Module 2

INTRODUCTION TO CLOUD COMPUTING

System Models for Distributed and Cloud Computing

- **Distributed and Cloud computing systems :**
  - Built over a large number of autonomous computer nodes.
  - Interconnected by SANs, LANs, or WANs in a hierarchical manner.
  - LAN switches → connect hundreds of machines as a working cluster.
  - WAN → connect many local clusters to form a very large cluster of clusters.

- A massive system with millions of computers connected to edge networks can be build in this way.

- Massive systems are considered highly scalable, and can reach web-scale connectivity – physically or logically.

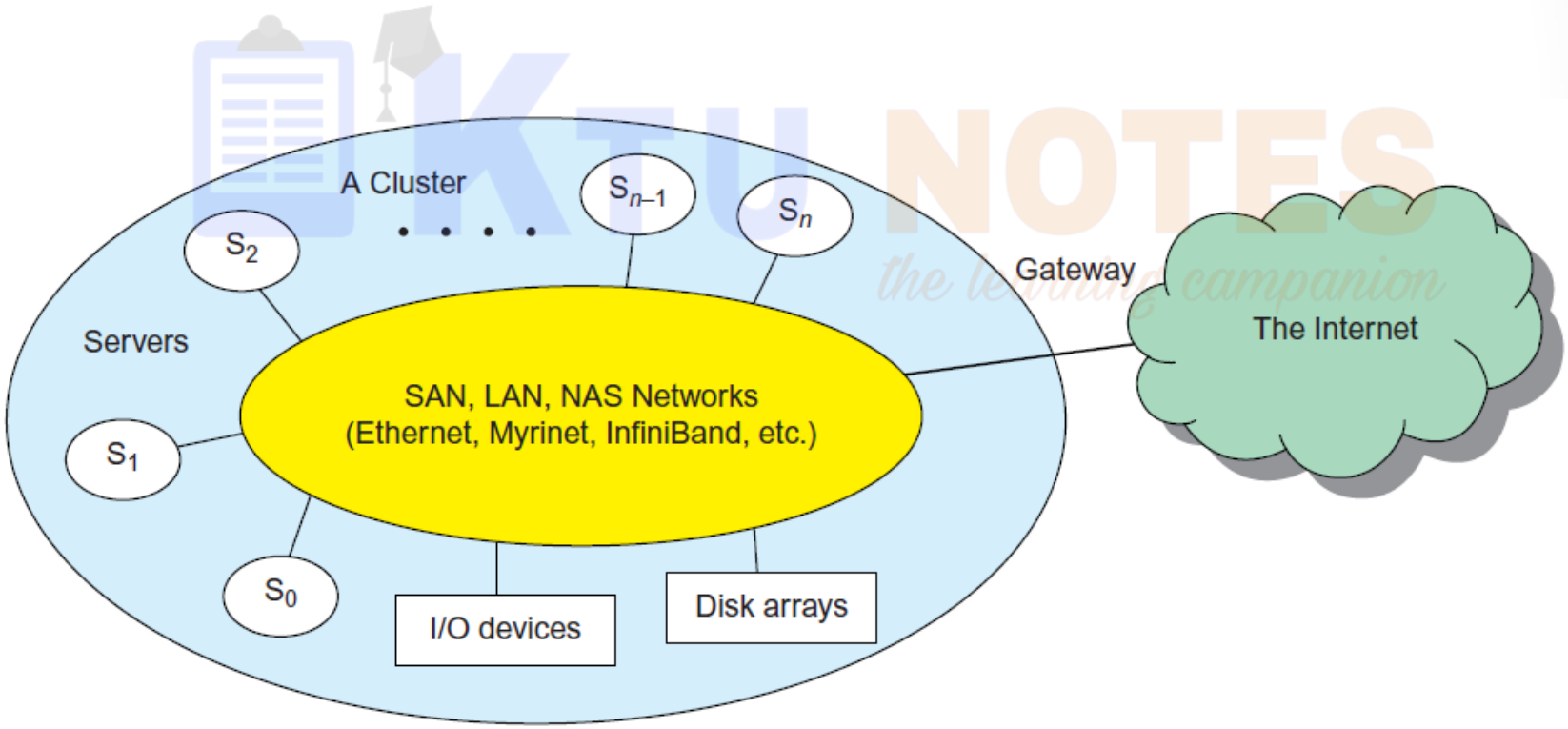
- Massive systems are classified into four groups:
  - Clusters
  - P2P networks
  - Computing grids
  - Internet clouds over huge data centers

- These four system classes may involve hundreds, thousands, or even millions of computers as participating nodes.

Clusters of Cooperative Computers

- Consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource.

- **Cluster Architecture**



- A typical server cluster built around a low-latency, high bandwidth interconnection network.

- Network can be:
  - a simple SAN (e.g., Myrinet)
  - a LAN (e.g., Ethernet)

- To build a larger cluster with more nodes, the interconnection network can be built with multiple levels of Gigabit Ethernet, Myrinet, or InfiniBand switches.

