Cloud Computing - Jaish Khan

- 1. Cloud Computing
  - 1.1. Definition
  - 1.2. Characteristics
    - 1.2.1. Essential Characteristics
    - 1.2.2. Benefits
    - 1.2.3. Risks & Challenges
    - 1.2.4. Roles and Boundaries
  - 1.3. History and Background
    - 1.3.1. Grid Computing
    - 1.3.2. Cluster Computing
    - 1.3.3. Differences between Cluster, Grid and Cloud
  - 1.4. Cloud Service Models
  - 1.5. Cloud Deployment Models
  - 1.6. Basics of Computing
  - 1.7. Other Services of Cloud Computing
    - 1.7.1 Cloud-Based Data Storage
    - 1.7.2. Cloud Backup Systems
    - 1.7.3. Identity as a Service (IDaaS)
    - 1.7.4. Collaboration
  - 1.8. Cloud Security Threats
  - 1.9. Service Oriented Architecture
- 2. Computer Networking
  - 2.1. Data Communication
  - 2.2. Network Topologies
  - 2.3. Network Types
  - 2.4. Connecting Devices
    - 2.4.1. Routing
  - 2.6. TCP/IP Suite
    - 2.6.1. IP Address
    - 2.6.2. Ethernet
    - 2.6.3. Wireless Network
  - 2.7. Advanced Topics
    - 2.7.1. Broadband Networks & Internet
    - 2.7.2. Internet Architecture & Cloud Deployment
    - 2.7.3. Technologies for Network-Based Systems
  - 2.8. Virtual Private Network (VPN)

- 3. Virtualization
  - 3.1. Virtualization Architectures

1. Cloud Computing

According to the National Institute of Science and Technology (NIST)

1.1. Definition

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

1.2. Characteristics

1.2.1. Essential Characteristics

1. *On-demand self-service* → Users can automatically allocate computing resources without manual operations, except for the initial sign-up. The cloud management software handles resource management and provisioning.
2. *Broad network access* → Cloud resources can be accessed over a network through a wide range of wired and wireless devices using various connectivity technologies. All clouds are inherently dependent upon the internet for remote provisioning.
3. *Resource pooling* → The provider's computing resources are pooled to serve multiple consumers based on demand. Customers generally have no control or knowledge over the exact location of these resources, which can include storage, processing, memory, and network bandwidth. Multiple users can simultaneously share these resources through dynamic allocation and reallocation.
4. *Rapid elasticity* → Capabilities can be elastically provisioned and released as per demand. To the consumer, the available capabilities often appear unlimited and can be appropriated in any quantity at any time.
5. *Measured service* → Cloud systems automatically control and optimize resource use by leveraging a metering capability, typically on a pay-per-use or charge-per-use basis.

Multitenancy

A cloud computing architecture where multiple users (tenants) share the same physical and virtual resources, but remain isolated from each other. It allows the cloud provider to efficiently pool and dynamically allocate resources based on demand using technologies like virtualization and statistical multiplexing.

1.2.2. Benefits

1. **Cost Efficiency**
   - Reduces upfront capital expenses (hardware, facilities)
   - Pay-as-you-go and right-sizing eliminate wasted capacity
2. **Elastic Scalability & Flexibility**
   - On-demand provisioning of virtually unlimited compute, storage and network
   - Fine-grained autoscaling (up/down) to match workload
3. **Global Reach & High Availability**
   - Deploy across multiple regions for disaster-resilience and low-latency
   - Built-in redundancy, failover and load balancing
4. **Benefits of Cloud Service Models**
   - *SaaS*: Centralized management, minimal local footprint, efficient licensing
   - *PaaS*: Faster dev & deployment (no tool upgrades), lower TCO, built-in scaling
   - *IaaS*: Full VM control, configurable network/storage, still no hardware capex

