Mid sem notes -CC

Module 1-Cloud Computing Architecture

1. What is Cloud Computing?

Definition: for anything that involves delivering hosted services over the Internet.
Cloud computing is the delivery of computing services (storage, processing, networking, software) over the internet ("the cloud"), enabling on-demand access to shared resources without direct user management.

Core Characteristics:

1. **Storing/accessing data & programs** on remote servers (e.g., Dropbox for file storage).
2. **Internet-based computing** (services accessed via browsers/APIs).
3. **Resources provided as a service** (e.g., renting servers from AWS instead of buying physical hardware).
4. **Transparency, scalability, security & intelligent monitoring** (automatic resource allocation, threat detection).

Real-life Example:
Netflix uses AWS cloud services to stream content globally. Users access movies via browsers/apps (frontend), while AWS handles storage, servers, and security (backend).

A Service Level Agreement (SLA) is a formal contract between a service provider and a client that defines the expected level of service. In cloud computing, SLAs are crucial for setting clear expectations and ensuring accountability.

2. Cloud Architecture Foundation

Combines two paradigms:

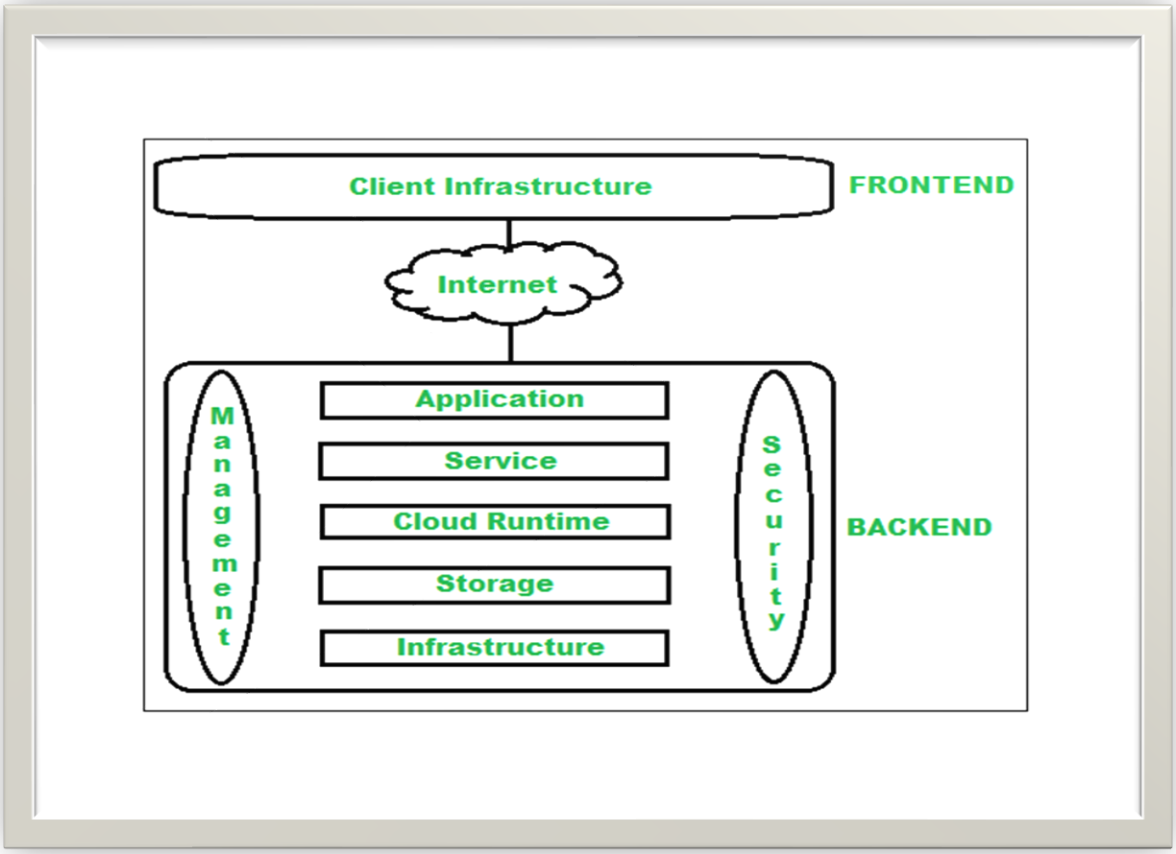
- **SOA (Service-Oriented Architecture)**:
    - Breaks down services into reusable components (e.g., "authentication service" used by multiple apps).
- **EDA (Event-Driven Architecture)**:
    - Responds to real-time events (e.g., processing a payment triggers an order-confirmation email).

Example: Uber uses SOA for modular services (maps, payments) and EDA to dispatch drivers when ride requests occur.

When a Cloud is made available in a pay-as-you-go manner to the public... The service being is Utility Computing."
This statement means that cloud providers (like AWS, Azure, Google Cloud) are operating lik utility company, and their product—computing power—is a utility similar to electricity, wate natural gas.

3. Cloud Architecture Components

Divided into **frontend** (client-facing) and **backend** (cloud infrastructure):



Frontend

- **Definition**: Interfaces users interact with to access cloud services.
- **Components**:
    1. **Client Infrastructure**:
        - Applications/GUIs used to access the cloud (e.g., web browser, mobile app).
        - *Example*: Using Chrome to access Google Docs.
    2. **User Interfaces**:
        - Dashboards, APIs, or command-line tools (e.g., AWS Management Console).

Backend

- **Definition**: The cloud itself, managing resources, security, and data.
- **Components**:
    1. **Application**:
        - Software/platform accessed by users (e.g., Salesforce CRM).
    2. **Service**:
        - **SaaS** (Software as a Service): Ready-to-use apps (e.g., Gmail).
        - **PaaS** (Platform as a Service): Development platforms (e.g., Heroku for app deployment).
        - **IaaS** (Infrastructure as a Service): Virtualized hardware (e.g., AWS EC2 virtual servers).

        eg. google and kaggle offers GPU in colab notebooks
    3. **Runtime Cloud**:
        - Execution environment for apps (e.g., Java apps running on Google App Engine).
    4. **Storage**:
        - Scalable storage (e.g., Amazon S3 for storing user files).
    5. **Infrastructure**:
        - Hardware/software (servers, virtualization, network devices).
    6. **Management**:
        - Coordinates resources (e.g., auto-scaling in Azure during traffic spikes).
    7. **Security**:
        - Tools like firewalls, encryption (e.g., AWS IAM for access control).
    8. **Database**:
        - Managed databases (e.g., Google Cloud SQL for structured data).
    9. **Networking**:
        - Connectivity services (e.g., AWS VPC for isolated cloud networks).
    10. **Internet**:

- Bridge between frontend and backend.

Real-life Workflow:
A user uploads a photo to Instagram (frontend). The backend processes it:

- Storage saves the image (AWS S3).
- Database records metadata (Google Cloud SQL).
- CDN (Networking) delivers it globally.

---

4. Benefits of Cloud Architecture

Benefit	Description	Example
Simplifies System	Abstracts complexity; single interface for users	AWS Management Console controls all services
Improves Data Processing	Scalable compute for big data tasks	Spotify analyzes user data for recommendations
High Security	Centralized mechanisms (encryption, monitoring)	Bank apps use Azure Security Center
Modularity	Independent components for easy updates	Updating a payment service without downtime
Disaster Recovery	Automated backups across regions	Slack restores data after outages via Google Cloud
Accessibility	Access services anywhere via internet	Remote teams collaborate on Microsoft 365
Cost Reduction	Pay-as-you-go model; no physical hardware	Startups use AWS instead of data centers

Benefit	Description	Example
Reliability	99.9% uptime SLAs	Netflix streams 24/7 via AWS
Scalability	Instantly handle demand spikes	Airbnb scales servers during holiday seasons

5. Real-World Applications

- **Healthcare**: Hospitals use **SaaS** (e.g., Epic EHR) for patient records with **backend security** (HIPAA compliance).
- **E-commerce**: Shopify (**PaaS**) hosts online stores; scales during Black Friday sales.
- **IoT**: Smart home devices send data to **cloud storage** (e.g., Google Cloud IoT) for analysis.

Key Takeaways:

- **Frontend** = User access points (GUI, apps).
- **Backend** = Cloud infrastructure (services, storage, security).
- Cloud architecture enables **flexibility**, **cost savings**, and **innovation** (e.g., AI/ML via cloud GPUs).

Control Automation

Four functional areas :
Self-Configuration
Automatic configuration of components.

Self-Healing
Automatic discovery, and correction of faults.

Self-Optimization
Automatic monitoring and control of resources to ensure the optimal functioning with re
the defined requirements.

Self-Protection
Proactive identification and protection from arbitrary attacks.

Detailed Notes on Cloud Computing Framework

1. What is a Cloud Computing Framework?

Definition:
A structured approach providing tools and technologies to *design, deploy, manage, and optimize* cloud-based applications and services. It acts as a blueprint for building cloud solutions.