- Through hierarchical construction using a SAN, LAN, or WAN, one can build scalable clusters with an increasing number of nodes.

- The cluster is connected to the Internet via a virtual private network (VPN) gateway.

- The gateway IP address locates the cluster.

- Most clusters have loosely coupled node computers and their resources are managed by their own OS.
- So most clusters have multiple system images.
- **Single System Image (SSI):**
- An ideal cluster should merge multiple system images into a single-system image.
- A cluster operating system or some middleware is required to support SSI at various levels, including the sharing of CPUs, memory, and I/O across all cluster nodes.
- **SSI** → illusion created by software or hardware that presents a collection of resources as one integrated, powerful resource.
- SSI makes the cluster appear like a single machine to the user.
- A cluster with multiple system images is nothing but a collection of independent computers.

- **Hardware, Software, and Middleware Support:**

- Hardware:
  - PCs, workstations, servers, or
  - SMP

- Software:
  - Special communication software such as PVM or MPI
  - Network interface card in each computer node
  - Most clusters run under the Linux OS.

- The computer nodes are interconnected by a high-bandwidth network (such as Gigabit Ethernet, Myrinet, InfiniBand, etc.).

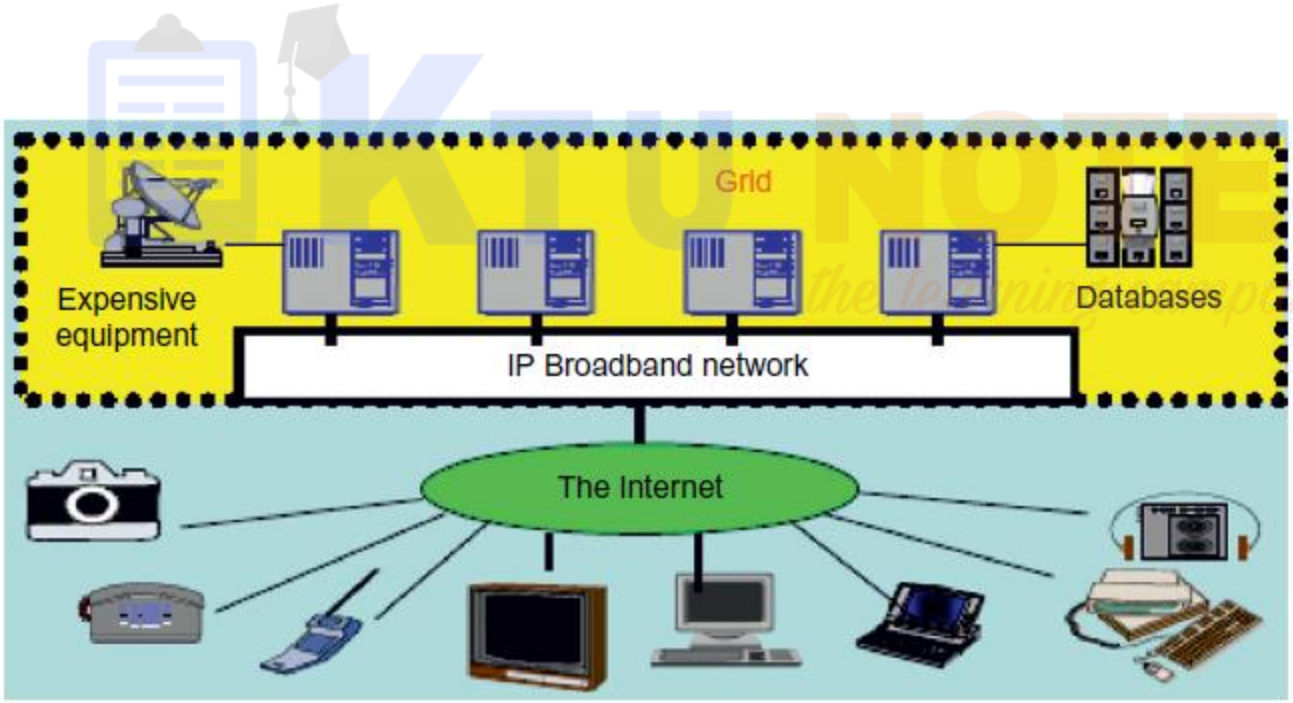
- Middleware:
  - Special cluster middleware supports are needed to create SSI.

Grid Computing Infrastructures

- An infrastructure that couples computers, software/middleware, special instruments, and people and sensors together.
- Constructed across LAN, WAN, or Internet backbone networks at a regional, national, or global scale.
- Mainly uses workstations, servers, clusters, and supercomputers.
- Personal computers, laptops, and PDAs can be used as access devices to a grid system.
- Enterprises or organizations present grids as integrated computing resources

- Computational grid built over multiple resource sites owned by different organizations.

- The resource sites offer complementary computing resources, including workstations, large servers, a mesh of processors, and Linux clusters to satisfy a chain of computational needs.