1.2.3. Risks & Challenges

1. **Security & Privacy**
   - Data breaches, DoS attacks, multi-tenant isolation failures
   - Traffic eavesdropping, flawed implementations, malicious intermediaries
   - User-control gaps, inadequate monitoring/audit, backup vulnerabilities
2. **Compliance & Governance**
   - Multi-region legal requirements and data-sovereignty laws
   - Contractual and SLA complexities, lack of automated compliance checks
   - Risk management and policy disparity across providers
3. **Vendor Lock-in & Portability**
   - Proprietary APIs, data-format differences, limited workload migration tools
   - Long-term data egress costs (“data lock-in”)
4. **Performance & Reliability**
   - Network dependence and unpredictable latency or I/O bottlenecks
   - Measuring and guaranteeing uptime across heterogeneous subsystems
   - Cloud “spikes” in workloads can lead to contention
5. **Operational Complexity**
   - Cloud-native architecture learning curve, management tool fragmentation
   - Legacy integration challenges; orphaned data from retired services
   - Cost-management (monitoring spend, unexpected long-term storage fees)

1.2.4. Roles and Boundaries

Role in Cloud Computing	What it is
Cloud Provider	A company that owns or leases the servers, storage, and networks you use in the cloud—and makes sure those resources stay up and running per their SLA.
Cloud Consumer	The person or organization (or their app) that signs up to rent and use the cloud resources via a web console or API.
Cloud Service Owner	Whoever "owns" a particular service or application running in the cloud—either the renter (consumer) if they built it on leased machines, or the provider if they built and host it themselves.
Cloud Resource Administrator	The person or team who actually configures, manages and monitors the cloud resources and services—this could be the consumer's ops staff, the provider's ops staff, or an outside contractor.
Cloud Auditor	An independent reviewer who checks that the provider's security, privacy and performance controls really match what they claim in their SLA and reports.
Cloud Broker	A middleman who helps buyers compare and purchase services from one or more cloud providers, often handling billing or integration for you.
Cloud Carrier	The network or internet service (think your ISP or backbone provider) that carries data back and forth between you (the consumer) and the provider's data center.

Boundaries of Cloud Computing

Organizational Boundary → What IT resources an organization owns and controls. Cloud providers have their own separate boundaries for the infrastructure they manage.
Trust Boundary → The edge of what an organization considers secure and under its control. When using cloud services, this boundary extends into the provider's infrastructure, even though it's not owned by the organization—making trust, security, and transparency very important.

1.3. History and Background

Cloud computing has evolved from fields like **Cluster Computing** and **Grid Computing**.

- *Cluster computing* → involves interconnected stand-alone computers cooperating as a single resource pool.
- *Grid computing* → utilize the processing power of networked PCs.

Time	What Happened
1961	Computer Scientist John McCarthy is attributed with delivering the idea that computations will be provisioned as utilities in future
1960s	Computer networking was conceived soon after the invention of computers.
1960-70s	The mainframes (giant powerful computers) were leased out by the manufacturers.
1990s	Clusters became popular, when mainframes and traditional supercomputers were becoming less cost-effective for high performance computing (HPC).
1990s	The idea of grid computing emerged, to use the processing power of networked PCs for scientific calculations during idle times.
1990s	Salesforce.com started bringing remotely provisioned software services to the enterprises.
2002	Amazon Web Services (AWS) were launched.
2006	The term “cloud computing” emerged; allowed organizations to “lease” the computing capacity and processing power from cloud providers.
2010	Out of top 500 supercomputers, 85% were computer clusters built with homogeneous nodes.

1.3.1. Grid Computing

> The grid is an integrated computing infrastructure for bringing together computers to create a large collection of compute, storage, and network resources.

- Used to solve large-scale computation problems or to enable fast information retrieval by registered users or user groups.
- **Grid Architecture** (top-to-bottom) `not-important`
  - *Application* → The top layer consisting of user applications to be run on grid.
  - *Collective Services* → Focus on interaction among the resources. implements functions such as resource discovery, scheduling, brokering etc.
  - *Resource Service* → Deals with the aggregated computing resources available for user applications in collective operations.
  - *Connectivity Layer* → Provides the core networking among the computational resources of fabric layer through physical or virtual networking.
  - *Grid Fabric* → Consists of all the computational resources such as storage systems, catalogs, network resources, servers and their network connections.

1.3.2. Cluster Computing

> A computer cluster is a collection of interconnected stand-alone computers which cooperate to work as a single resource pool of computing resources.

