (Thermal Design power) it consumes less power.
- Supports upto 64GB memory
- Designed for software defined networking (SDN), storage, and networking platforms.
- Best suited for light workload like web hosting and entry-level servers.

Use case:

Hosting companies (eg., I&I, Ericsson) use atom C2000 in their servers to handle multiple customer workload best effectively.

⑤ Web 2.0

Web 2.0 refers to the second generation of the world wide web where the internet moved from being static & read-only (web 1.0) to dynamic, interactive and user-driven.

→ Actually web 2.0 is the "social and participatory web" where users can create, share, and interact with content instead of just reading it which means "read-write".

* The term web 2.0 was "Tim O'Reilly (2004)". popularized.

→ One of the most significant difference b/w web 2.0 & web 1.0 is that web 2.0 facilitates greater collaboration and information sharing among internet users, content providers, and enterprises.

→ Technically, this change/shift was enabled by:

AJAX (Asynchronous JavaScript and XML)

RSS (Really Simple Syndication)

→ web 1.0: mostly static, one-way publishing ("read-only").

* → web 2.0: dynamic, many-to-many interaction ("read-write").

Real-Time examples:

1) Social Media platforms:

Facebook → post photos, comment, like, share

Instagram → share reels, stories, interact with each other.

Twitter → post tweets, reply, follow trends.

2) Video sharing & streaming:

Youtube → users upload, comment, subscribe and interact

TikTok → short video creation and sharing with global interaction.

3) collaboration & knowledge sharing:

wikipedia → Anyone can edit and add knowledge.

Quora → users ask, answer and discuss topics.

4) Communication Tools:

WhatsApp/Telegram → Instant chatting, voice/video calls, groups.

Zoom/Google meet → It is a Real-time video conferencing.

5.1 characteristics of web 2.0:

In the web 2.0 websites, instead of reading the contents from a web page, a user is allowed to write or contribute to the content available to everyone in an effective & user-friendly manner. "web 2.0 is also called network as a platform computing" as it provides software, computing and storage facilities to the user all through the browser.

The major applications of web 2.0 include

- Social networking sites
- Self-publishing platform
- Tagging
- Social bookmarking.

Key features of web 2.0 include:

- a) Folksonomy: Users tag and classify content (hashtags, keywords), it makes search and discovery easier.
- b) Rich user Experience: It is interactive, responsive, app-like web pages like (AJAX, dynamic content).
- c) User as contributor: Users create and upload content (blogs, vlogs, review).
- d) User participation: Users actively collaborate, comment, rate and improve content.

Some of the characteristic features are:

- a) Blogging: It allows users to publish posts (like online diaries or Journals).
- b) Usage of AJAX and new web technologies: Ajax is a way of developing web applications that combines XHTML and CSS.
- c) RSS - generated syndication: It is a format for syndicating web content. It allows to feed the freshly published web content to the users through the RSS reader.
- d) Social bookmarking: It is a user-defined taxonomy system for storing tags to web contents. The taxonomy is called "Folksonomy". Instead of storing bookmarks in a folder on the user's computer, tagged pages are (29) stored on the web.

5.2 Difference b/w web 1.0 and web 2.0

Feature	web 1.0	web 2.0
Authoring mechanism	It has personal websites (few authors, manual updates)	Blogging (many authors, frequent posts, comments)
Information Sources	Encyclopedias like Britannica (Centralized, curated)	Wikipedia (crowd-created, continuously updated)
Content Creation & maintenance	Managed by site owners/editors, often via CMS (Content management system)	Created/edited openly via wikis and also user tools.
Data storage	Local disk (on your machine/server)	Online storage/cloud (Google drive)
Online advertising	Done by static Banners	Done by contextual/ad-network models (Google AdSense)
Online Payment	Direct bank transfers	Pay Pal and similar web payment gateways

5.3 Applications of web 2.0

1) Social Media:

- It is a key application of web 2.0.
- It changes how people communicate, interact, and share information.
- It provides platforms (Facebook, Twitter, Instagram, etc) for users to share data, ideas, opinions with communities.

2) Marketing:

- web 2.0 offers excellent opportunities for marketing by engaging customers in various stages of the product development cycle.
- web 2.0 allows direct engagement with customers. Companies can collaborate with consumers for feedback, service improvements, promotions.
- Social n/w (Twitter, Yelp, Facebook, etc) help in multichannel promotion of products. The information becomes user-friendly, and spreads quickly, reaching more customers efficiently (Eg: Amazon reviews, viral influencer ads).

