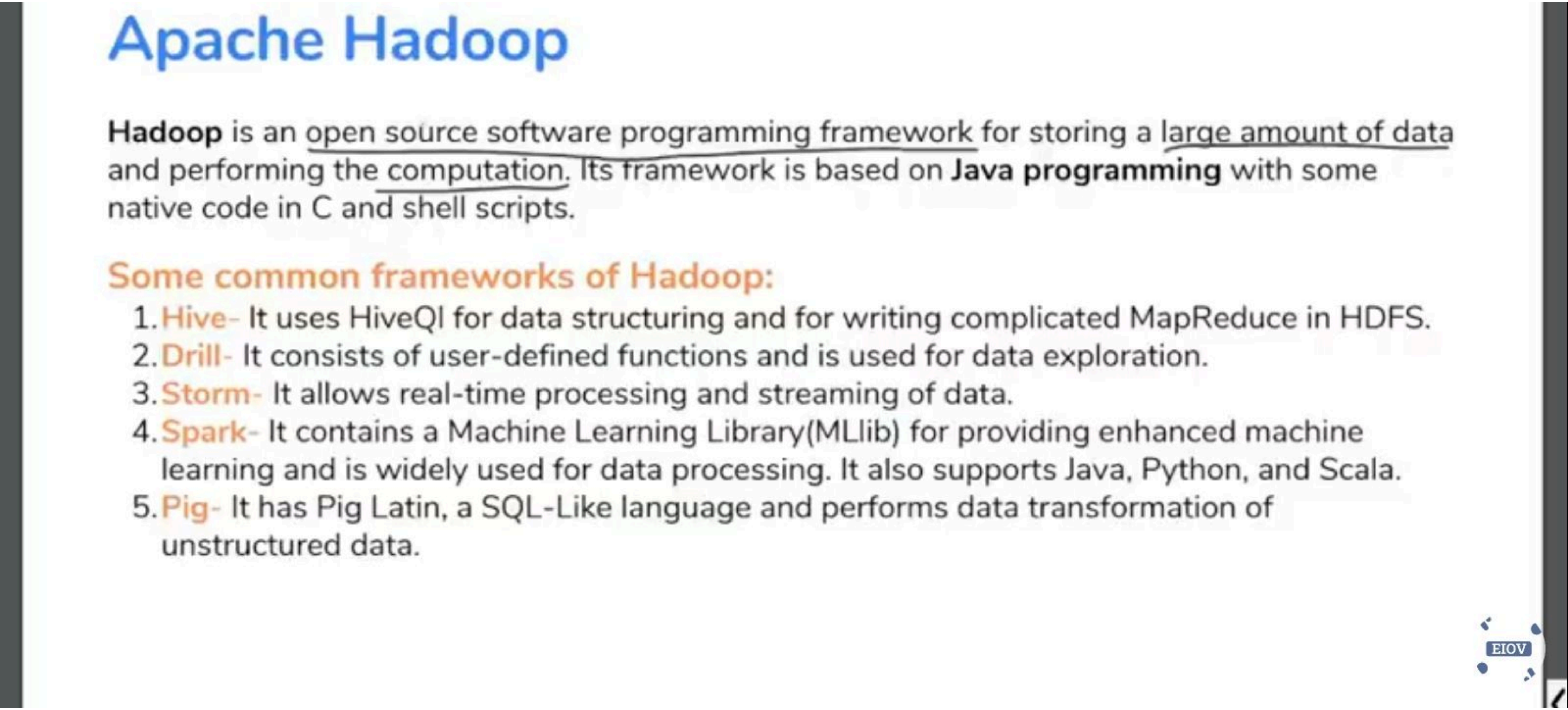
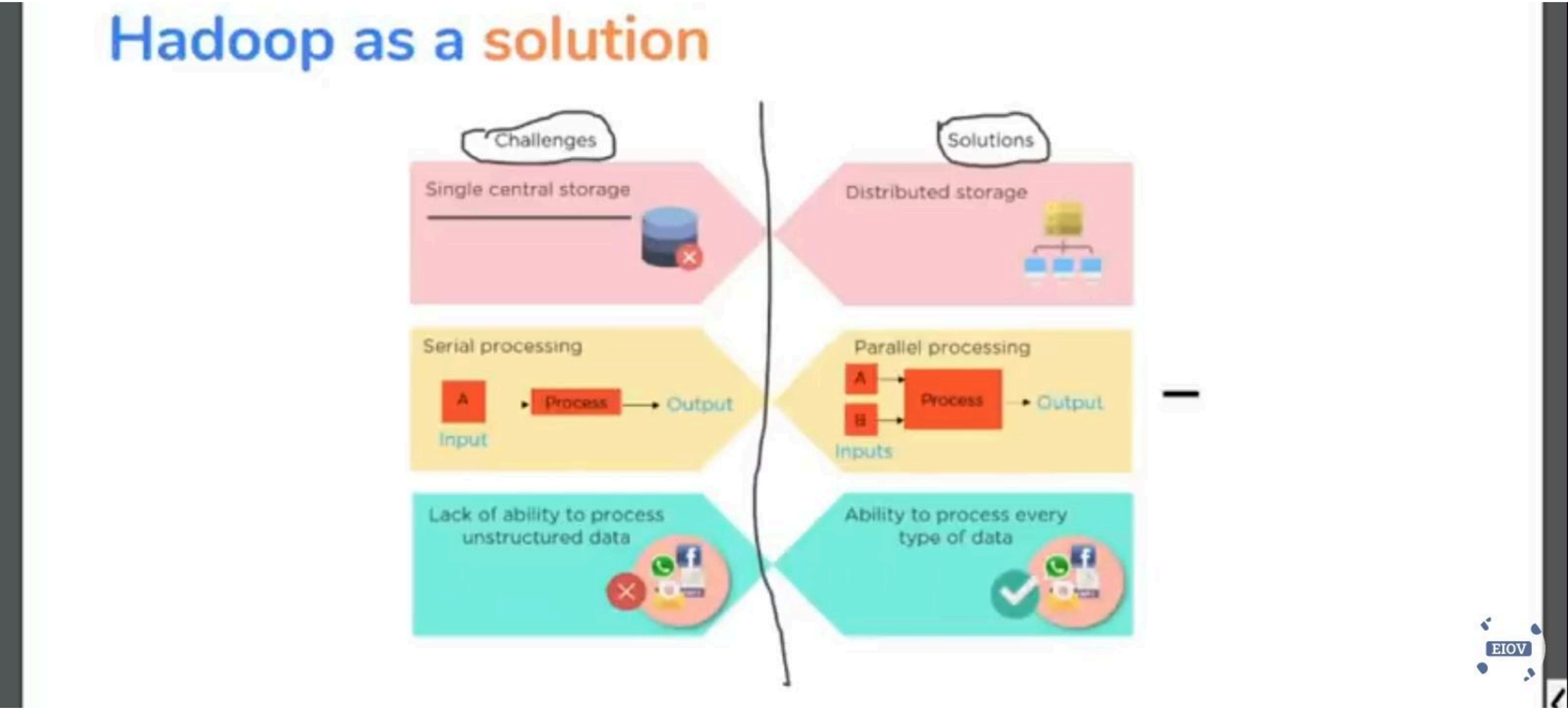


# Apache Hadoop

**Hadoop** is an open source software programming framework for storing a large amount of data and performing the computation. Its framework is based on **Java programming** with some native code in C and shell scripts.