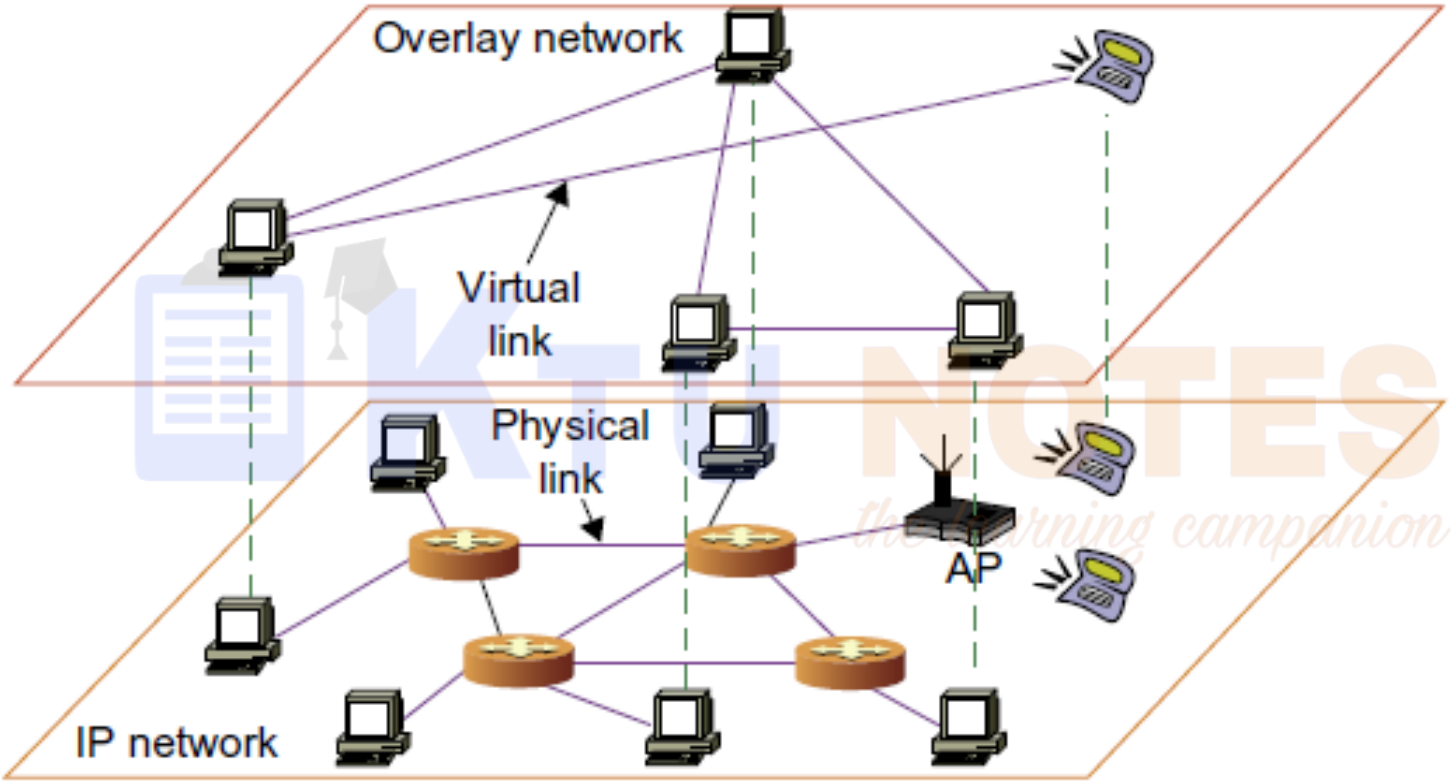


- The grid is built across various IP broadband networks including LANs and WANs already used by enterprises or organizations over the Internet.
- The grid is presented to users as an integrated resource pool
- Special instruments may be involved such as using the radio telescope in SETI@Home search of life in the galaxy
- At the client end →wired or wireless terminal devices.
- The grid integrates the computing, communication, contents, and transactions as rented services.
- Enterprises and consumers form the user base.
- Industrial grid platform development by IBM, Microsoft, Sun, HP, Dell, Cisco

Peer-to-Peer Network Families

- The P2P architecture offers a distributed model of networked systems.
- A P2P network is client-oriented instead of server-oriented.
- P2P systems are introduced at the physical level and overlay networks at the logical level.
- **P2P Systems:**
- Every node acts as both a client and a server, providing part of the system resources.
- Peer machines → client computers connected to the Internet.

- All client machines act autonomously to join or leave the system freely.
- No master-slave relationship exists among the peers.
- No central coordination or central database is needed.
- No peer machine has a global view of the entire P2P system.
- The system is self-organizing with distributed control.
- **Physical Network:**
- The participating peers form the physical network at any time.
- Unlike the cluster or grid, a P2P network does not use a dedicated interconnection network.
- The physical network is simply an ad hoc network formed at various Internet domains randomly using the TCP/IP and NAI protocols



- **Overlay Network:**

- Based on communication or file-sharing needs, the peer IDs form an overlay network at the logical level.

