Introduction To Big Data Analytics INSY 8413

ADVENTIST UNIVERSITY OF CENTRAL AFRICA

Instructor:

- Eric Maniraguha | eric.maniraguha@auca.ac.rw | LinkedIn Profile

6h00 pm – 8h50 pm

- Monday A-G104
- Tuesday E-G108
- Wednesday A-G104
- Thursday E-G108

June 2025

Reference reading

- Big Data Ecosystem
- What is a Data Ecosystem?
- Article: Big data ecosystem
- Journal Paper: Big data analysis and cloud computing for smart transportation system integration

Learning Objectives

Learning Objectives
By the end of this lecture, students will be able to:

- **Explain** what the Hadoop ecosystem is and describe the roles of HDFS, MapReduce, and YARN.
- **Understand** the basic concepts and advantages of Apache Spark for big data processing.
- **Identify** different types of NoSQL databases (MongoDB, Cassandra, HBase) and explain when to use them.
- **Compare** how NoSQL databases differ from traditional relational databases.
- **Describe** how cloud platforms like AWS, Microsoft Azure, and Google Cloud support big data storage and processing.
- **Recognize** the main benefits and challenges of using cloud services for big data solutions.
- **Decide** which big data tools or platforms are most suitable for different types of data problems.

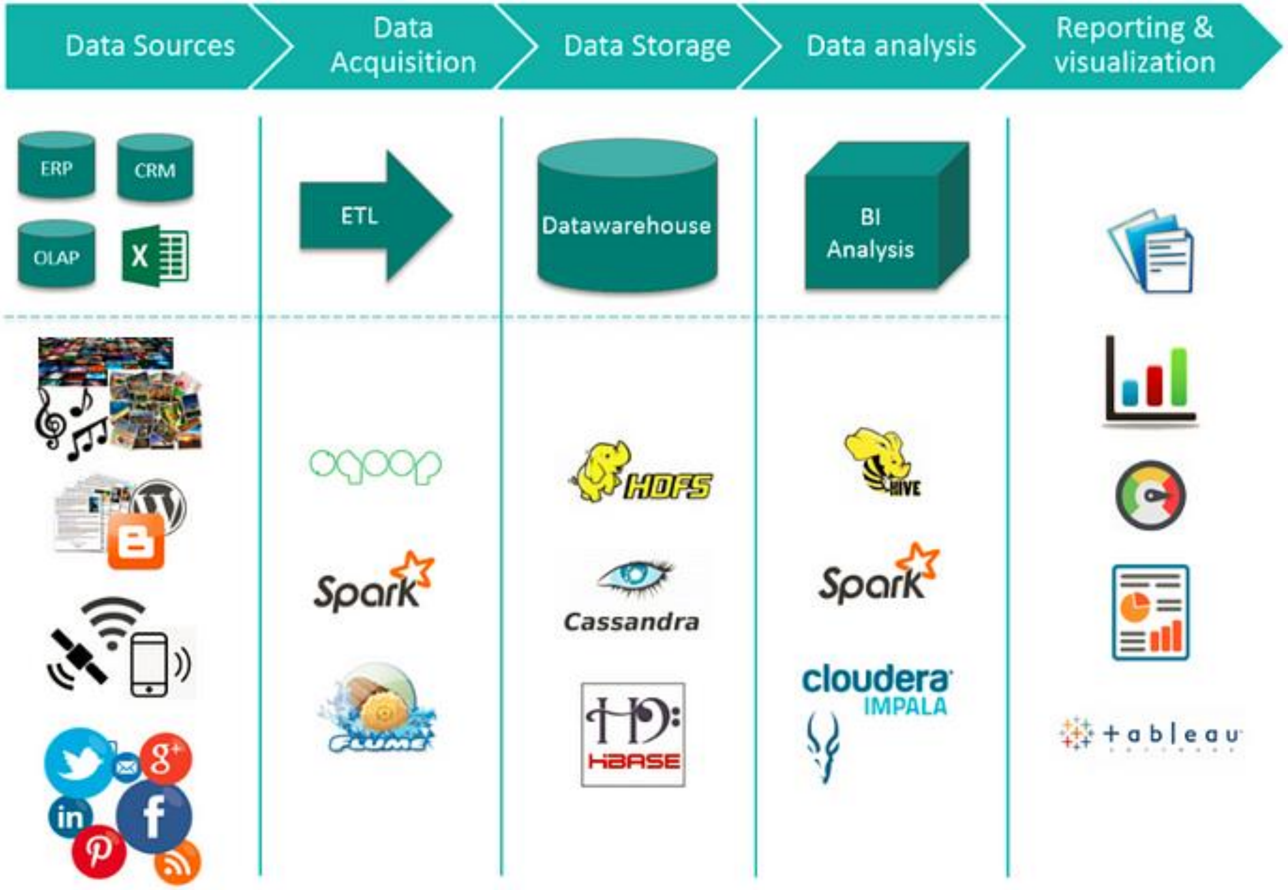
What is a Data Ecosystem?