## Some common frameworks of Hadoop:

1. Hive- It uses HiveQl for data structuring and for writing complicated MapReduce in HDFS.
2. Drill- It consists of user-defined functions and is used for data exploration.
3. Storm- It allows real-time processing and streaming of data.
4. Spark- It contains a Machine Learning Library(MLlib) for providing enhanced machine learning and is widely used for data processing. It also supports Java, Python, and Scala.
5. Pig- It has Pig Latin, a SQL-Like language and performs data transformation of unstructured data.



# Hadoop as a solution

Challenges

Single central storage

Serial processing

A

Process

Output

Input

Lack of ability to process unstructured data

Solutions

Distributed storage

Parallel processing

A

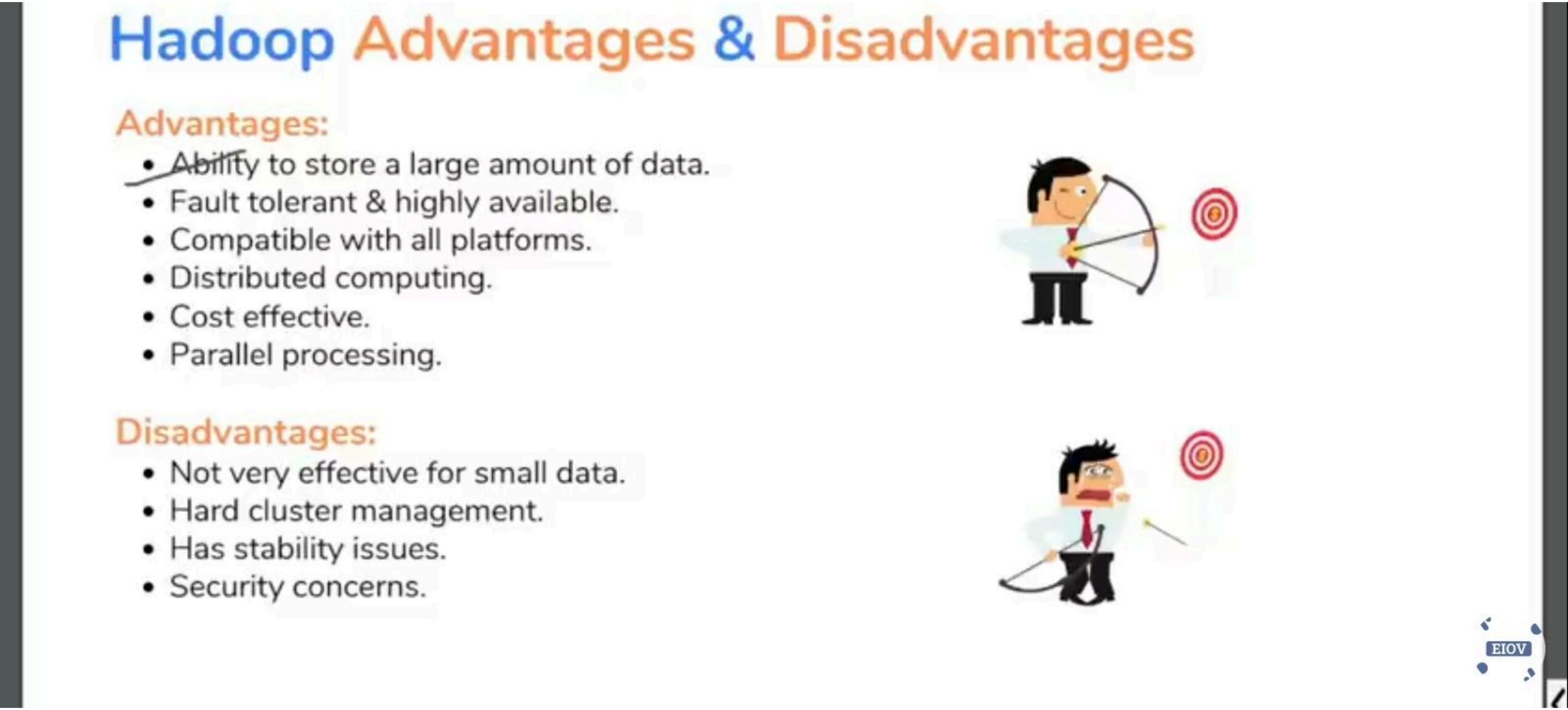