Core Components:

Component	Purpose	Examples
Development Tools	Build/test cloud apps	AWS Cloud9, Azure DevOps
Middleware	Connects apps/data across cloud environments	Red Hat JBoss, MuleSoft Anypoint Platform n8n
Administration Software	Monitors/manages cloud resources	VMware vRealize, IBM Cloud Pak

Real-World Analogy:
Like a "factory assembly line" for cloud apps:

- **Development tools** = Raw materials (code, APIs)
- **Middleware** = Conveyor belts (data integration)
- **Admin software** = Quality control robots (performance monitoring)

2. Framework Phases BAE

Phase 1: Analysis

Evaluates feasibility and requirements:

- **Cost Analysis**:
    - *Example*: Netflix migrated to AWS to save $1B/year vs. maintaining data centers.
- **Security Analysis**:

- *Example*: Banks use Azure Security Center to audit compliance (GDPR, HIPAA).

- **Accounting Analysis**:
  - Tracks usage-based billing (e.g., Google Cloud's per-second VM pricing).
- **Risk/Benefit Analysis**:
  - *Trade-off*: Cloud scalability vs. dependency on internet connectivity.

Phase 2: Evaluation

Assesses solutions against business needs:

Evaluation Type	Focus	Real Application
Investment	ROI of cloud migration	Dropbox saved $75M over 2 years by moving to AWS
Risk	Downtime/data loss probability	Slack uses AWS multi-region backups for 99.99% uptime
ROI	Cost savings vs. on-premises	Capital One reduced TCO by 30% with AWS
Scenario	"What-if" testing (e.g., traffic spikes)	Zoom scales servers during global events
Security	Vulnerability assessments	Shopify uses automated penetration testing

3. Why Businesses Adopt Cloud Frameworks

Key Drivers:

1. **Cost Reduction (61% of large IT companies)**:
   - *Mechanism*: Pay-as-you-go model eliminates upfront hardware costs.
   - *Example*: Airbnb avoids $200M+ in data center expenses using AWS.

2. **Enhanced Security**:
    - *Mechanism*: Enterprise-grade firewalls + encryption + access controls.
    - *Example*: Pfizer stores COVID vaccine data in IBM Cloud with end-to-end encryption.
3. **Remote Work Enablement**:
    - *Mechanism*: Centralized cloud access from any device/location.
    - *Example*: GitLab's 1,500+ remote employees collaborate via Google Workspace.
4. **High Reliability**:
    - *Mechanism*: Geographically distributed servers + failover systems.
    - *Example*: Salesforce guarantees 99.999% uptime for financial clients.
5. **Elastic Scalability**:
    - *Mechanism*: Instantly add/remove resources (CPU, storage).
    - *Example*: Instagram handles 4M+ uploads/hour by auto-scaling on AWS.

---

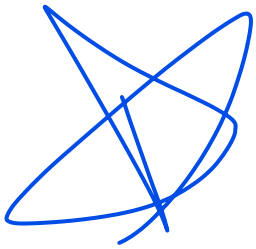
4. Real-World Industry Applications

Industry	Cloud Framework Use Case	Outcome
Healthcare	Epic EHR on Azure: Secure patient data sharing	250M+ patient records accessed globally
Retail	Shopify (PaaS): E-commerce store hosting	1M+ stores scale during Black Friday sales
Finance	Capital One on AWS: Fraud detection algorithms	Reduced false positives by 70%
Manufacturing	Siemens MindSphere (IoT cloud): Predictive maintenance	30% fewer machine failures

5. Challenges & Mitigations

Challenge	Framework Solution
Data Privacy Concerns	Encryption-at-rest + regional compliance (e.g., EU data in Azure Germany)
Vendor Lock-in	Hybrid/multi-cloud strategies (e.g., Anthos on AWS + GCP)
Skill Gaps	Managed services (e.g., AWS Managed Services)

Service Models



Cloud Service Models (SaaS, PaaS, IaaS)

1. Software as a Service (SaaS)

Definition:

- Software delivered over the internet on a subscription basis.
- **Key Feature**: No local installation/maintenance (accessed via web browser).
- **Billing Model**: Pay-as-you-go.
- **Nicknames**: *Web-based software*, *On-demand software*, *Hosted software*.

Real-World Examples:

- **Salesforce**: CRM for sales teams
- **Microsoft Office 365**: Productivity suite
- **Dropbox**: Cloud file storage

Advantages:

Benefit	Explanation	Real Application
Cost-Effective	No hardware costs; pay per user/month	Startups use Gmail instead of Exchange servers
Zero Installation	Accessible via browser instantly	Doctors access patient records on Epic EHR from any hospital computer
Automatic Updates	Provider handles patches/upgrades	Adobe Creative Cloud users get new features automatically

Benefit	Explanation	Real Application
Accessibility	Use anywhere with internet	Remote teams collaborate on Google Docs
Scalability	Add/remove users instantly	Zoom scales licenses during conference season

Disadvantages:

1. **Limited Customization**
    - *Issue*: Can't modify core functionality (e.g., Shopify stores can't alter checkout code).
    - *Workaround*: Use APIs for partial integrations (e.g., connect Mailchimp to Salesforce).
2. **Internet Dependency**
    - *Impact*: Offline work impossible (e.g., construction sites with poor connectivity can't access Autodesk BIM 360).
3. **Security Risks**
    - *Incident*: 2023 Microsoft breach exposed SaaS customer data.
    - *Mitigation*: Enable MFA and data encryption (e.g., Box Enterprise Key Management).
4. **Data Control Concerns**
    - *Regulatory Challenge*: HIPAA-compliant healthcare orgs avoid SaaS for sensitive patient data processing.

---

2. Platform as a Service (PaaS)

Definition:

- Cloud platform for developing, testing, and deploying applications.
- **Key Feature**: Manages OS, servers, storage – developers focus *only* on code.
- **Analogy**: Renting a fully equipped kitchen (PaaS) vs. building one (IaaS).

Real-World Examples:

- **AWS Elastic Beanstalk**: Deploy web apps without server config
- **Google App Engine**: Build scalable Python/Java apps
- **Heroku**: Deploy container-based apps

Advantages:

Benefit	Explanation	Real Application
Faster Development	Pre-configured tools (DBs, SDKs, runtimes)	Spotify built backends in days using Google App Engine
Cost Reduction	No server maintenance costs	Duolingo saved 60% vs. on-premises servers
Lifecycle Support	End-to-end: build → test → deploy → update	Netflix uses PaaS for continuous deployment
High-Level Abstraction	Focus on business logic, not infrastructure	Airbnb developers ignore server scaling rules

Disadvantages:

1. **Vendor Lock-in**
    - *Issue*: Apps built on Salesforce PaaS can't easily migrate to Azure.
    - *Solution*: Use Kubernetes for hybrid cloud portability.
2. **Limited Infrastructure Control**

- *Consequence*: Can't optimize OS/kernel for high-frequency trading apps.

3. **Provider Dependency**
   - *Outage Impact*: 2021 AWS outage paralyzed PaaS users like Slack for 5 hours.

---

3. Infrastructure as a Service (IaaS)

Definition:

- Virtualized computing resources (servers, storage, networking) over the internet.
- **Key Feature**: Full control over OS/apps; provider manages *physical hardware only*.
- **Nickname**: *Hardware as a Service (HaaS)*.

Real-World Examples:

- **AWS EC2**: Virtual servers
- **Azure Virtual Machines**: Windows/Linux VMs
- **Google Cloud Storage**: Scalable object storage

Advantages:

Benefit	Explanation	Real Application
Cost Efficiency	Pay per hour/GB (e.g., $0.10/hr for Linux VM)	Pinterest saved $20M/year migrating to AWS
Hosting Flexibility	Custom web server configurations	NASA hosts Mars imagery on AWS with custom CDN

Benefit	Explanation	Real Application
Enterprise Security	Better than most on-premises setups	JPMorgan uses Azure for FedRAMP-compliant banking apps
Zero Maintenance	Provider handles hardware failures/upgrades	Tesla avoids data center staff for Autopilot training

Disadvantages:

1. **Steep Learning Curve** Non technical can't handle
   - *Challenge*: Requires DevOps skills (e.g., configuring AWS VPC networks).
   - *Solution*: Use managed services like AWS Lightsail.
2. **Security Responsibility**
   - *Shared Model*: Provider secures hardware; *you* secure OS/apps/data.
   - *Mistake*: 2022 Uber breach occurred due to misconfigured IAM permissions.
3. **Geographic Limitations**
   - *Restriction*: Chinese companies can't use AWS in Shanghai due to GFW policies.