A **data ecosystem** is the combination of infrastructure, applications, people, and processes used by an organization to collect, manage, and analyze data.

- Helps organizations understand customers and optimize operations
- Every organization has a unique ecosystem
- May use internal or public data sources

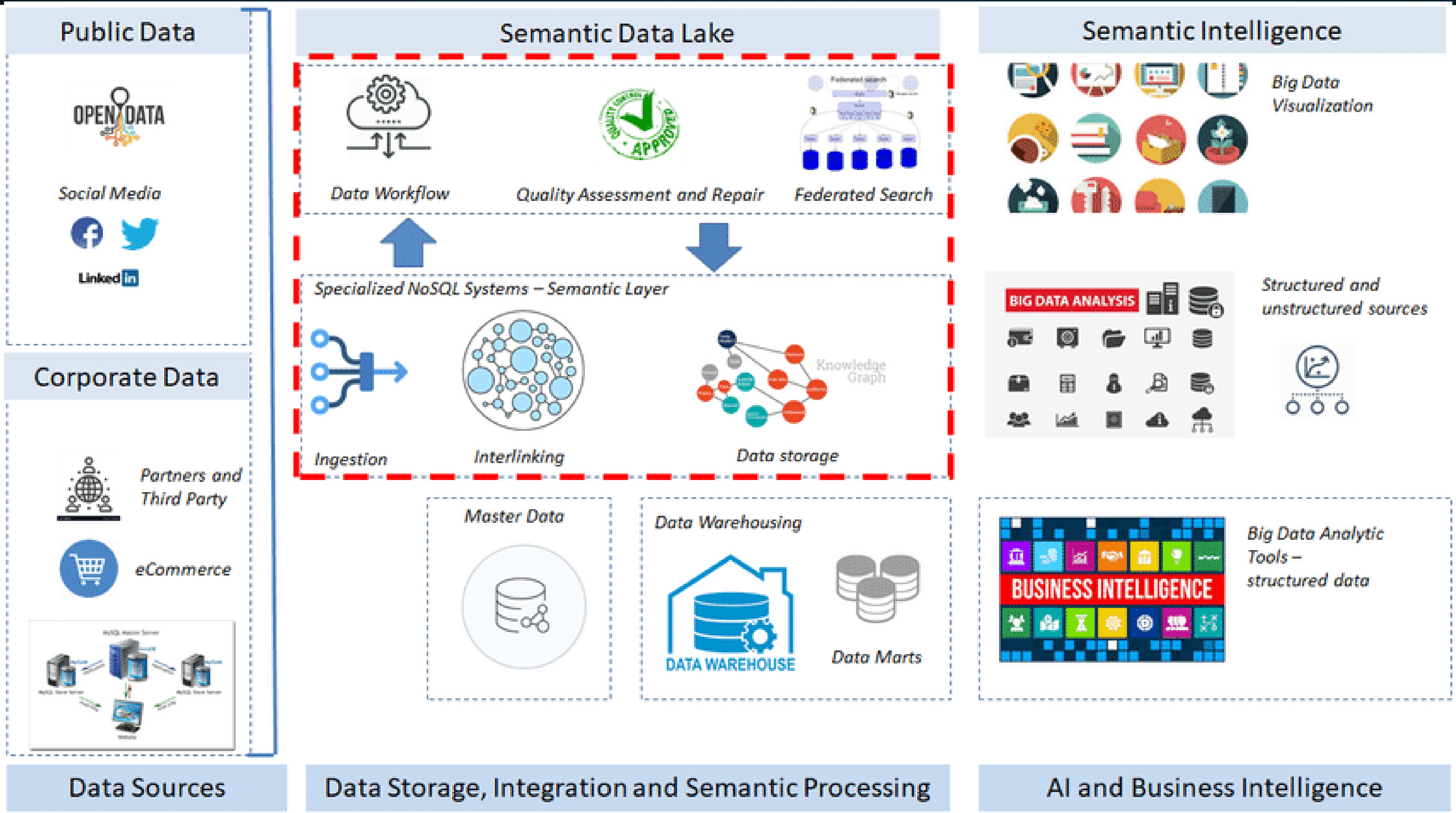
Key Components of a Data Ecosystem

- **People** – Users who generate, analyze, and make decisions using data
- **Technology** – Tools and platforms for data collection, storage, and analysis
- **Processes** – Workflows that define how data is managed and used