B

Process

Output

Inputs

Ability to process every type of data



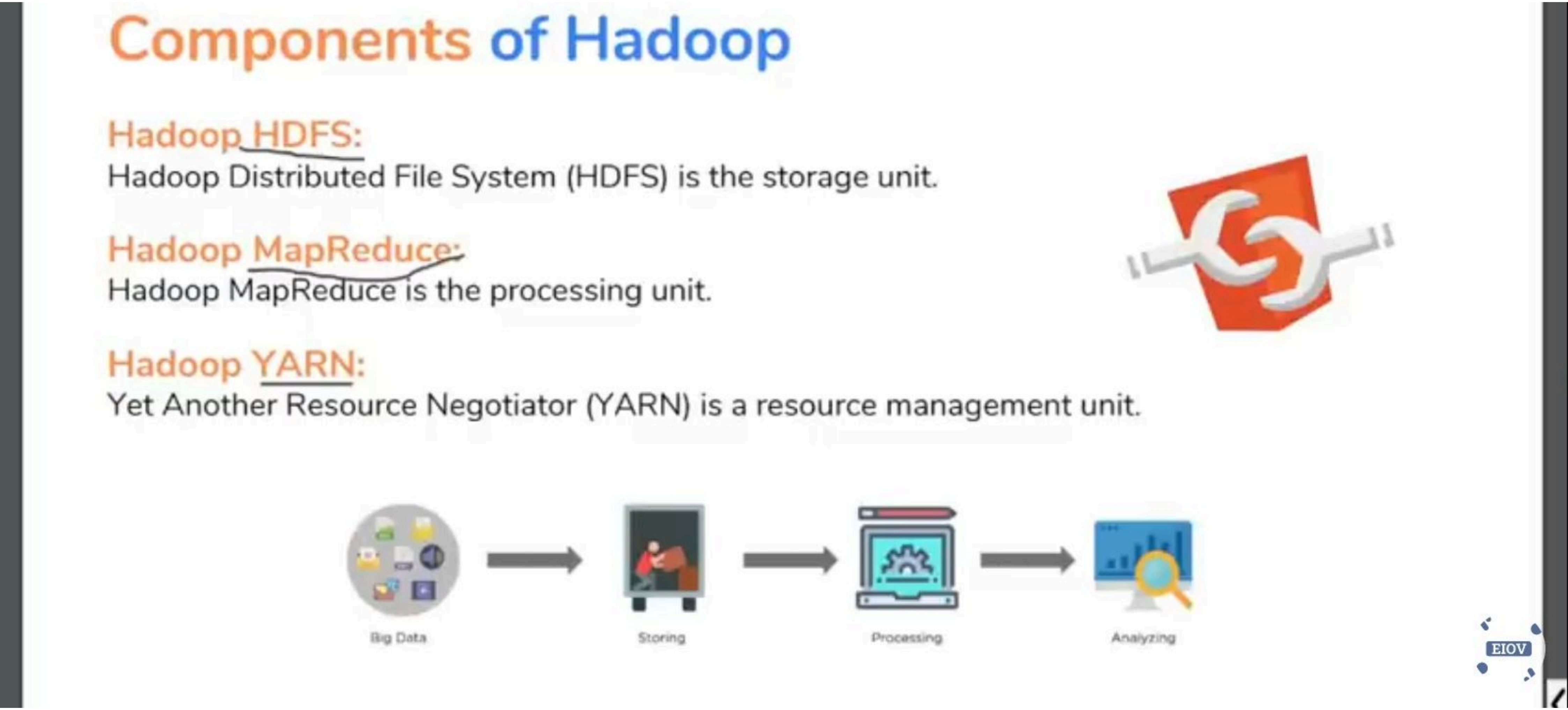
# Hadoop Advantages & Disadvantages

## Advantages:

- Ability to store a large amount of data.
- Fault tolerant & highly available.
- Compatible with all platforms.
- Distributed computing.
- Cost effective.
- Parallel processing.

## Disadvantages:

- Not very effective for small data.
- Hard cluster management.
- Has stability issues.
- Security concerns.



# Components of Hadoop

## Hadoop HDFS:

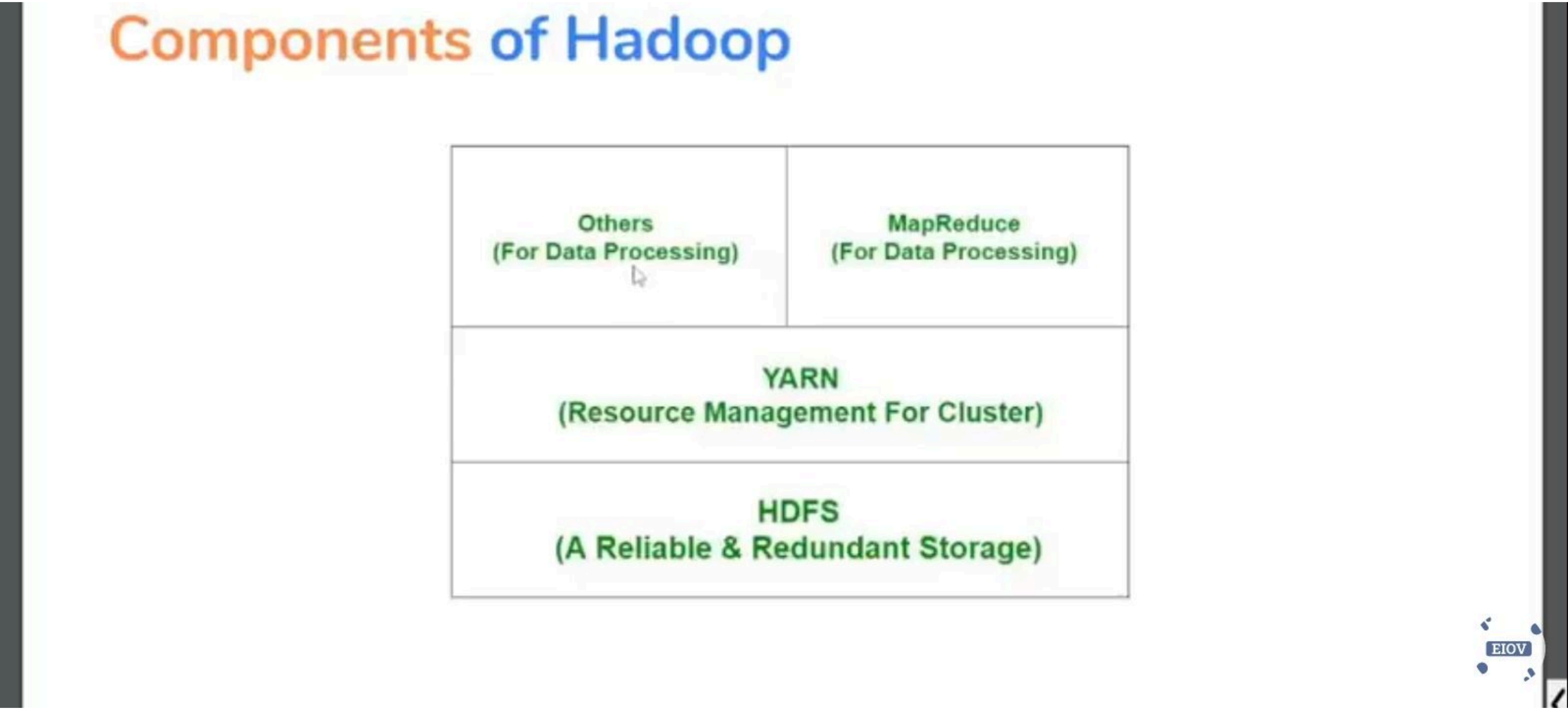
Hadoop Distributed File System (HDFS) is the storage unit.

## Hadoop MapReduce:

Hadoop MapReduce is the processing unit.

## Hadoop YARN:

Yet Another Resource Negotiator (YARN) is a resource management unit.