3) Education:

- web 2.0 improves education by providing collaboration opportunities for students and teachers.
- Students can share learning with peers easily, making education more interactive.
- Ex: Google classroom, online academies (30)

5.4 web 2.0 and Cloud Computing

1) web 2.0:

- web 2.0 introduced interactive and collaborative features on the web
- The data is described using XML, which computers can read automatically
- protocols like SOAP, WSDL, UDDI allow integration of applications developed in different programming languages & platforms.
- The Applications can be hosted on the web and accessed by "geographically distributed users".
- These applications are "interoperable" and can be discovered dynamically (without pre-configuration).

2) Cloud Computing Business model:

- Provides computing resources (processors, storage, memory, OS, software) as services over the internet.
- It works on a "pay-per-use model", reducing cost for users.
- Resources (hardware/software) are shared by multiple client on demand
 - ↳ Based on:
 - * SOA (Service-oriented Architecture) of web 2.0
 - * Virtualization of hardware/software resources.

3) SaaS and web 2.0 influence:

- web 2.0 apps (like social n/w, online tools) encouraged SaaS.
- SaaS allows:
 - ↳ online collaboration (e.g., Google Docs, Office 365),
 - ↳ sharing & distribution of data/information.
- the cloud provides the platform to access hardware, s/w, & data resources via the internet.

[note]

web 2.0 made the internet interactive & collaborative, it introduced technologies and standards (like XML, SOAP, WSDL, UDDI) that allowed diff applications to work together across platforms.

the cloud computing is cheaper, faster, and easier to maintain, which is why it is considered the future of internet-based computing.

- SaaS applications (like Google Docs, Dropbox, Zoom) are examples of this model in action.

(6) Web 3.0

The name web 3.0 was given by "John Markoff" to this third generation of the web. The web 3.0 is also called the "intelligent web or semantic web".

Web 1.0: (Read-only web):

- It was the first generation of the web.
- It focus on building the web, making it accessible and also commercializing it.
- Technologies: HTTP, HTML, XML, ISPs, web browsers, Java, JavaScript.
- In this (web 1.0) users could only read content, little to no interactivity.

Web 2.0 (Read/write web):

- It was coined by Tim O'Reilly.
- It emphasis for collaboration, sharing, and interaction.
- Ex: Social networks, wikis, blogs, communication tools.
- The features are user-generated content, tagging, blogs, wikis.

Web 3.0 (Read/write/Execute web-intelligent web):

- It was coined by John Markoff (NY Times) → New York Times
- Technologies such as Semantic web, NLP, ML, & AI to achieve machine-facilitated understanding of information in order to provide a more productive & intuitive experience of the users.

Real time examples:

- 1) Payments (Crypto in Daily life):
 - ↳ Starbucks (USA) → Accepts Bitcoin Payments through Bakkt App
- 2) Fashion & Shopping:
 - ↳ Nike Cryptokicks (NFT sneakers) → You can own digital sneakers
- 3) Healthcare:
 - ↳ BuylstIQ → manages & secures patient health records.

6.1 Components of web 3.0

Semantic web:

- The term web 3.0, also known as the "semantic web", describes sites wherein the computers will be generating raw data on their own without direct user interaction.
- The semantic web provides the web user a common framework that could be used to share and reuse the data across various applications, enterprises and community boundaries.
- The semantic web is a vision of IT, that allows the data and information to be readily interpreted by machines so that the machines are able to take contextual decisions on their own by finding, combining, and acting upon relevant information on the web.
- The semantic web is originally envisioned, is a system enabling the machines to understand the context and meaning of complex human requests & respond to them appropriately.

web Services:

- A web service is a software system that supports computer-to-computer interaction over the internet.
- web services are usually represented as API (Application Programming Interface / API) → it is not an app of a complete sw, it is the messenger or interface that allows this app to talk to other apps, services or devices.

Ex: Google Maps (App/software)
↳ The app you install on our phone.
Google Maps API (API/web service)
↳ Let's other apps (like uber, swiggy, etc) use maps, route, distance calculations inside their own software.

→ Flickr is the best example of web service, because it is a photo & video sharing platform. Millions of users upload & tag photos. Every photo has metadata: title, description, tags, geolocation, date, camera details. This service is help for developers want to reuse this data in ⁽³³⁾ their apps.

6-2 characteristics of web 3.0

- a) Ubiquitous connectivity: In this internet access anytime, anywhere across devices (PC, mobile, IoT). users and machines are continuously connected.
- b) Network computing: Applications and data move from the local computer to the cloud, software is delivered as SaaS, users subscribe to services instead of installing SW locally.
- c) Open Technologies: It built on "open standards & protocols" like (HTTP, HTML5, RDF, XML, JSON). It encourages interoperability among applications.
- d) Open Identity: It helps to port the user account from one service to another effectively. It means users can have one digital identity across services.
- e) The intelligent web: It uses AI, ML, NLP, reasoning engines. In this Applications can understand meaning, context, behind queries.