Source Image: https://medium.com/@jain6968/big-data-ecosystem-b0e4c923d7aa

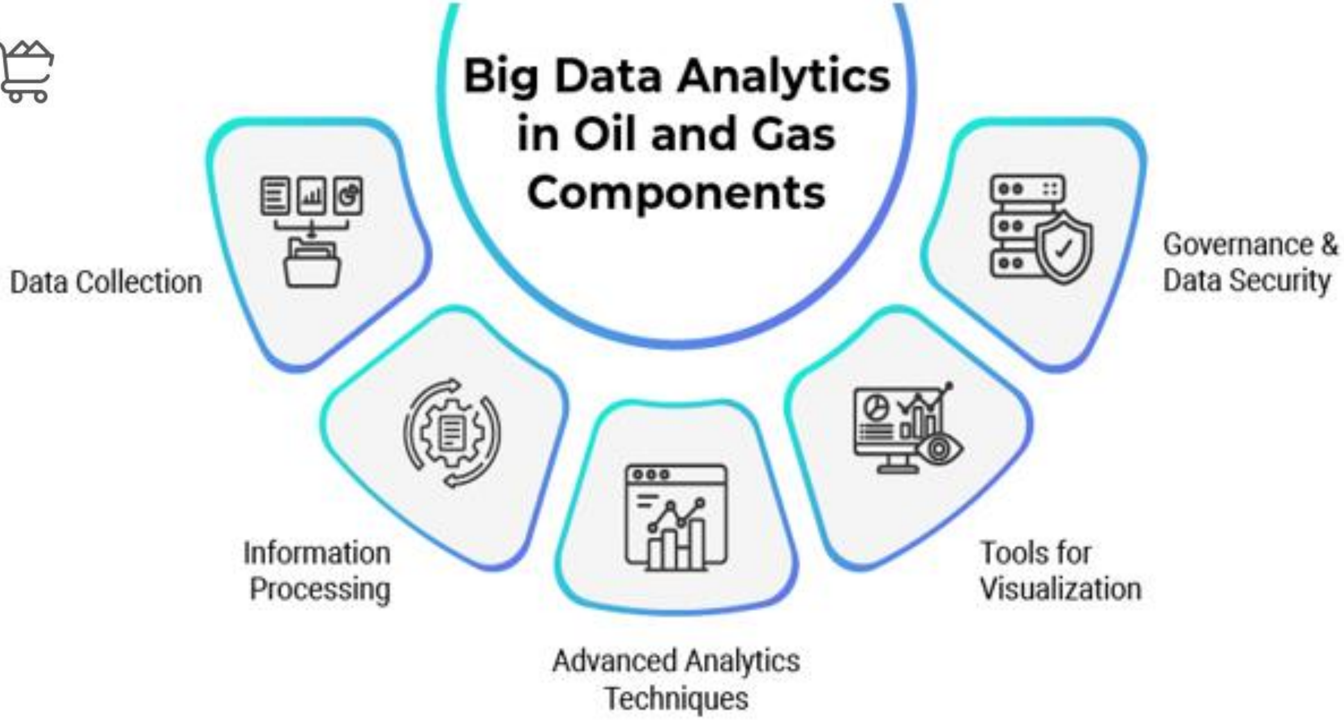
Big Data Ecosystem



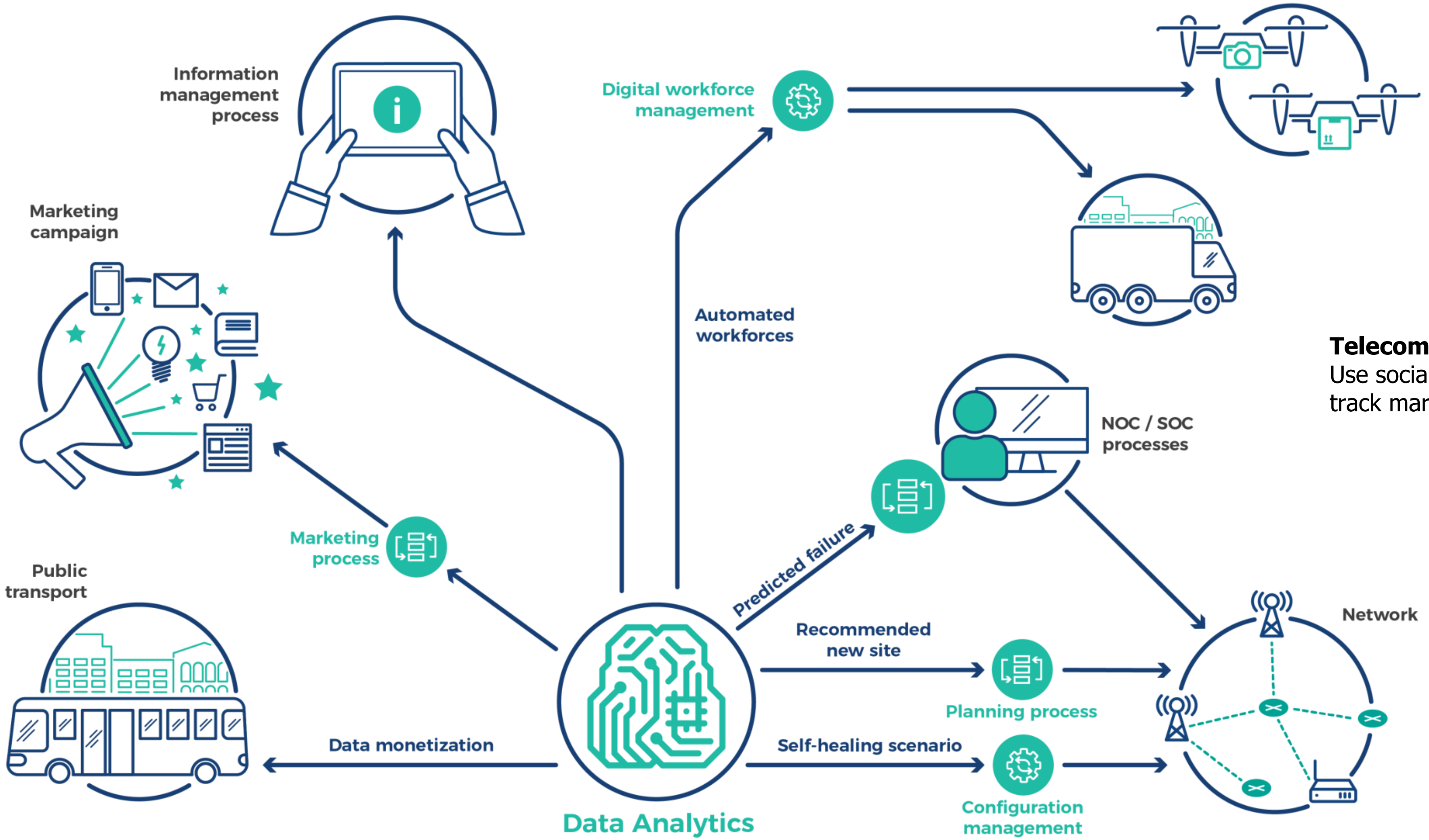
Real-Life Use Cases – Retails & BDA Oil & Gas