- This overlay is a virtual network formed by mapping each physical machine with its ID, logically, through a virtual mapping .

- When a new peer joins the system, its peer ID is added as a node in the overlay network and is removed from the overlay network automatically when it leaves.

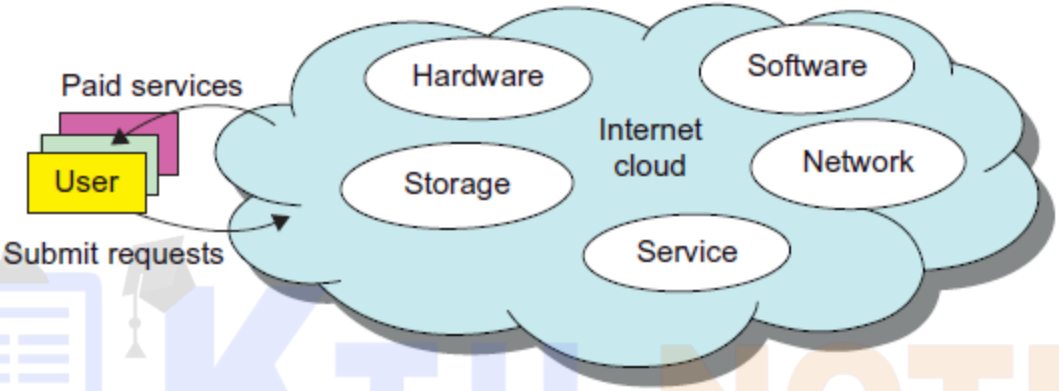
- Therefore, it is the P2P overlay network that characterizes the logical connectivity among the peers.

- Two types of overlay networks:
  - unstructured and structured
- An unstructured overlay network is characterized by a random graph.
- There is no fixed route to send messages or files among the nodes.
- Often, flooding is applied to send a query to all nodes in an unstructured overlay, thus resulting in heavy network traffic and nondeterministic search results.
- Structured overlay networks follow certain connectivity topology and rules for inserting and removing nodes (peer IDs) from the overlay graph.
- Routing mechanisms are developed to take advantage of the structured overlays.

Cloud Computing over the Internet

- **Definition of Cloud Computing by IBM:**
  - A cloud is a pool of virtualized computer resources. A cloud can host a variety of different workloads, including batch-style backend jobs and interactive and user-facing applications
- i.e. a cloud allows workloads to be deployed and scaled out quickly through rapid provisioning of virtual or physical machines.
- The cloud supports redundant, self-recovering, highly scalable programming models that allow workloads to recover from many unavoidable hardware/software failures.
- Finally, the cloud system should be able to monitor resource use in real time to enable rebalancing of allocations when needed.

- **Internet Clouds:**



- Cloud computing applies a virtualized platform with elastic resources on demand by provisioning hardware, software, and data sets dynamically.

- Cloud computing intends to satisfy many user applications simultaneously.

- The cloud ecosystem must be designed to be secure, trustworthy, and dependable.

Software Environments for Distributed Systems and Clouds

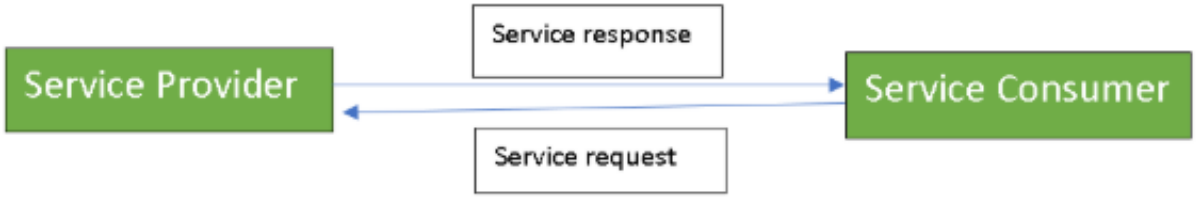
- **Service-Oriented Architecture (SOA) :**
  - An architectural approach in which applications make use of services available in the network.
  - An application's business logic or individual functions are modularized and presented as services for consumer/client applications.
  - Loosely coupled nature → the service interface is independent of the implementation.
  - Application developers or system integrators can build applications by composing one or more services without knowing the services' underlying implementations.
  - For example, a service can be implemented either in .Net or J2EE, and the application consuming the service can be on a different platform or language.

- There are two major roles within Service-oriented Architecture:

- **Service provider:** The service provider is the maintainer of the service and the organization that makes available one or more services for others to use.

- To advertise services, the provider can publish them in a registry, together with a service contract that specifies the nature of the service, how to use it, the requirements for the service, and the fees charged.

- **Service consumer:** The service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service.



- **Distributed Operating Systems:**

  - Tanenbaum identifies 3 approaches for distributing resource management functions in a distributed computer system.

  - The first approach is to build a **network OS** over a large number of heterogeneous OS platforms. Such an OS offers the lowest transparency to users, and is essentially a distributed file system, with independent computers relying on file sharing as a means of communication.