- Has laid the foundation of modern day super computers, computational grids and cloud computing.
- **Benefits** → Scalability, High availability, Fault Tolerance and Use of commodity computers.

1.3.3. Differences between Cluster, Grid and Cloud

1. **Cluster Computing**: Best for single organizations needing raw power for HPC tasks (like weather modeling, rendering).
2. **Grid Computing**: Ideal for collaborative, large-scale projects with distributed resources (like scientific research).
3. **Cloud Computing**: Offers flexibility, scalability, and cost-efficiency for dynamic workloads (like startups, SaaS apps).

Aspect	Cluster Computing	Grid Computing	Cloud Computing
Architecture	Tightly coupled, homogeneous nodes in a single system.	Loosely coupled, heterogeneous nodes across multiple organizations.	Virtualized, pooled resources accessed over the internet.
Resource Location	Localized.	Geographically distributed.	Centralized.
Management	Centralized control.	Decentralized.	Provider-managed.
Use Cases	High-performance computing (HPC) tasks.	Distributed, parallel tasks.	Scalable, on-demand services.
Resource Allocation	Dedicated to a single organization.	Shared across organizations for specific projects.	Dynamically allocated via virtualization.
Scalability	Vertical scaling.	Horizontal scaling.	Elastic scaling.
Cost Model	High upfront capital.	Volunteer-based or project-funded.	Pay-as-you-go.
Fault Tolerance	Redundant nodes/HA configurations.	Tasks resubmitted on failure.	Built-in redundancy.
Ownership	Single entity owns the cluster.	Multiple entities contribute resources.	Provider-owned; users lease resources.

1.4. Cloud Service Models

1. **Software as a Service (SaaS)** → You use ready-made apps over the internet (like Gmail or Zoom), with no need to manage the underlying hardware or software.
2. **Platform as a Service (PaaS)** → You get a setup to build and run your own apps without worrying about the servers or system setup (like using Firebase or Heroku).
3. **Infrastructure as a Service (IaaS)** → You rent virtual machines, storage, and networks, and you're responsible for installing and managing your own software (like with AWS EC2 or Google Cloud Compute Engine).

1.5. Cloud Deployment Models

1. **Private cloud** → For exclusive use by a single organization with multiple consumers. It can be owned, managed, and operated by the organization, a third party, and it can be on or off-premises.
2. **Community cloud** → For exclusive use by a specific community of consumers with shared concerns. Ownership, management, and operation can be by one or more organizations in the community, a third party, and it can be on or off-premises.
3. **Public cloud** → For open use by the general public. It may be owned, managed, and operated by businesses, academic, or government organizations, and it exists on the premises of the cloud provider.
4. **Hybrid cloud** → A composition of two or more distinct cloud infrastructures that remain unique entities but are bound together by technology enabling data and application portability.

Business Drivers for Cloud Computing → *IT Capacity Planning*, *Cost Reduction*, *Organizational Agility* and *Access to virtually unlimited resources and rapid elasticity*.

1.6. Basics of Computing

> **Mainframe** → A large, expensive, powerful server that can handle hundreds or thousands of connected users/servers simultaneously.

- The customers were charged on monthly basis for the use of hardware such as CPU, memory and peripheral devices. The software usage was charged for the time of usage.
- The mainframe leasers used to develop customized software exclusively for a client organization and charged for it. The client was also charged for the maintenance of those customized software.
- This model still exists in the form of cloud computing.

> **Server** → A computer which provides services to other computers and/or devices connected to it.

Services provided by a server include the controlled access to hardware and software resources and storage. A server can support thousands of users at the same time.

- *Web Server* → stores websites and web apps and provides them on your desktops and mobiles through web browsers.
- *Domain Name Server* (DNS) → Stores domain names and the corresponding IP addresses.
- *Database Server* → Hosts database and provides access to data and provides data manipulation functionality.

> **Desktop** → A computer which is designed to remain in a stationary position. It is used as a personal computer.

1.7. Other Services of Cloud Computing

1.7.1 Cloud-Based Data Storage

> The modern form of network storage, allowing users to store data on cloud infrastructure and access it via the internet. It offers scalable and cost-effective solutions with virtually unlimited space.