Retail & Supply Chain:
Use supplier and economic data to forecast demand and reduce stockouts

Oil & Gas: Use traffic, GPS, and weather data to optimize route planning and improve delivery performance.

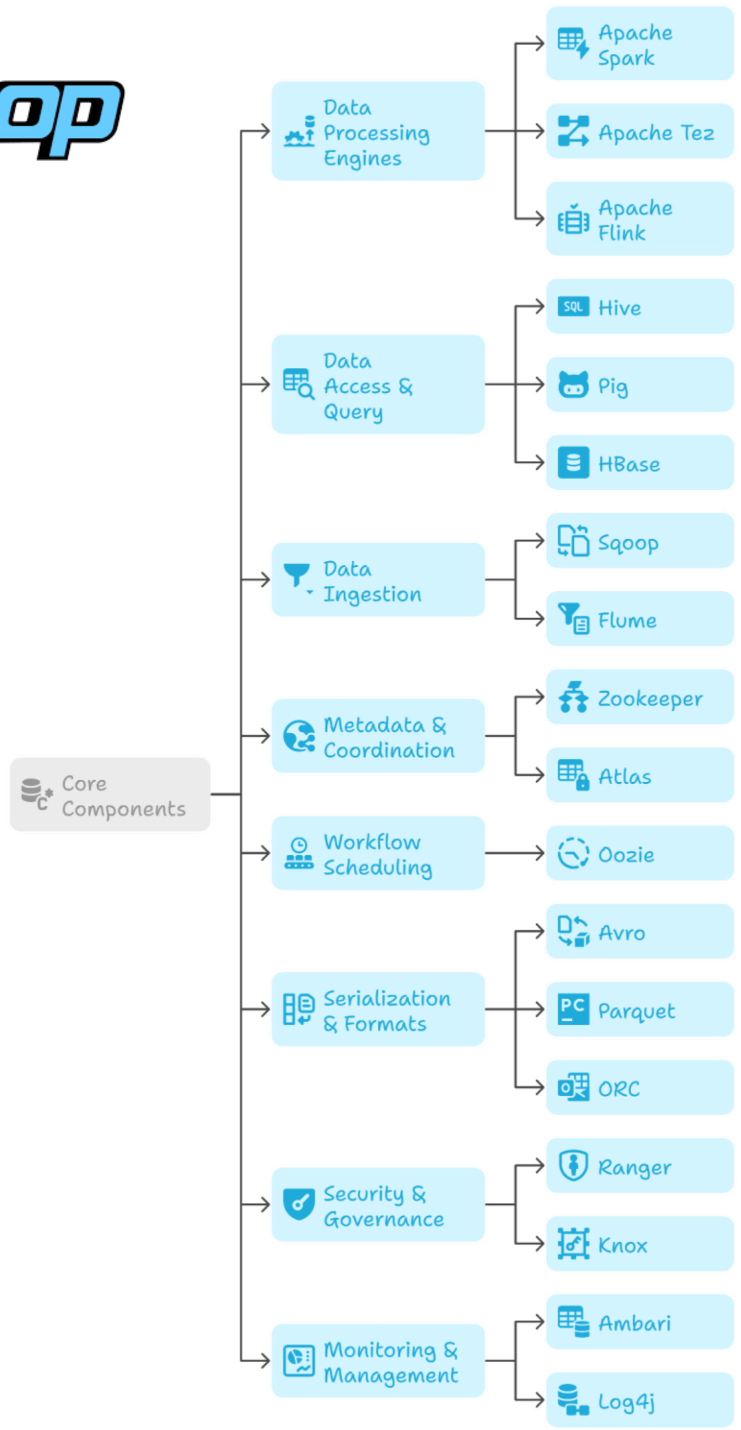


Real-Life Use Cases - Telecommunications



Telecommunications:
Use social media, customer, and competitor data to track market trends.

Hadoop Ecosystem Overview



Building the Hadoop Powerhouse

Hadoop is an open-source software framework used for storing and processing large datasets using distributed computing.

HDFS (Hadoop Distributed File System)
Scalable, fault-tolerant storage system for big data.

MapReduce