  - The second approach is to develop **middleware** to offer a limited degree of resource sharing, similar to the MOSIX/OS developed for clustered systems.

  - The third approach is to develop a truly **distributed OS** to achieve higher use or system transparency.

- A **distributed operating system** is a software over a collection of independent, networked, communicating, and physically separate computational nodes.
- They handle jobs which are serviced by multiple CPUs.
- Each individual node holds a specific software subset of the global aggregate operating system.
- Each subset is a composite of two distinct service provisioners.
- The first is a ubiquitous minimal kernel, or microkernel, that directly controls that node's hardware.
- Second is a higher-level collection of system management components that coordinate the node's individual and collaborative activities.
- These components abstract microkernel functions and support user applications

- **Parallel and Distributed Programming Models:**

- **Message-Passing Interface (MPI):**

- Primary programming standard used to develop parallel and concurrent programs to run on a distributed system.

- MPI is essentially a library of subprograms that can be called from C or FORTRAN to write parallel programs running on a distributed system.

- Synchronous or asynchronous point-to-point and collective communication commands and I/O operations in user programs for message-passing execution.

- MPI's goals are high performance, scalability, and portability.

- MPI is not agreed upon by any standards body, but it is the most widely used.

- **MapReduce:**

- Web programming model for scalable data processing on large clusters over large data sets.

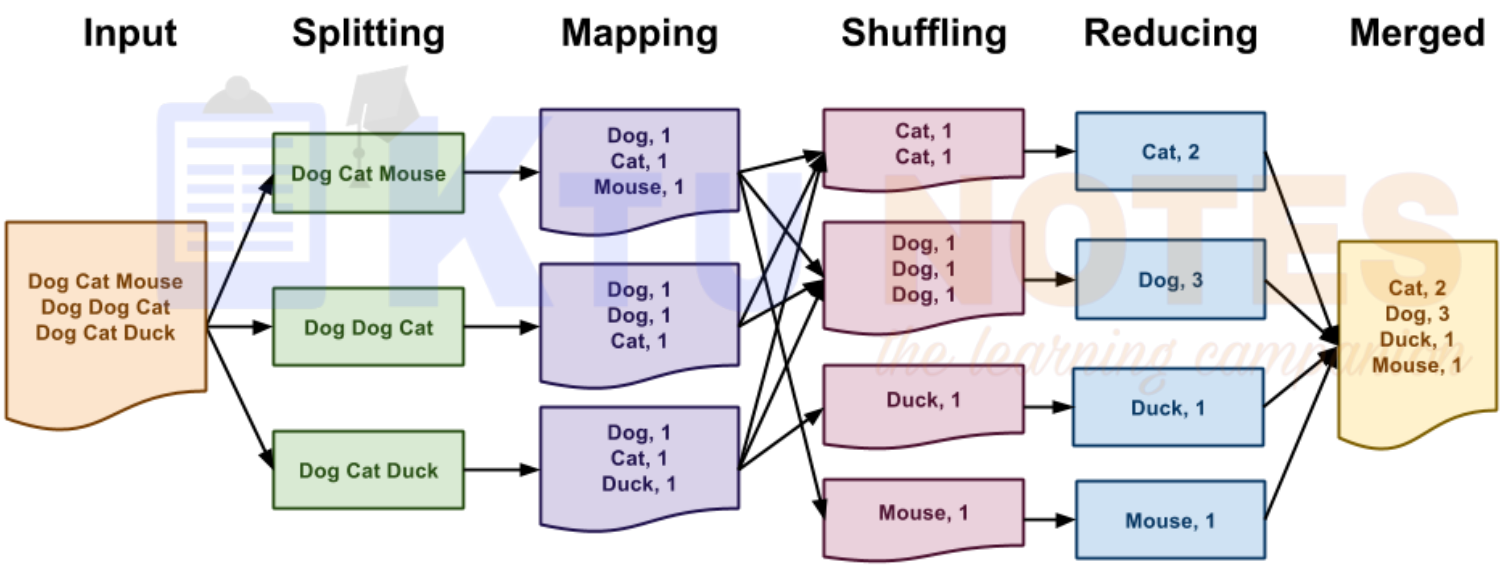
- Applied mainly in web-scale search and cloud computing applications.

- The user specifies a Map function to generate a set of intermediate key/value pairs.

- Then applies a Reduce function to merge all intermediate values with the same intermediate key.

- MapReduce is highly scalable to explore high degrees of parallelism at different job levels.

- A typical MapReduce computation process can handle terabytes of data on tens of thousands or more client machines.



- **Hadoop Library:**

- Software platform that was originally developed by a Yahoo! group.

- The package enables users to write and run applications over vast amounts of distributed data.