Property	Detail
Advantages	Scalability, Flexible Pricing, Reliability, Global Access and Multiple Access Methods (Web Interfaces, Mounted Drives and APIs).
Disadvantages	Performance, Security Concerns and Data Deletion.
Popular Providers	Dropbox, pCloud, Carbonite, ElephantDrive

Block Storage

Low-level storage ideal for structured data and custom filesystems.

1.7.2. Cloud Backup Systems

> Cloud backups send encrypted data copies to remote cloud servers on a schedule using client software. While convenient, long-term and high-volume storage can be expensive, and local backups are sometimes preferred for sensitive data.

Cloud Databases

Accessible over the internet, these can be hosted on VMs or as managed services. Examples: Oracle, Amazon, Microsoft. Benefits include scalability and redundancy.

Technical Aspects

- **Resource Pooling**: Shared storage dynamically allocated.
- **Virtualization**: Virtual disks created over VMs.
- **Data Tiering**: Optimizes performance on a single device using varied disks.
- **De-duplication**: Prevents redundant data storage.
- **Maintenance Architecture**: Ensures availability during maintenance.
- **Elastic Disk Provisioning**: Thin-provisioning with usage-based billing.

Storing data in the cloud introduces risks like → Unauthorized access, Limited control over data lifecycle, Provider access to data and Lack of transparency in security policies

1.7.3. Identity as a Service (IDaaS)

> A cloud-based solution that manages user authentication and access for both on-premises and cloud applications.

- **Benefits** → Simplifies login management by allowing users to access multiple services with a single sign-in and Automatically revokes access when employees leave, reducing security risks.
- **Why It's Needed** → Organizations often use a mix of cloud and on-premise systems and managing multiple logins across various platforms can be complex and error-prone. Also, manual user deactivation upon employee exit is tedious and risky if overlooked.
- **Examples** → *OpenID*, *Ping Identity* and *PasswordBank*.

1.7.4. Collaboration

Being able to edit shared files

Cloud supports real-time collaboration on documents, spreadsheets on services provided by Google, Microsoft, Dropbox etc.

Collaborative meetings can be conducted using software hosted in the cloud. This offers organizations a *cost-effective virtual meeting option* as an alternative to traditional face-to-face meetings.

- **Streaming video** to enable face-to-face interaction.
- **Shared whiteboards** for controlling presentations.
- **Shared applications** to demonstrate software in a live environment.
- **Meeting recordings** for playback and sharing.

1.8. Cloud Security Threats

1. **Network-level Attacks**
   - *Traffic Eavesdropping* → Stealing or reading data in transit without detection.
   - *Man-in-the-Middle* → Intercepting and altering messages before relaying them.
   - *Denial of Service* → Flooding compute, memory or network to exhaust resources.
2. **Virtualization-specific Threats**
   - *Hypervisor/VM Escape* -→ Exploiting hypervisor bugs or weak VM isolation to break out of one VM and control the host or other VMs.
   - *VM Sprawl & Unpatched Guests* -→ Forgotten, unpatched VMs become entry points for malware.
   - *Malicious VM Migration* -→ Moving an infected VM between hosts, spreading compromise.
3. **Platform- & Model-Specific Issues**
   - *Browser-Based Threats* -→ XSS, CSRF or malicious add-ons compromising SaaS apps.
   - *Legacy Software Compatibility* -→ Old apps on IaaS may carry known vulnerabilities.
   - *Portability & Vendor Lock-In* -→ Difficulty migrating data/services can trap insecure deployments.
4. **Multi-Tenancy & Trust Boundaries** → *Overlapping Trust Boundaries* and *Isolation Failures*.
5. **Implementation & Configuration Flaws** → *Flawed Provider Software/Hardware*, *Disparity in Security Policy* and *Inadequate Monitoring & Auditing*.
6. **Data-centric Risks** → *Unauthorized Access*, *Data Persistence & Deletion* and *Backup Vulnerabilities*.
7. **Governance, SLAs & Risk Mismanagement**

1.9. Service Oriented Architecture

> An architectural style in which application functionality is exposed as discrete, loosely coupled services that communicate over a network. Each service encapsulates a business capability, is discoverable, reusable, and can be orchestrated to form larger business processes.

