

# HADOOP

2CEIT702 - Big Data Analytics

**Book:**

**Big Data and Analytics** by Seema Acharya, Subhashini Chellappan, Paperback

**Reference book:**

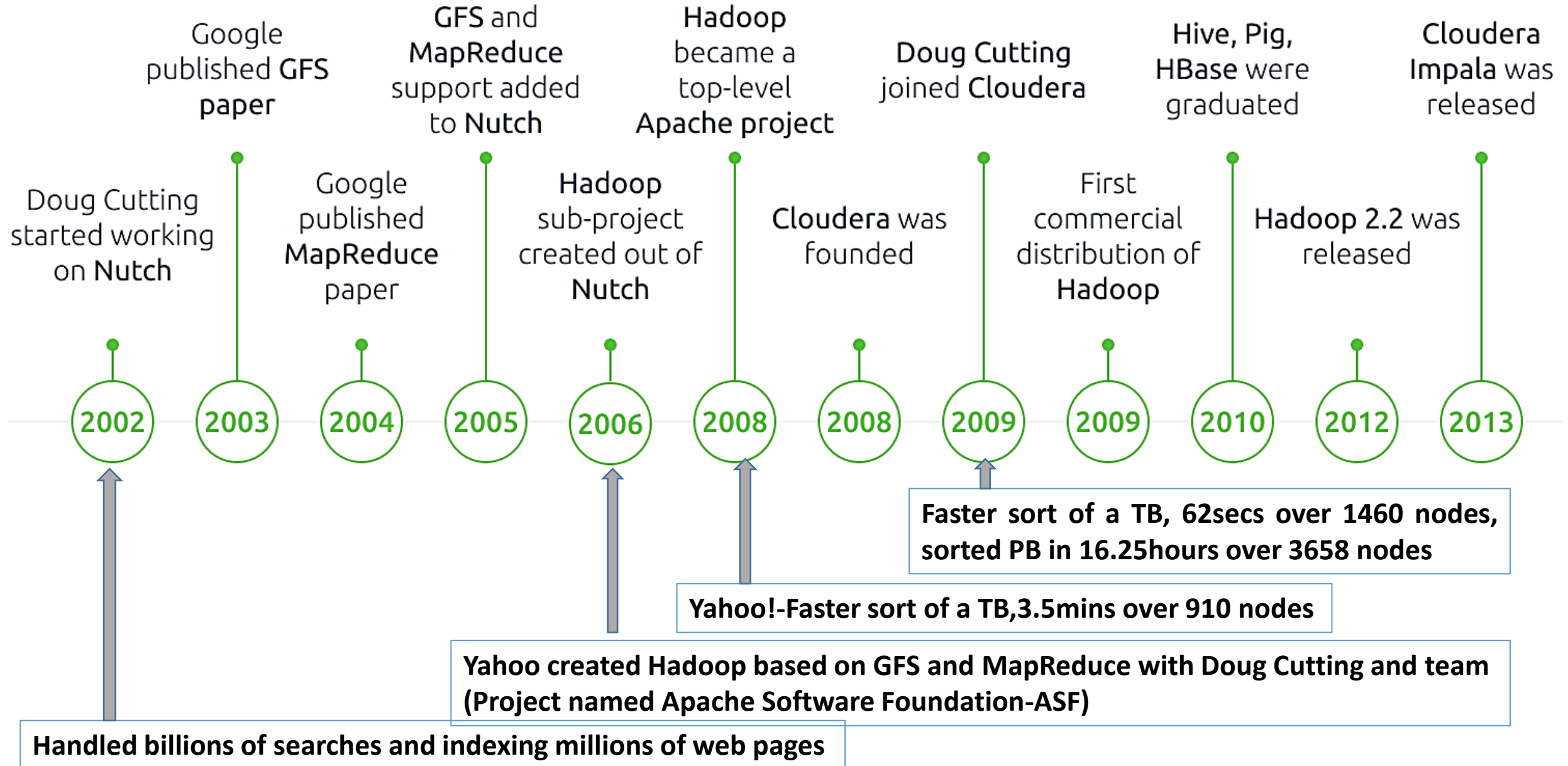
<https://www.uvpce.ac.in/content/syllabus-ce>

# Introduction

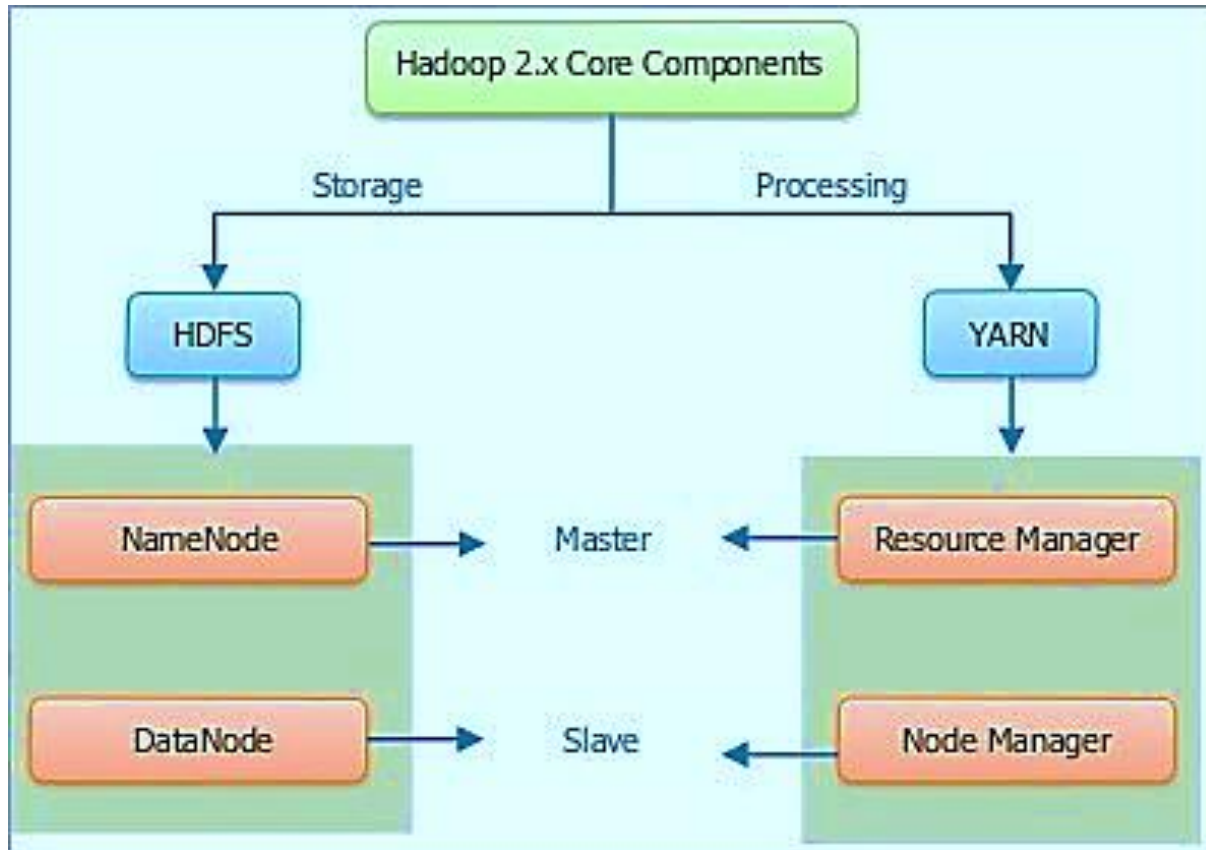
- **Big data brings with it two fundamental challenges:**
  - How to store and work with voluminous data sizes?
  - How to understand data and turn it into a competitive advantage?
- To process, analyse, and make sense of these different kinds of data, we need a system that scales and addresses the challenges.