Parallel processing framework for large datasets.

YARN (Yet Another Resource Negotiator))
Resource management and job scheduling platform.

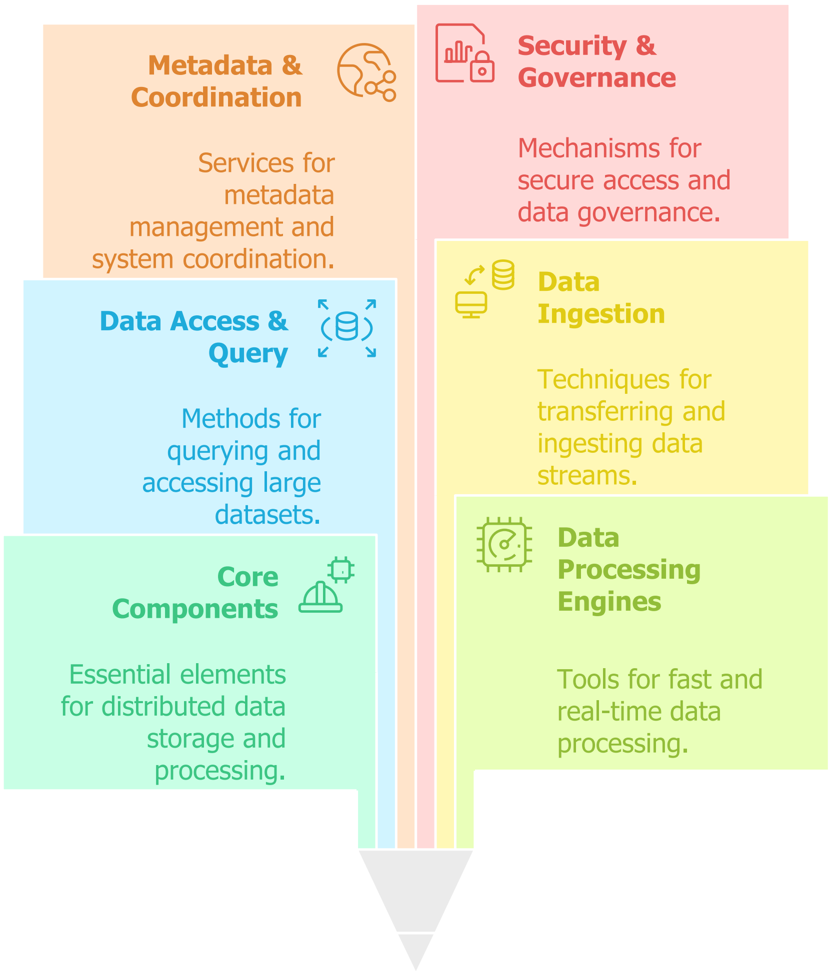
Hive
Data warehousing tool for querying and analyzing data.

HBase
NoSQL database for real-time data access.

Zookeeper
Centralized service for maintaining configuration information.

Hadoop Ecosystem

Building a Unified Hadoop Platform



BD Ecosystem - Apache Spark fundamentals

Apache Spark is an **open-source distributed computing system** designed for **fast, in-memory big data processing**. It is widely used for **data analytics, machine learning, and graph processing**.