Web Services as SOA Building Blocks → Independent software components that enable machine-to-machine interaction over a network. Accessed via API calls in any language.

- **Core Web-Service Technologies**
  - **WSDL** → XML-based language describing a service's operations and message formats
  - **XML Schema** → Defines the structure of input/output messages
  - **SOAP or REST** → Standard protocols/formats for exchanging those XML messages
  - **UDDI** → Service registry standard where WSDLs are published and discovered

> **Cloud vs. Web Services**
>
> **Cloud Services** (SaaS, PaaS, IaaS) ≠ **Web Services** (APIs)
>
> Web services often serve as the "front door" to cloud functionality.

- **SOA in Web Applications** → Follow a three-tier model: presentation (HTML/HTTP/URLs), application logic, and data storage. Each tier can expose or consume web services, making the app part of a larger SOA.

2. Computer Networking

> **Network** → A collection of computers and devices connected together through transmission media.

> **Internet** → A network of interconnected networks.

Devices → Hosts and *Connecting Devices*:

- *Router* → Connects the network with other networks.
- *Switch* → Connects devices within the network.
- *Modem* → Changes the form of data (modulates-demodulates).

Network Criteria

1. *Performance* → Measured by "Throughput" (bulk of data transmitted in unit of time) and delay.
2. *Reliability* → Measured in terms of frequency of network failure, time to recover from a failure and robustness from disasters.
3. *Security* → Protecting data from unauthorized access and damage.

Physical Structures

Communication can only take place if the devices are simultaneously connected to the same communication-path or link or connection. A link can be dedicated link (Point to Point) or shared among devices (multipoint).

2.1. Data Communication

> **Data Communication** ⟶ Exchange of data over some transmission medium between two devices.

Data must be delivered to correct destination, there must be timely delivery of the data and there must not be uneven delay among the packet arrival time during audio or video transmission.

Components ⟶ *Sender*, *Receiver*, *Message* (The data to be sent), *Transmission Medium* (The physical path through which a message travels) and *Protocol* (The set of agreed-upon communication-rules).

Data Representation

1. *Text* ⟶ Represented by bit pattern called code (Unicode and ASCII).
2. *Numbers* ⟶ Directly converted binary of the number.
3. *Images* ⟶ Represented by a matrix of pixels. A pixel is a small dot.
4. *Audio* ⟶ A continuous stream of sound data.
5. *Video* ⟶ Can be a continuous stream or a sequence of image combinations.

Data Flow

1. *Simplex* ⟶ Only one device can send and other can receive.
2. *Half Duplex*: Both devices can communicate but one at a time.
3. *Full Duplex*: Both devices can send and receive at the same time.

2.2. Network Topologies

Topology	What it is?	Advantage	Disadvantage
Mesh	Every device has a dedicated point to point link to every other device.	Robustness of network from failure of any link.	Requires too many wires/cables and is expensive to make the more devices you add.
Star	All devices are connected to a central device.	Requires only one I/O port in each device.	If the central device fails, the whole network fails.

Topology	What it is?	Advantage	Disadvantage
Bus	A multipoint topology in which one long cable is used a network backbone.	Ease of installation and requires less cabling.	Breaking of backbone cable isolates the network segments and introduces noise.
Ring	The devices are connected in the form of ring.	Easy to expand and alter the network.	Failure of a single device can disable the entire network.

2.3. Network Types

> **Local Area Network** (LAN) ⟶ A network that has a scope of an office, building or a campus. A LAN can even extend throughout a company.

Each host in a LAN has a unique identifier or address. The communication packets between any two hosts in a LAN contain the source and destination addresses.

- *Ethernet* (CSMA/CD): Carrier Sense with Multiple Access with Collision Detection (retransmission after collision detection)
- *Local Talk* (CSMA/CA): Carrier Sense with Multiple Access with Collision Avoidance (reserve the media before transmission)
- *Wireless LAN* (WiFi): IEEE 802.11
- *Token Ring/FDDI*: A token travels around the ring.