(Answer: Hadoop)

# Evolution of Hadoop



# What is Hadoop?



- Hadoop is an open-source java based framework to store and process Big Data in a distributed environment.
- There are basically two components in Hadoop: HDFS and YARN
- Data is stored on inexpensive commodity servers that run as clusters.
- It is a distributed file system allows concurrent processing and fault tolerance.
- Hadoop MapReduce programming model is used for faster storage and retrieval of data from its nodes.
- It is based on the Google File System

# Why Hadoop?

- **Low Cost**

- The setup cost of Hadoop clusters is quite less as compared to other data storage and processing units.
- The reason is the low cost of the commodity hardware that is part of the cluster.

- **Computing Power**

- Hadoop is based on distributed computing model which process very large volumes of data fairly quickly.
- The more the number of computing nodes, the more the processing power at hand.

- **Scalability**

- Hadoop clusters come with limitless scalability.
- Unlike RDBMS that isn't as scalable, Hadoop clusters give you the power to expand the network capacity by adding more commodity hardware.

# Why Hadoop?

- **Storage flexibility & Processing**

- It is one of the primary benefits of Hadoop clusters.
- Process any type or form of data.
- Unlike other such clusters that may face a problem with different types of data, Hadoop clusters can be used to process structured, unstructured, as well as semi-structured data.
- This is the reason Hadoop is so popular when it comes to processing data from social media.

- **Inherent data protection**

- Hadoop protects data and executing applications against hardware failure.

# RDBMS versus HADOOP

## #1. Data Variety

Hadoop



Used for Structured, Semi Structured and Unstructured data.

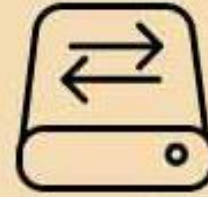
RDBMS



Mainly for Structured data.

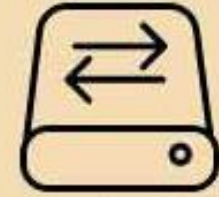
## #2. Data Storage

Hadoop



Use for large data set (Tbs and Pbs).

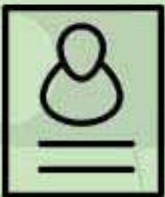
RDBMS



Average size data (Gbs).

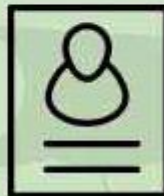
## #3. Querying

Hadoop



HQL (Hive Query Language).

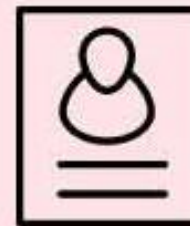
RDBMS



SQL Language.

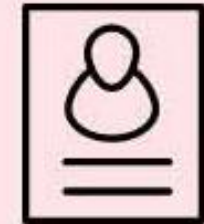
## #4. Schema

Hadoop



Required on read (dynamic schema).

RDBMS


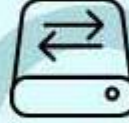


Required on write (static schema).



# RDBMS versus HADOOP

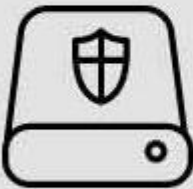

## #5. Speed

Hadoop	RDBMS
 <p>Both read and writes are fast.</p>	 <p>Reads are fast.</p>

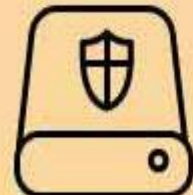
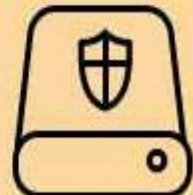
## #6. Cost

Hadoop	RDBMS
 <p>Free .</p>	 <p>License .</p>

## #7. Use Case

Hadoop	RDBMS
 <p>Analytics (Audio, video, logs etc), Data Discovery.</p>	 <p>OLTP (Online transaction processing).</p>

## #8. Data Objects

Hadoop	RDBMS
 <p>Works on Key/Value Pair.</p>	 <p>Works on Relational Tables.</p>



# RDBMS versus HADOOP

## #9. Throughput

Hadoop



High.

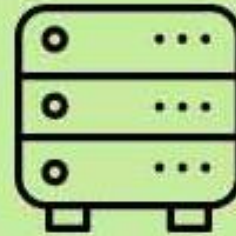
RDBMS



Low.