6-3 Convergence of cloud & web 3.0

- > web 2.0 → Made internet more interactive (social networking, sharing, communication).
- (semantic web, smart applications)
- > web 3.0 → Adds intelligence (Personalization, data portability, user-centric identity).
- > web 3.0 + cloud: cloud provides storage and computing power, web 3.0 makes services intelligent & user-focused, users can access data and applications anytime, anywhere and across multiple devices.

Examples:

- 1) Google Drive + AI (Docs/Sheets/Translate)
- 2) Facebook (Meta) → Open Graph + cloud
- 3) Netflix / Amazon Prime Video

① Pervasive Computing

Pervasive → Exact meaning existing everywhere, spreading through
Pervasive computing is a combination of technologies, such as Internet, AI, networking, voice recognition and wireless computing, used to make computing anywhere possible.

→ It's main goal is to make computing available everywhere, anytime, and with minimal user effort.

* → The computing means the process of using computers and related technology to solve problems, perform tasks, or process information. It involves both hardware (machines, devices) and software (programs, instructions) work together.

* → The pervasive computing is also known as "Ubiquitous Computing" which means computing that exists everywhere.

* → Pervasive computing devices are completely connected & constantly available.

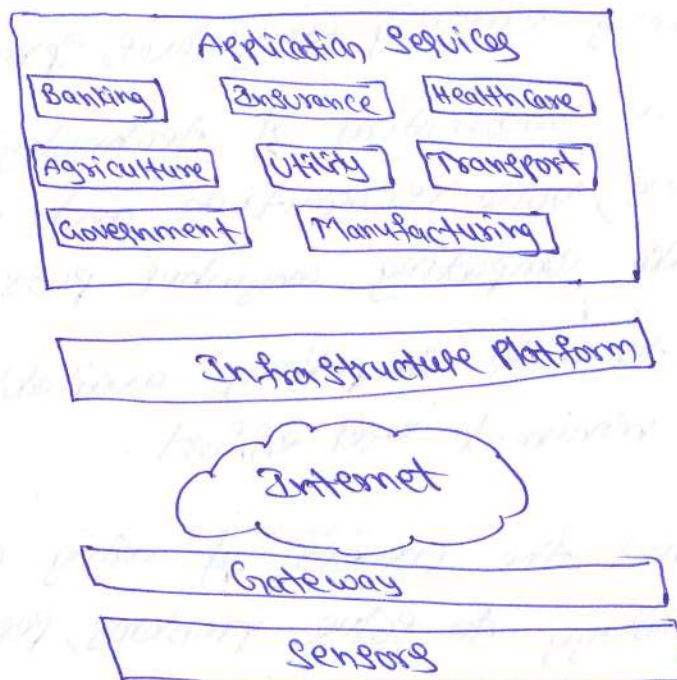
* → Pervasive computing relies on the convergence of wireless technologies, advance ~~for~~ electronics, and the Internet.

* → The goal of researchers working in pervasive computing is to create smart products that communicate ubiquitously.

Examples:

- 1) Smart meters, Smart lights (turn on only when motion is detected)
- 2) Bio metric Attendance (Face / fingerprint recognition instead of manual punch)
- 3) Smart Speakers (Amazon Alexa / Google Home Control appliances)
- 4) Smart watches (Track heart rate, steps, sleep)
- 5) Smart Glasses (Display directions, notifications in front of our eyes)
- 6) Tesla Self driving cars
- 7) Metro Cards (Contactless payments for transport)

7.1 How pervasive Computing works



[Fig: Pervasive Computing Stack]

1. Physical Layer (Sensors & Device)

> At the base, tiny sensors are attached to people, objects, animals, vehicle, or buildings. These sensors capture data from the environment like microphone, camera, compass, GPS, etc.
Ex: A smartwatch with a heart-rate sensor tracks your pulse continuously.

2. Wireless Communication Layer

> Sensors need to send collected data, it will be done using wireless communication. Through WiFi/Bluetooth it sends the data to mobile.

3. Gateway Layer

> Gateway will forward the data to the cloud. Gateways collect and process data from multiple sensors before sending it to the cloud. It acts as a "Bridge" between sensors and the Internet.

4. Internet & Cloud Layer

> The data moves from gateways to the "Internet & cloud servers". The cloud computing provides data storage, processing power (AI, analytics), and remote accessibility.

5. ~~App~~ Infrastructure Platform

> It is a backbone, because it provides computing, storage, networking, and scalability for processing sensor data.

6. Application Services

> It delivers end-user services across domains like healthcare, transport, banking, insurance, etc.