> **Wide Area Network** (WAN) ⟶ A network that spans large geographical area such as town, cities, states or even countries. Usually interconnects multiple LANs.

It is normally created and run by communication companies and is leased to the user organizations.

- *P2P WAN* ⟶ Connecting two devices through wired or wireless media.
- *Switched WAN* ⟶ A combination of several P2P WANs connected by switches.
    1. **Circuit-Switched Network** ⟶ A dedicated physical connection (circuit) is established, maintained and terminated through a carrier network for each communication session.
    2. **Packet-Switched Network** ⟶ A single link is shared among multiple network devices.

> **Metropolitan Area Network** (MAN) ⟶ A network covering a large geographical area bigger than LAN and smaller than WAN.

MAN is not owned by a single organization generally just like WAN. The MAN equipment are usually owned by a service provider.

2.4. Connecting Devices

1. **Hub** (Layer 1)
    - Basic device that sends incoming signals to all connected devices.
    - It can't understand addresses, so it broadcasts everything.
    - Works like a multi-port repeater in a star topology.
2. **Switch** (Layer 1-2)
    - Smarter than a hub — it learns MAC addresses and sends data only to the correct device.
    - Reduces collisions and supports devices with different speeds.
    - Still forwards broadcasts to all devices.
3. **Router** (Layer 1-3)
    - Connects different networks (e.g., your home Wi-Fi to the internet).
    - Uses IP addresses (Layer 3) and MAC addresses (Layer 2) to route data efficiently.
    - Each port has its own MAC and IP.
    - Chooses the best path using routing tables and protocols.

> **Virtual LAN (VLAN)**
>
> A logical (not physical) segment of a physical LAN. VLANs are defined by software. Each VLAN is a work group in an organization, has a VLAN ID and receives the broadcast messages addressed to its own ID. A VLAN may span over multiple switches in a LAN. No need to update the physical topology to relocate a person from one VLAN to other, just the software configuration is to be updated.

2.4.1. Routing

> A service of Network layer to find the best route. It is performed by applying routing protocols and using the decision tables called routing tables in each router.

Forwarding is the action performed by a router on the basis of routing protocol like **Distance-Vector** or **Link-State** routing protocol and routing table according to the destination address of each packet received at any interface.

Routing can be

- **Unicast** → forwards the packet to only one of the attached networks.
- **Multicast** → forwards the packet to multiple attached networks.

Routing can also be

- **Connectionless** → All packets of the same message are treated independently and may or may not follow the same route.
- **Connection Oriented** → All the packets of same message are labeled and routed through a virtual circuit or a fixed route.

2.6. TCP/IP Suite

Transmission Control Protocol (TCP) was proposed in 1973 to ensure a reliable, end-to-end and error free transmission control. It was latter split into TCP and Internet Protocol (IP) layers with IP handling the message routing and TCP performing the error control.

1. **Physical Layer** → Deals with transmission of bits into signals and transmission of signals over the link.
2. **Data-link Layer** → Creates the frames of data. Each frame contains the data and is addressed with the MAC address of the receiving device and also contains the MAC address of sending device.
3. **Network Layer**: Is responsible for host to host communication through their IP addresses and related protocols. No control for error and congestion is performed. Packets are called datagrams.
4. **Transport Layer** → Responsible for transporting a message from application program running over source host to corresponding application program on destination host. Works on port numbers on corresponding hosts.
    1. *Transmission Control Protocol* (TCP) → Provides flow control, congestion control and error control as it is a connection oriented protocol.
    2. *User Datagram Protocol* (UDP) → Is light weight and is not connection oriented.
5. **Application Layer** → Consist of programs running on two hosts and exchanging messages. Applications use these protocols: HTTP, FTP and SMTP.

2.6.1. IP Address

The **IP Address** Identifies the network connection of sender and receiver; changes if you move to a different network. A device with multiple internet connections can have multiple IPs.

IPv4 → 32-bit address space (2^{32} addresses), shown as dotted decimals. Divided into Network ID and Host ID.

- *Classful Addressing*: A (0-127) `8-bit prefix`, B (128-191) `16-bit prefix`, C (192-223) `24-bit prefix`, D (224-239) `multicast` and E (240-255) `reserved`.
- *Classless Addressing*: Done by CIDR, variable-length prefixes `/24` to avoid wasted addresses.
- **DHCP** → Automatically leases IPs from a pool.