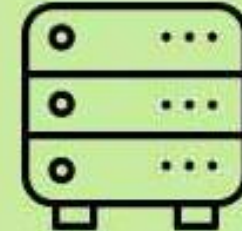
## #10. Scalability

Hadoop



Horizontal.

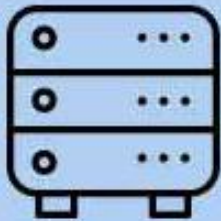
RDBMS



Vertical.

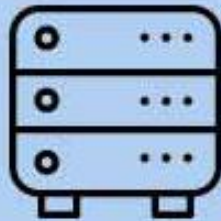
## #11. Hardware Profile

Hadoop



Commodity/Utility Hardware.

RDBMS



High End Servers.

## #12. Integrity

Hadoop



Low.

RDBMS



High (ACID).

# Distributed computing challenges

- There are several challenges with distributed computing, we will focus on two major challenges:

- **Hardware failure:**

Hadoop  
Answer:

Replication Factor

Number of data copies to be  
stored across the network

**Example**

RF=2 (Implies we have two replicas of the data)

- **Processing on gigantic data:**

Hadoop  
Answer:

MapReduce Programming

- **Key aspects of Hadoop:** Open source software, Framework, Distributed, Massive storage, and Faster Processing

## Hadoop Principle

A system to move computation, where the data is.

# Hadoop Ecosystem

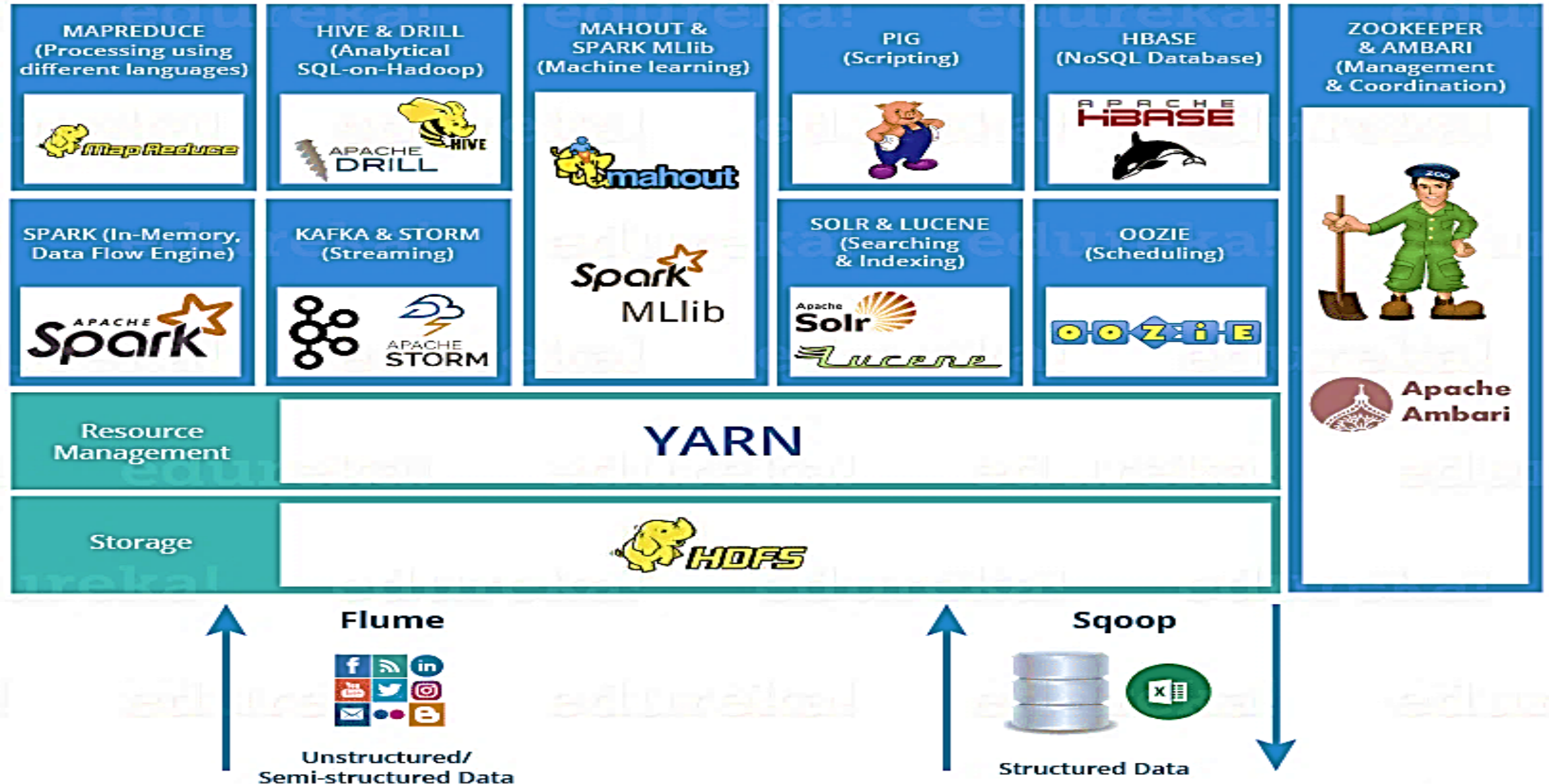
- The Ecosystem of Hadoop is not an actual programming language or service, it's an application or framework that solves big data-related problems.
- It can be described as a complete suite that includes various services like **ingesting data, storing, analysing** and **maintaining data** within it.

Hadoop Components	Use-case
HDFS	Hadoop Distributed File System
YARN	Yet Another Resource Negotiator
MapReduce	Data processing using programming
Spark	In-memory Data Processing

# Hadoop Ecosystem

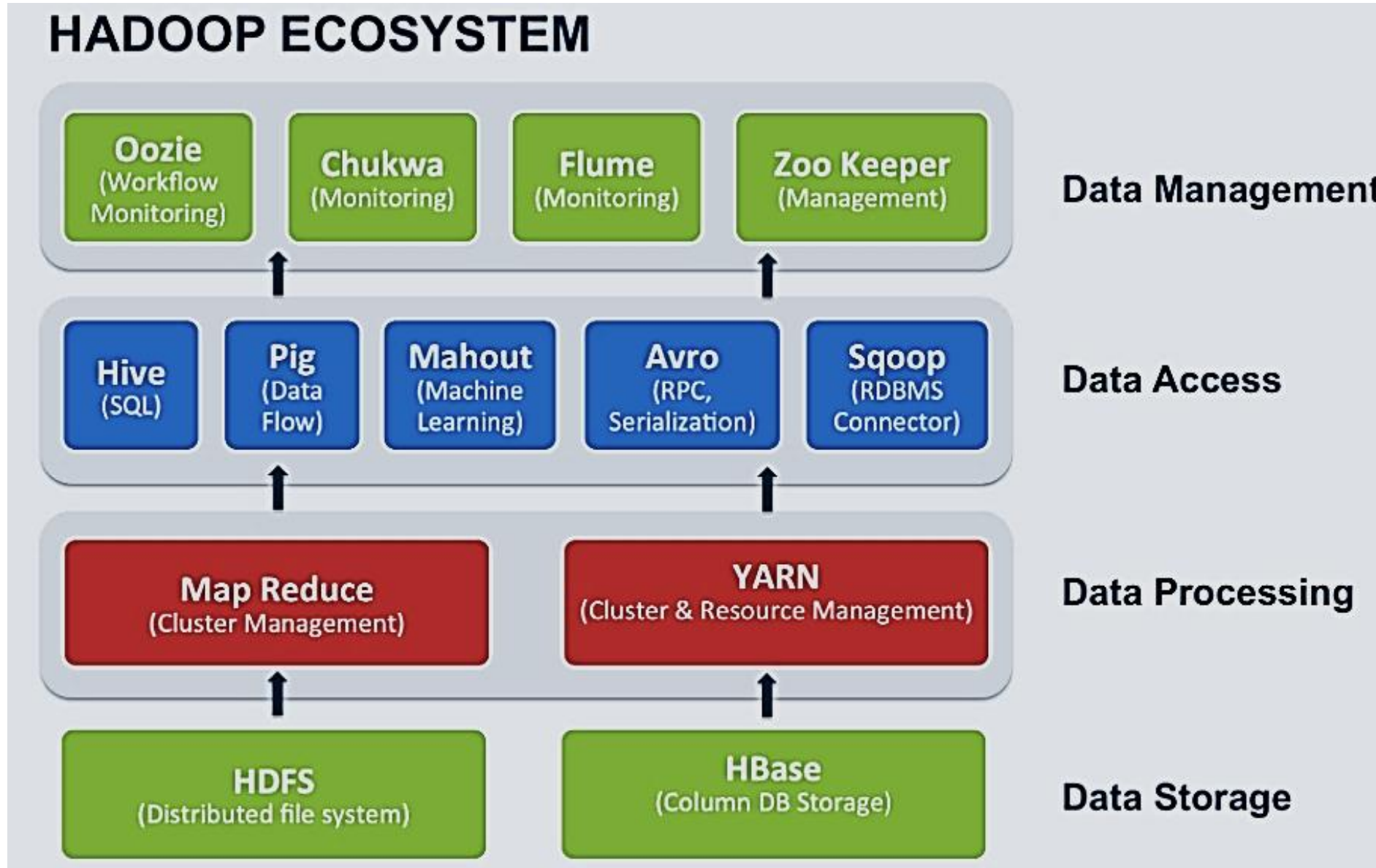
Hadoop Components	Use-case
PIG, HIVE	Data Processing Services using SQL like query
Hbase	NoSQL Database
Apache Drill	SQL on the top of Hadoop
ZooKeeper	Managing Cluster
Oozie	Job Scheduling
Flume, Sqoop	Data Ingesting Services
Solr And Lucene	Searching & Indexing
Ambari	Provision, Monitor and Maintain cluster
Spark Mlib	Machine Learning library

# Hadoop Ecosystem

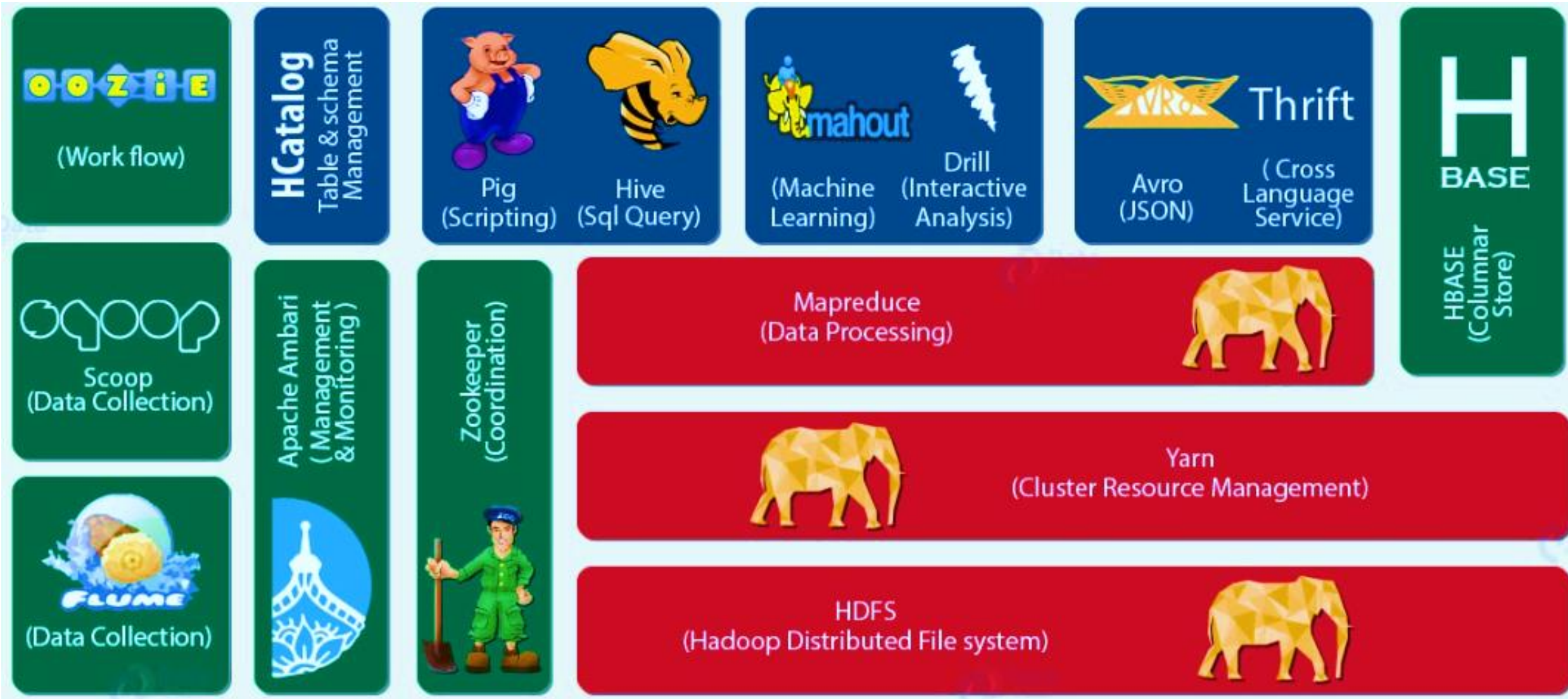




# Hadoop Ecosystem



# Hadoop Ecosystem





# Hadoop Core Components

There are three components of Hadoop:

- **Hadoop HDFS** - Hadoop Distributed File System (HDFS) is the storage unit.
- **Hadoop MapReduce** - Hadoop MapReduce is the processing unit.
- **Hadoop YARN** - Yet Another Resource Negotiator (YARN) is a resource management unit.

# HDFS (Hadoop Distributed File System)

- HDFS is a Java based distributed file system that allows you to store large data across multiple nodes in a Hadoop cluster.
- HDFS is the core component or you can say, the backbone of Hadoop Ecosystem.
- If you install Hadoop, you get HDFS as an underlying storage system for storing the data in the distributed environment.
- HDFS is the one, which makes it possible to store different types of large data sets (i.e. structured, unstructured and semi structured data).
- HDFS creates a level of abstraction over the resources, from where we can see the whole HDFS as a single unit.
- It helps us in storing our data across various nodes and maintaining the log file about the stored data (metadata).
- HDFS follows master-slave architecture.

## How HDFS resolves all the three major issues of traditional file systems?

- **Cost:**

- HDFS is open-source software so that it can be used with zero licensing and support costs.

- **Speed:**

- Large Hadoop clusters can read or write more than a terabyte of data per second.
- HDFS can easily deliver more than two gigabytes of data per second, per computer to MapReduce, which is a data processing framework of Hadoop.

- **Reliability:**

- HDFS copies the data multiple times and distributes the copies to individual nodes.
- Regular file system, like a Linux file system, data block size is small (512 bytes). In HDFS, each block size is 64MB default but industry use 128MB block size.
- A regular file system provides access to large data but may suffer from disk input/output problems mainly due to multiple seek operations. On the other hand, HDFS can read large quantities of data sequentially after a single seek operation. This makes HDFS unique since all of these operations are performed in a distributed mode.

# Characteristics of HDFS

- **HDFS has high fault-tolerance**

- HDFS may consist of thousands of server machines. Each machine stores a part of the file system data. HDFS detects faults that can occur on any of the machines and recovers it quickly and automatically.

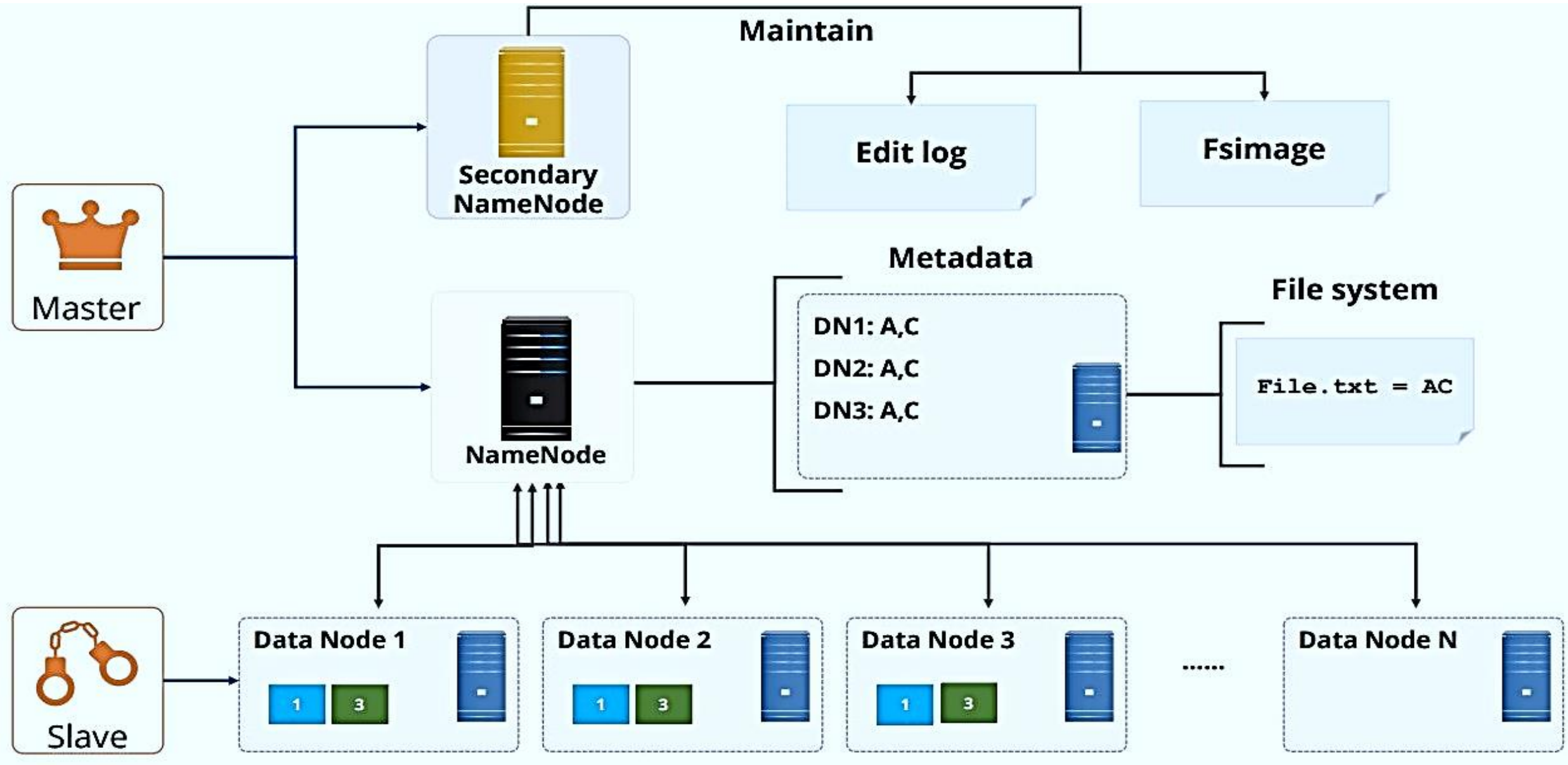
- **HDFS has high throughput**

- HDFS is designed to store and scan millions of rows of data and to count or add some subsets of the data. The time required in this process is dependent on the complexities involved.
- It has been designed to support large datasets in batch-style jobs. However, the emphasis is on high throughput of data access rather than low latency.

- **HDFS is economical**

- HDFS is designed in such a way that it can be built on commodity hardware and heterogeneous platforms, which is low-priced and easily available.

# HDFS



# HDFS

- HDFS has two core components, i.e. **NameNode** and **DataNode**
- **NameNode**
  - It is the main/master node and it doesn't store the actual data.
  - It contains metadata, just like a log file or you can say as a table of content like which data block is stored in which data node, where are the replications of the data block kept etc.
  - The actual data is not stored in NameNode Therefore, NameNode requires less storage and high computational resources.
- **DataNode**
  - The actual data is stored in Data Nodes.
  - It requires more storage resources. we actually replicate the data blocks present in Data Nodes, and the default replication factor is 3.
  - These DataNodes are commodity hardware (like your laptops and desktops) in the distributed environment.

# HDFS

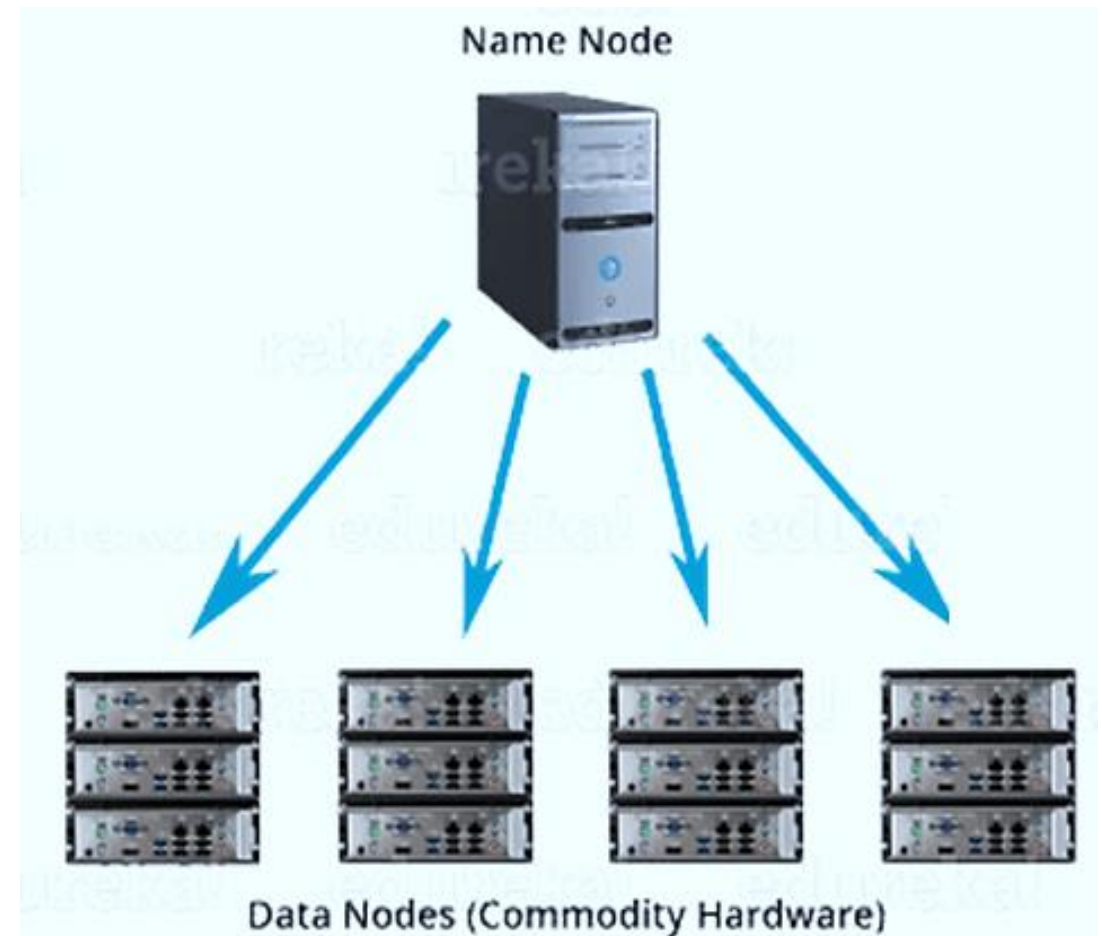
- You always communicate to the NameNode while writing the data. Then, it internally sends a request to the client to store and replicate data on various DataNodes.

## Advantages of HDFS

1. Distributed Storage
2. Distributed & Parallel Computation
3. Horizontal Scalability

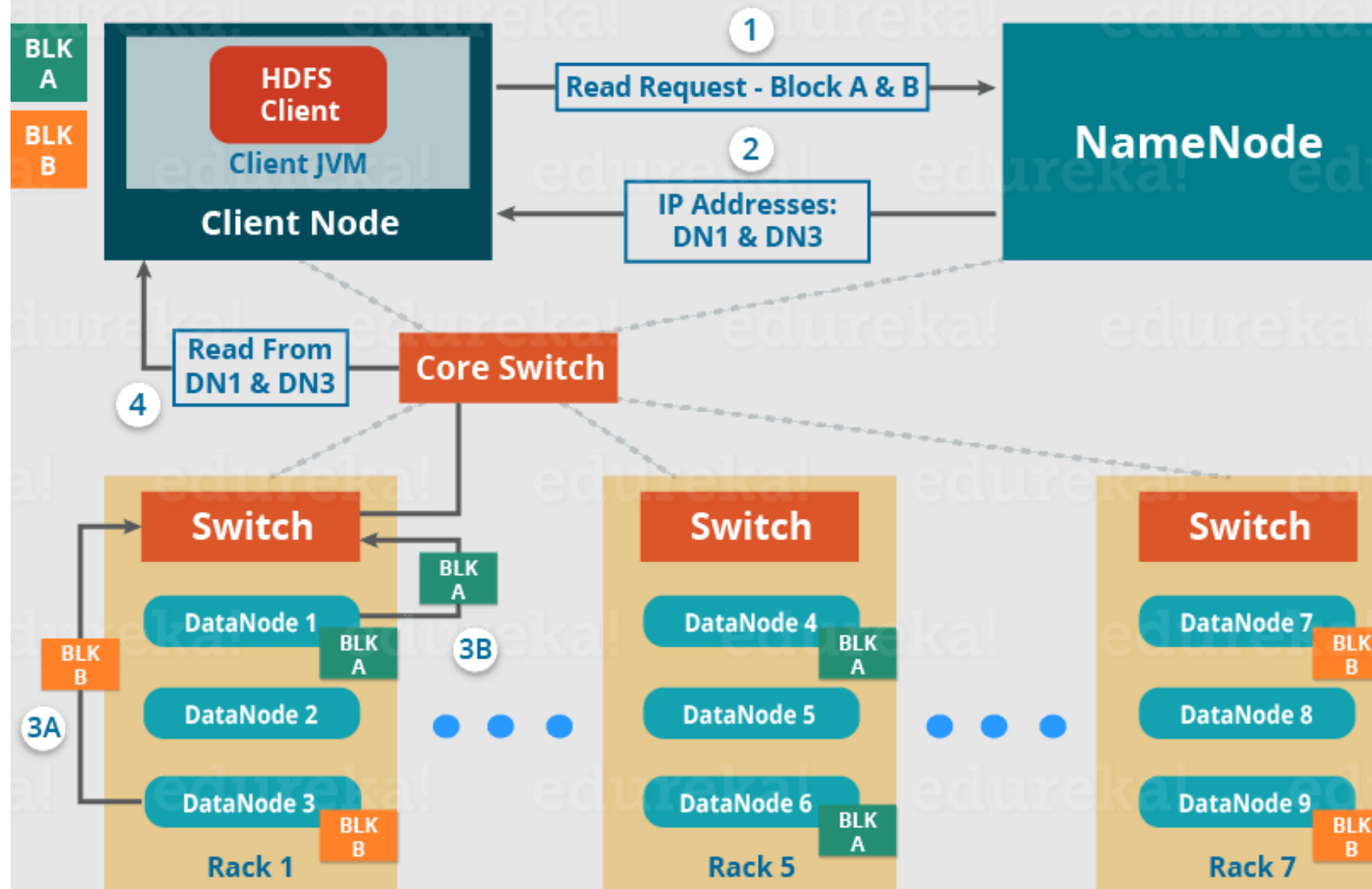
## Questions:

1. How is the data distributed in HDFS?
2. Who keeps the track of the distributed files?

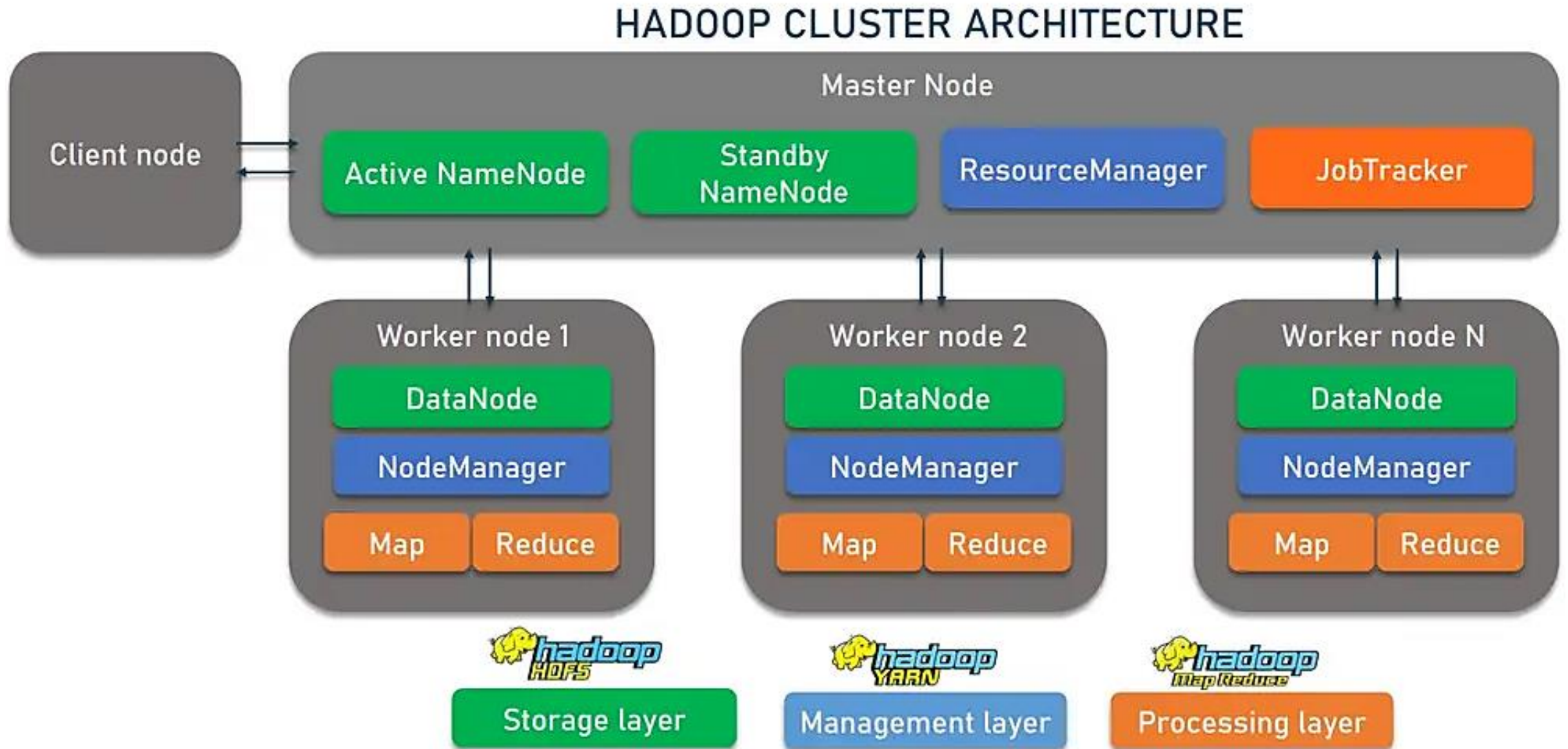




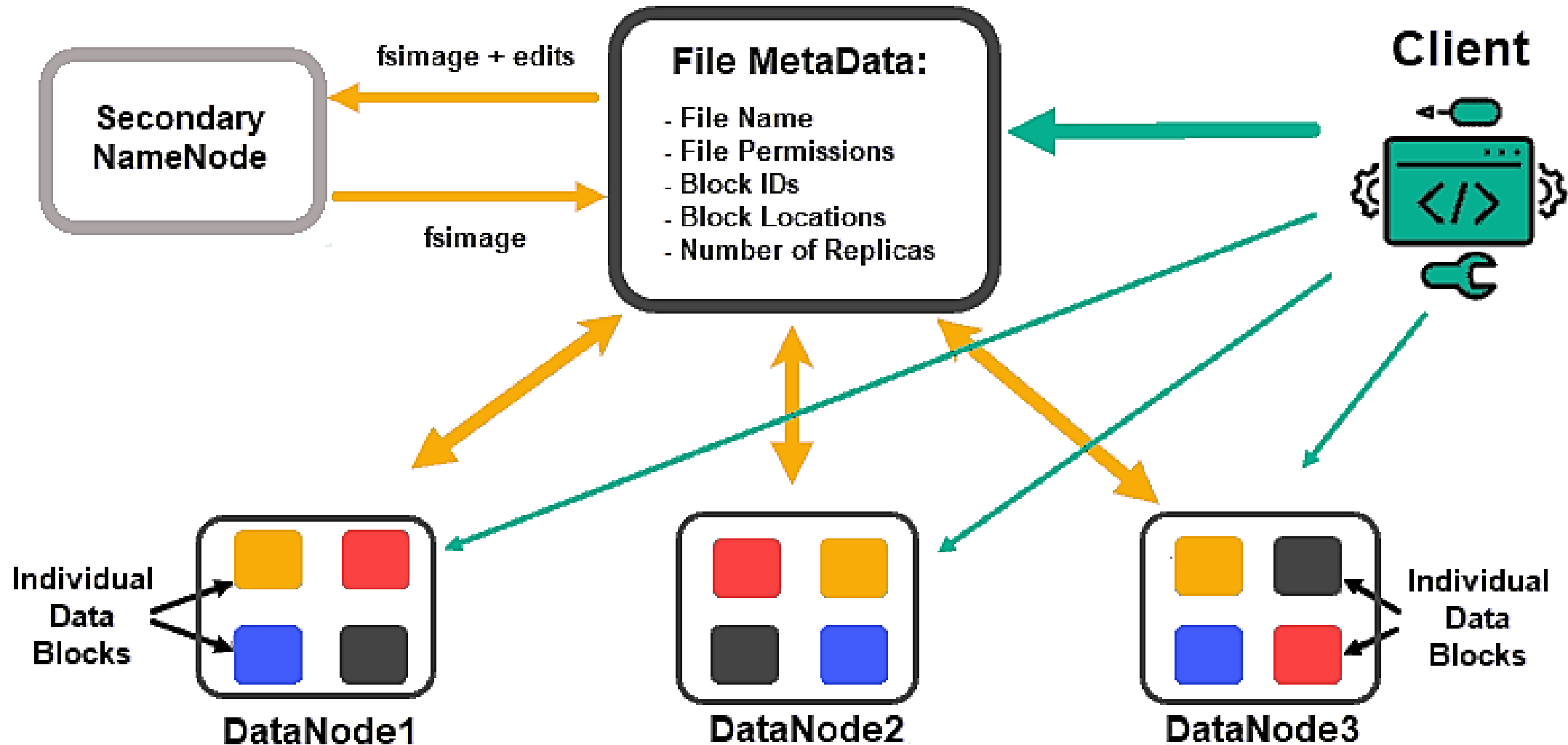
## HDFS - Read Architecture



# Hadoop Architecture



# NameNode



# Disadvantages of Hadoop

- Problems with Small Files.
- Processing Speed
- Resource Management
- Limited to Batch Processing