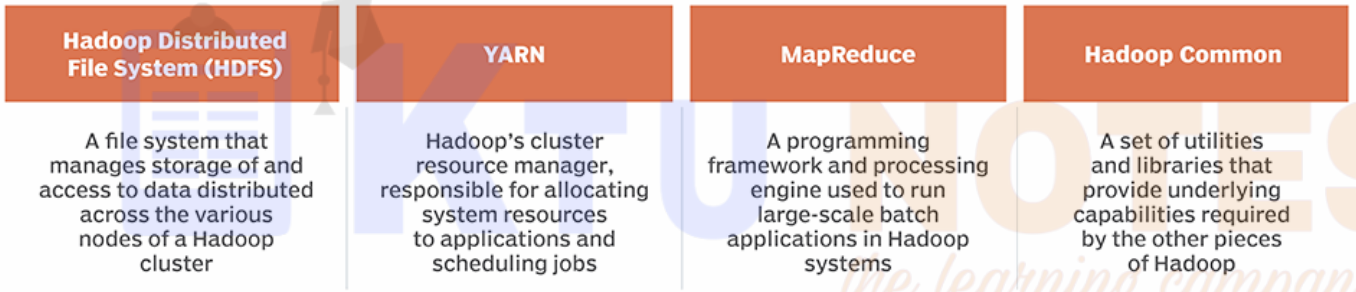
- **Scalability:** Users can easily scale Hadoop to store and process petabytes of data in the web space.

- **Economical**: Comes with an open source version of MapReduce that minimizes overhead in task spawning and massive data communication.

- **Efficient:** Processes data with a high degree of parallelism across a large number of commodity nodes.

- **Reliable:** Automatically keeps multiple data copies to facilitate redeployment of computing tasks upon unexpected system failures.

Hadoop's core components



Cloud Computing and Service Models

- **Public, Private, and Hybrid Clouds:**
  - Cloud computing has evolved from cluster, grid, and utility computing.
  - Cluster and grid computing leverage the use of many computers in parallel to solve problems of any size.
  - Utility and Software as a Service (SaaS) provide computing resources as a service with the notion of pay per use.
  - Cloud computing is a high-throughput computing (HTC) paradigm whereby the infrastructure provides the services through a large data center or server farms.

- **Public Clouds:**

  - A **public cloud** is built over the Internet and can be accessed by any user who has paid for the service.
  - Public clouds are owned by service providers and are accessible through a subscription.
  - Google App Engine (GAE), Amazon Web Services (AWS), Microsoft Azure, IBM Blue Cloud, and Salesforce.com's Force.com.
  - Commercial providers offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure.
  - A public cloud delivers a selected set of business processes.
  - The application and infrastructure services are offered on a flexible price-per-use basis.

- **Private Clouds:**

  - A **private cloud** is built within the domain of an intranet owned by a single organization.
  - Client owned and managed, and access is limited to the owning clients and their partners.
  - NOT meant to sell capacity over the Internet through publicly accessible interfaces.
  - Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains.
  - A private cloud is supposed to deliver more efficient and convenient cloud services.
  - It may impact the cloud standardization, while retaining greater customization and organizational control.