Comparison: SaaS vs. PaaS vs. IaaS

Criteria	SaaS	PaaS	IaaS
User Control	None (pre-built app)	Code & data only	OS, apps, data
Technical Skill	Low (end users)	Medium (developers)	High (sysadmins)
Maintenance	Fully managed	Platform managed	Hardware managed only
Scalability	User-based	Automatic (app-level)	Manual (VM-level)
Cost Model	Per user/month	Per resource consumption	Per hour/GB
Use Case	CRM, Email (Salesforce)	App development (Heroku)	Custom servers (AWS)

Industry-Specific Applications

1. **Healthcare**:
    - *SaaS*: Epic EHR for patient records
    - *IaaS*: AWS for genomic data processing
2. **E-commerce**:
    - *PaaS*: Shopify for store hosting
    - *SaaS*: Zendesk for customer support

3. **Gaming**:
    - *IaaS*: Azure VMs for multiplayer backends
    - *PaaS*: Google App Engine for leaderboards

💡 **Case Study**: Netflix uses all three:

- **SaaS** (Slack for internal comms)
- **PaaS** (Jenkins for CI/CD)
- **IaaS** (AWS for video streaming infrastructure)

Key Takeaway:

- **SaaS** = "Ready-to-eat meal" (least control, easiest use)
- **PaaS** = "Kitchen with prepped ingredients" (build apps faster)
- **IaaS** = "Raw ingredients + kitchen" (maximum flexibility, maximum effort)

Cloud Deployment Models (Public, Private, Hybrid)

1. Public Cloud

Definition:

- Infrastructure/services owned by third-party providers, available to the public over the internet.
- **Core Principle**: Shared resources ("multi-tenant"), pay-per-use pricing.

Real-World Providers:

- **AWS** (Amazon): 33% market share
- **Azure** (Microsoft): 22% market share
- **GCP** (Google): 11% market share

Advantages:

Benefit	Explanation	Example
Cost Efficiency	No capital expenditure; pay only for usage	Startups launch apps on AWS for $0.01/hr
Automatic Updates	Provider handles security patches/upgrades	Azure automatically updates SQL databases
Global Accessibility	Access resources anywhere via internet	Remote teams use GCP BigQuery worldwide
Elastic Scalability	Instantly handle traffic spikes	Netflix scales on AWS during new releases

Disadvantages:

1. **Security Concerns**:
    - *Risk*: Shared infrastructure vulnerabilities (e.g., 2022 AWS SSRF breach).
    - *Mitigation*: Encryption + VPC (e.g., HIPAA-compliant apps on Azure).
2. **Limited Control**:
    - *Constraint*: Can't customize physical hardware (e.g., GPU models on GCP).
3. **Internet Dependency**:
    - *Impact*: Offline work impossible (e.g., field engineers without connectivity).
4. **Compliance Challenges**:
    - *Issue*: GDPR restricts EU data storage to local regions (e.g., Azure Germany).

Best For:

- Web apps, big data analytics, disaster recovery (e.g., Spotify on GCP).

---

2. Private Cloud

Definition:

- Dedicated infrastructure operated solely for one organization.
- **Deployment Options**:
    - *On-premises*: Self-managed data centers (e.g., VMware vSphere).
    - *Hosted*: Managed by third-party (e.g., IBM Cloud Private).

Real-World Examples:

- **Capital One**: On-premises private cloud for banking compliance

- **NASA Nebula**: Government research cloud
- **Siemens MindSphere**: Industrial IoT cloud

Advantages:

Benefit	Explanation	Example
Enhanced Security	Isolated infrastructure for sensitive data	Lockheed Martin stores defense IP on-premises
Full Customization	Tailor hardware/software to exact needs	BMW optimizes factory robots with custom OS
Regulatory Compliance	Meet strict data sovereignty laws	Swiss banks use on-prem clouds for GDPR
Performance Control	Dedicated resources for latency-critical apps	High-frequency trading systems at Goldman Sachs

Disadvantages:

1. **High Costs**:
    - *Expense*: Upfront hardware + ongoing maintenance (e.g., $500k+ for small data center).
2. **Limited Scalability**:
    - *Constraint*: Physical hardware expansion required (vs. cloud's click-to-scale).
3. **Technical Complexity**:
    - *Challenge*: Requires in-house IT expertise (e.g., OpenStack engineers).
4. **Maintenance Burden**:

- *Overhead*: Organizations handle all updates/patches (e.g., hospital IT teams managing EHR systems).

Best For:

- Government, healthcare, finance (e.g., JPMorgan private cloud for transaction processing).

---

3. Hybrid Cloud

Definition:

- Integrates public + private clouds, allowing data/apps to move between them.
- **Key Drivers**: Flexibility, cost optimization, regulatory compliance.

Real-World Architectures:

- **AWS Outposts**: Run AWS services on-premises
- **Azure Stack**: Azure services in private data centers
- **Google Anthos**: Manage apps across clouds

Advantages:

Benefit	Explanation	Example
Flexibility	Sensitive data in private cloud; public for scalability	Walmart: Customer data on-prem, inventory on Azure
Cost Optimization	"Burst" to public cloud during peak demand	United Airlines uses AWS during holiday rushes

Benefit	Explanation	Example
Regulatory Agility	Keep regulated data private; use public for analytics	Pfizer: Drug research on-prem, sales data on AWS
Disaster Recovery	Backup private cloud data to public cloud	Nasdaq backs up trading data to GCP

Disadvantages:

1. **Complexity**:
    - *Challenge*: Requires integration tools (e.g., Kubernetes) and skilled architects.
2. **Cost Management**:
    - *Risk*: Unplanned public cloud spending (e.g., 37% of enterprises exceed budgets).
3. **Security Fragmentation**:
    - *Threat*: Inconsistent policies across environments (e.g., misconfigured S3 buckets).
4. **Data Governance**:
    - *Compliance Risk*: Data movement across regions (e.g., EU→US transfers violating GDPR).

Best For:

- Enterprises with variable workloads (e.g., Airbnb: Core app on AWS, payment processing on-prem).

Comparison: Public vs. Private vs. Hybrid

Criteria	Public Cloud	Private Cloud	Hybrid Cloud
Cost	OpEx (pay-as-you-go)	CapEx + OpEx (high)	Mixed (optimized)
Control	Limited (provider-managed)	Full customization	Balanced
Security	Shared responsibility	Fully self-controlled	Segmentation required
Scalability	Instant & unlimited	Limited by hardware	Burst to public cloud
Compliance	Limited (provider-dependent)	Full control	Flexible for regulations
Use Case	Web apps, startups	Banks, government	Enterprise, healthcare

Industry Case Studies

1. **Healthcare (Hybrid)**:
    - *Scenario*: Patient records in private cloud (HIPAA), research data on public cloud.
    - *Example*: Mayo Clinic uses private cloud for EHRs + AWS for AI-driven diagnostics.
2. **Retail (Hybrid)**:
    - *Scenario*: POS systems on-premises, e-commerce on public cloud.

    - *Example*: Target uses Azure for online sales + on-prem for in-store inventory.

3. **Finance (Private)**:
    - *Scenario*: Core banking systems on private cloud for compliance.
    - *Example*: HSBC hosts trading platforms on IBM Cloud Private.

💡 **Did You Know?**
58% of enterprises adopt hybrid cloud (2023 Flexera Report). NASA uses:

- **Public**: AWS for satellite imagery processing
- **Private**: On-premises for classified research

Key Takeaway:

- **Public Cloud** = "Renting an apartment" (shared, low maintenance).
- **Private Cloud** = "Owning a house" (full control, high responsibility).
- **Hybrid Cloud** = "Owning a house + co-working space" (balance of control & flexibility).

Knowledge as a service

Big Data Concepts, Storage, and Management

1. What is Big Data?

Big Data in Healthcare, Education, E-commerce ,Media and Entertainment,Finance ,Travel Industry, Telecom, Automobile

Definition:
Extremely large, complex datasets (structured, semi-structured, unstructured) that cannot be processed by traditional systems. Used to uncover patterns for business decisions, product improvements, and growth.

Real-World Application:
Walmart analyzes 2.5 PB customer data hourly to optimize inventory and pricing.

The 5 Vs of Big Data:

1. **Volume**: Scale (terabytes to exabytes)
   Example: Facebook processes 4 PB/day of user data.
2. **Velocity**: Speed of generation
   Example: Twitter handles 500M tweets/day (5,787/sec).
3. **Variety**: Diverse formats
   - **Structured**: SQL tables (sales records)
   - **Semi-structured**: JSON/XML (sensor logs)
   - **Unstructured**: Text/images/video (social media posts)

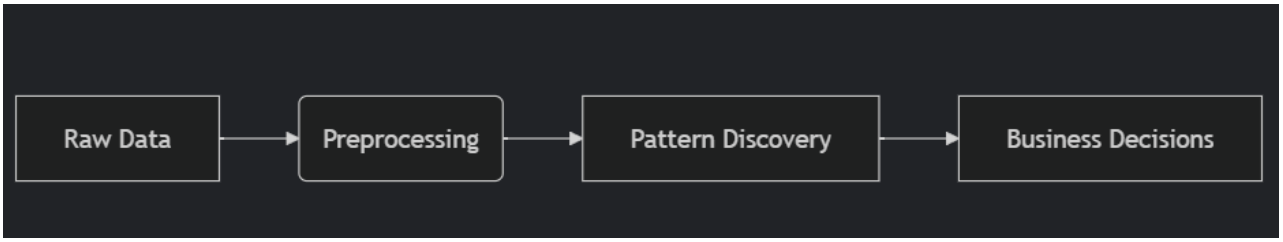
   need flexible data pipelines and schema-less databases to handle this diversity.
4. **Veracity**: The trustworthiness and quality of data eg medical records
5. **Value**: Business insights, The usefulness of data in driving business decisions and outcomes