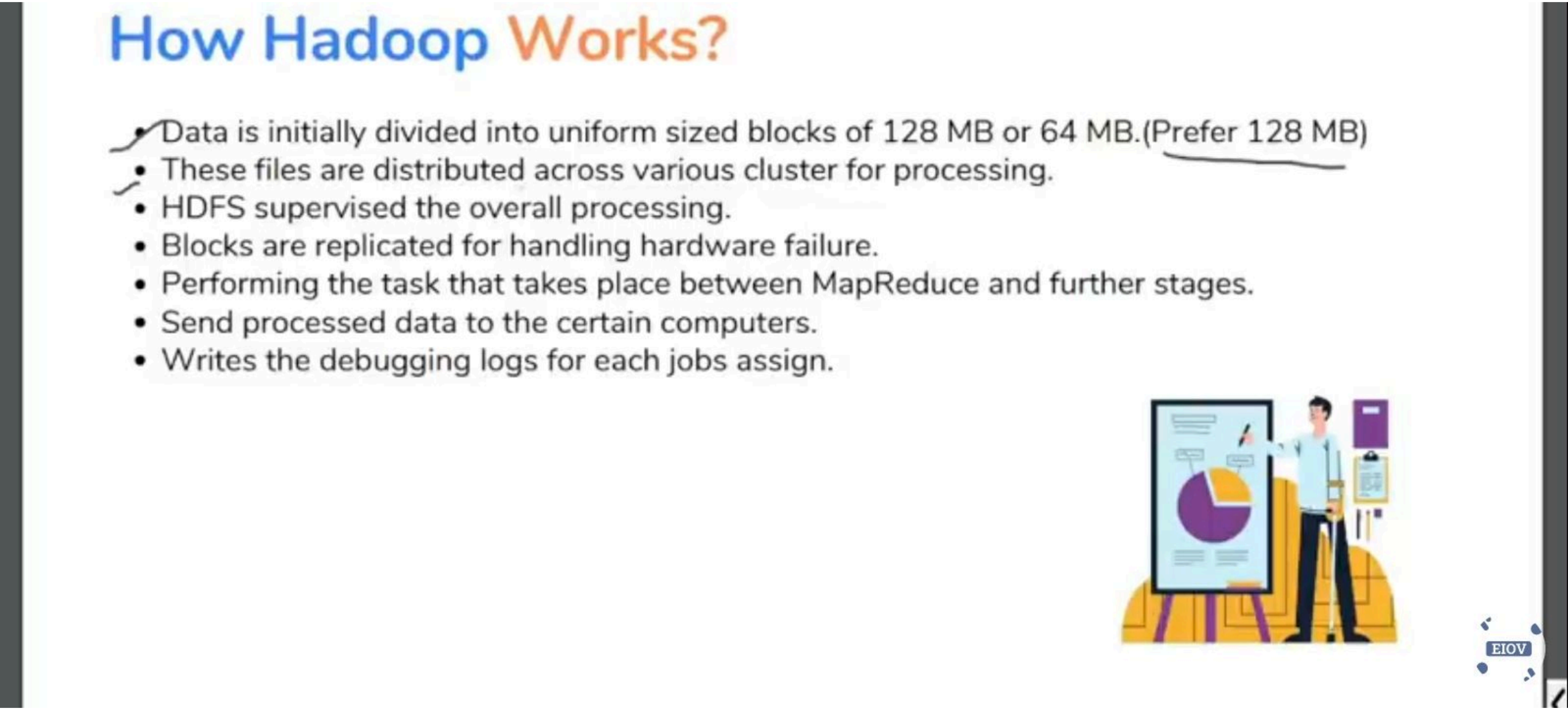
# Components of Hadoop

Others
(For Data Processing)

MapReduce
(For Data Processing)

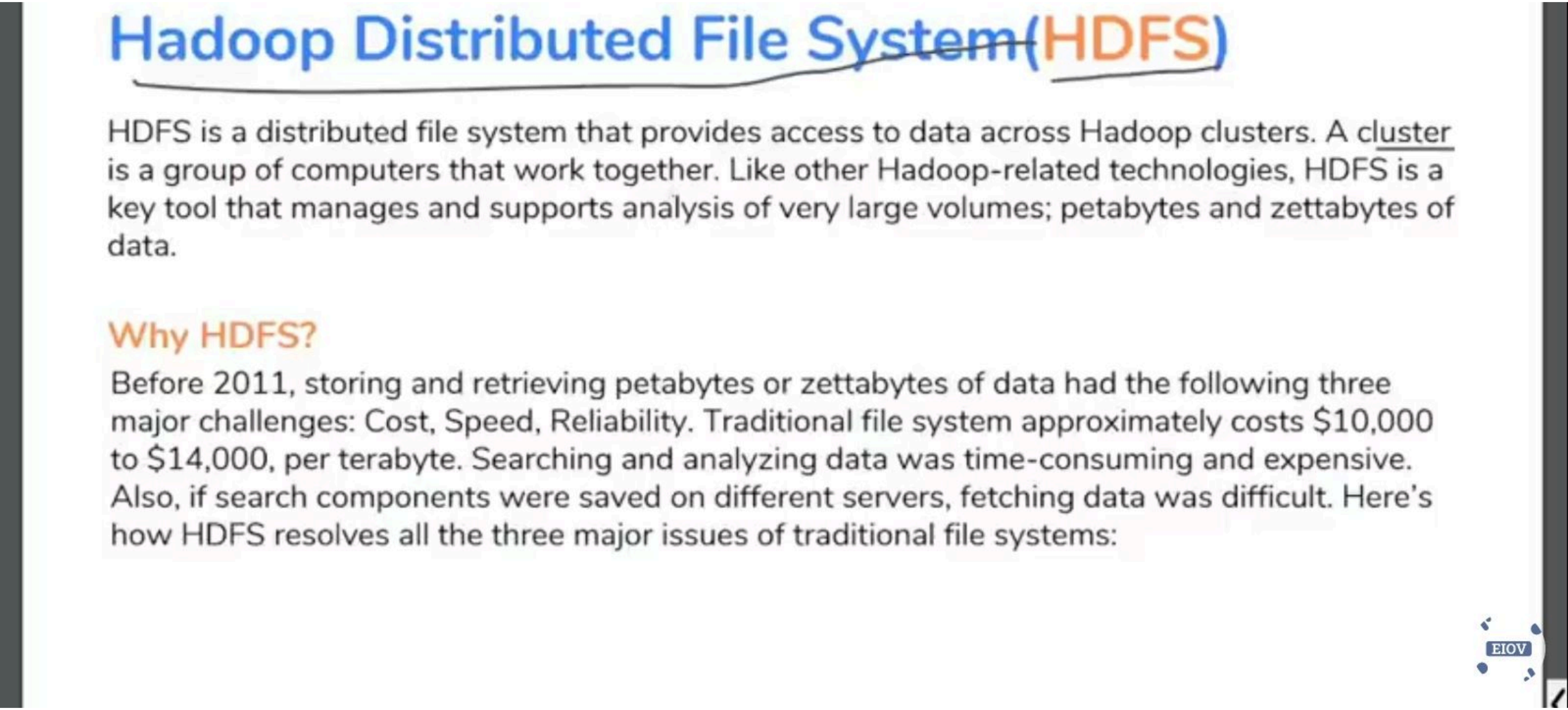
YARN
(Resource Management For Cluster)

HDFS
(A Reliable & Redundant Storage)



# How Hadoop Works?

- Data is initially divided into uniform sized blocks of 128 MB or 64 MB.(Prefer 128 MB)
- These files are distributed across various cluster for processing.
- HDFS supervised the overall processing.
- Blocks are replicated for handling hardware failure.
- Performing the task that takes place between MapReduce and further stages.
- Send processed data to the certain computers.
- Writes the debugging logs for each jobs assign.