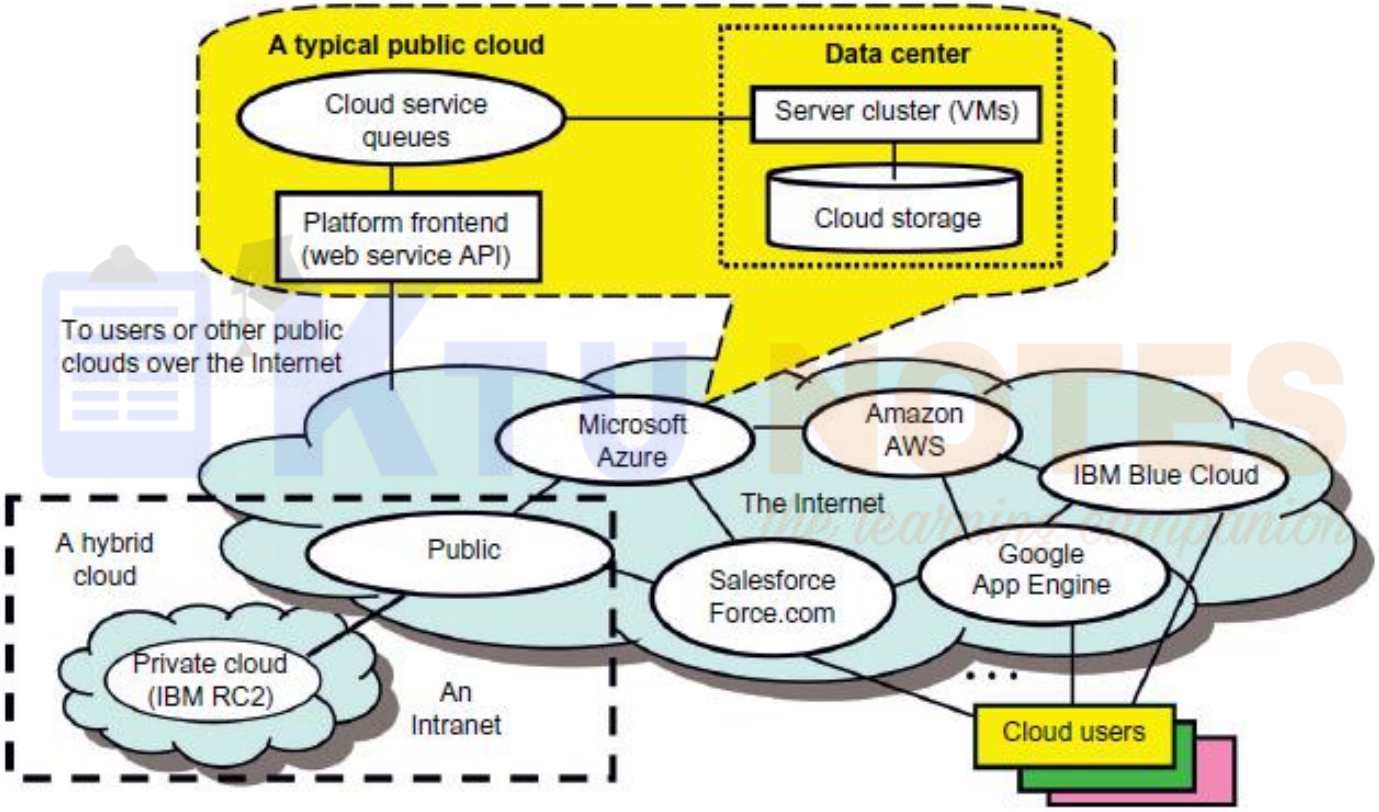
- **Hybrid Clouds:**

  - A hybrid cloud is built with both public and private clouds.

  - Private clouds can also support a hybrid cloud model by supplementing local infrastructure with computing capacity from an external public cloud.

  - The Research Compute Cloud (RC2) is a private cloud, built by IBM, that interconnects the computing and IT resources at eight IBM Research Centers scattered throughout the United States, Europe, and Asia.

  - A hybrid cloud provides access to clients, the partner network, and third parties.



- **Summary**

  - Public clouds promote standardization, preserve capital investment, and offer application flexibility.

  - Private clouds attempt to achieve customization and offer higher efficiency, resiliency, security, and privacy.

  - Hybrid clouds operate in the middle, with many compromises in terms of resource sharing.

- **Cloud Service Models:**

  - The services provided over the cloud can be generally categorized into three different service models:

    - **Infrastructure as a Service (IaaS)**
    - **Platform as a Service (PaaS)**
    - **Software as a Service (SaaS)**

  - These services are available as subscription-based services in a pay-as-you-go model to consumers.

  - All three models allow users to access services over the Internet

- **Infrastructure-as-a-Service (IaaS):**

  - This model allows users to use virtualized IT resources for computing, storage, and networking.

  - In short, the service is performed by rented cloud infrastructure.

  - The user can deploy and run his applications over his chosen OS environment.

  - The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components.

  - This IaaS model encompasses:
    - storage as a service, compute instances as a service, and communication as a service.

- **Infrastructure-as-a-Service (IaaS):**

  - This model allows users to use virtualized IT resources for computing, storage, and networking.

  - In short, the service is performed by rented cloud infrastructure.

  - The user can deploy and run his applications over his chosen OS environment.

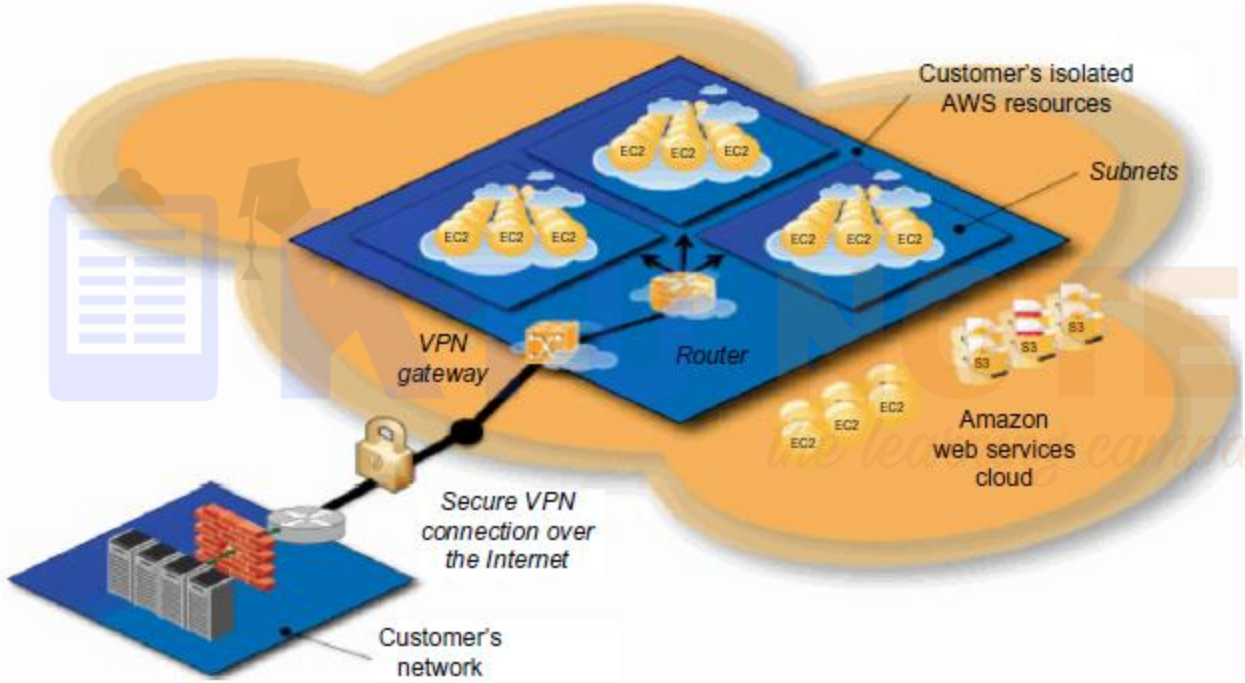
  - The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components.