Common Use Cases

Illustrates applications in big data processing and machine learning

Key Features

Highlights Spark's speed, ease of use, and fault tolerance

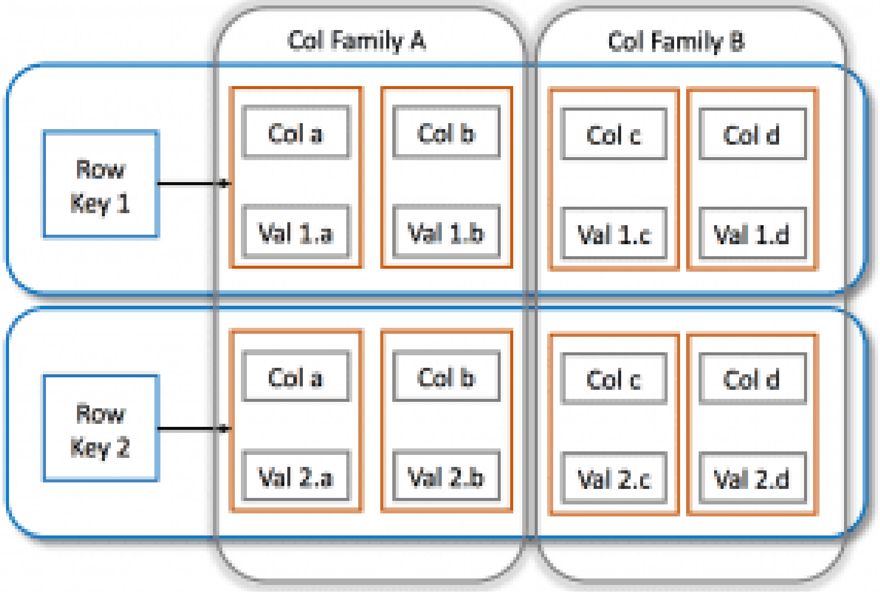
Architecture

Describes the components like Driver Program and RDDs

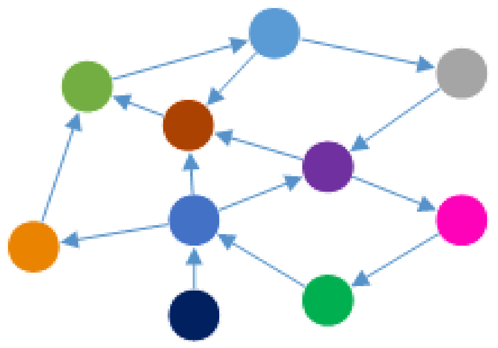
BD Ecosystem - NoSQL databases (MongoDB, Cassandra, HBase)

NoSQL databases are **non-relational**, meaning they **do not use the traditional table-based structure**. Instead, they store data in formats like:

Wide-column stores



Graph Database

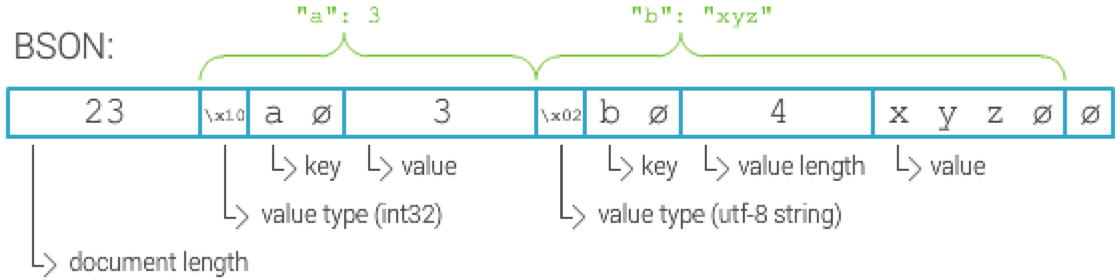


Documents (JSON, BSON)

JSON:

```
{
  "a": 3,
  "b": "xyz"
}
```

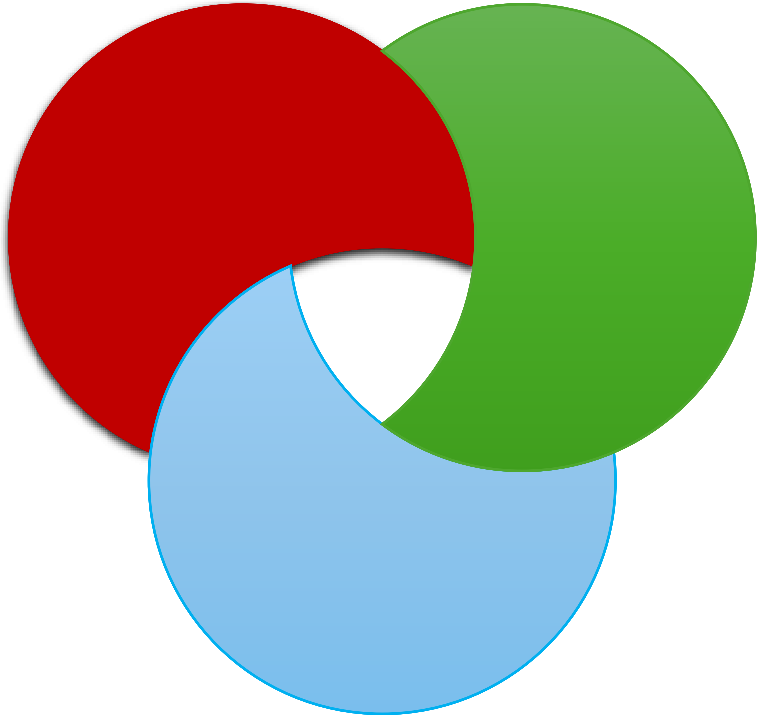
BSON:



Key-Value Store

NoSQL databases (MongoDB, Cassandra, HBase)

Column-oriented database built on Hadoop



Document-oriented database with flexible schemas

Wide-column store for high availability and scalability

Use Cases:

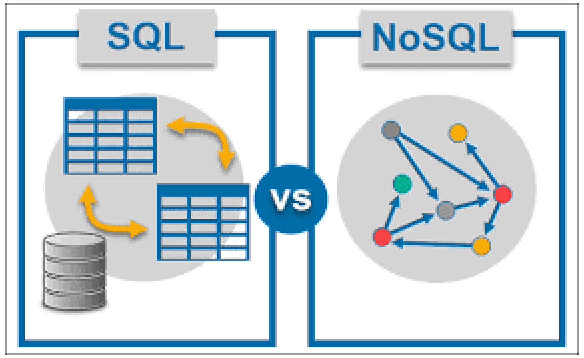
- Real-time analytics → Redis, Cassandra
- IoT data → InfluxDB, MongoDB
- Social network relationships → Neo4j
- Search and logs → Couchbase, Elasticsearch (though not a DB per se)

SQL vs NoSQL Databases

Feature	**SQL (Relational DB)**	**NoSQL (Non-relational DB)**
Data Model	Tables (rows & columns)	Key-Value, Document, Column, Graph
Schema	Fixed, predefined schema	Flexible, dynamic schema
Scalability	Vertical (scale-up)	Horizontal (scale-out)
Transactions	Strong ACID compliance	BASE model; eventual consistency
Best for	Structured data, complex queries	Unstructured/semi-structured data, fast development
Examples	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, Redis, Couchbase
Use Case Example	Banking system, inventory management	Real-time analytics, social media feeds, IoT apps

When to Use SQL

- Complex queries and joins
- Strict data integrity
- Structured data



When to Use NoSQL

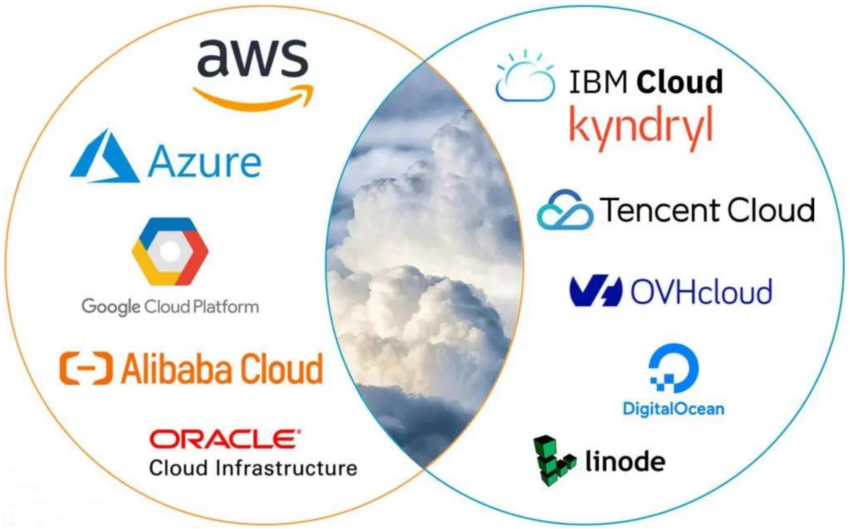
- Flexible or evolving data models
- High-speed, high-volume workloads
- Scalable distributed systems (IoT, social, mobile apps)

The Cloud

The Cloud, or **cloud computing**, refers to the **delivery of computing services over the internet** ("the cloud")—including:

- **Servers**
- **Storage**
- **Databases**
- **Networking**
- **Software**
- **Analytics**
- **Artificial Intelligence** and more...

Instead of owning and maintaining physical servers or data centers, users can **access and pay for what they need**, when they need it.



Key Characteristics

On-Demand Self-Service
Access computing resources automatically without human interaction with the service provider.

Broad Network Access
Services are available over the internet from any device—PCs, phones, tablets.

Resource Pooling
Resources are shared across multiple users via a **multi-tenant model** (e.g., virtual machines running on the same physical servers).

Rapid Elasticity
Scale resources up or down automatically based on demand.