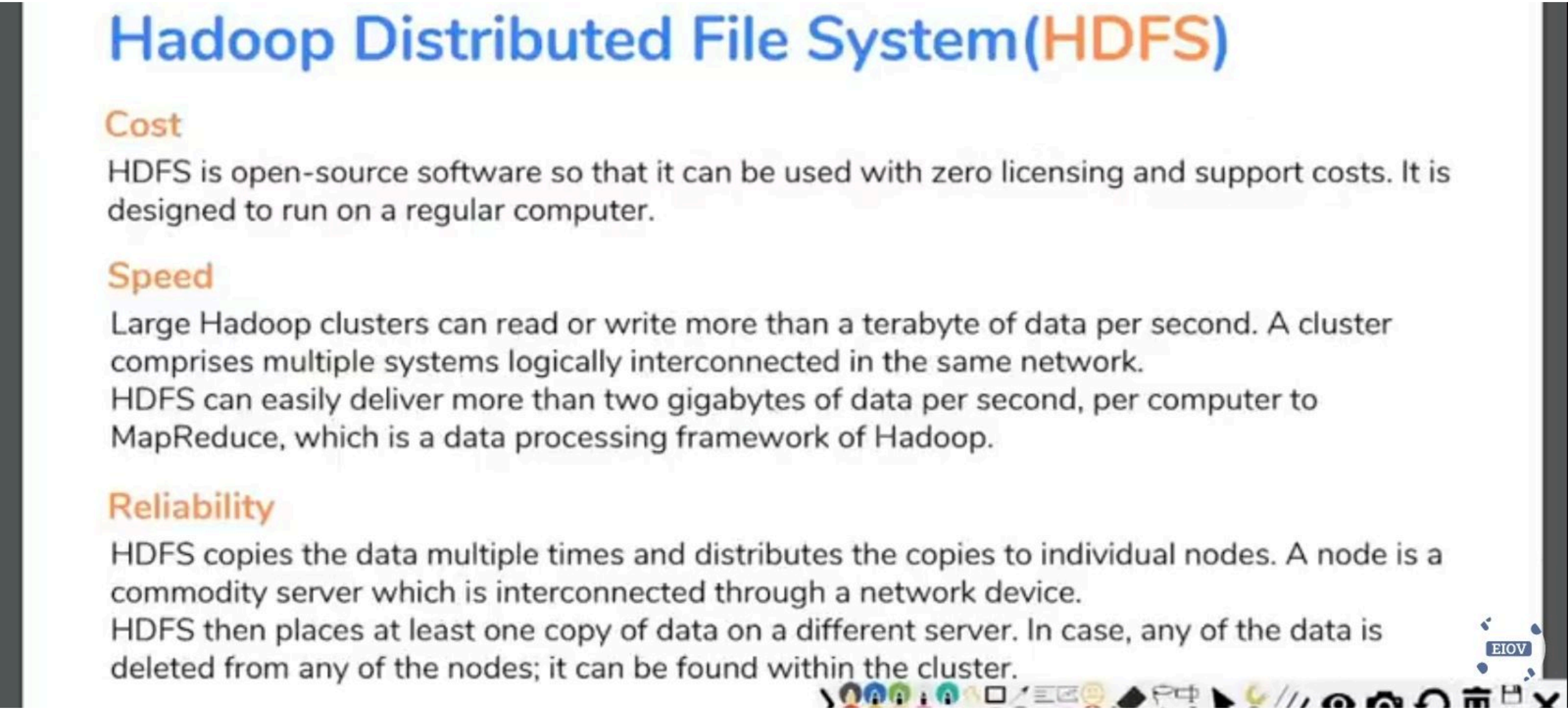


# Hadoop Distributed File System(HDFS)

HDFS is a distributed file system that provides access to data across Hadoop clusters. A cluster is a group of computers that work together. Like other Hadoop-related technologies, HDFS is a key tool that manages and supports analysis of very large volumes; petabytes and zettabytes of data.

## Why HDFS?

Before 2011, storing and retrieving petabytes or zettabytes of data had the following three major challenges: Cost, Speed, Reliability. Traditional file system approximately costs $10,000 to $14,000, per terabyte. Searching and analyzing data was time-consuming and expensive. Also, if search components were saved on different servers, fetching data was difficult. Here's how HDFS resolves all the three major issues of traditional file systems:



# Hadoop Distributed File System(HDFS)

## Cost

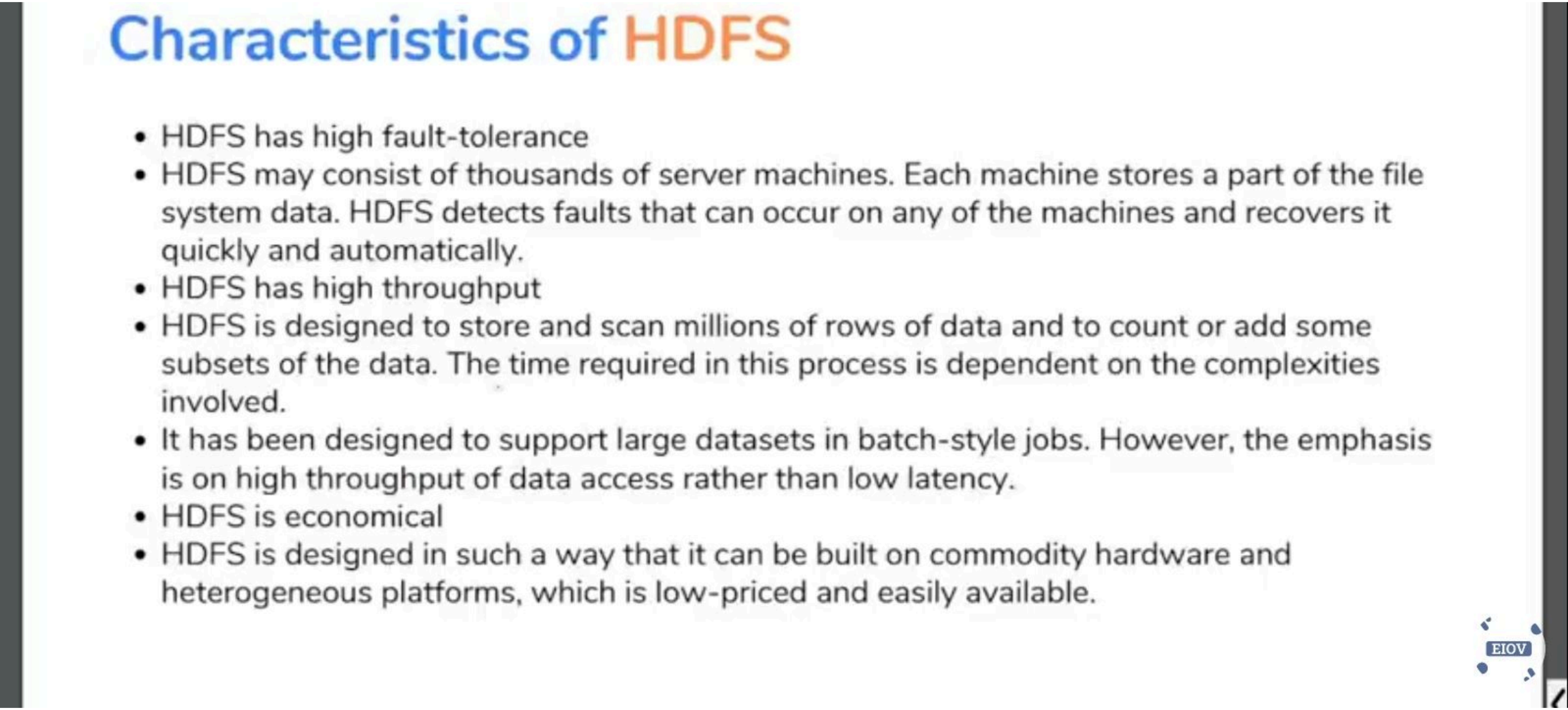
HDFS is open-source software so that it can be used with zero licensing and support costs. It is designed to run on a regular computer.

## Speed

Large Hadoop clusters can read or write more than a terabyte of data per second. A cluster comprises multiple systems logically interconnected in the same network.
HDFS can easily deliver more than two gigabytes of data per second, per computer to MapReduce, which is a data processing framework of Hadoop.

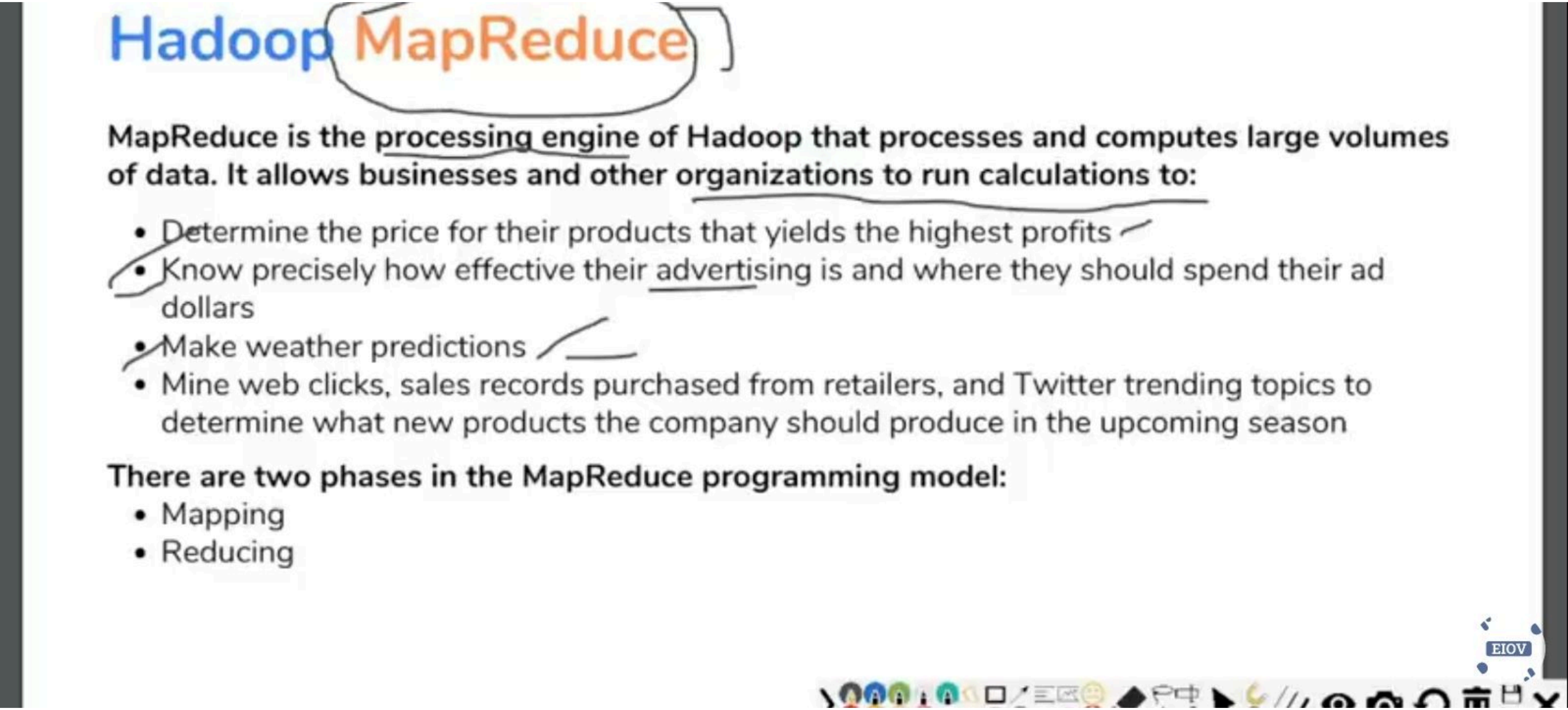
## Reliability

HDFS copies the data multiple times and distributes the copies to individual nodes. A node is a commodity server which is interconnected through a network device.
HDFS then places at least one copy of data on a different server. In case, any of the data is deleted from any of the nodes; it can be found within the cluster.