   Example: Netflix uses viewing data to recommend shows

2. Data Mining

Definition:
It is a process of extracting insight meaning, hidden patterns from collected data that is useful to take a business decision for the purpose of decreasing expenditure and increasing revenue

Process:

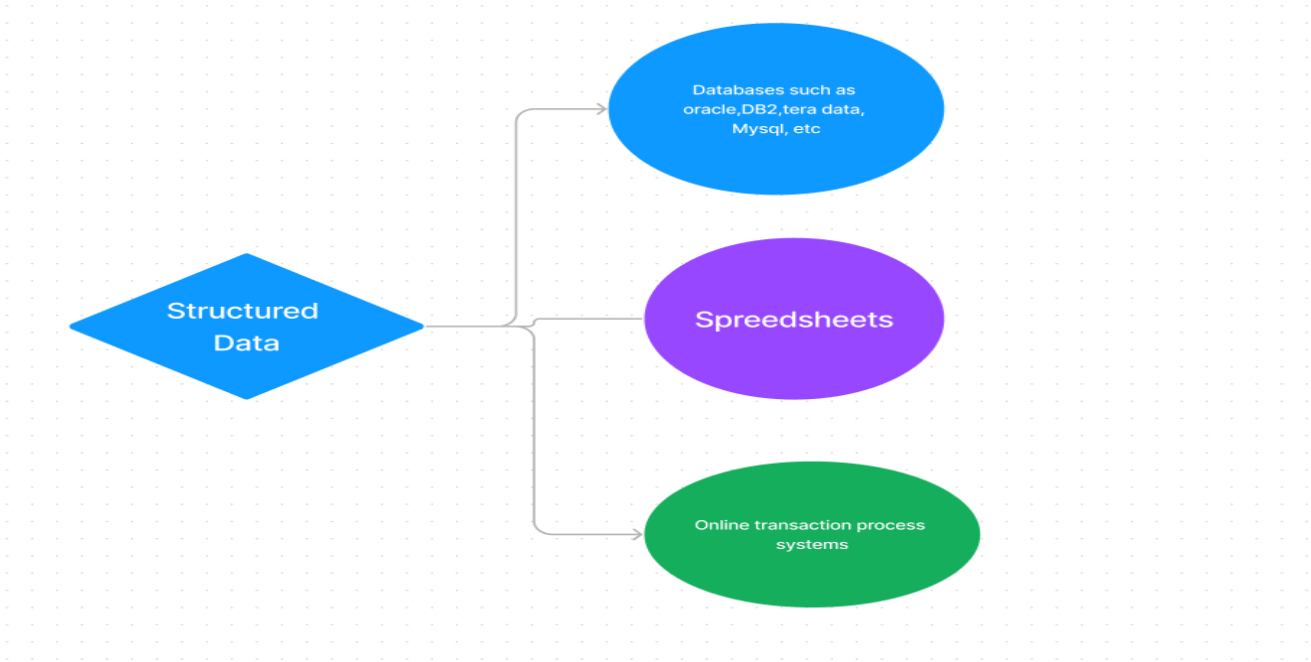


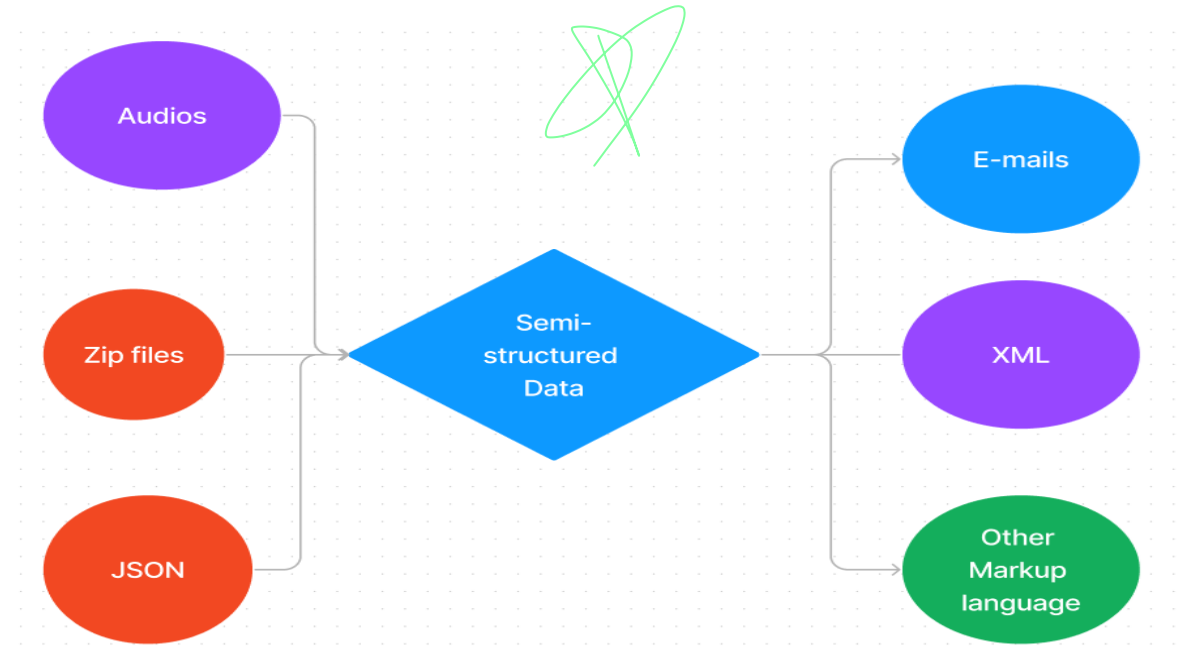
3. Big Data Sources

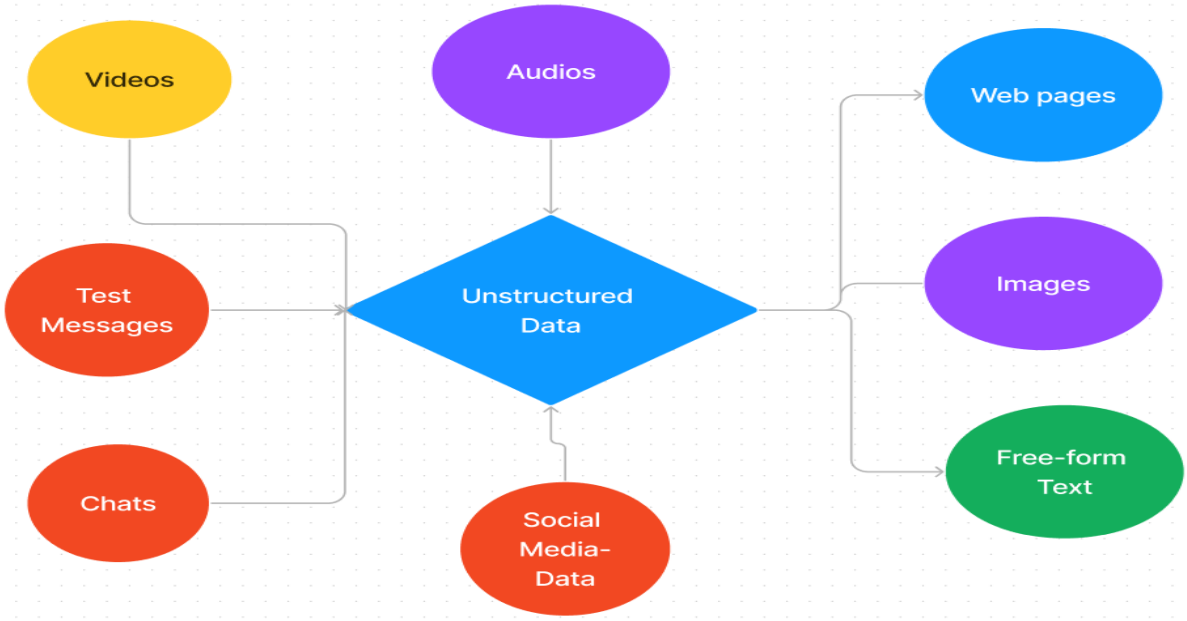
Source	Data Generated	Real Example
Social Media	Posts, likes, shares	TikTok: 1B+ videos analyzed daily for trends
IoT Sensors	Temperature, motion, pressure	Tesla cars: 10M+ data points/car/day

Source	Data Generated	Real Example
E-commerce	Clicks, cart history, purchases	Alibaba: 1.4B transactions/day during Singles' Day
GPS	Location trails, traffic patterns	Uber: 20M+ trips/day mapped globally

Types of Big Data:




Audios
Zip files
JSON
Semi-
structured
Data
E-mails
XML
Other
Markup
language


Videos
Audios
Test
Messages
Chats
Unstructured
Data
Social
Media-
Data
Web pages
Images
Free-form
Text

4. Big Data Storage Challenges

Challenge	Impact	Solution Example
Volume Management	Storing 175 ZB global data by 2025	Distributed storage (HDFS)
Data Velocity Handling	Real-time processing needs	Apache Kafka for stream processing
Scalability	Sudden traffic spikes (e.g., product launch)	Cloud auto-scaling (AWS)
Cost Efficiency	$20K/TB/year for on-prem vs. $2K on cloud	Google Cloud Storage Nearline

- **Data velocity**: Your data must be able to move quickly between processing centers and databases for it to be helpful in real-time applications.

---

5. Big Data Storage Solutions

Technology	Purpose	Real Implementation
Distributed File Systems	Split data across clusters	Facebook: 300+ PB on Hadoop HDFS
NoSQL Databases	Handle unstructured data	Netflix: Cassandra DB for user profiles

Technology	Purpose	Real Implementation
Columnar Databases	Fast analytics queries	Amazon Redshift for business intelligence
Cloud Storage	Scalable, pay-as-you-go	Spotify: 100+ PB on Google Cloud

---

6. Top Management Challenges & Solutions

1. **Data Quality Issues**
    - *Problem*: 30% of business data is inaccurate.
    - *Solution*: Automated cleansing tools (Talend, Trifacta).
    - *Case*: HSBC reduced errors by 60% with data quality pipelines.
2. **Data Integration**
    - *Problem*: Merging CRM, ERP, and social media data.
    - *Solution*: Apache NiFi for data flow automation.
3. **Governance**
    - *Problem*: GDPR/CCPA compliance across 100+ sources.
    - *Solution*: Collibra for metadata management.
4. **Analytics Preparation**
    - *Problem*: Months spent cleaning data for ML.
    - *Solution*: Databricks for unified analytics.

---

7. Benefits of Big Data Management

Benefit	Impact	Case Study
Cost Savings	Cloud storage: 60% cheaper than on-prem	Pinterest: $20M/year saved with AWS S3
Personalized Marketing	Targeted campaigns boost conversion	Starbucks: 23% sales increase via AI recommendations
Improved Accuracy	Better forecasting models	FedEx: 99% on-time delivery with route optimization

9. Future Trends

1. **AI/ML Integration**
    - *Example*: Google's BERT analyzes search queries in real-time.
2. **Cloud-Native Storage**
    - Multi-cloud solutions (AWS + Azure + GCP).
3. **Enhanced Analytics**
    - Real-time dashboards: Tesla factory monitors production via Splunk.
4. **Security Innovations**
    - Homomorphic encryption (processing encrypted data).

Industry Applications

- **Healthcare**: IBM Watson analyzes 200M medical papers for cancer treatment.

- **Agriculture**: John Deere sensors optimize crop yields using soil data.
- **Finance**: Mastercard AI detects fraud in 250M transactions/day.

Key Takeaway:
Big Data transforms raw information into actionable intelligence. For example:

- **Structured Data**: SQL analysis of sales → inventory optimization
- **Unstructured Data**: NLP on customer reviews → product improvements
- **Semi-structured Data**: IoT sensor patterns → predictive maintenance

Big Data Security

---

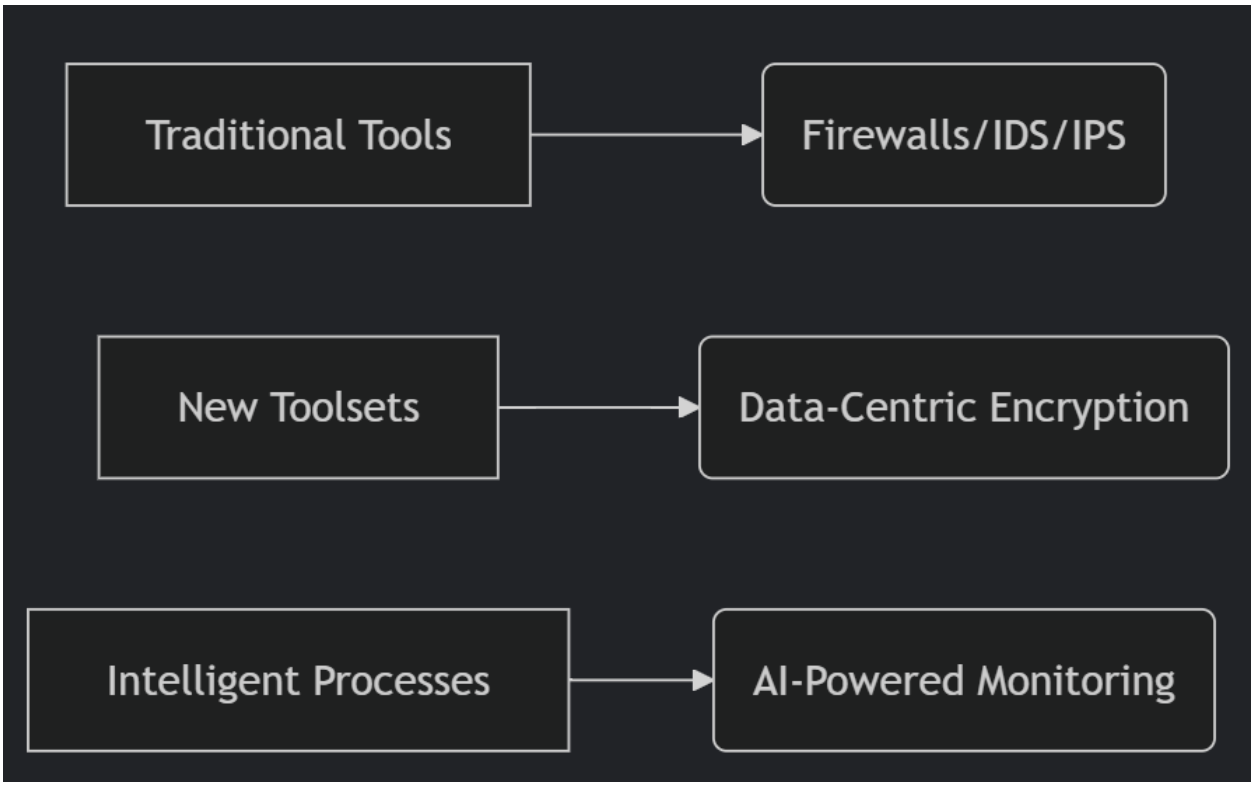
1. What is Big Data Security?

Definition:
Protecting large-scale datasets throughout their lifecycle (ingestion → storage → output) to prevent breaches, ensure compliance, and maintain operational integrity.

Core Challenge:
Big Data environments are prime targets due to their high-value insights:

- *Ransomware Threat*: 2023 MOVEit attack compromised 2,000+ organizations
- *Data Theft*: Facebook-Cambridge Analytica scandal exposed 87M profiles

Security Approach:



How Big Data Security Works?

❑ Big data security's mission is clear enough: keep out on unauthorized users and intrusions with firewalls, strong user authentication, end-user training, and intrusion protection systems (IPS) and intrusion detection systems (IDS)

---

2. The 3-Stage Security Framework

Stage 1: Data Ingress (Input)

- **Sources**: CRM (customer relationship management) systems, IoT sensors, social media, transaction logs
- **Threats**: Malicious data injection, phishing attacks
- **Protection**:
    - *Example*: AWS Kinesis Firehose encrypts streaming data from 10M+ Tesla vehicles
    - *Tool*: Apache NiFi for secure data pipeline automation

Stage 2: Stored Data

- **Challenges**:
    - Distributed clusters (100s of nodes)
    - Multi-format data (JSON, Parquet, AVRO)
- **Protection**:
    - *Encryption-at-rest*: AES-256 in Hadoop HDFS
    - *Authentication*: Kerberos for user access control

Stage 3: Data Output

->The entire reason for the complexity and expense of the big data platform is so it can run meaningful analytics across massive data volumes and different types of data to fetch relevant data

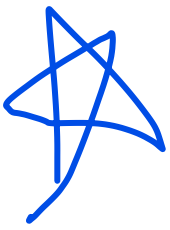
- **Risks**:

    - Sensitive insights leakage (e.g., healthcare analytics)
    - Compliance violations (GDPR/HIPAA)
- **Protection**:
    - *Data Masking*: Anonymize PII in dashboards
    - *Example*: United Airlines redacts passenger info in operational reports

---

3. Emerging Trends & Countermeasures

Trend	Security Risk	Solution
IoT Proliferation	5.5M attacks/day on smart devices (2023)	Network segmentation + TLS 1.3 encryption
Consumer Data Ownership	GDPR fines up to 4% of revenue	Blockchain-based consent management (IBM Food Trust)
Cloud Misconfiguration	80% of breaches due to human error	Automated compliance checks (Azure Policy)
AI Data Collection	Deepfake voice scams costing $2.6B/year	Synthetic data for ML training (Mostly AI)



Two Conflicting Trends in Big Data

1. **Proliferation of Big Data for Smart Technology**
    - Technologies like IoT, AI, machine learning, and CRM systems are generating and leveraging massive volumes of data.

- This data fuels intelligent systems, enabling personalized services, predictive analytics, and automation.
- Enterprises benefit from this by tailoring products and services more effectively to consumer needs.

2. **Consumer Data Ownership and Privacy Movement**
   - At the same time, there's a growing push for individuals to control how their personal data is collected, stored, and used.
   - This includes demands for transparency, consent, and ethical data handling.
   - Regulations and public awareness are pressuring companies to adopt responsible data governance.

➔ i.e. I want services like semantic searches and summarized data from my database but I am reluctant to make my data available to models for training for even better results.

4. Critical Security Practices

1. **Cloud Infrastructure Updates**

- Cloud security is often established based on old legacy security principles, and as a result, cloud security features are misconfigured and open to attack. Thus it's recommended to update old infrastructure
  - *Fix*: Enable AWS GuardDuty + automated remediation

2. **Mobile/IoT Device Management**

- Set strict policies for how employees can engage with corporate data on personal devices, and be sure to set additional layers of security in order to manage which devices can access sensitive data
  - *Policy*: BYOD restrictions + containerization (VMware Workspace ONE)

- *Example*: Hospitals use MDM to separate patient data on doctors' phones

3. **Employee Training**

- Most often, big data is compromised as the result of a successful phishing attack or other personalized attack targeted at an unknowing employee. Company/service is reliable however incompetent employees may leak the data.
    - *Threat*: 95% of breaches start with phishing
    - *Defense*: Simulated attacks + mandatory MFA (Multi-Factor Authentication)
    - *Stat*: Google reduced compromise by 50% with Security Keys

---

5. Benefits of Big Data Security

Benefit	Mechanism	Business Impact
Customer Retention	Secure personalization builds trust	Amazon: 35% repeat customers via recommendations
Risk Identification	Anomaly detection in real-time	JPMorgan: $150M saved stopping fraud
Business Innovation	Safe data sharing for R&D	Pfizer: Accelerated vaccine trials by 60%
Cost Optimization	Efficient storage/processing	Spotify: 30% lower cloud costs with encryption

6. Key Challenges & Solutions

Challenge	Scenario	Mitigation Strategy
Vulnerable New Tech	It can be difficult for security software and processes to protect these new toolsets.	Runtime application self-protection (RASP)
Unauthorized Data Mining	data administrators may decide to mine data without permission or notification	your security tools need to monitor and alert on suspicious access no matter where it comes from
Audit Complexity	1 PB = 20M filing cabinets of data	Automated auditing (Splunk Security Orchestration)
Update Fatigue	Equifax breach due to unpatched Apache	Immutable infrastructure (Docker + Kubernetes)

7. Security Technologies

Technology	Function	Industry Example
Encryption	Data protection in transit/at rest	Apple iMessage: End-to-end encryption
Centralized Key Management	Unified control of encryption keys	AWS KMS managing 1B+ keys for Nasdaq

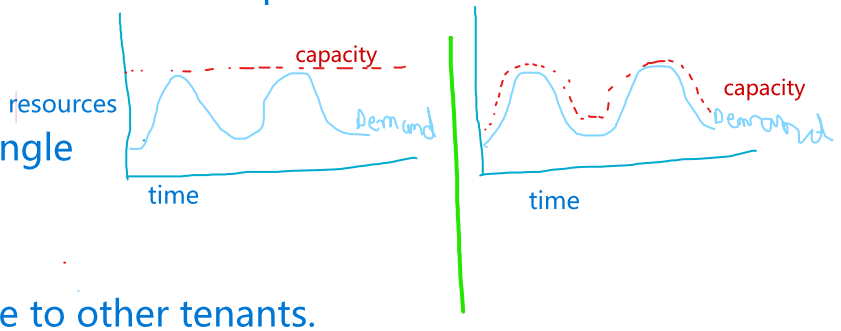
Technology	Function	Industry Example
User Access Control	Least-privilege permissions	Salesforce: Dynamic roles for 50K employees
Intrusion Detection (IDS)	Network anomaly alerts	Darktrace AI stopping ransomware at Maersk
Physical Security	Data center protections	Google's biometric access for server farms

Key Takeaway:
Big Data security requires a *layered approach*:

1. **Prevent** with encryption/access controls
2. **Detect** via AI-powered monitoring
3. **Respond** through automated incident playbooks
4. **Govern** using compliance-as-code frameworks

Q How to achieve Scalability and Elastic sol. using Dynamic provisioning

Dynamic Provisioning of resources based on requirement



Multi-tenant design:

This is an architectural pattern where a single instance of a software application and its underlying infrastructure serves multiple customers ("tenants").
Each tenant's data is isolated and invisible to other tenants.
(Virtualization)

Hadoop and mapreducer

1. What is Hadoop?

Definition:
An open-source framework for distributed storage and processing of large datasets across in a parallel and distributed manner. Written in Java, it handles **big data** using the **MapReduce** programming model.

Unlike rows and columns which are limited to text , Hadoop uses cloud storage by distributing it's database across different clusters

Historical Context:

- Created by **Doug Cutting** and **Mike Cafarella** (2006), named after Cutting's son's toy elephant.
- Inspired by Google's **GFS** (2003) and **MapReduce** paper.
- Apache Foundation maintains it today.

---

2. Hadoop Architecture & Components

Core Components:

1. **HDFS (Hadoop Distributed File System)**
    - *Purpose*: Distributed storage across multiple machines.
    - *Design*:
        - Splits files into **blocks** (default 128 MB).
        - Replicates blocks across nodes (default 3x) for fault tolerance.
    - *Example*: A 1 GB file is split into 8 blocks stored on different servers.
2. **YARN (Yet Another Resource Negotiator)**
    - *Purpose*: Manages cluster resources (CPU, memory) and schedules tasks.

- *Workflow*:
    - **ResourceManager**: Allocates resources to applications.
    - **NodeManager**: Monitors resources on individual nodes.

3. **Hadoop Common**
    - Libraries and utilities supporting other modules.

---

4. Hadoop Limitations

- **Not for Small Data**: Overhead outweighs benefits for datasets <100 GB.
- **Complex Management**: Requires dedicated DevOps team (e.g., Cloudera Manager).
- **Security Risks**: No built-in encryption (requires Apache Ranger).
- **Latency**: Batch-oriented → unsuitable for real-time processing.

---

5. What is MapReduce? (CPU of Hadoop)

Definition:
A programming model for parallel processing of large datasets in two phases:

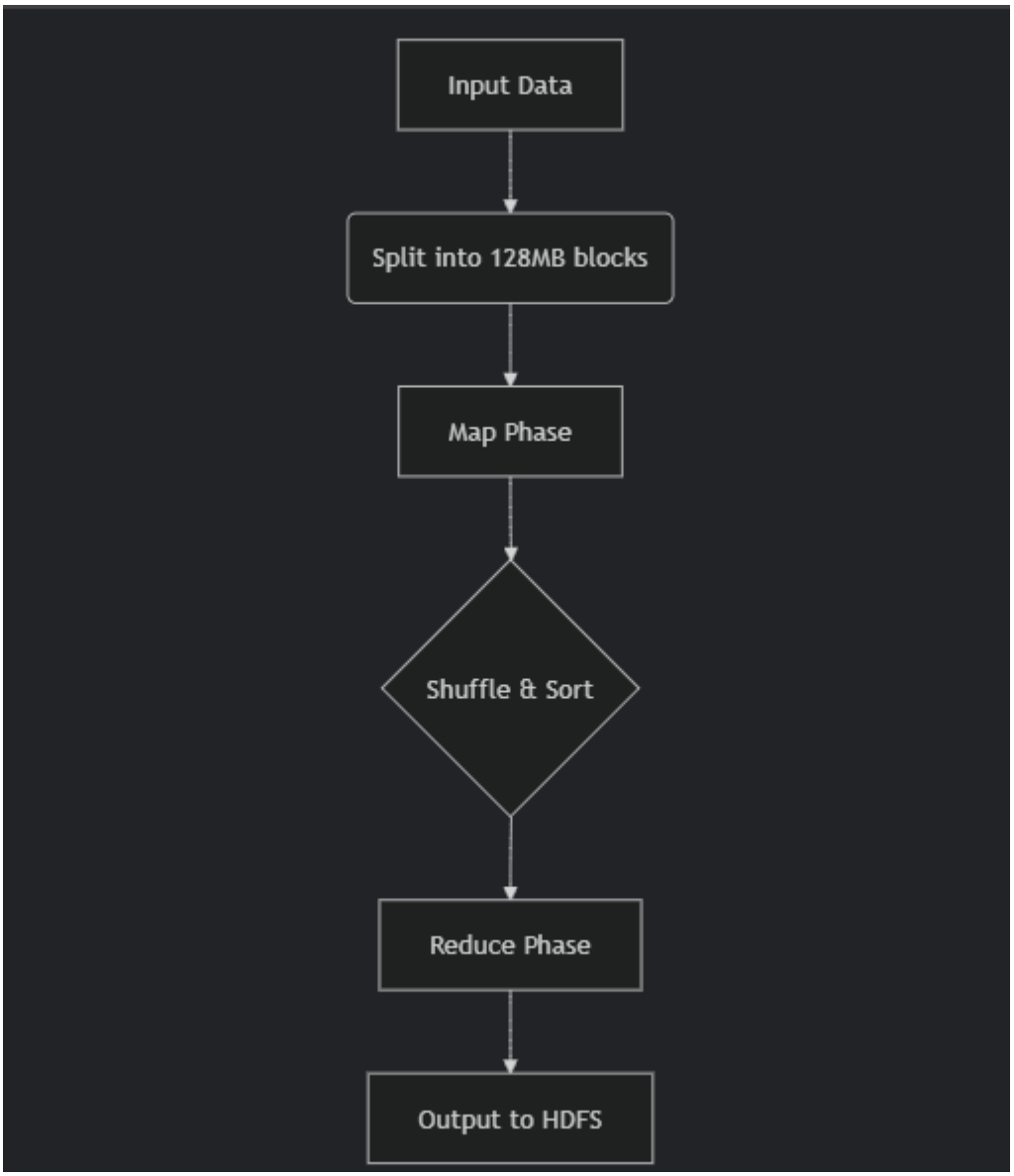
1. **Map**: Filters/sorts data (e.g., counting words in a document).
2. **Reduce**: Aggregates results (e.g., summing word counts).

Analogy:
Census data collection:

- **Mappers** = Surveyors collecting data from each neighborhood.
- **Reducers** = Statisticians compiling city-wide totals.

---

6. MapReduce Workflow



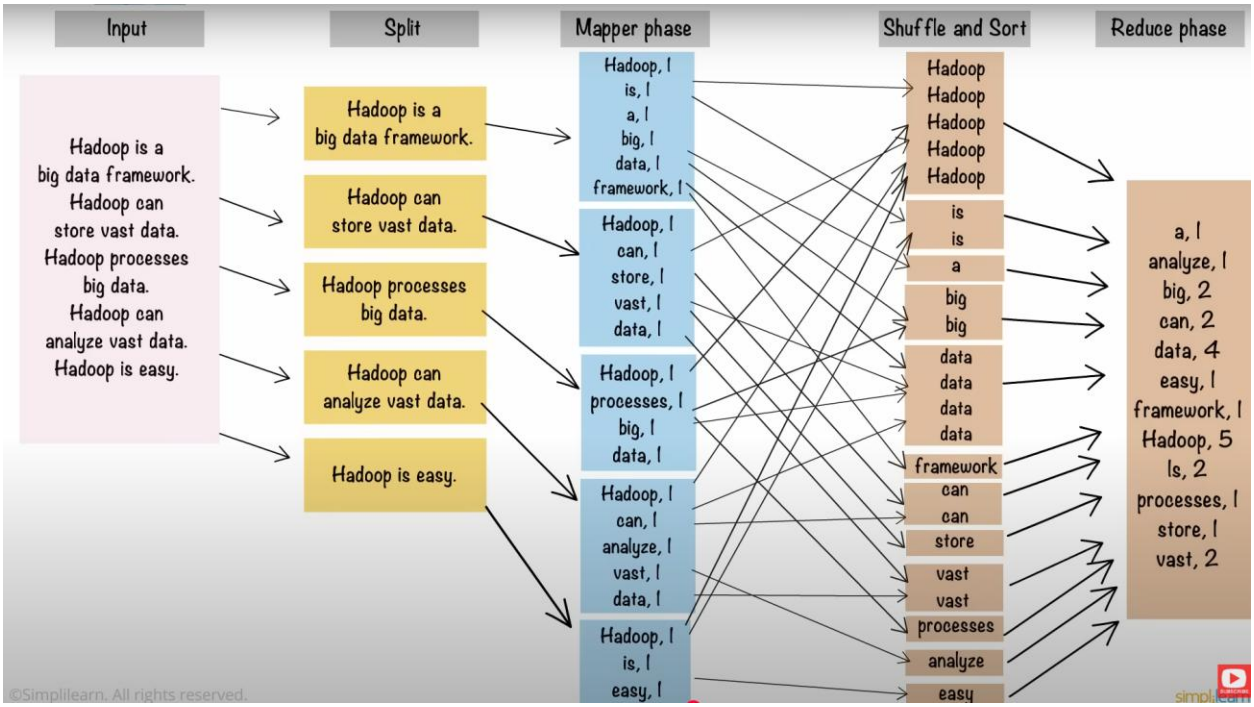
Step-by-Step Process:

1. **Input Splits**:
    - File divided into blocks (e.g., 128 MB each).
2. **Mapping**:
    - Each block processed by a **mapper** task.
    - Outputs key-value pairs (e.g., 〈"apple", 1〉).
3. **Shuffling**:

- Groups values by key (e.g., 〈"apple", [1,1]〉).

4. **Reducing**:
   - Aggregates values (e.g., 〈"apple", 2〉).
5. **Output**:
   - Saved to HDFS (e.g., part-r-00000).

Eg.



7. Real-World Example: Word Count

Input Text:

text

"Hello World"

"Hello Hadoop"

Map Phase:

- Mapper 1: 〈"Hello",1〉, 〈"World",1〉
- Mapper 2: 〈"Hello",1〉, 〈"Hadoop",1〉

Shuffle & Sort:

- 〈"Hello", [1,1]〉, 〈"Hadoop", [1]〉, 〈"World", [1]〉

Reduce Phase:

- Reducer 1: 〈"Hello", 2〉
- Reducer 2: 〈"Hadoop", 1〉, 〈"World", 1〉

Output:

text

Hello 2

Hadoop 1

World 1

`9. Hadoop vs. Modern Alternatives

Criteria	Hadoop MapReduce	Apache Spark
Speed	Batch (minutes-hours)	In-memory (seconds)
Ease of Use	Low (requires Java)	High (Python/SQL APIs)
Use Case	ETL, historical analysis	ML, real-time streaming

💡 **Did You Know?**

LinkedIn uses Hadoop to process **1.5 TB of data daily** for job recommendations.

---

10. Key Takeaways

- **Hadoop** = Scalable, fault-tolerant storage (HDFS) + resource management (YARN).
- **MapReduce** = Batch processing model for large-scale data.
- **Use When**: Data >100 TB, batch processing acceptable, budget constrained.
- **Avoid When**: Real-time analytics needed or data is small.

Future Trend: Hybrid architectures (e.g., Hadoop + Spark) for batch + real-time processing.

End of Mid-sem syllabus

Virtualization

1. What is Virtualization?

Definition:
Virtualization is the technology that creates an abstract, software-based (or *virtual*) representation of physical IT resources like servers, storage, networks, and applications. It allows multiple virtual environments to run simultaneously on a single physical machine.

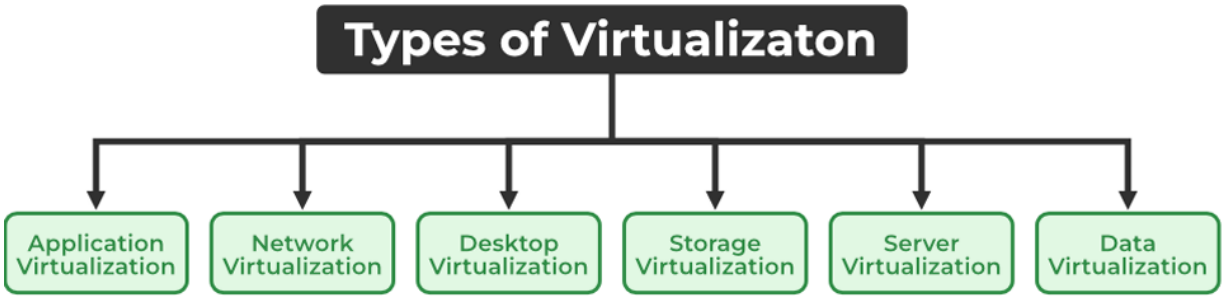
Core Concept: Decoupling the software from the underlying hardware.

- **Host Machine**: The physical server that hosts the virtual environments.
- **Guest Machine**: The virtual environment (Virtual Machine, container, etc.) that runs on the host.

Historical Context: Originally developed for mainframe computers to maximize the use of expensive hardware.

---

3. **Types of Virtualizations (With Real-World Examples)**



1. Application Virtualization: Application virtualization helps a user to have remote access to an application from a server. The server stores all personal information and other characteristics of the application but can still run on a local workstation

through the internet. The application is packaged to run in a self-contained, isolated environment ("bubble" or "sandbox") on the host OS, without being installed in the traditional way.

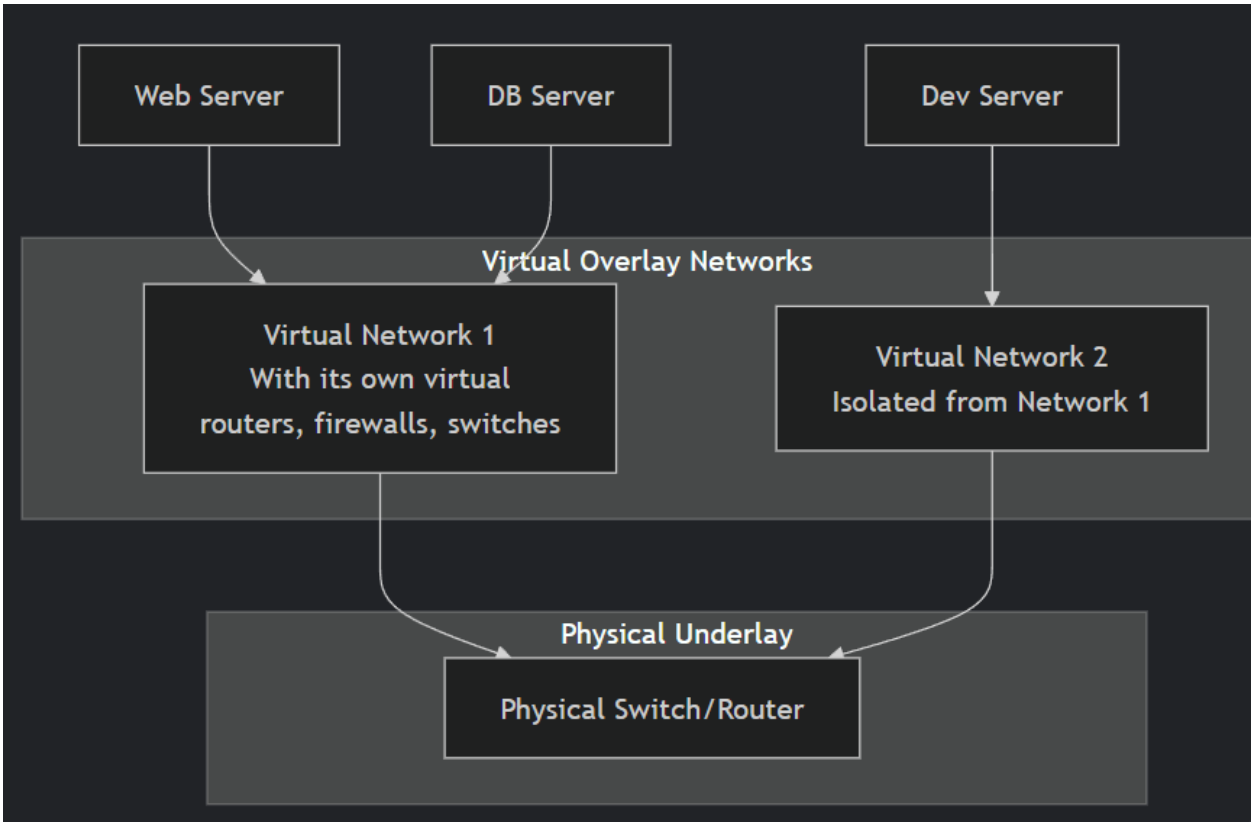
Real-World Example:

- **Microsoft Office 365 Click-to-Run:** It uses application virtualization to download and run Office in a managed state, allowing for easy updates and coexistence with older versions.

2. Network Virtualization

Concept: This abstracts the physical network hardware (switches, routers, firewalls) into software. It allows you to create multiple, isolated virtual networks on top of a single physical network infrastructure.

Network virtualization provides a facility to create and provision virtual networks, logical switches, routers, firewalls, load balancers, Virtual Private Networks (VPN), and workload security within days or even weeks.



3. Desktop Virtualization (VDI - Virtual Desktop Infrastructure)

Concept: The desktop environment (operating system, applications, data) is hosted on a centralized server in a data center. Users access their personal desktop remotely from any device.

Eg. call center desktops accessed by virtual desktop on personal machines

4. Storage Virtualization: Storage virtualization is an array of servers that are managed by a virtual storage system.

Pools storage from multiple physical devices into a single, centralized virtual storage pool that is managed from a central console.

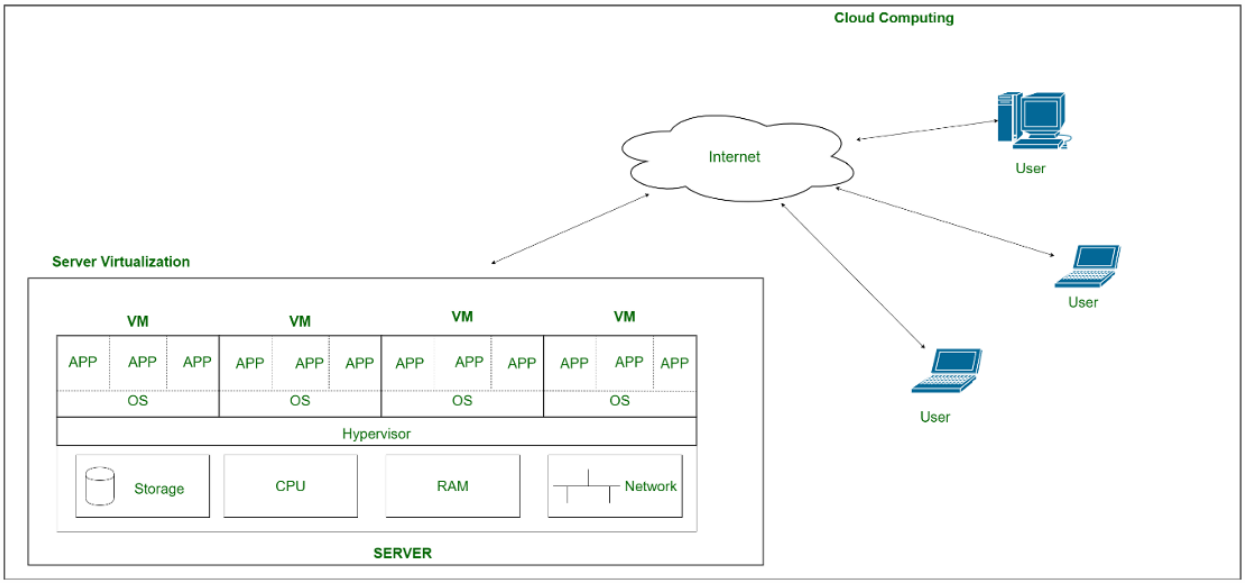
Eg. Google Drive/Dropbox: You see a single, continuous storage space, but your files are actually distributed across many physical hard drives in multiple data centers.

5. Server Virtualization

Concept: It involves partitioning a physical server into multiple, isolated virtual servers using a hypervisor. Each virtual server can run its own operating system and applications independently.

It causes an increase in performance and reduces the operating cost by the deployment of main server resources into a sub-server resource. It's beneficial in virtual migration, reducing energy consumption, reducing infrastructural costs, etc.

How it works: The hypervisor sits between the hardware and the operating systems, allocating physical resources (CPU, RAM, storage, network) to each virtual machine (VM).



6. Data Virtualization: This is the kind of virtualization in which the data is collected from various sources and managed at a single place without knowing more about the technical information like how data is collected, stored & formatted.

Example. A dashboard of any database

---

4. Key Benefits & Drawbacks

Benefits	Drawbacks
✅ **Cost Reduction:** Drastically reduces hardware costs (server sprawl), energy consumption, and data center cooling needs.	❌ **High Initial Investment:** Requires powerful servers and purchasing hypervisor licenses (e.g., VMware).
✅ **Increased Efficiency & Utilization:** Boosts hardware utilization from 5-15% to 80% or higher.	❌ **Complexity & Skill Gap:** Requires specialized IT skills to manage and troubleshoot a virtualized environment.
✅ **Agility & Scalability:** New virtual servers can be provisioned in minutes, not days/weeks ("spin up a VM").	❌ **Security Risks:** A compromised hypervisor could potentially expose all guest VMs on that host.
✅ **Improved Disaster Recovery & Business Continuity:** VMs are portable files that can be easily backed up and moved to another host with minimal downtime.	❌ **Performance Overhead:** The hypervisor layer adds a small performance penalty, though this is often negligible on modern hardware.
✅ **Enhanced Security:** VMs are isolated from each other. A crash or malware infection in one VM is contained.	❌ **Single Point of Failure:** If the physical host server fails, all VMs running on it go down (mitigated by clustering).

Real World applications of Cloud Computing

Google Cloud excels in ML-based content moderation with its Vertex AI platform, which supports custom model training (tensorflow, sklearn)and automated workflows for detecting inappropriate content in images, videos, and text. It integrates seamlessly with BigQuery and Cloud Storage for data handling and analytics, enhancing the accuracy and transparency of moderation models.

AWS complements this with its reliable storage service (like Netflix), Amazon S3, for hosting large volumes of user-generated content securely and efficiently. AWS also offers AI moderation tools like Rekognition for image/video analysis and Comprehend for natural language processing, enabling real-time content scanning. Serverless components like Lambda help orchestrate moderation workflows smoothly.

Azure's strength lies in content delivery via its extensive global CDN (content delivery network), which caches moderated content close to users for low latency and high availability. It supports advanced features like HTTPS, custom domains, and robust security including DDoS protection. Integrating Azure CDN with AWS storage and Google Cloud moderation enables a performant, secure, and scalable multi-cloud content moderation and distribution system.