- **NAT** (Network Address Translation) → Enables many private LAN addresses to share a global IP. Router rewrites local IP:port to its public IP:port and Maintains a NAT table mapping internal ↔ external connections.

IPv6 → 128-bit address space (2^{128} addresses), Eliminates NAT and private-address conflicts with built-in authentication/privacy, no DHCP required, simpler routing.

> ⑦ **MAC Address Resolution (ARP)**
>
> When a device needs a link-layer (MAC) address for an IP on its LAN: It broadcasts an ARP request with the target IP and its own addresses and the target replies with its MAC address.

2.6.2. Ethernet

> A widely-used Wired LAN technology for Layers 1-2, also known as IEEE 802.3.

A *connectionless protocol* with no flow control, error control, retransmission or acknowledgements. This makes it inherently unreliable—similar to IP and UDP.

- *48-bit MAC addresses* for identifying devices at the link layer. Each Ethernet frame ranges from `64 to 1518` bytes, with `46 to 1500` bytes allocated for data.
- Uses **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)** for managing data transmission, with either *Unicast*, *Multicast* or *Broadcast*.

1. **Standard Ethernet** → 10 Mbps
2. **Fast Ethernet** → 100 Mbps
3. **Gigabit Ethernet** → 1 Gbps
4. **10 Gigabit Ethernet** → 10 Gbps

2.6.3. Wireless Network

> A local network using radio signals instead of cables. No physical connection or switch required instead devices are connected via an Access Point (AP).

1. **IEEE 802.11 (Wi-Fi)** → The WLAN standard for wireless networks. The devices are connected to the nearest AP. It uses CSMA/CA for collision handling.
2. **IEEE 802.15 (Bluetooth)** → The WLAN standard for adhoc networks. Short-range wireless tech for connecting personal devices. Its range is about 10 meters.
3. **IEEE 802.16 (WiMAX)** → The WLAN/WMAN standard for long-distance internet access.
4. **Cellular Networks** → Radio-based mobile communication system. Data is sent as electromagnetic radio waves via antennas. It has had 5 generations (1G to 5G).

2.7. Advanced Topics

2.7.1. Broadband Networks & Internet

> Cloud services depend on the Internet for remote provisioning. Providers and consumers both connect via ISPs into a dynamic hierarchy of backbone networks linked by core routers. Worldwide connectivity is organized into three tiers of ISPs, where cloud providers and end-users attach at the local level.

- **Tier 1:** Large international backbone providers
- **Tier 2:** Regional ISPs peered with Tier 1
- **Tier 3:** Local ISPs serving end-users (cloud providers and consumers connect here)

2.7.2. Internet Architecture & Cloud Deployment

> Cloud resources sit on centralized infrastructure and rely on Internet connectivity. Deployment can be either on-premises or Internet-based, affecting how users reach services.

- **On-premises (on-prem)** ⟶ Provider installs a corporate LAN and Internet link, Internal users connect over the LAN and Remote users connect via VPN.
- **Internet-based** ⟶ Provider hosts resources online, All users access via the provider's Internet connection.

2.7.3. Technologies for Network-Based Systems

> Explosive growth in processor speeds and network bandwidth led to new architectures and storage solutions.

Compute Growth

- 10 MHz (1970s) → 4 GHz (2010s) clock speed.
- Single Core → Multi Code CPUs (dual, quad, octa).
- CPUs/GPUs now multithreaded for high MIPS.
- 16 KB (1976) SRAM → 64 GB (2011) DRAM
- 260 MB (1981) HDD → 3 TB (recent) HDD
- Floppy/Tape → Flash/SSD.

Network storage ⟶ *SAN* (Block-level storage over a dedicated network) and *NAS* (File-level storage over Ethernet).

Cloud data centers host tens of thousands of servers (physical & virtual), organized into a layered network fabric that handles compute, control, and external access.