  - This IaaS model encompasses:
    - storage as a service, compute instances as a service, and communication as a service.

- **Key features**

  - Instead of purchasing hardware outright, users pay for IaaS on demand.
  - Infrastructure is scalable depending on processing and storage needs.
  - Saves enterprises the costs of buying and maintaining their own hardware.
  - Because data is on the cloud, there can be no single point of failure.
  - Enables the virtualization of administrative tasks, freeing up time for other work.

- **Amazon Virtual Private Cloud (VPC)**



- **Public Cloud Offerings of IaaS**

Cloud Name	VM Instance Capacity	API and Access Tools	Hypervisor, Guest OS
Amazon EC2	Each instance has 1–20 EC2 processors, 1.7–15 GB of memory, and 160–1.69 TB of storage.	CLI or web Service (WS) portal	Xen, Linux, Windows
GoGrid	Each instance has 1–6 CPUs, 0.5–8 GB of memory, and 30–480 GB of storage.	REST, Java, PHP, Python, Ruby	Xen, Linux, Windows
Rackspace Cloud	Each instance has a four-core CPU, 0.25–16 GB of memory, and 10–620 GB of storage.	REST, Python, PHP, Java, C#, .NET	Xen, Linux
FlexiScale in the UK	Each instance has 1–4 CPUs, 0.5–16 GB of memory, and 20–270 GB of storage.	web console	Xen, Linux, Windows
Joyent Cloud	Each instance has up to eight CPUs, 0.25–32 GB of memory, and 30–480 GB of storage.	No specific API, SSH, Virtual/Min	OS-level virtualization, OpenSolaris

- **Platform as-a-Service (PaaS):**

  - This model provides users with a cloud environment in which they can develop, manage and deliver applications

  - Platform includes operating system and runtime library support

  - An integrated computer system consisting of both hardware and software infrastructure.

  - In addition to storage and other computing resources, users are able to use a suite of prebuilt tools to develop, customize and test their own applications.

  - The user application can be developed on this virtualized cloud platform using some programming languages and software tools supported by the provider (e.g., Java, Python, .NET).

- The user does not manage the underlying cloud infrastructure.
- Enables a collaborated software development platform for users from different parts of the world

- **Key Features:**
  - PaaS provides a platform with tools to test, develop and host applications in the same environment.
  - Enables organizations to focus on development without having to worry about underlying infrastructure.
  - Providers manage security, operating systems, server software and backups.
  - Facilitates collaborative work even if teams work remotely

- **Public Cloud Offerings of PaaS**

Cloud Name	Languages and Developer Tools	Programming Models Supported by Provider	Target Applications and Storage Option
Google App Engine	Python, Java, and Eclipse-based IDE	MapReduce, web programming on demand	Web applications and BigTable storage
Salesforce.com's Force.com	Apex, Eclipse-based IDE, web-based Wizard	Workflow, Excel-like formula, Web programming on demand	Business applications such as CRM
Microsoft Azure	.NET, Azure tools for MS Visual Studio	Unrestricted model	Enterprise and web applications
Amazon Elastic MapReduce	Hive, Pig, Cascading, Java, Ruby, Perl, Python, PHP, R, C++	MapReduce	Data processing and e-commerce
Aneka	.NET, stand-alone SDK	Threads, task, MapReduce	.NET enterprise applications, HPC

- **Software as-a-Service (SaaS):**

  - The SaaS model provides software applications as a service
  - Provides users with access to a vendor's cloud-based software.
  - Users do not install applications on their local devices.
  - Instead, the applications reside on a remote cloud network accessed through the web or an API.
  - Through the application, users can store and analyze data and collaborate on projects.
  - Example: Google Gmail and docs, Microsoft SharePoint, and the CRM software from Salesforce.com

- **Key features**

  - SaaS vendors provide users with software and applications via a subscription model.
  - Users do not have to manage, install or upgrade software; SaaS providers manage this.
  - Data is secure in the cloud; equipment failure does not result in loss of data.
  - Use of resources can be scaled depending on service needs.
  - Applications are accessible from almost any internet-connected device, from virtually anywhere in the world.