# 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.



# Hadoop MapReduce

**MapReduce is the processing engine of Hadoop that processes and computes large volumes of data. It allows businesses and other organizations to run calculations to:**

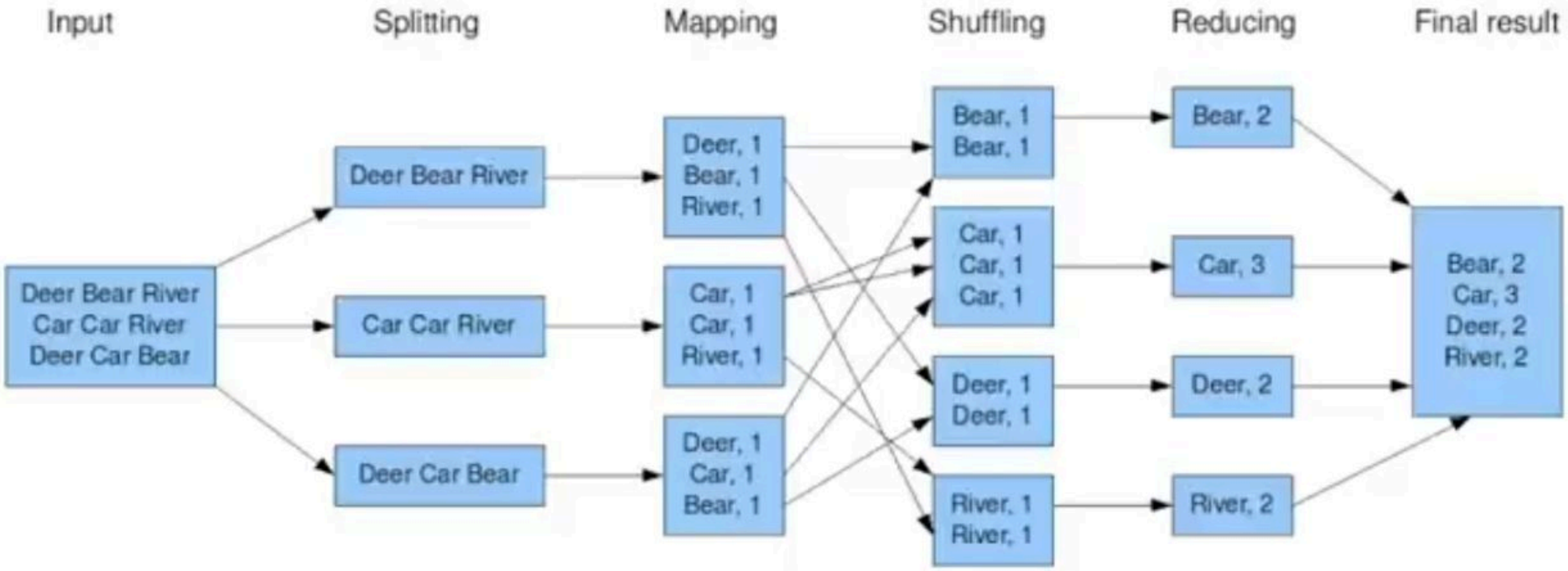
- Determine the price for their products that yields the highest profits
- Know precisely how effective their advertising is and where they should spend their ad dollars
- Make weather predictions
- Mine web clicks, sales records purchased from retailers, and Twitter trending topics to determine what new products the company should produce in the upcoming season