1. Servers
2. Top-of-Rack L2 switches
3. Aggregate switches (AGS)
4. Access routers (AR)
5. Border routers (BR)
6. Internet

> **What is Web 2.0?**
>
> The second-generation Web shifted pages from static to interactive, enabling user collaboration and content sharing.
>
> **Key technologies** → AJAX (Asynchronous JavaScript & XML) for dynamic, partial page updates and RSS feeds for continuous news/updates delivery.

2.8. Virtual Private Network (VPN)

> A VPN creates an encrypted tunnel (IP Tunneling) over the public Internet, making remote devices appear as if on a private LAN. Encryption adds security but can impact performance. It is implementable at Layers 1–3.

- **Remote-access VPN**: Client device ↔ enterprise VPN gateway
- **Site-to-site VPN**: Gateway ↔ gateway between two networks

3. Virtualization

> Enables one physical machine to host multiple isolated virtual machines, improving resource utilization. A hypervisor manages hardware resources and enforces isolation.

Implementation Levels

1. *Instruction-Set Architecture (ISA) Level* -→ Emulate legacy code via interpreters translating source instructions to target machine instructions.
2. *Hardware Abstraction Level* -→ Hypervisor virtualizes CPU, RAM, disk, and NIC, sharing them among VMs.
3. *Operating System Level* -→ Host OS provides containers or VMs, acting as the abstraction layer.
4. *Library Support Level* -→ Hardware-accelerated APIs (e.g., vCUDA for GPUs) exposed within VMs.
5. *Application Level* -→ Individual applications run within an abstraction layer (e.g., JVM) isolating them from the OS.

3.1. Virtualization Architectures

Three main VM architectures transform physical hardware into virtual hardware:

- **Hypervisor (Bare-Metal)** -→ Runs directly on hardware, managing VMs (e.g., Xen, VMware).
- **Full Virtualization** -→ Guest OS is unaware of virtualization. Hypervisor provides hardware acceleration; can sit on bare-metal or host OS, with critical instructions executed on hardware.
- **Paravirtualization** -→ Guest OS is modified to interface with hypervisor for hardware calls (e.g., KVM).

Other architectural styles: *Generic*, *Monolithic*, *Microkernel*.

Modern processors support hardware-assisted virtualization, separating critical and non-critical instructions. They have two **Processor Modes**: Supervisor (Privileged) or User (Non-Privileged).

- **CPU Virtualization** → Hypervisor runs in supervisor mode, VMs in user mode, trapping privileged instructions.
- **Memory Virtualization** → Guest OS maps virtual to physical memory; hypervisor maps physical to machine memory using hardware MMU and TLB.
- **I/O Virtualization:**
  1. *Full Device Emulation* → Hypervisor emulates device in software.
  2. *Paravirtualized I/O* → Guest OS uses frontend driver; hypervisor provides backend driver.
  3. *Direct I/O* → VMs access hardware devices directly.

> **Virtual Cluster** ⟶ A set of VMs interconnected by a virtual network across one or more physical clusters.

Nodes can be physical or virtual and scale dynamically. Host failure may disrupt hosted VMs but not the entire virtual cluster. VM failure does not crash the host. Multiple virtual clusters can coexist on a single physical cluster, and a single virtual cluster can span multiple physical clusters.

VM Migration

Moving a VM between hosts supports load balancing, consolidation, failover, and scalability.

- VM States ⟶ Powered-off, Suspended, Paused, Powered-on
- Migration Types:
  - **Cold Migration** → VM is powered off.
  - **Warm Migration** → VM is suspended.

- **Live Migration** → VM powered-on, zero downtime.

> **Data Center Automation via Virtualization** ⟶ Virtualization enables efficient resource management in large data centers by supporting dynamic provisioning. It helps balance performance, utilization, and cost by consolidating servers—migrating VMs to fewer hosts and powering down idle ones. Virtual storage layers present virtual disks to VMs, and integration with cloud management automates provisioning and billing for greater efficiency.

> **Network Virtualization** ⟶ Abstracts physical infrastructure into multiple virtual networks. VMs connect via vNICs → vSwitch → pNIC under a hypervisor. Virtual networks, made of nodes and links, separate ISP infrastructure from virtual control. Key Technologies are *VLANs* and *VPNs*.

THE END