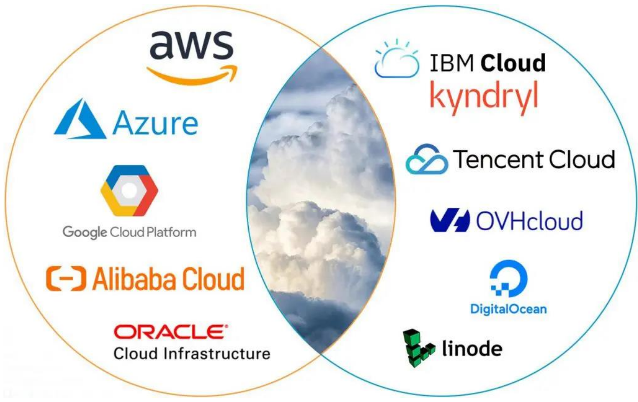
Measured Service (Pay-as-you-go)
You pay only for what you use (e.g., compute hours, storage GBs, etc.).

Cloud Service Models

Model	Description	Examples
IaaS (Infrastructure as a Service)	Basic computing resources like VMs, storage	AWS EC2, Google Compute Engine
PaaS (Platform as a Service)	Environment for app development & deployment	Google App Engine, Heroku
SaaS (Software as a Service)	Ready-to-use software applications	Gmail, Microsoft 365, Dropbox

Cloud Deployment Model

- **Public Cloud**
  - Services provided over the internet (e.g., AWS, Azure, Google Cloud)
- **Private Cloud**
  - Dedicated infrastructure used by one organization (e.g., on-premise or hosted)
- **Hybrid Cloud**
  - Combination of public and private for flexibility and security



Benefit of cloud computing

1. Cost-efficiency (No Hardware Costs)
You don't need to **buy, own, or maintain** physical servers or infrastructure.
Instead, you pay only for what you use (e.g., storage, computing time).
This reduces **capital expenses (CapEx)** and shifts costs to **operational expenses (OpEx)**.

Example: A startup can deploy an app without buying expensive servers, just renting space on AWS or Azure.

2. Scalability
Cloud systems can **scale resources up or down** automatically based on demand.
This ensures your application performs well under both low and high traffic.

Example: An e-commerce site can handle traffic spikes during Black Friday without downtime by scaling up resources instantly.

3. High Availability
Cloud providers offer **redundancy and failover** mechanisms across multiple data centers and regions.
This means your services remain available even if one server or data center fails.

Example: If a power outage affects a server in one region, your application can continue running from another region.

4. Global Reach
Cloud providers have **data centers around the world**.
You can deploy applications closer to users in different regions for faster response times and better user experience.

Example: A company can serve customers in Africa, Europe, and Asia from nearby cloud data centers to reduce latency.

5. Security (With Proper Controls)
Cloud platforms offer **advanced security features**, including:

- Data encryption (at rest and in transit)
- Access control (identity and role management)
- Regular security updates and monitoring

However, security is a **shared responsibility**:

- The **cloud provider** secures the infrastructure
- The **customer** must configure access, monitor usage, and secure their own data/applications

Example: Using AWS, you can encrypt sensitive healthcare data and restrict access to authorized users only.

Key Takeaways

1. **Diverse Data Sources Integration**
   Big Data ecosystems allow integration of **structured, semi-structured, and unstructured** data from varied sources like sensors, logs, social media, and transactional systems, enabling comprehensive analytics.

2. **Scalability and Distributed Computing**
   Technologies like **Hadoop, Spark, and Flink** support scalable, fault-tolerant, and distributed processing, making it possible to analyze petabytes of data efficiently.

3. **Real-time and Batch Processing**
   The ecosystem supports both **batch analytics** (e.g., Hadoop MapReduce) and **real-time streaming analytics** (e.g., Apache Kafka + Apache Flink/Spark Streaming).

4. **Advanced Analytical Capabilities**
   Enables **predictive analytics, machine learning, and AI** on massive datasets to uncover hidden patterns, trends, and insights that traditional systems can't handle.

5. **Data Lake Architecture**
   Centralized repositories like **data lakes** (e.g., on AWS S3 or Hadoop HDFS) allow storing raw data in its native format, supporting schema-on-read and flexible exploration.

6. **Interoperability and Tooling**
   Ecosystem supports multiple tools (Hive, Presto, Airflow, Superset, etc.) and languages (SQL, Python, Scala), ensuring flexibility for data engineers and analysts.

7. **Cost-effectiveness via Open Source & Cloud**
   Open-source tools and cloud-native services (e.g., AWS EMR, Azure Synapse, Google BigQuery) offer cost-effective, on-demand scaling for analytics workloads.

8. **Data Governance and Security Challenges**
   Managing **data quality, lineage, access control, and privacy** becomes more complex, requiring robust governance frameworks and tools like Apache Ranger, Atlas, or Lake Formation.

9. **Ecosystem Evolution and Tool Specialization**
   No single tool fits all needs. Understanding the **strengths and weaknesses** of each component in the ecosystem is essential for building efficient analytics pipelines.

10. **Value Extraction is Business-Driven**
    Technology alone isn't enough—analytics must be **aligned with business goals** to extract actionable insights that drive impact and ROI.

SQL*Plus

Thank you!

Stay Connected!