**There are two phases in the MapReduce programming model:**

- Mapping
- Reducing

# Hadoop MapReduce

The overall MapReduce word count process



Input

Deer Bear River
Car Car River
Deer Car Bear

Splitting

Deer Bear River

Car Car River

Deer Car Bear

Mapping

Deer, 1
Bear, 1
River, 1

Car, 1
Car, 1
River, 1

Deer, 1
Car, 1
Bear, 1

Shuffling

Bear, 1
Bear, 1

Car, 1
Car, 1
Car, 1

Deer, 1
Deer, 1

River, 1
River, 1

Reducing

Bear, 2

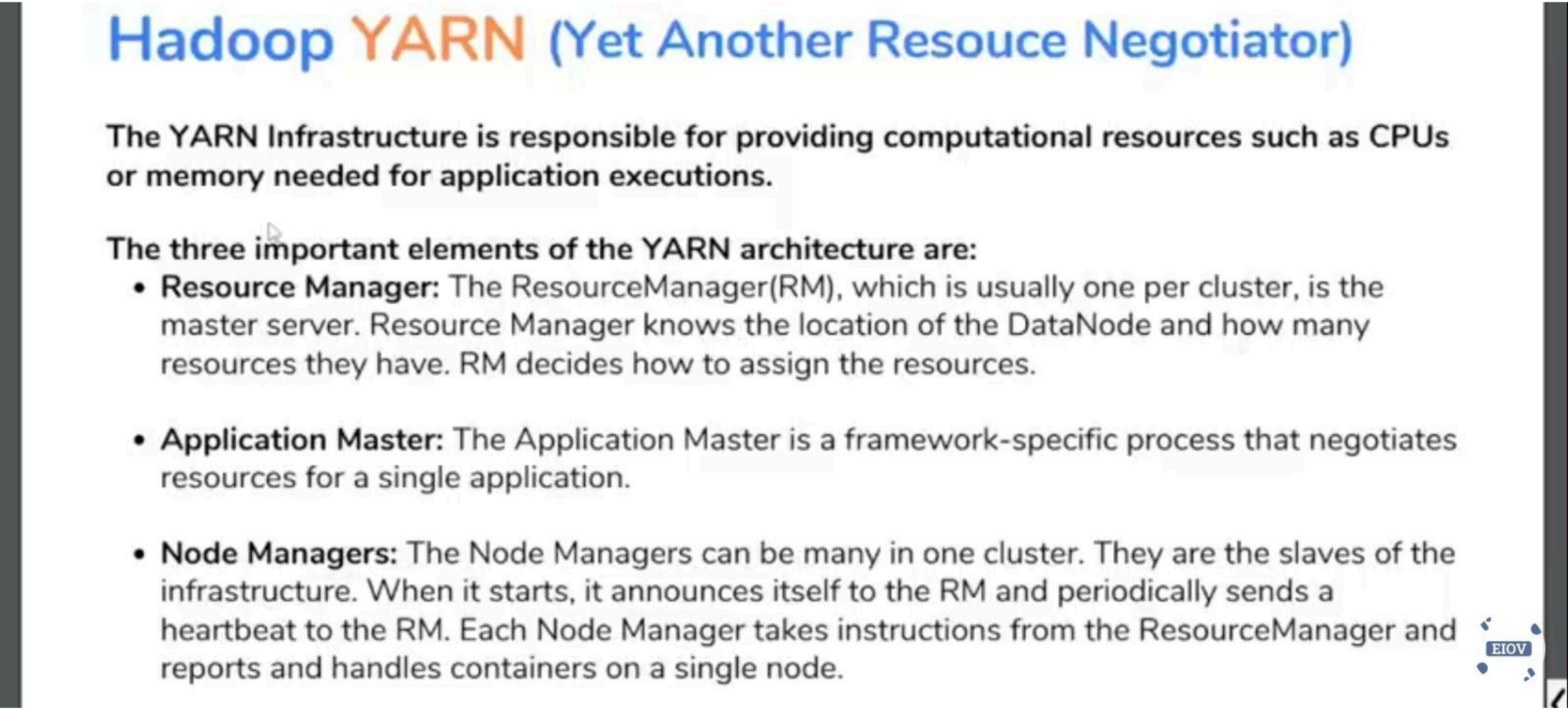
Car, 3

Deer, 2

River, 2

Final result

Bear, 2
Car, 3
Deer, 2
River, 2



# Hadoop YARN (Yet Another Resouce Negotiator)

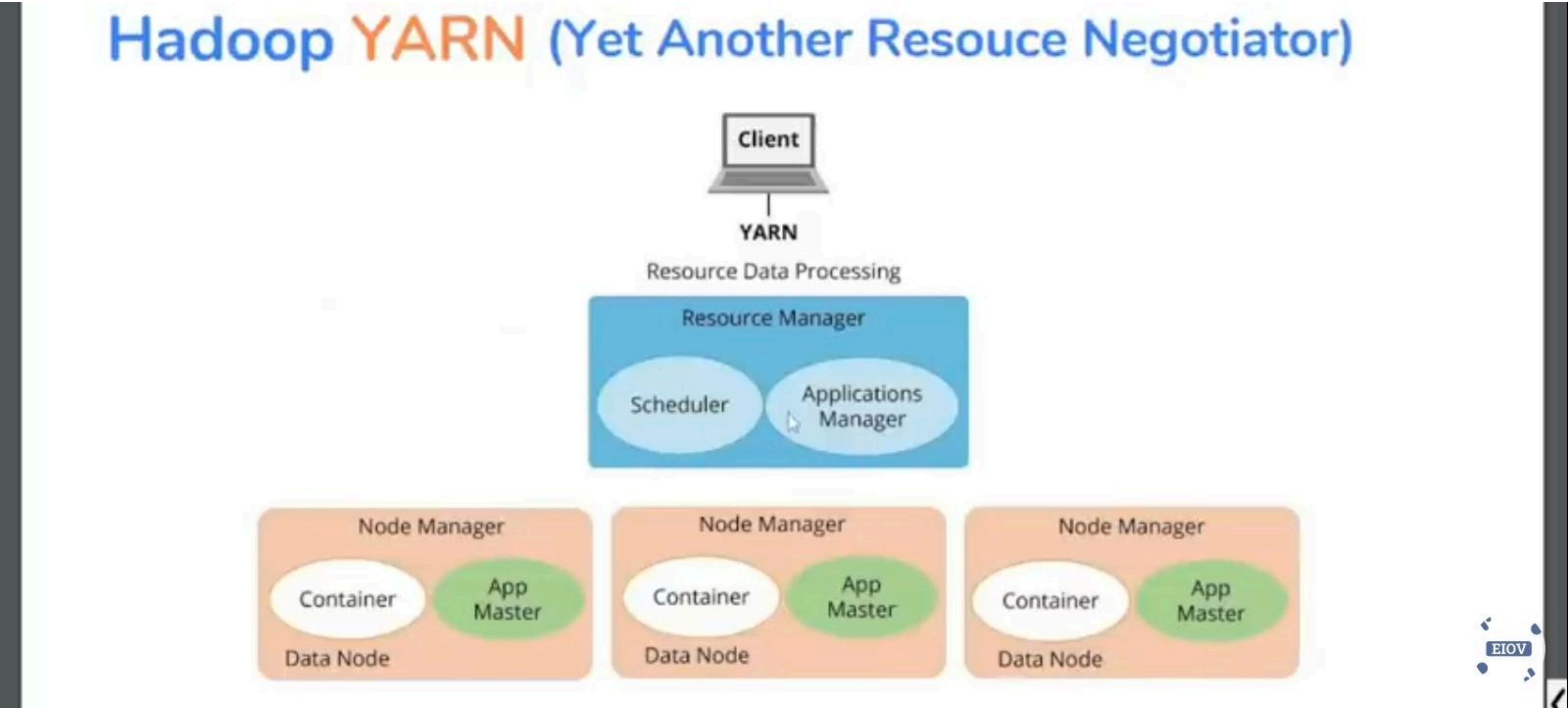
**The YARN Infrastructure is responsible for providing computational resources such as CPUs or memory needed for application executions.**

**The three important elements of the YARN architecture are:**

- **Resource Manager:** The ResourceManager(RM), which is usually one per cluster, is the master server. Resource Manager knows the location of the DataNode and how many resources they have. RM decides how to assign the resources.

- **Application Master:** The Application Master is a framework-specific process that negotiates resources for a single application.

- **Node Managers:** The Node Managers can be many in one cluster. They are the slaves of the infrastructure. When it starts, it announces itself to the RM and periodically sends a heartbeat to the RM. Each Node Manager takes instructions from the ResourceManager and reports and handles containers on a single node.



# Hadoop YARN (Yet Another Resouce Negotiator)

Client

YARN

Resource Data Processing

Resource Manager

Scheduler

Applications Manager

Node Manager

Container

App Master

Data Node

Node Manager

Container

App Master

Data Node

Node Manager

Container

App Master

Data Node

## ★ Data format in Hadoop

Hadoop uses 4 different file formats :-
1. Text files
2. Sequence file
3. Avro Data files
4. Parquet file format

Text files :- It is the most basic and a human readable file. It can be read or written in any programming language and is mostly delimited by comma or tab.

Sequence file :- It can be used to store an image in binary format. They store key value pairs in a binary container format and are more efficient than a text file. These are not human readable.

Avro Data files :- It has efficient storage due to optimized binary encoding. It is widely supported both inside and outside the Hadoop ecosystem.

Parquet file format :- It is a columnar format developed by Cloudera and Twitter. It reduce the storage space and increase performance. It is most efficient for adding multiple records at a time.

# Hadoop Ecosystem

1. HDFS ✓
2. YARN ✓
3. Map Reduce ✓
4. Spark ✓
5. Pig, Hive ✓
6. HBase ✓

(In syllabus)

7. Manaout
8. Zookeeper
9. Oozie

7. Manaout allows Machine learnability to a system or application. It provides various libraries or functionalities which are the concept of ML. It allows invoking algorithms as per our need with the help of its own libraries

8. There was a huge issue of management of coordination and synchronization among the resources which resulted in inconsistency. Zookeeper overcame all the problems by performing synchronization, grouping and maintenance.

9. Oozie performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit.