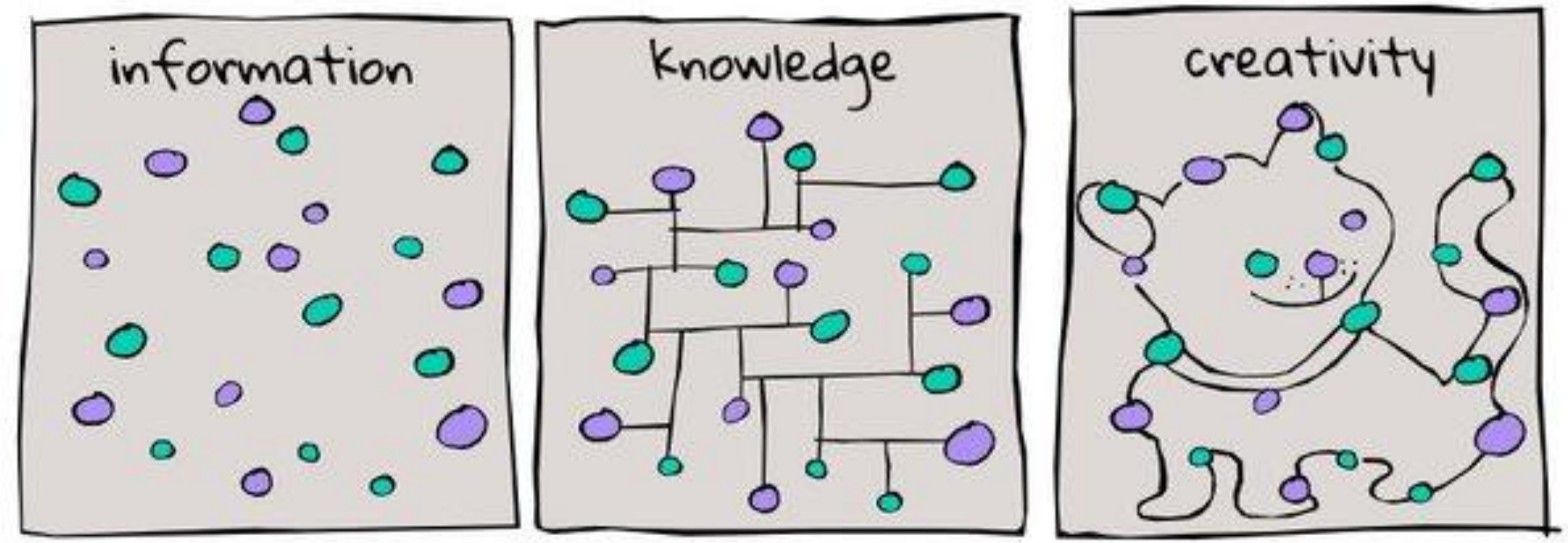
Big Data Analytics

Hive and Impala

Good Morning to all Big Data Analyst



Presenter

/arifmazumder

Mohammed Arif, PhD
Lead Data Scientist
Big Data | Machine Learning | AI

Mohammed Arif has more than thirteen (13) years of working experience in Information Communication and Technology (ICT) industry. The highlights of his career are more than six (7) years of holding various senior management and/or C-Level and had five (5) years of international ICT consultancy exposure in various countries (APAC and Australia), specially on Big Data, Data Engineering, Machine Learning and AI arena.

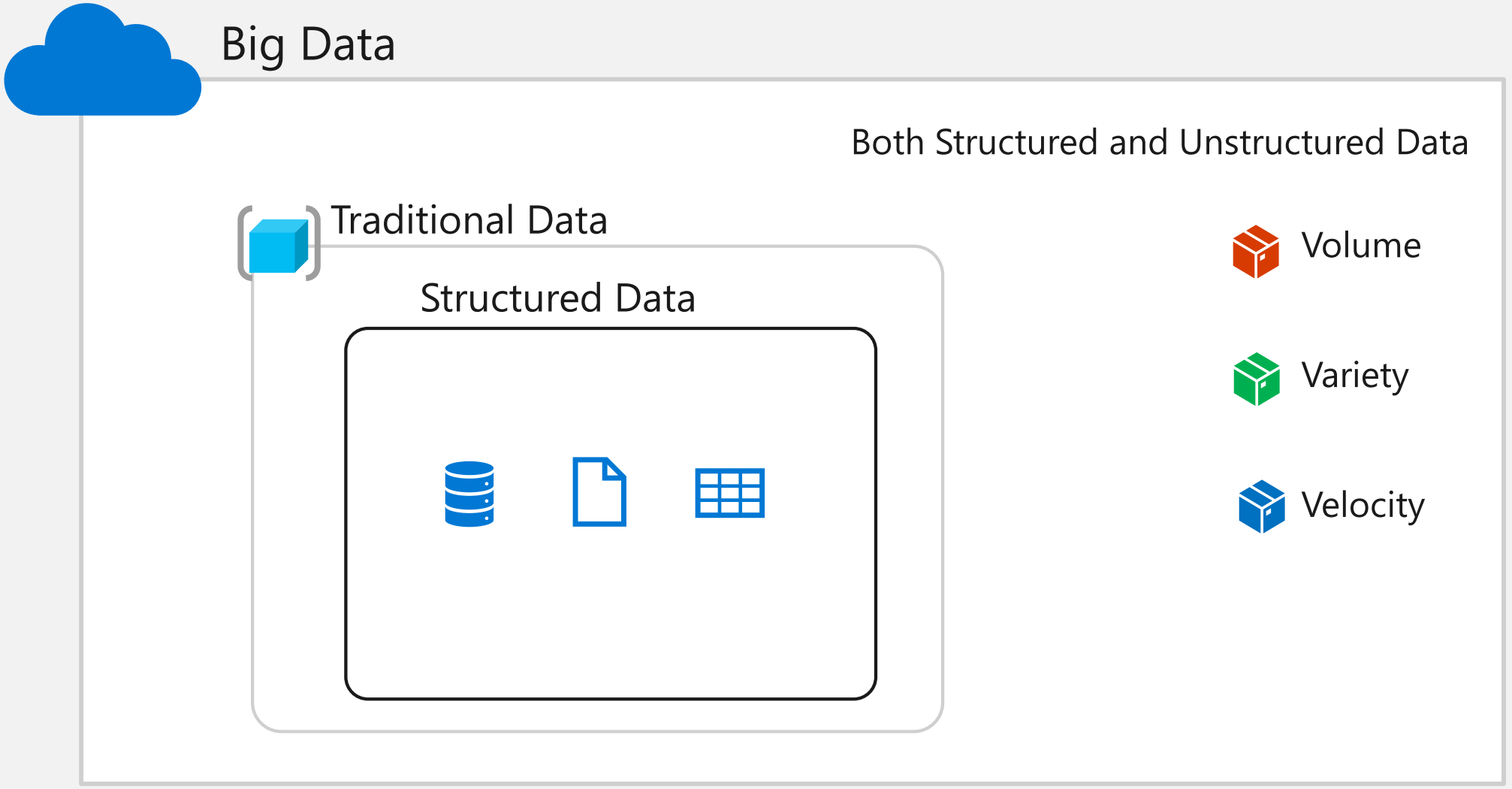
He is also Certified Trainer for HRDF and Microsoft

Agenda

- Big Data and its Ecosystem Components
- Reference Architecture
- Deploy Hadoop Distribution
- Introduction to Hive
- Architecture of Hive
- Hive Basic Commands
- Functions in Hive

Data vs Big Data

Data

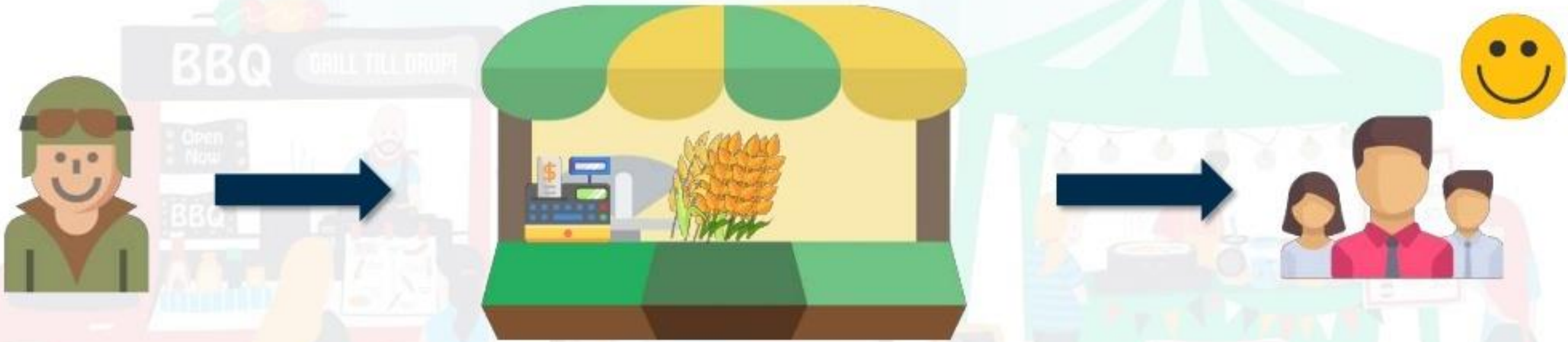

Big Data
Both Structured and Unstructured Data
Traditional Data
Structured Data
Volume
Variety
Velocity

Why Hadoop?

In a town far away..

Tim sells food grains in his shop

The customers were happy as Tim was very quick with the orders



Tim sensed a good demand for other products, so he thought of expanding his business

He started selling fruits, vegetables, meat, and dairy products in addition to food grains

But it wasn't as easy as he expected it to be. The number of customers increased, and he was not able to cater to their needs on time

He had to look into assisting his customers with each of their orders and billing. It was too difficult for him to manage alone

To start delivering orders on time and to manage the customers' demands, Tim hired 3 more people to work with him

Matt took care of the fruits and vegetable section. Luke handled the dairy and meat section. Ann was appointed as the cashier

Tim

Matt

Luke

Ann

However, this was still not a solution to Tim's problem as there was not enough space in the shop for all the items

The storage was a bottleneck since storing and accessing became more and more difficult with increased supply and demand

Storage area

Tim came up with an idea to overcome this issue. He decided to expand the storage area and distribute each category of product on different floors

Now, customers were happy, and after picking up their products from the respective sections, it was then billed

Now, customers were happy, and after picking up their products from the respective sections, it was then billed

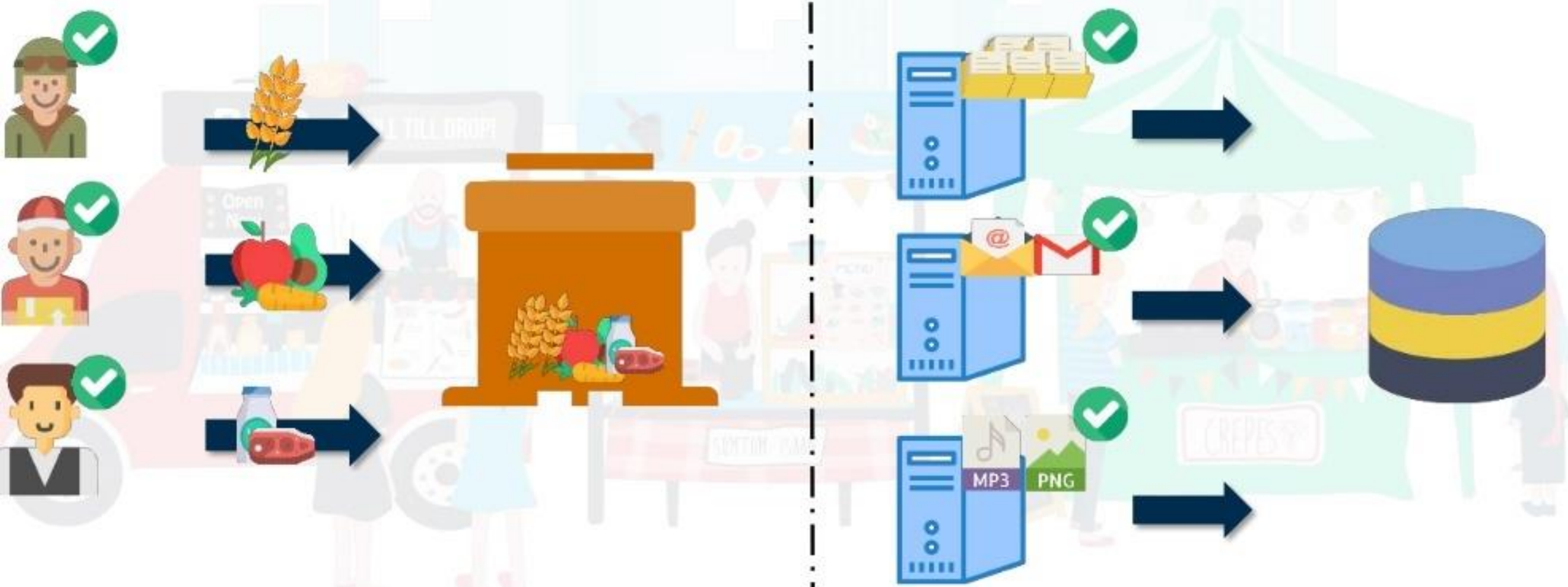
Now, let us compare this story to big data

Earlier, data was generated at a moderate rate, and all the data was structured in nature. One processor was enough to process all of it

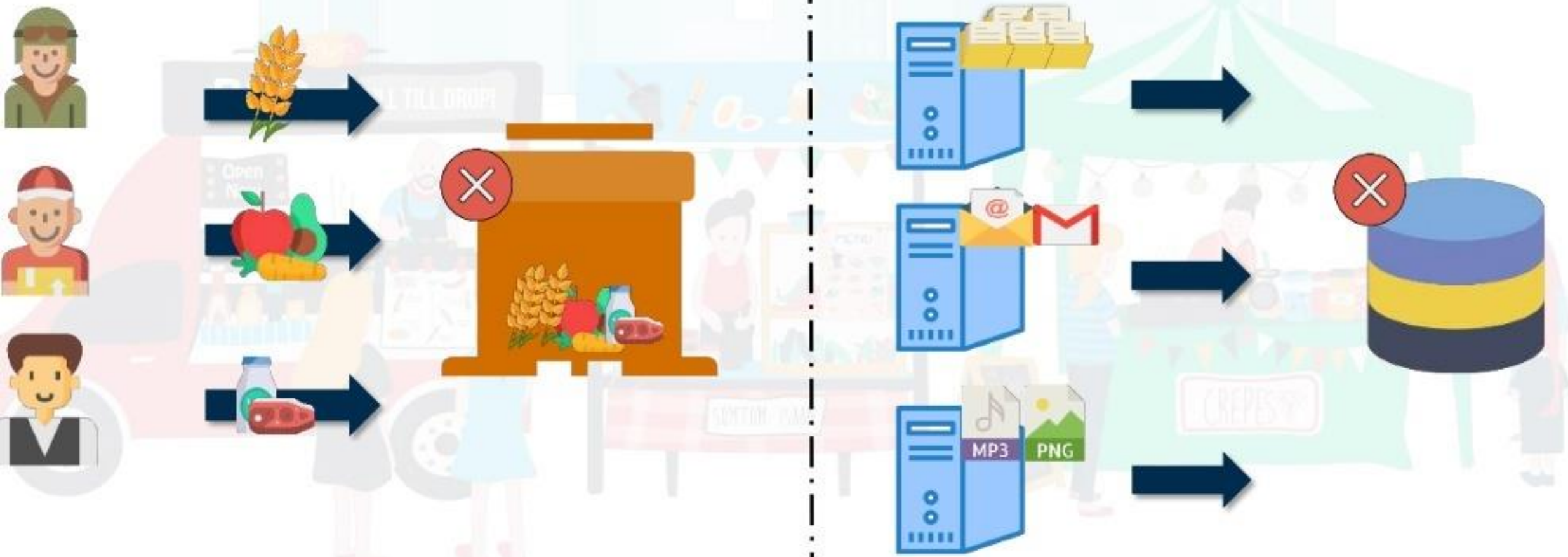
With the increase in data generation, different types of data were generated at high speed. It became difficult for a single processor to process different types of data

Massive amount of different types of data which cannot be processed and stored using traditional databases is known as big data

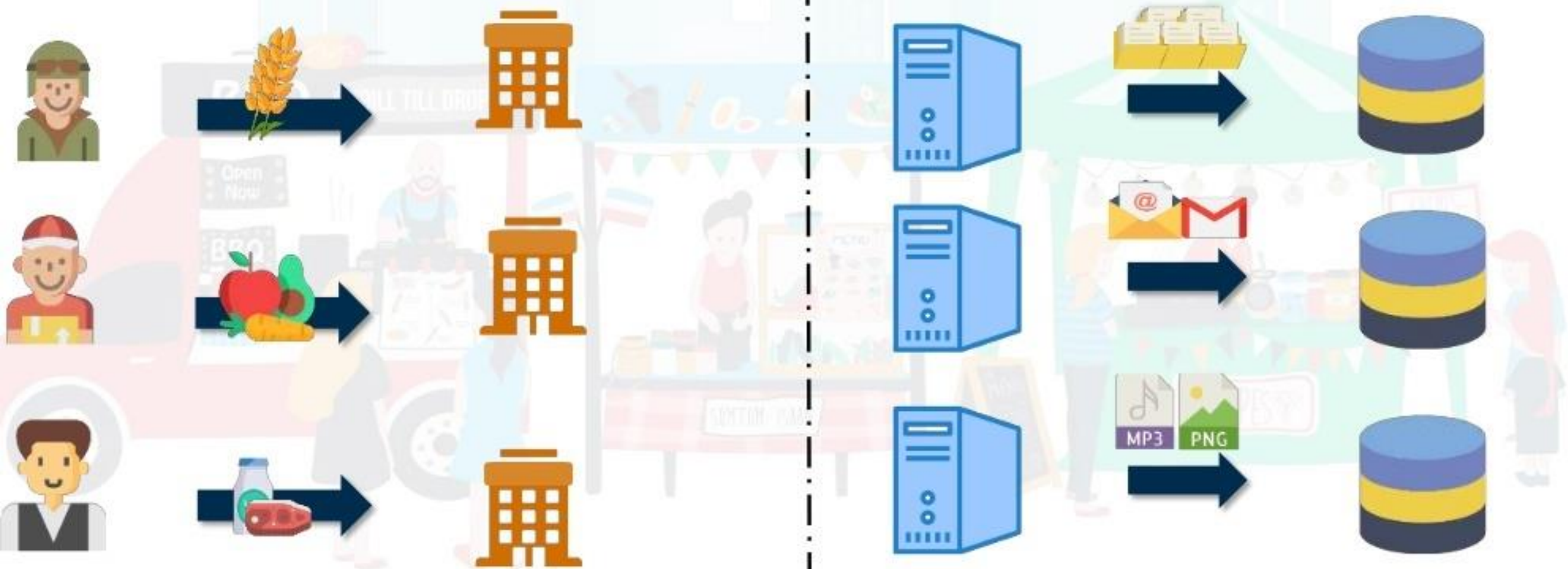
To overcome this issue, multiple processors were used to process each type of data



But now the problem was that one storage system was accessed by all the processors and the storage became the bottleneck



Just like how Tim adopted the distributed approach, the storage system was also distributed and by doing so, the data was stored in individual databases



Just like how Tim adopted the distributed approach, the storage system was also distributed and by doing so, the data was stored in individual databases

Through this story, we see the two approaches that are used by Hadoop that is HDFS and MapReduce

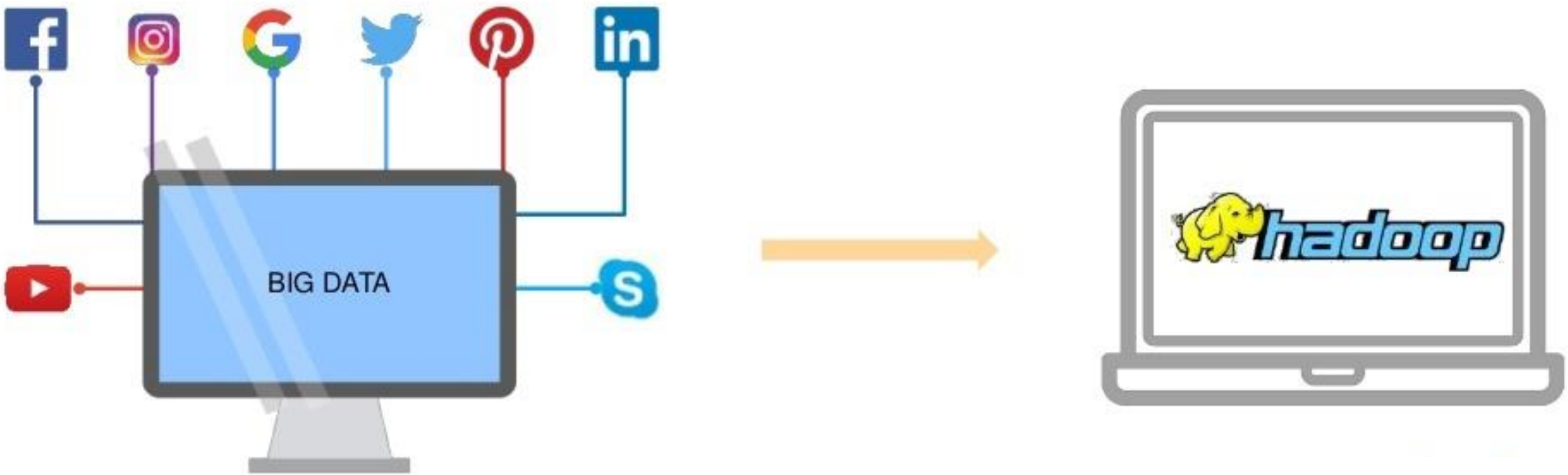
HDFS refers to the distributed storage space just like how Tim distributed the storage space amongst the various sections

Each person took care of a separate section and at the end the customers went to the cashier for the final billing, this sorted the process and made it easier. This is how Hadoop MapReduce works

What is Hadoop?

What is Hadoop?

Hadoop is a framework which stores and processes big data in a distributed and parallel fashion



One day in an office

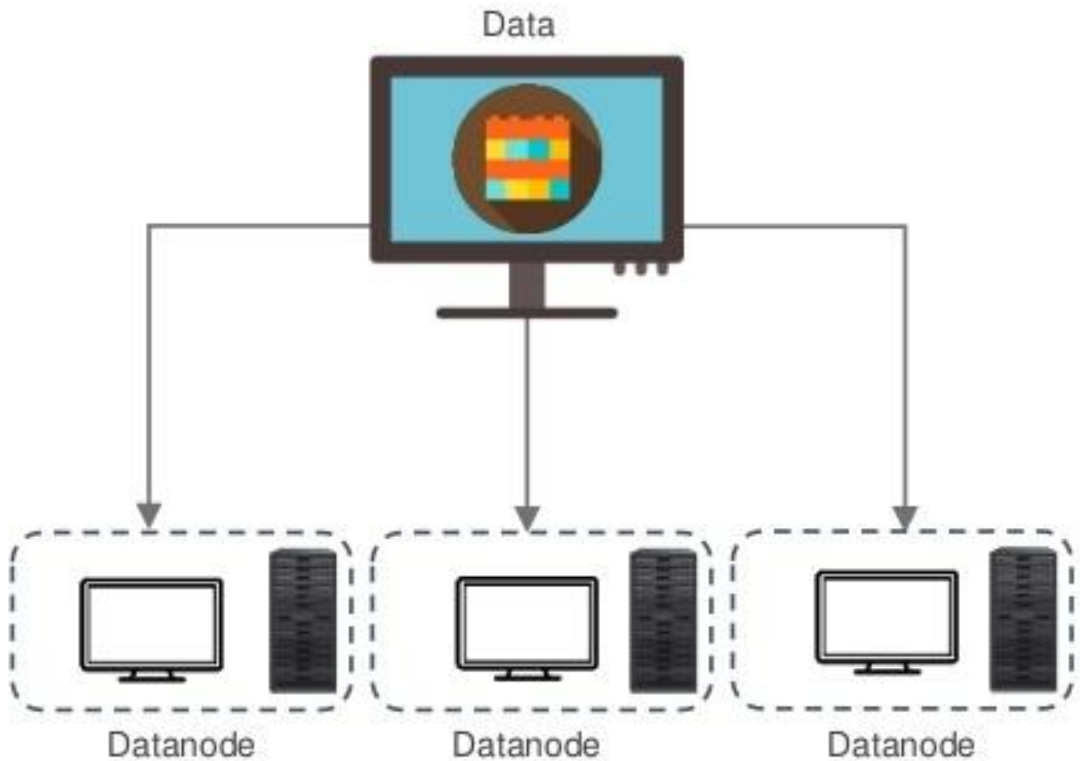
Components of Hadoop

HDFS	The storage unit of Hadoop
MapReduce	The processing unit of Hadoop
YARN	The resource management unit of Hadoop

Hadoop HDFS

What is HDFS?

Hadoop Distributed File System (HDFS) is known for its distributed storage method. It distributes the data amongst many computers. In addition to this, replication of data is also done to avoid loss of data



Each block of data is stored on multiple systems and by default has 128 MB of data

What is HDFS?

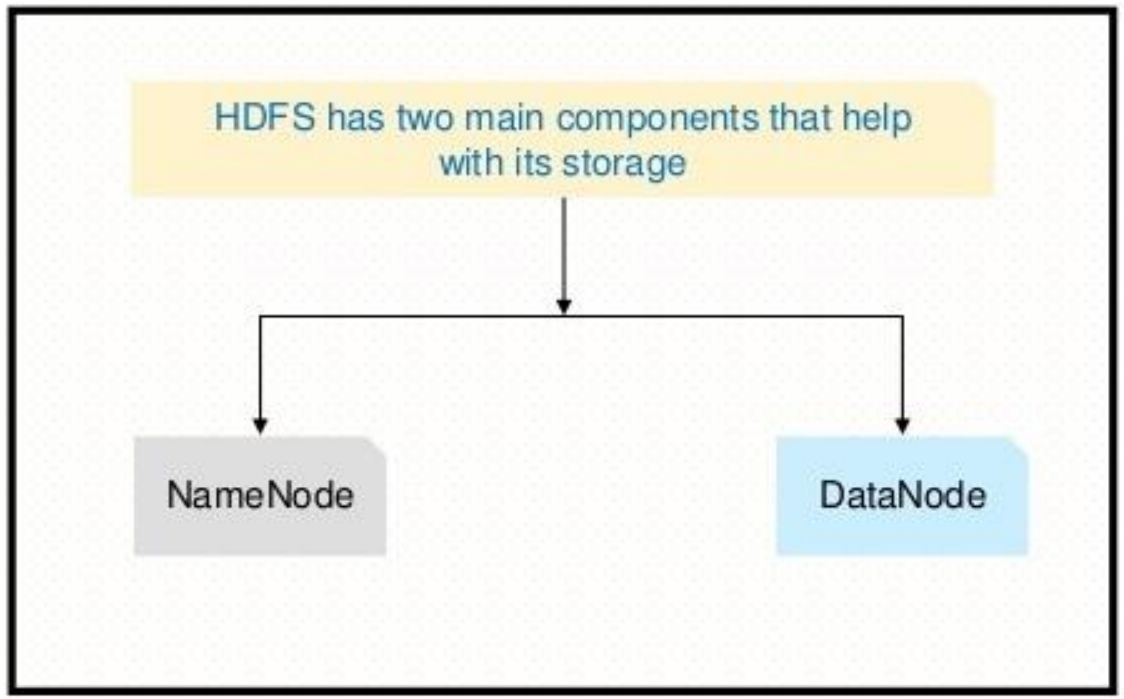
Let us now see how 500 MB of data is stored in the traditional method

What is HDFS?

Using Hadoop HDFS, this problem is taken care of as data is distributed amongst many systems

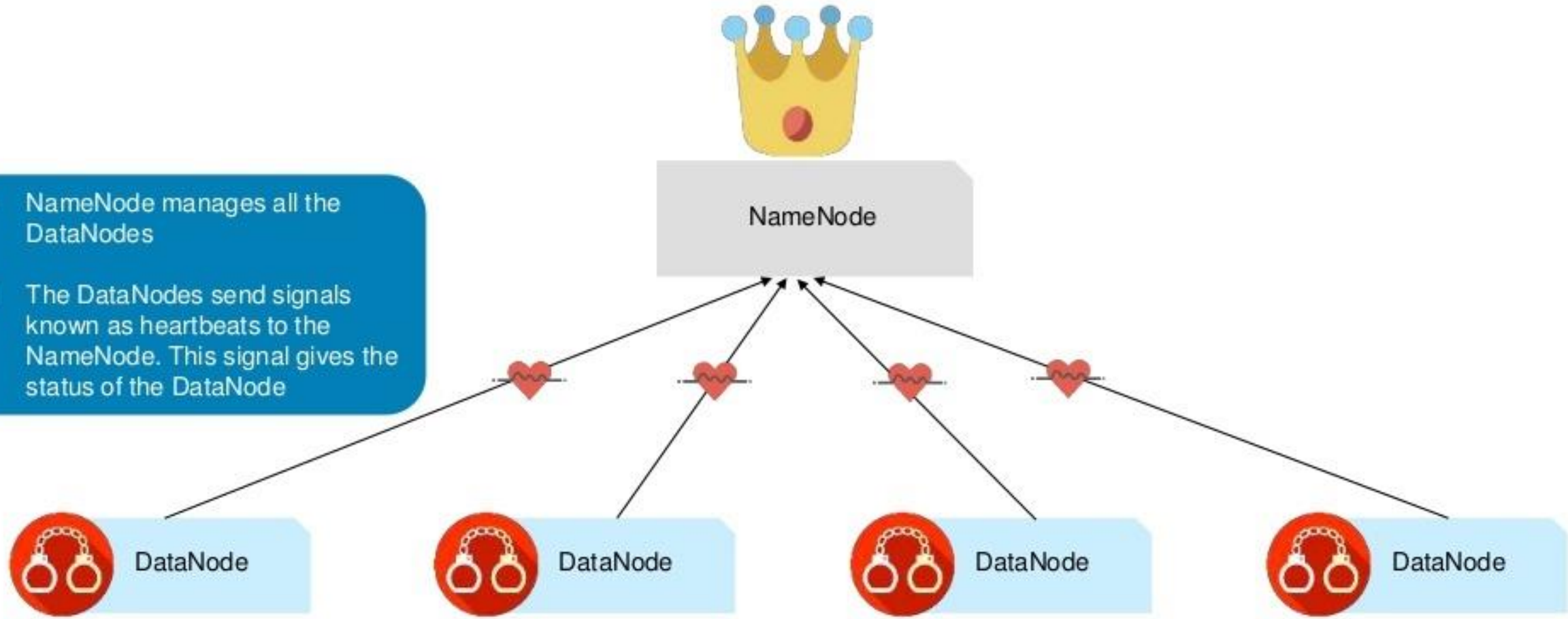
What is HDFS?

Hadoop Distributed File System (HDFS) is specially designed for storing massive datasets in commodity hardware



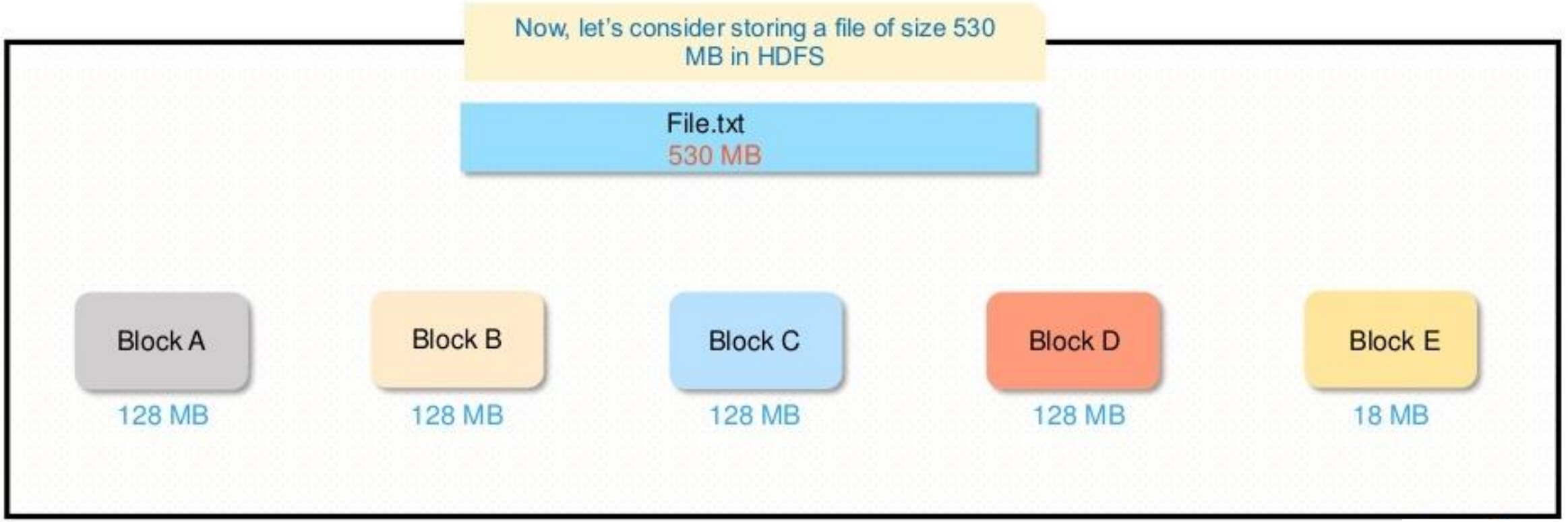
What is HDFS?

- NameNode manages all the DataNodes
- The DataNodes send signals known as heartbeats to the NameNode. This signal gives the status of the DataNode



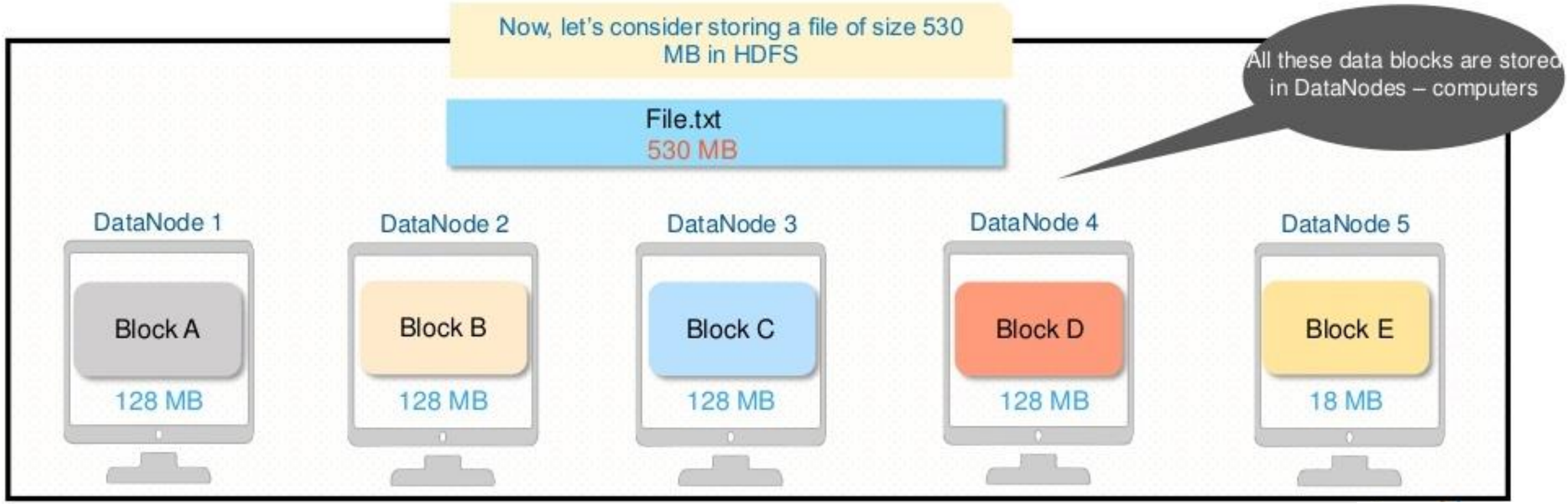
What is HDFS?

As mentioned earlier, the actual data is stored in DataNodes. Data is stored in the form of blocks here. The default size of each block is 128 MB



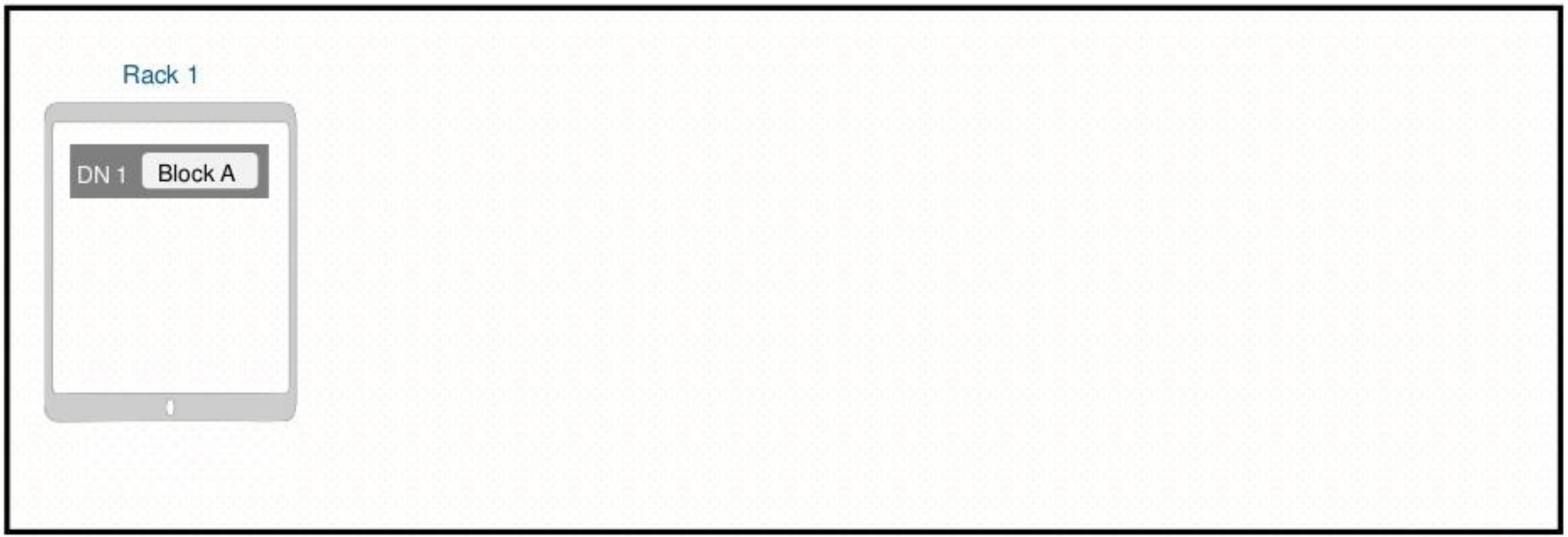
What is HDFS?

As mentioned earlier, the actual data is stored in DataNodes. Data is stored in the form of blocks here. The default size of each block is 128 MB



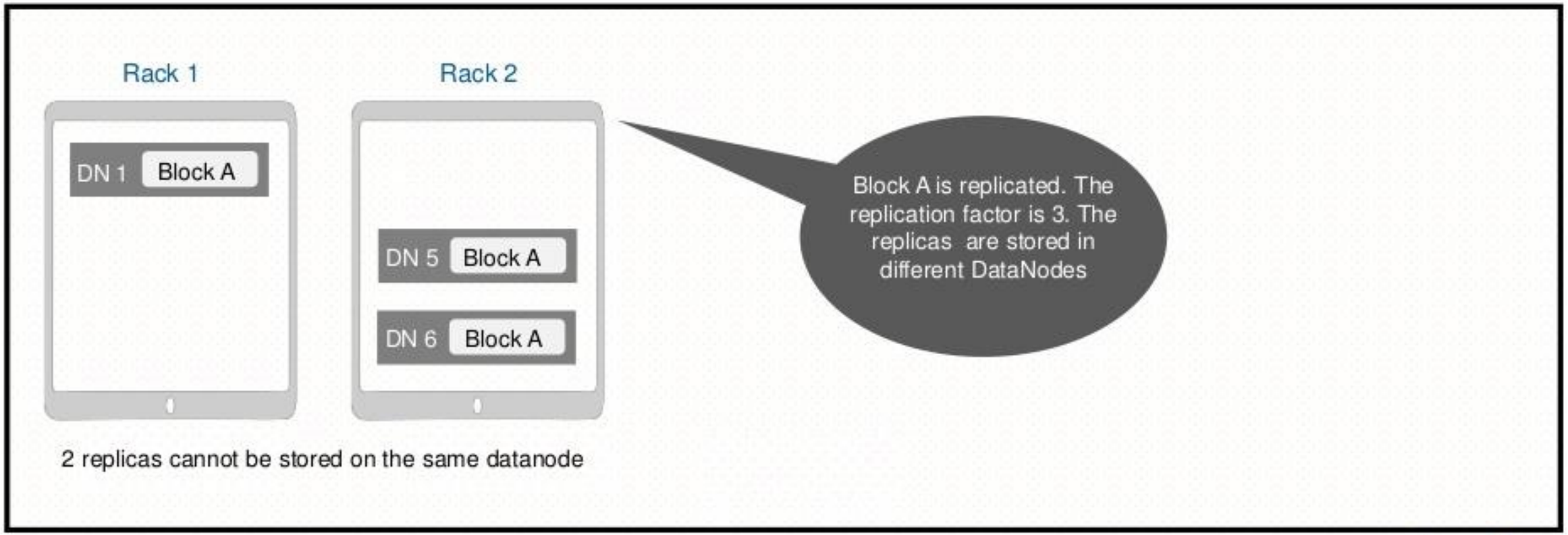
Replication in HDFS

HDFS overcomes the issue of DataNode failure by creating copies of the data; this is known as the replication method

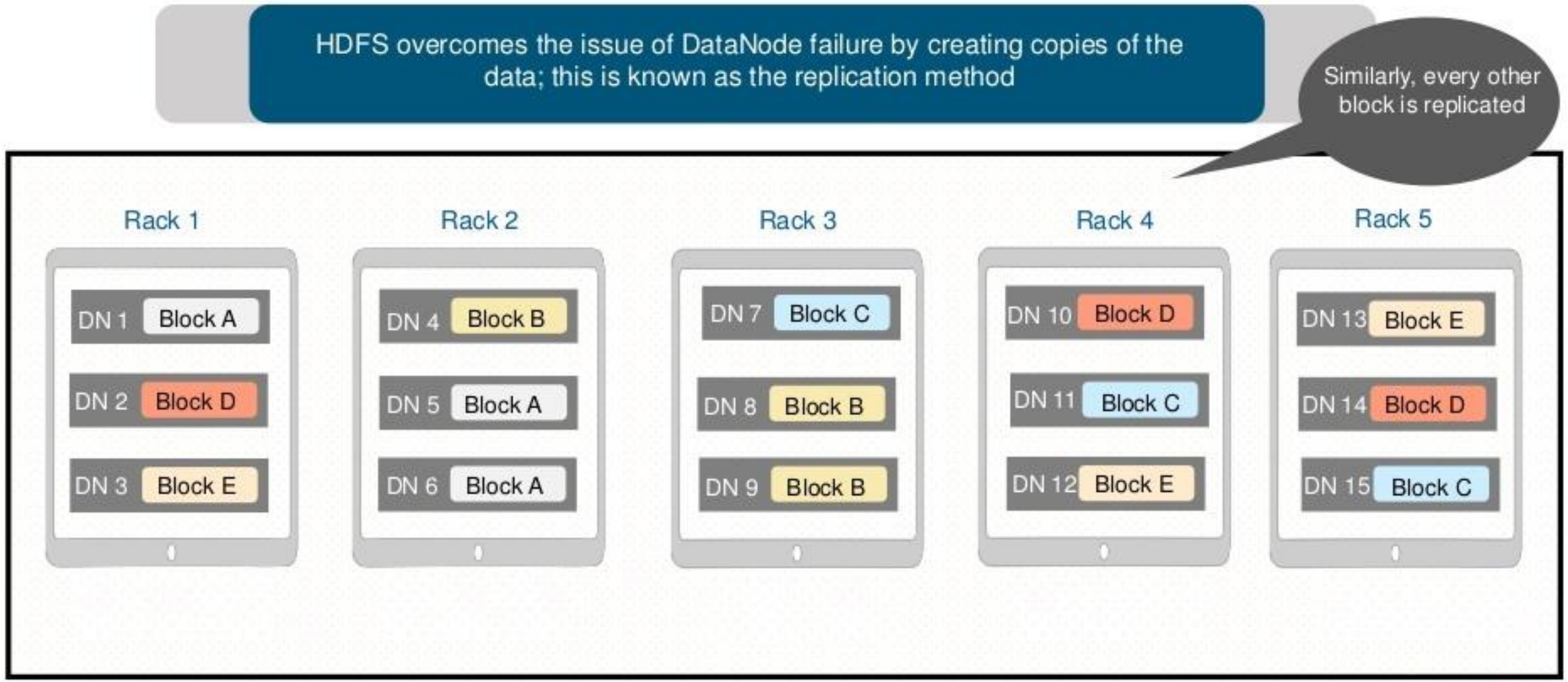


Replication in HDFS

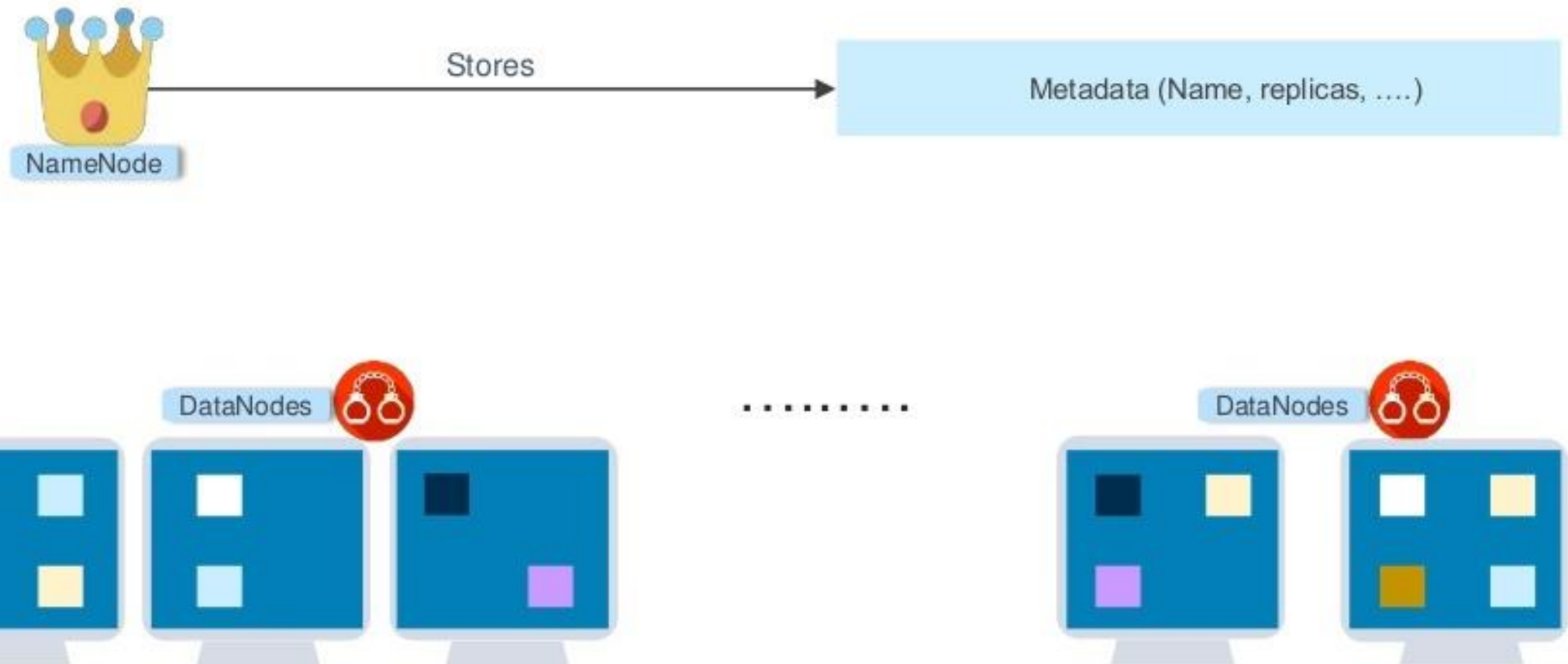
HDFS overcomes the issue of DataNode failure by creating copies of the data; this is known as the replication method



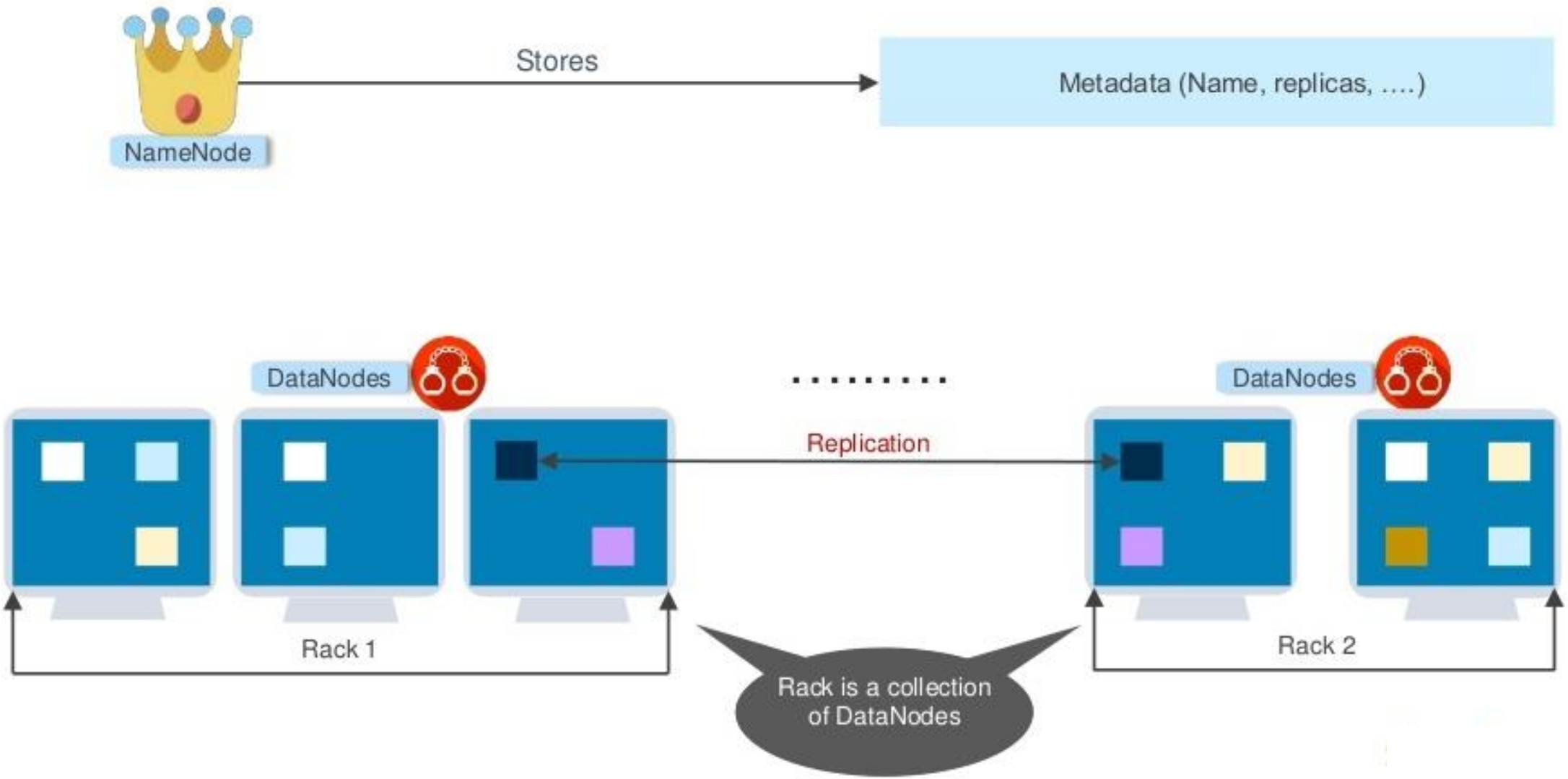
Replication in HDFS



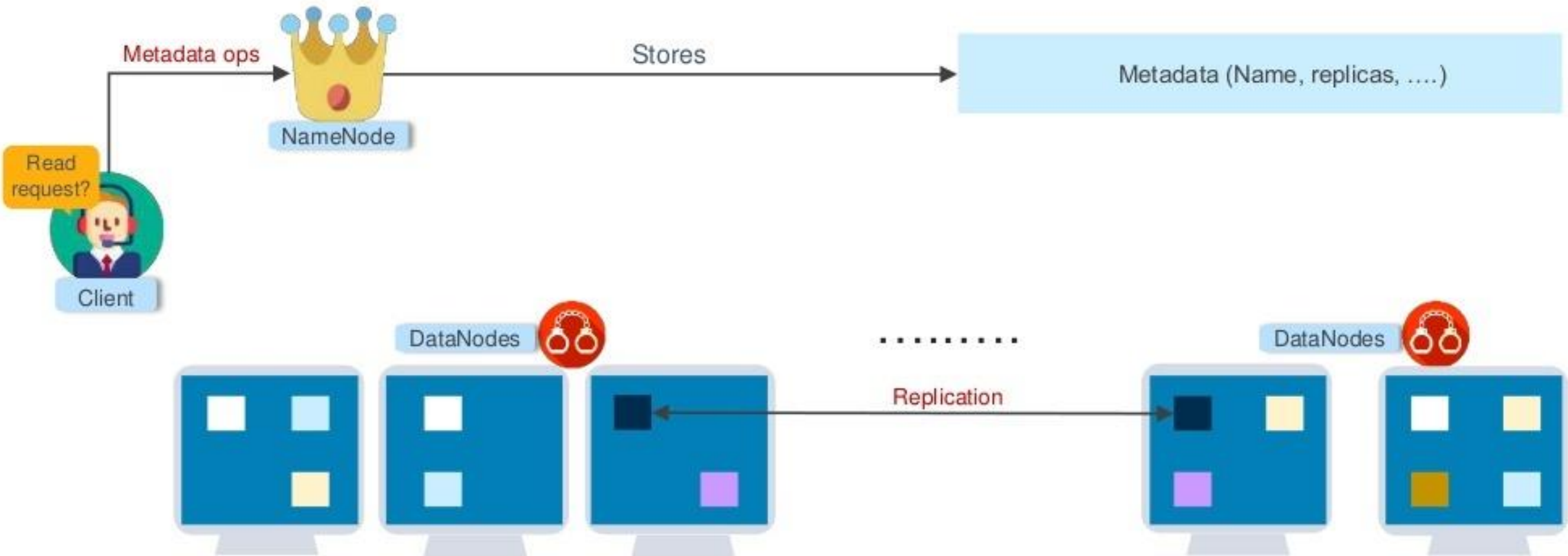
Architecture of HDFS



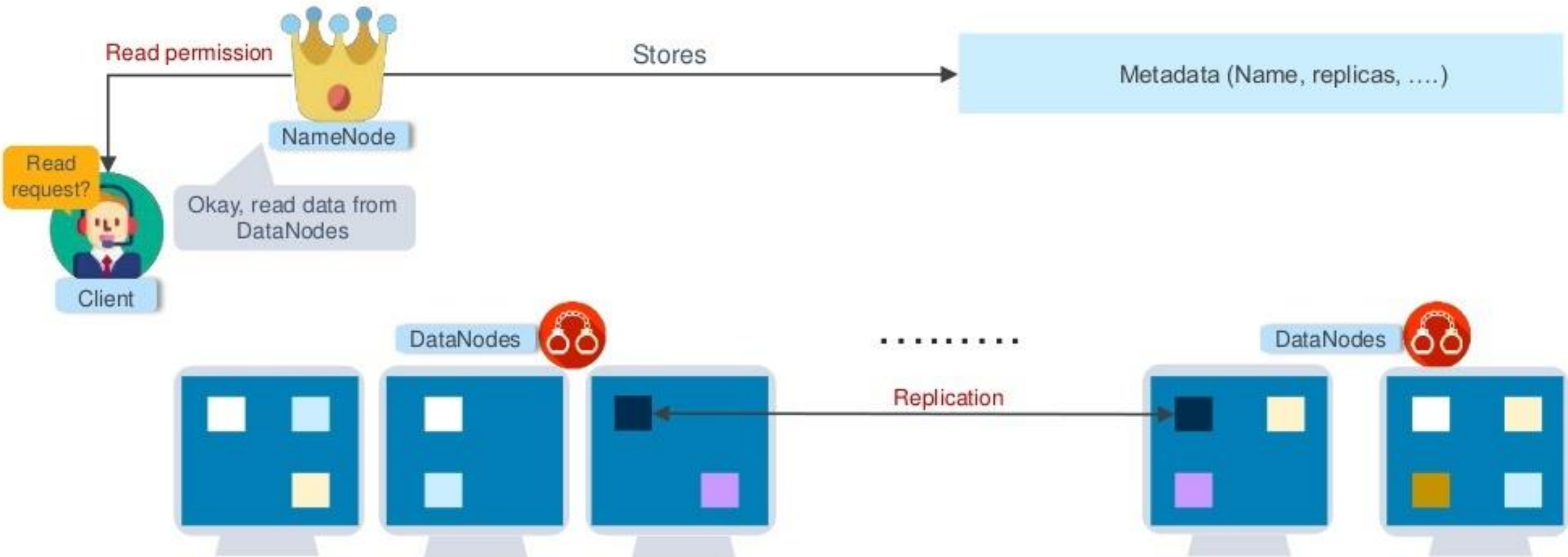
Architecture of HDFS



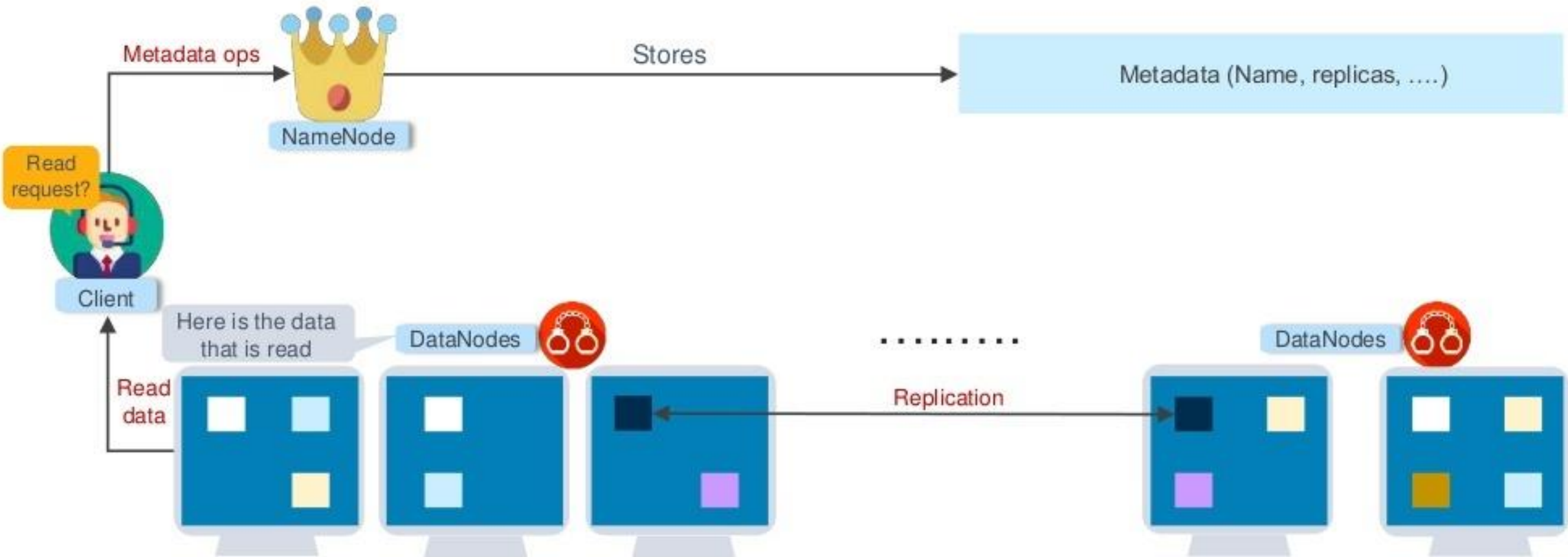
Architecture of HDFS



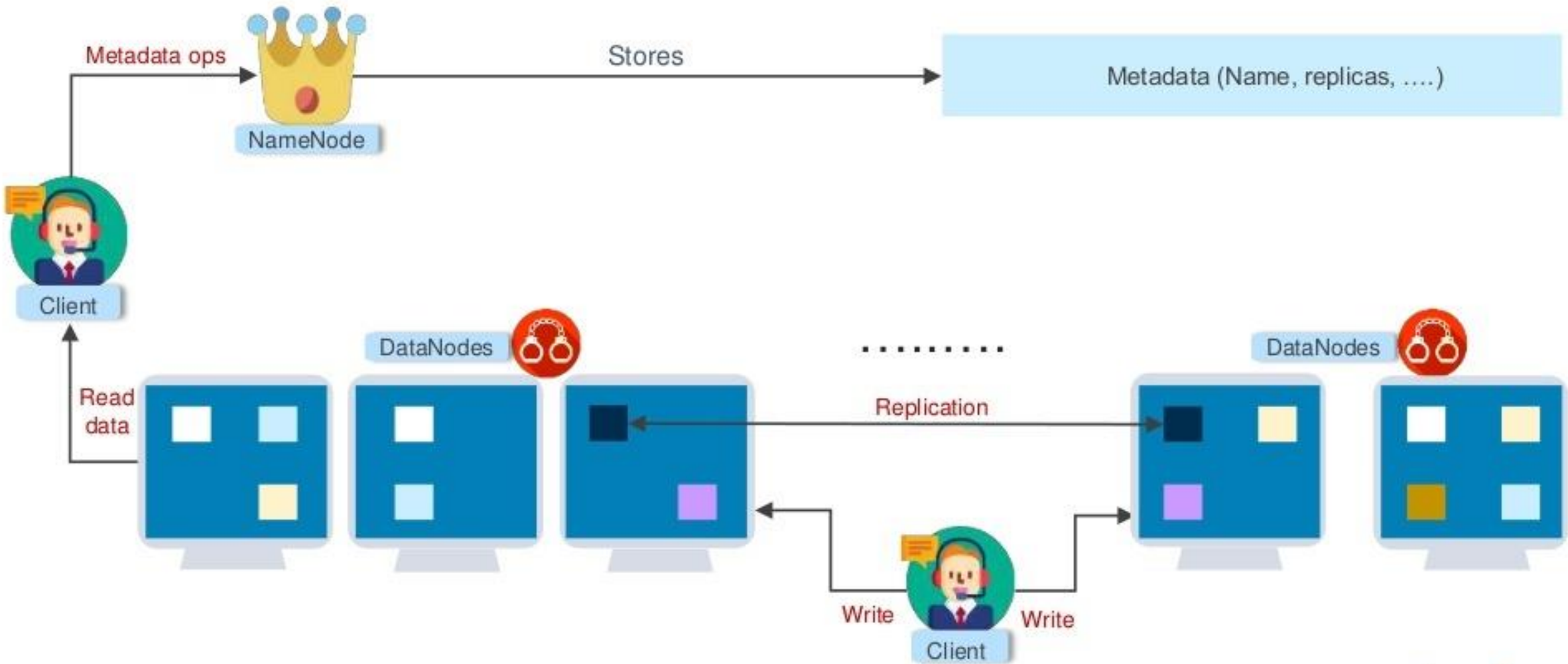
Architecture of HDFS



Architecture of HDFS



Architecture of HDFS



Features of HDFS

Fault tolerant

Data security

Scalability

Flexibility

Hadoop is flexible in storing any type of data, like structured, semi structured or unstructured data

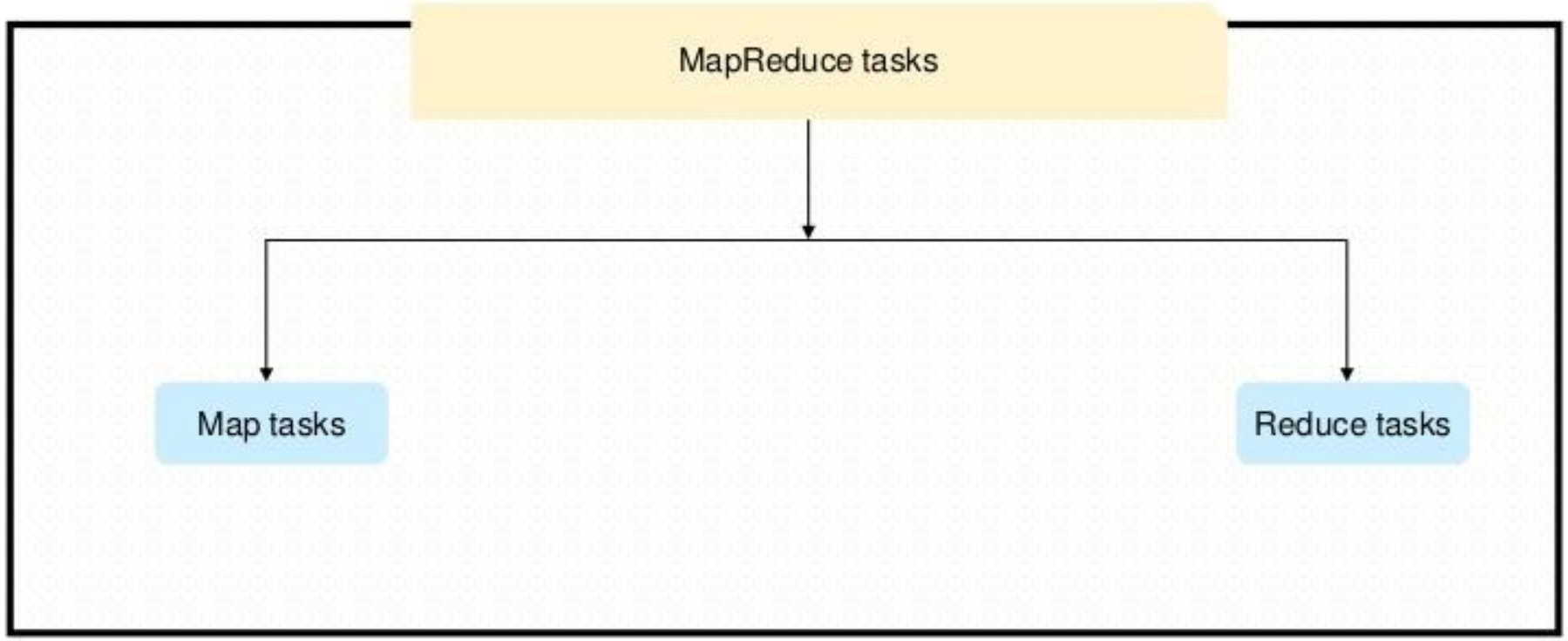
Hadoop MapReduce

What is MapReduce?

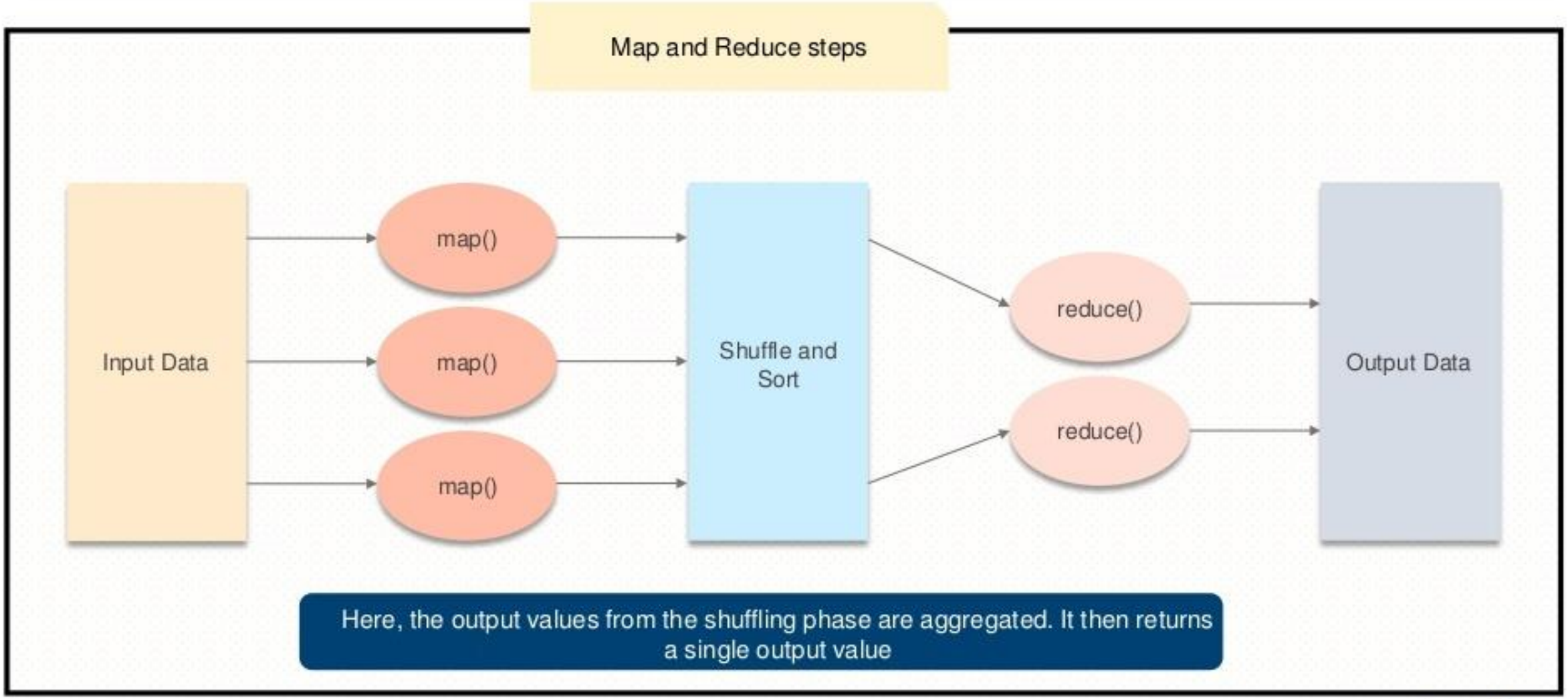
Programming technique where huge data is processed in a parallel and distributed fashion is known as Hadoop MapReduce

What is MapReduce?

Programming technique where huge data is processed in a parallel and distributed fashion is known as Hadoop MapReduce

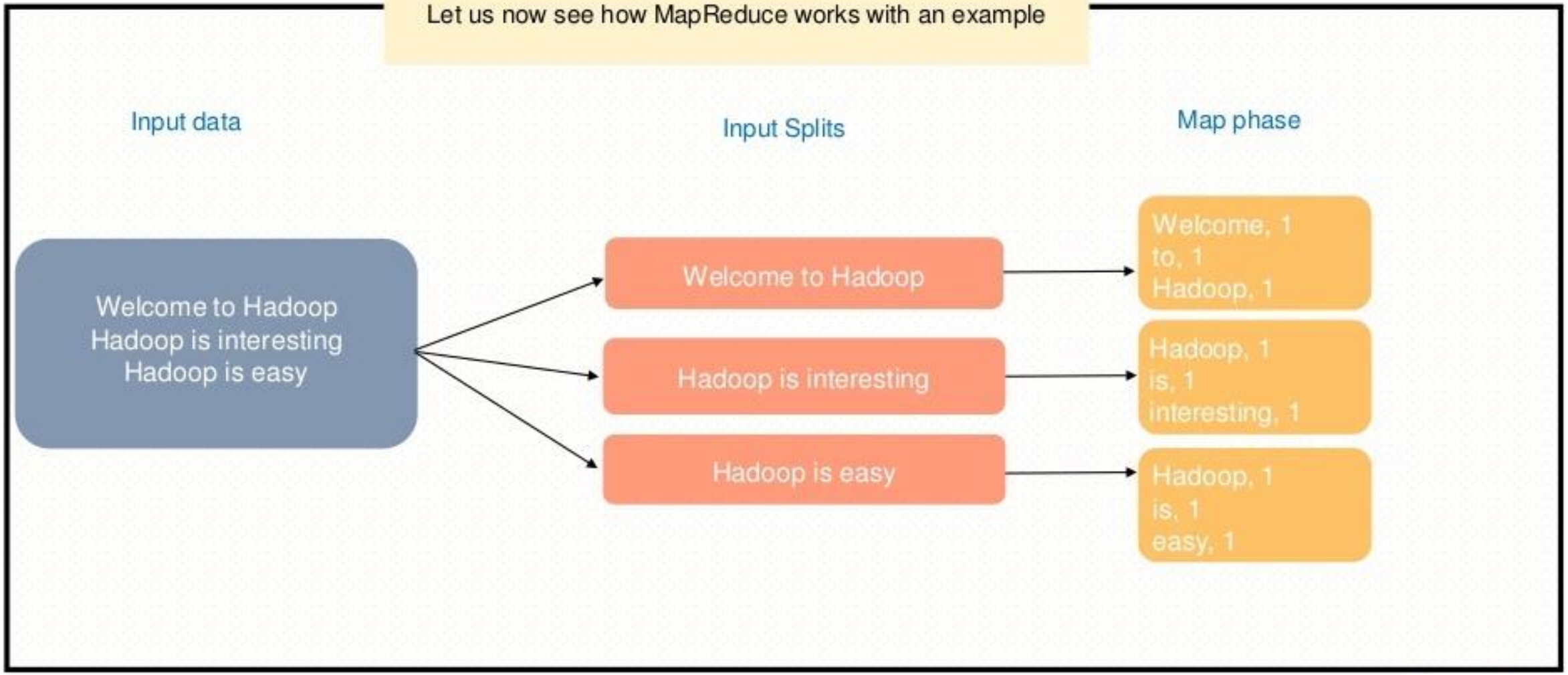


What is MapReduce?



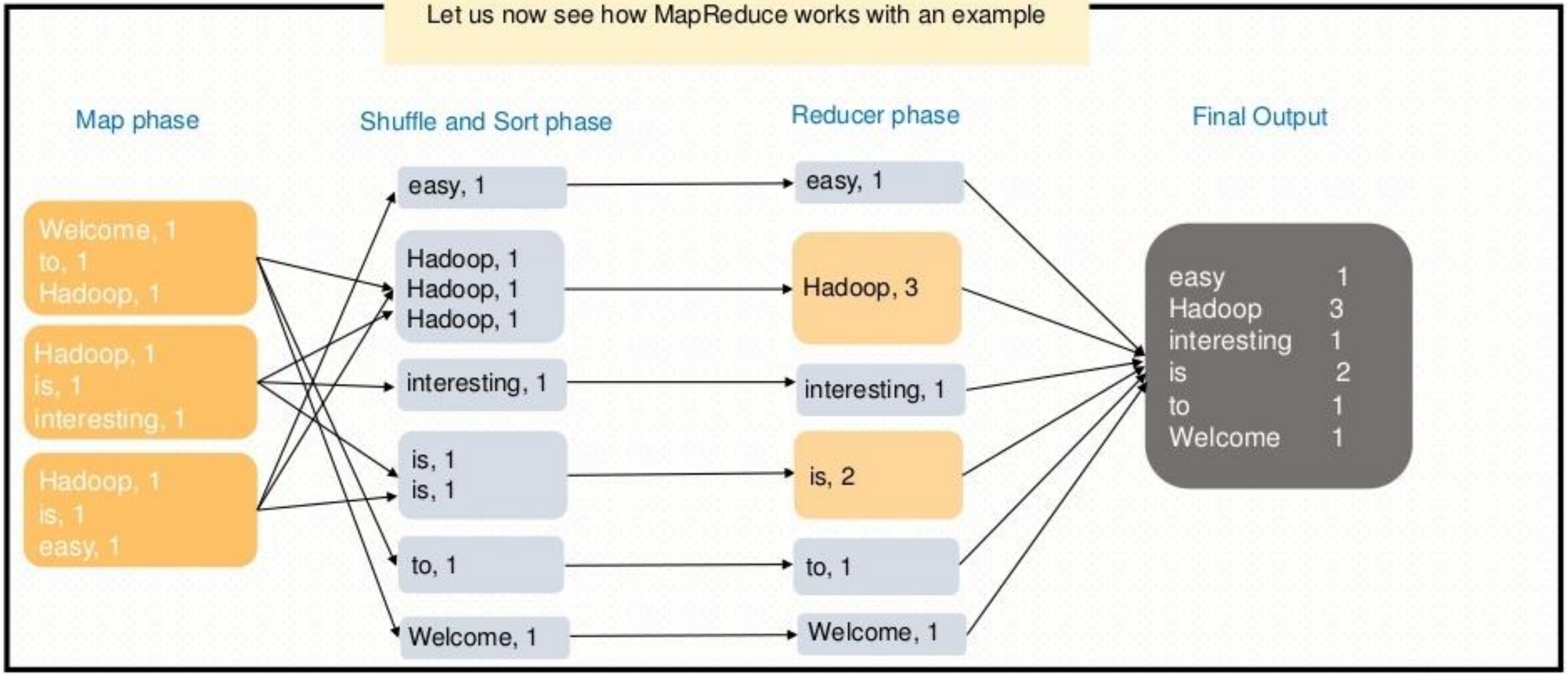
What is MapReduce?

Let us now see how MapReduce works with an example



What is MapReduce?

Let us now see how MapReduce works with an example



Features of MapReduce

Good load balancing

Map task + Reduce task

Re-execution of tasks

Simple programming model

MapReduce has one of the simplest programming model which is based on Java. Java is a very common programming language

Why MapReduce?

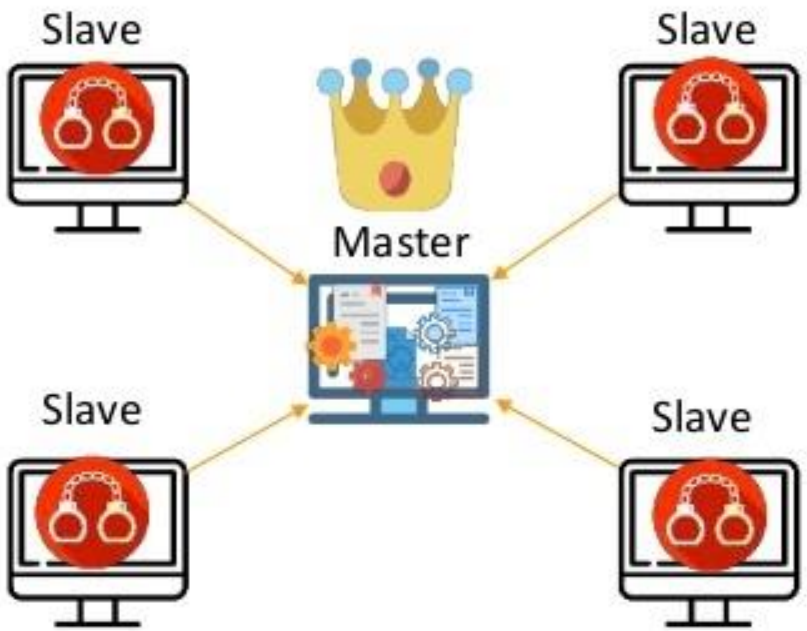
In the traditional approach, big data was processed at the master node

big data

Why MapReduce?

In the traditional approach, big data was processed at the master node

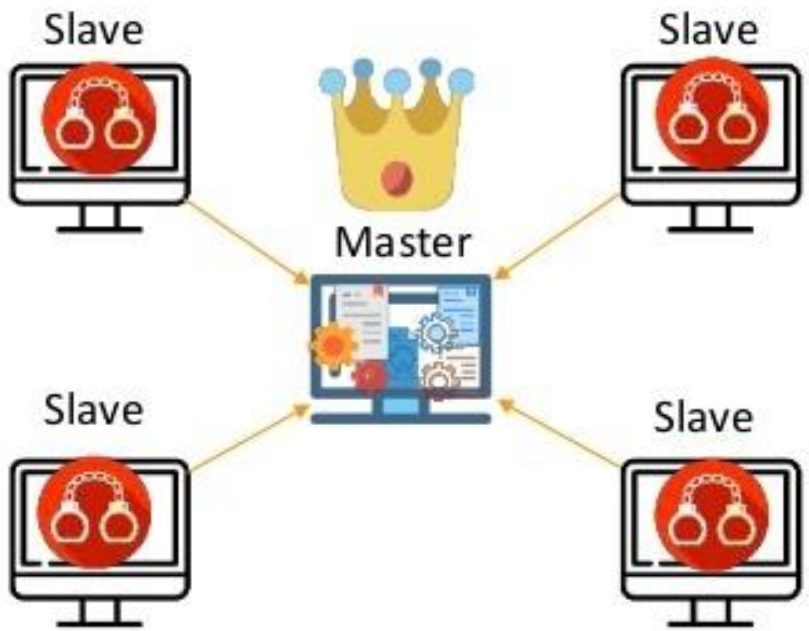
big data



Why MapReduce?

This was a disadvantage as it consumed more time to process various types of data

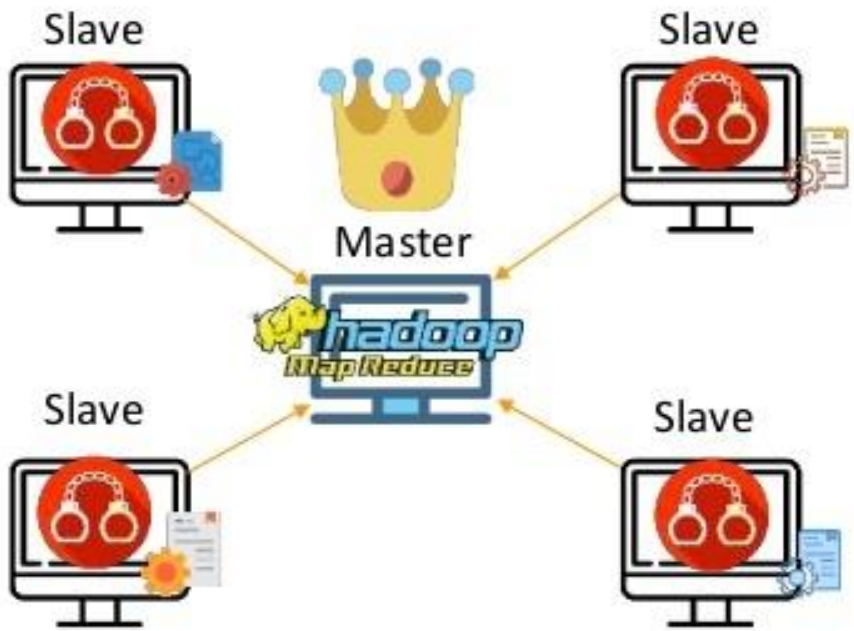
big data



Why MapReduce?

To overcome this issue, data was processed at each slave node. This approach is known as MapReduce

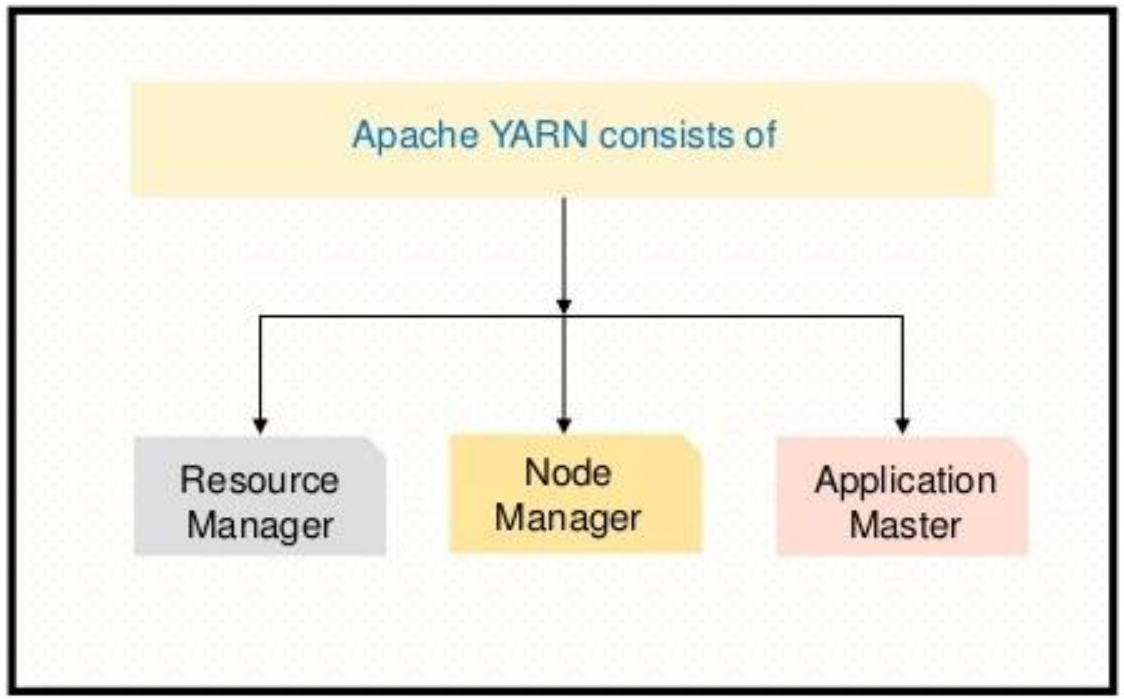
big data



Hadoop YARN

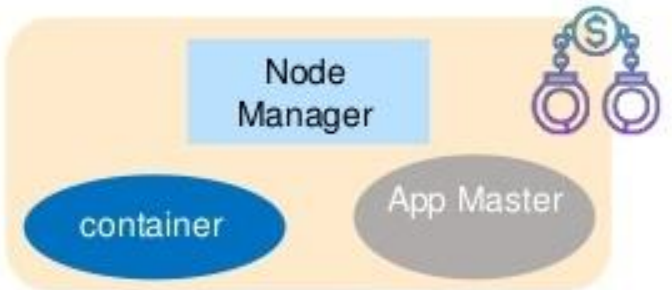
What is YARN?

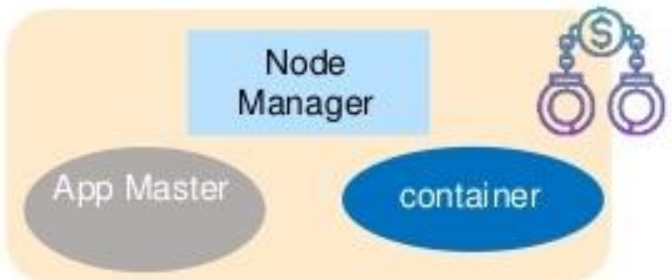
Yet Another Resource Negotiator (YARN) acts as the resource management unit of Hadoop

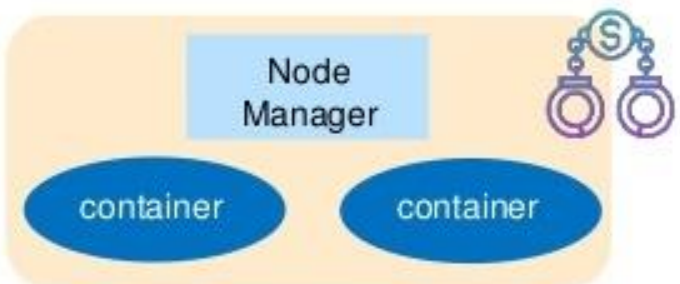


Works with the negotiation of resources from resource manager and works with node manager

What is YARN?





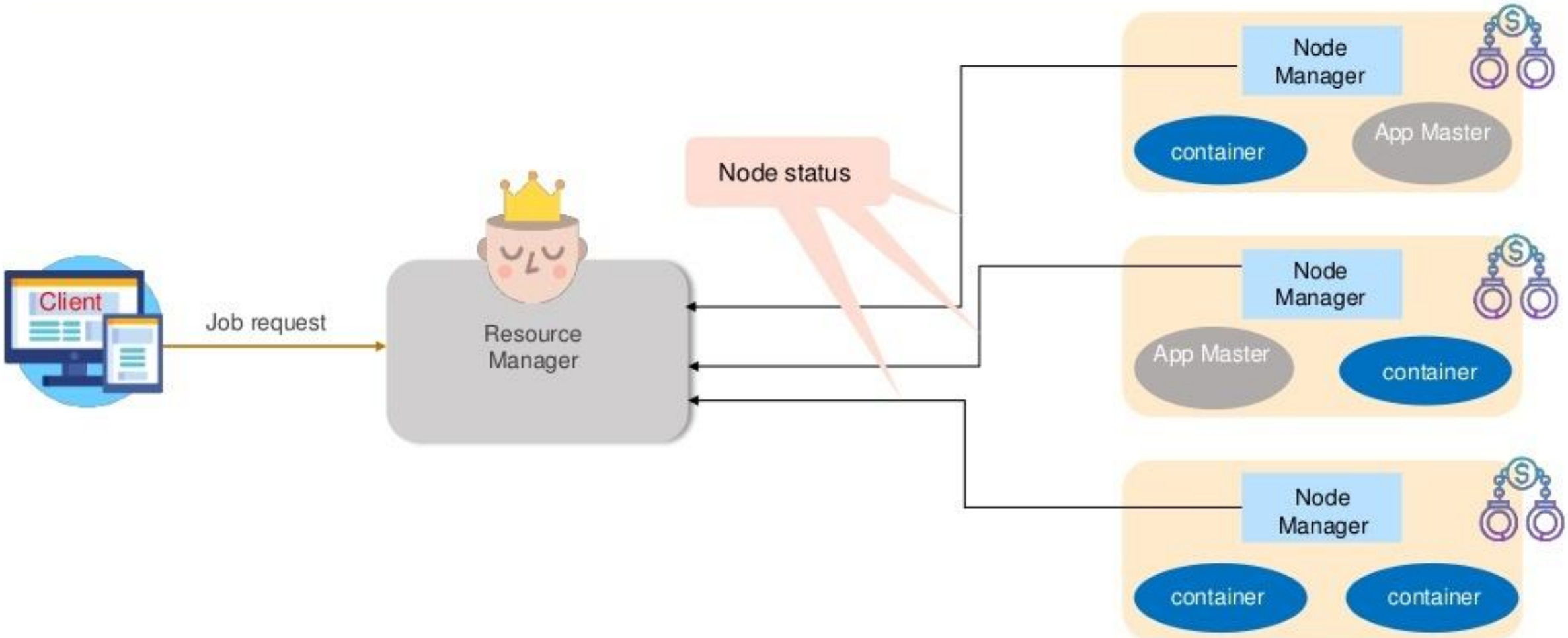


App Master requests container to Resource Manager. It uses container allocated by Node Manager

What is YARN?

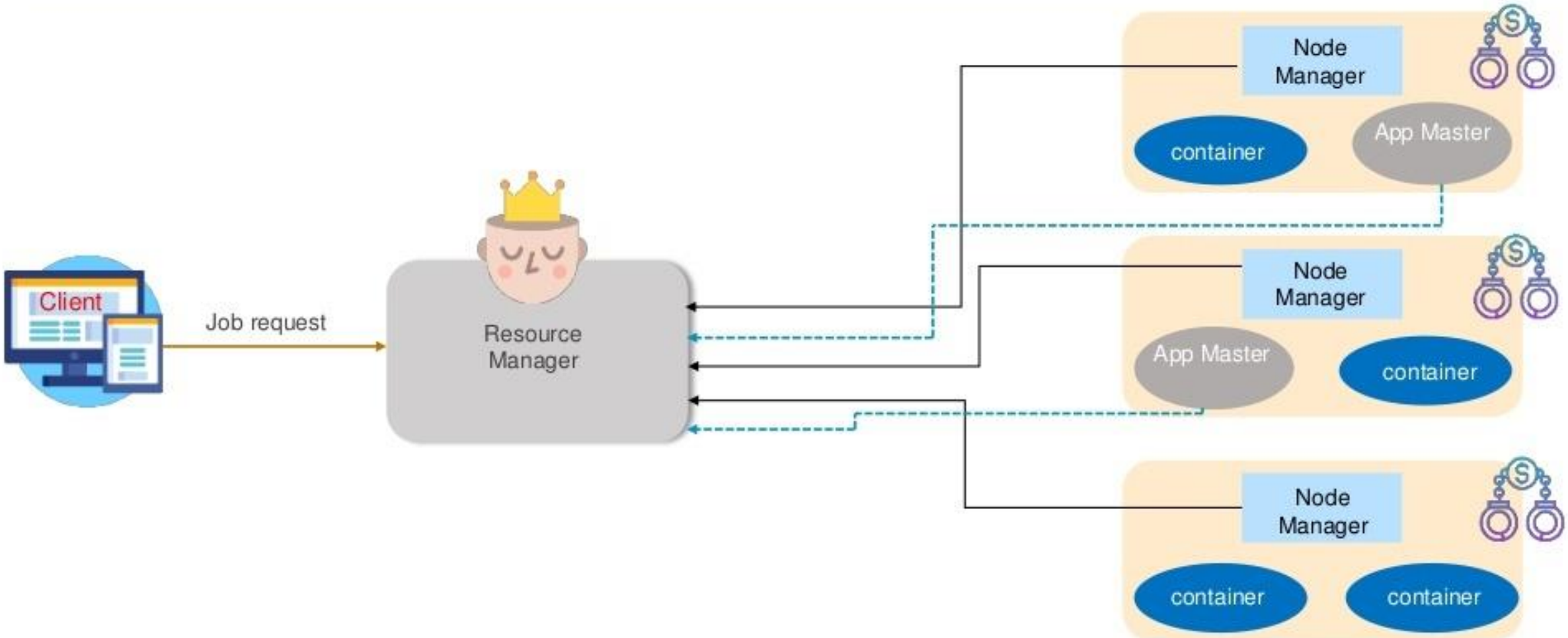
Client program sends application request to the resource manager

What is YARN?



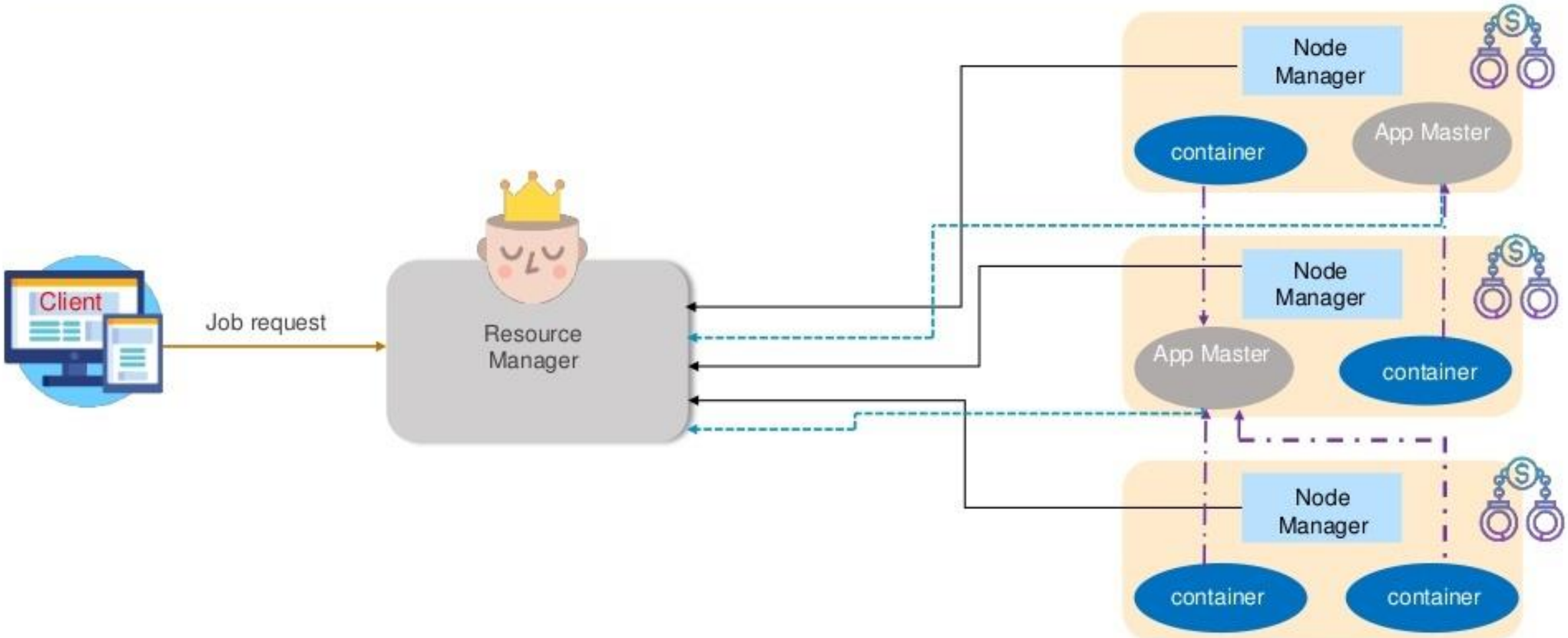
Node manager updates the status of the nodes to the resource manager

What is YARN?



Resource Manager contacts the Node Manager requesting for resources(containers). The Node Manager grants the request

What is YARN?



App Master contacts the Node Manager to use the container and runs in one of the container allocated on one of the nodes

Features of YARN

Job scheduling

Multitenancy

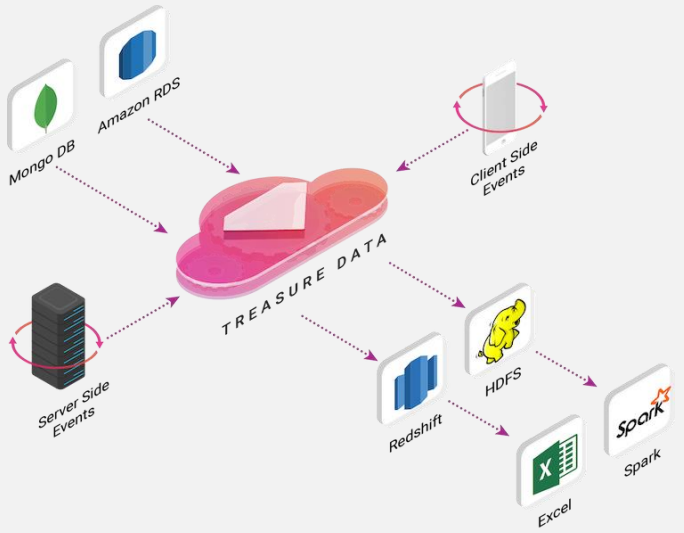
Scalability

Depending on the requirement, the number of nodes can be increased

Reference Architecture

What is Reference Architecture

A reference architecture shows which **functionality** is generally needed in a certain domain or the solve a certain class of problems. Also, how this functionality is divided and how **information flows** between the pieces (called the reference model).



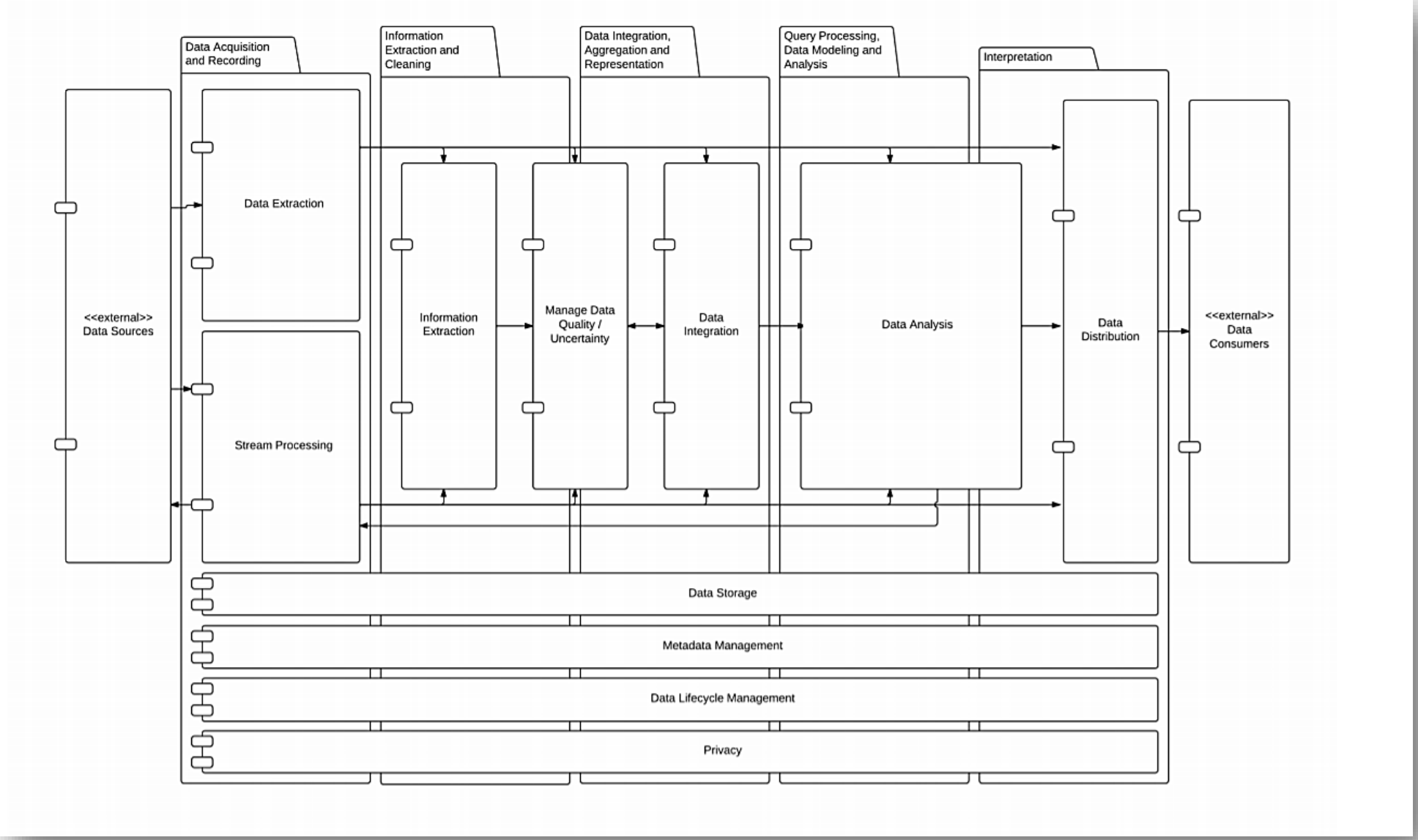
It then maps this functionality onto software elements and the data flows between them.

Reference Architecture: High-level functional view

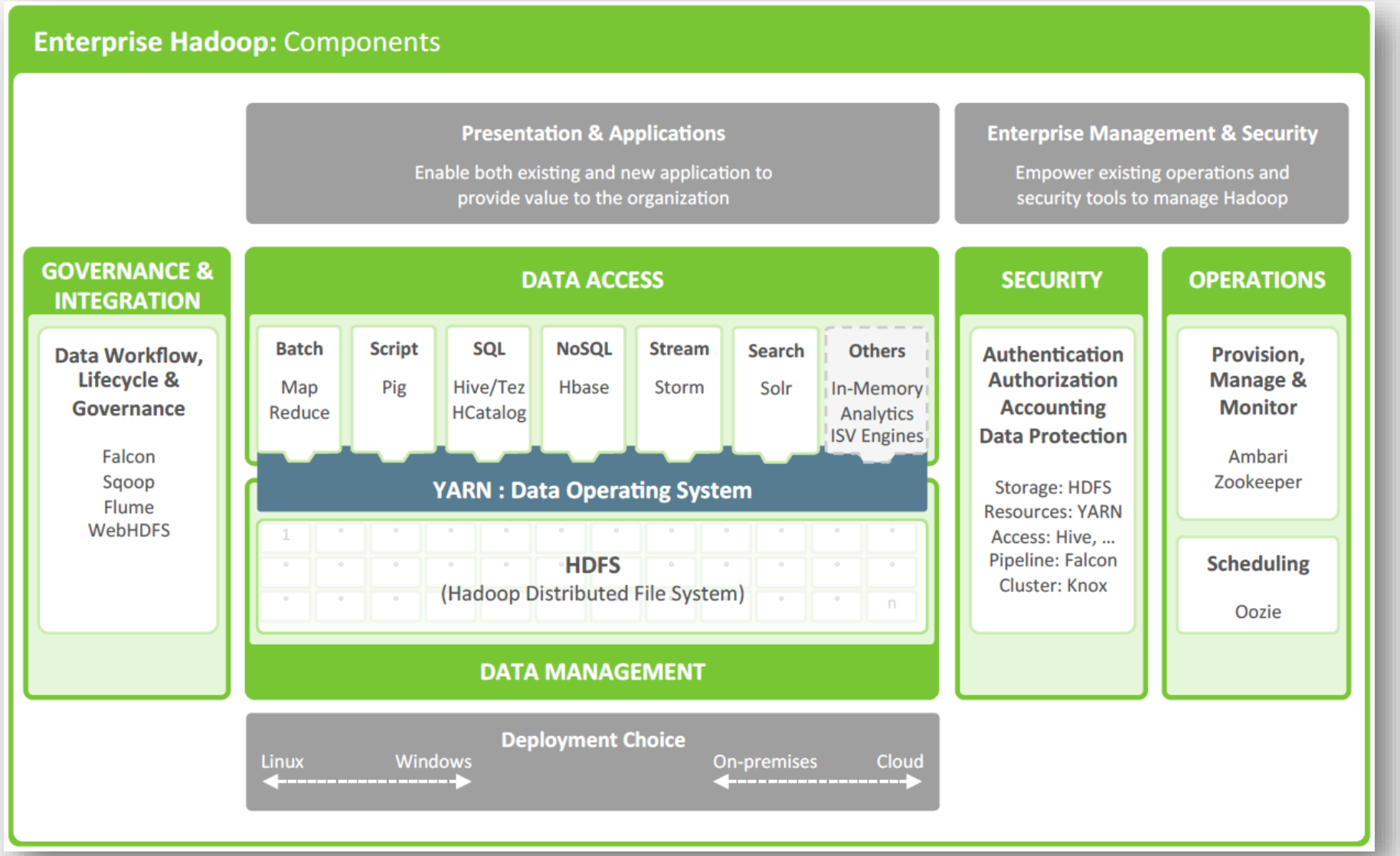
The analysis of ‘big data’ into distinct phases of a sequential processing **pipeline**:

- Acquisition / Recording,
- Extraction / Cleaning / Annotation,
- Integration / Aggregation / Representation,
- Analysis / Modeling, and
- Interpretation.

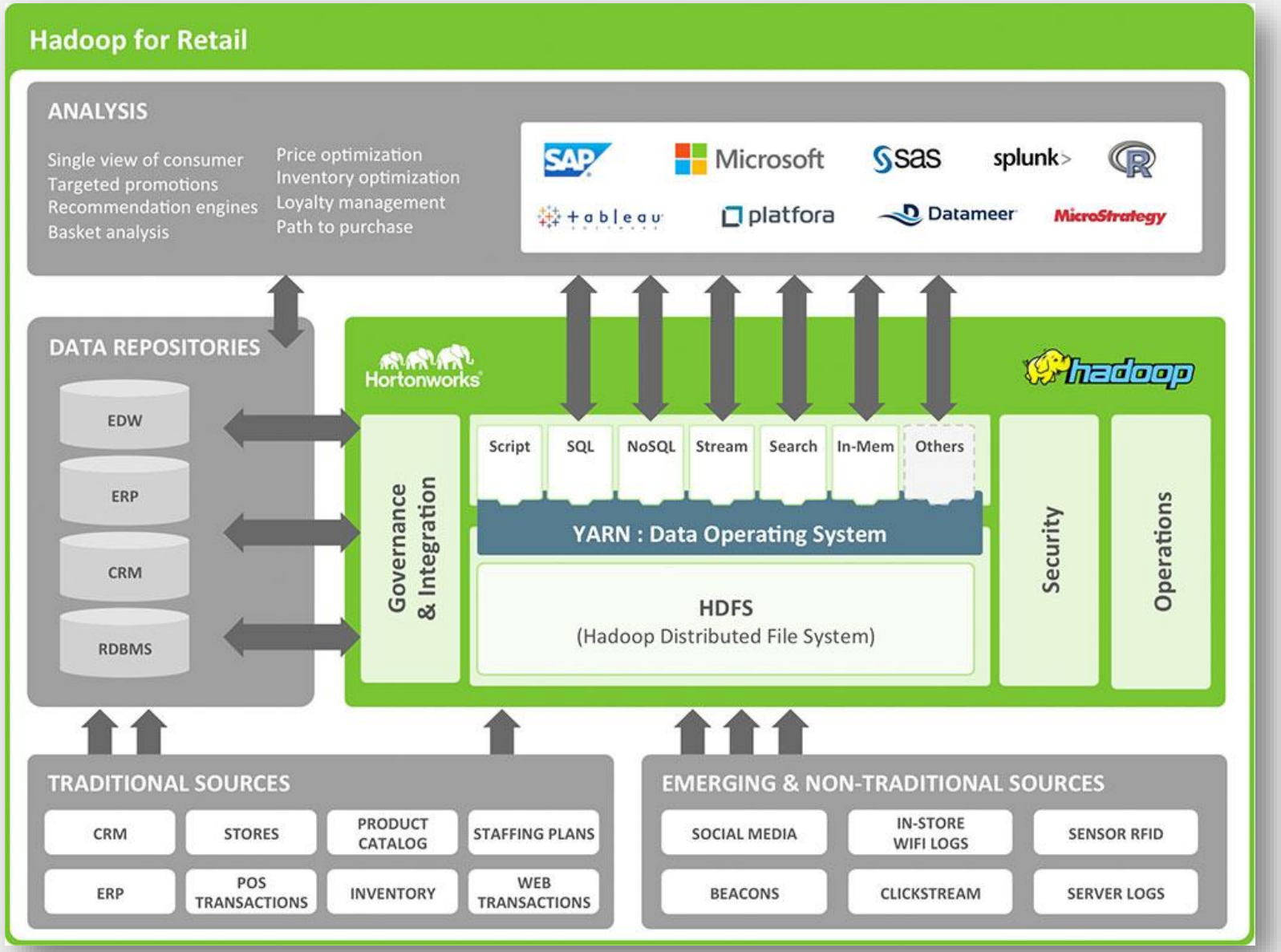
Reference Architecture: High-level functional view



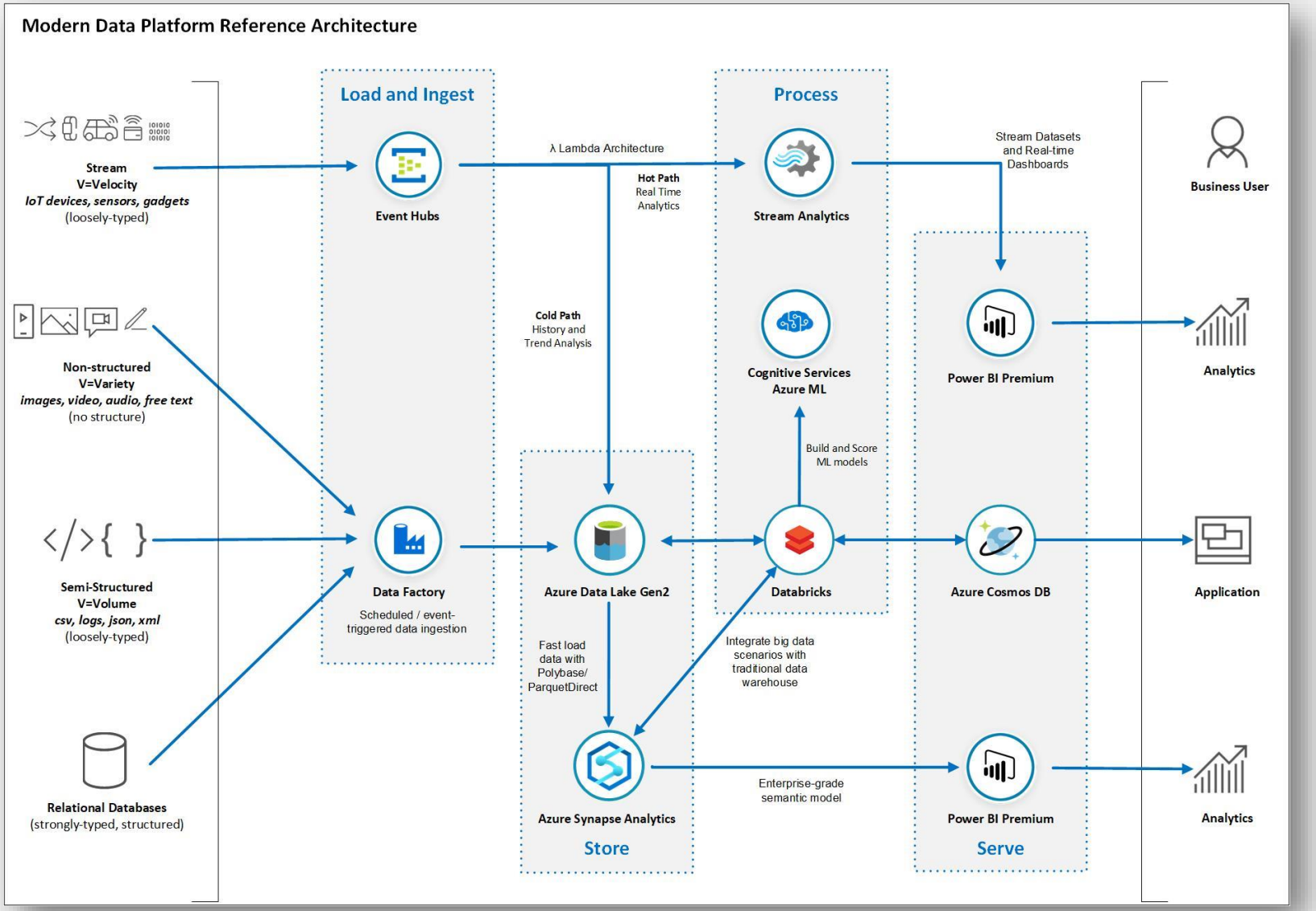
Reference Architecture: High-level functional view



Reference Architecture: Retail (Example)



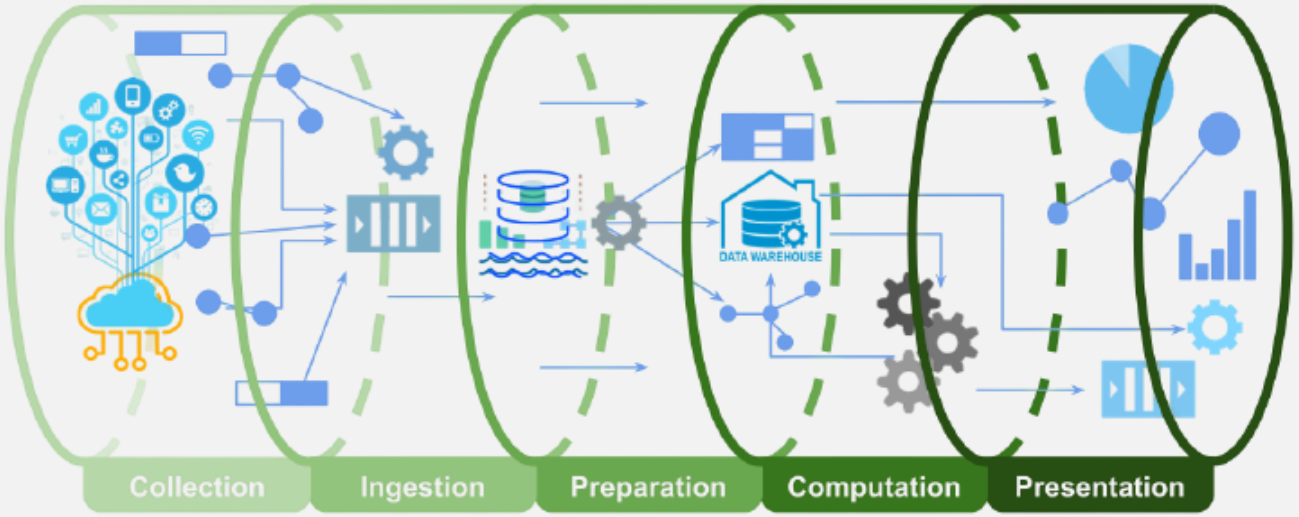
Reference Architecture: Modern Data Platform Azure



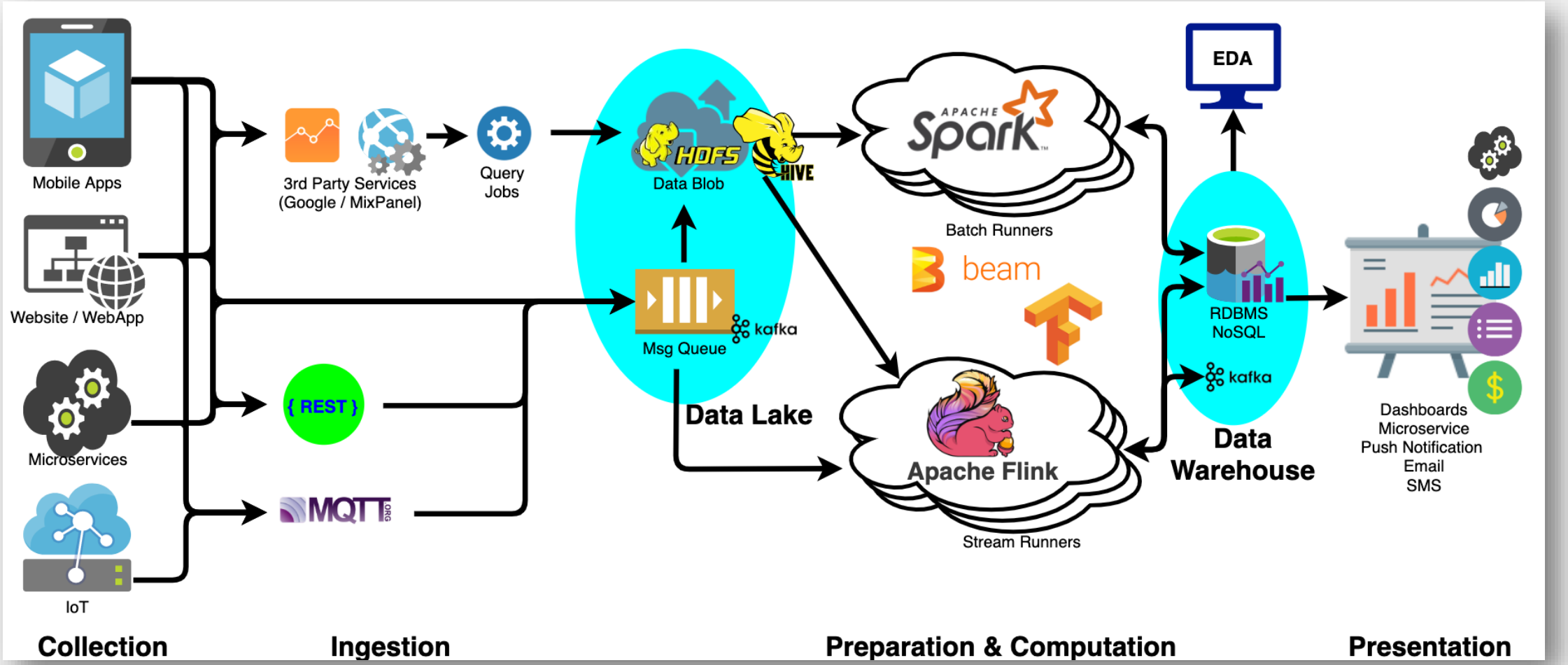
In summary,

Data Pipeline has five stages:

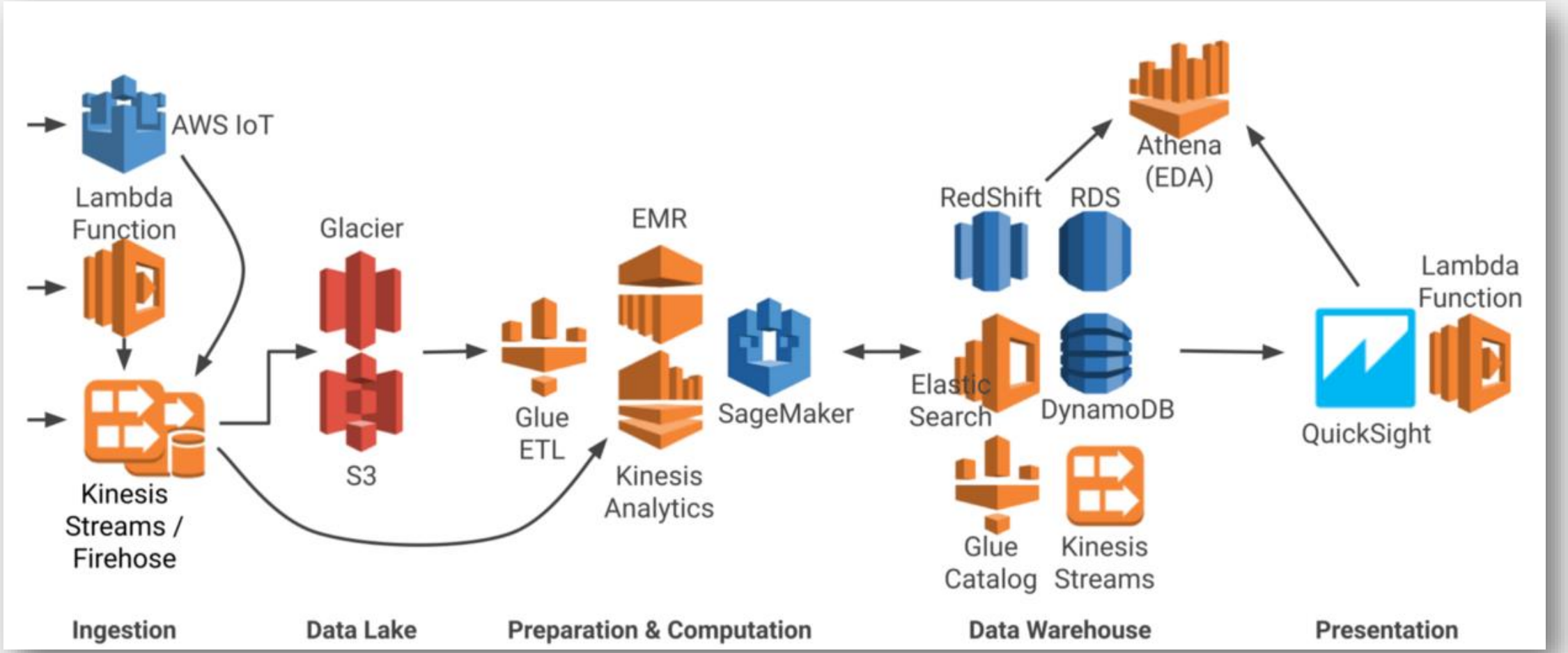
- Collection
- Ingestion
- Preparation
- Computation
- Presentation



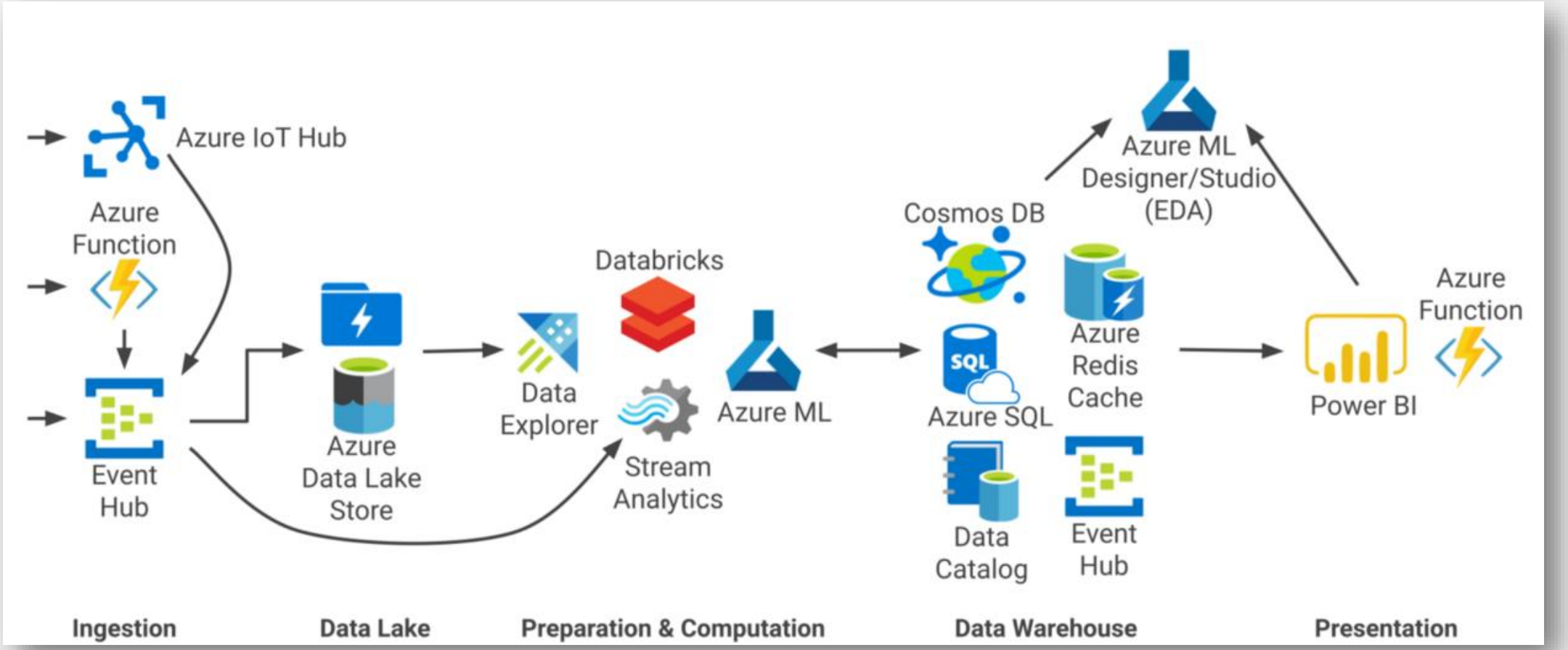
Big Data Architecture: Open Source Technologies



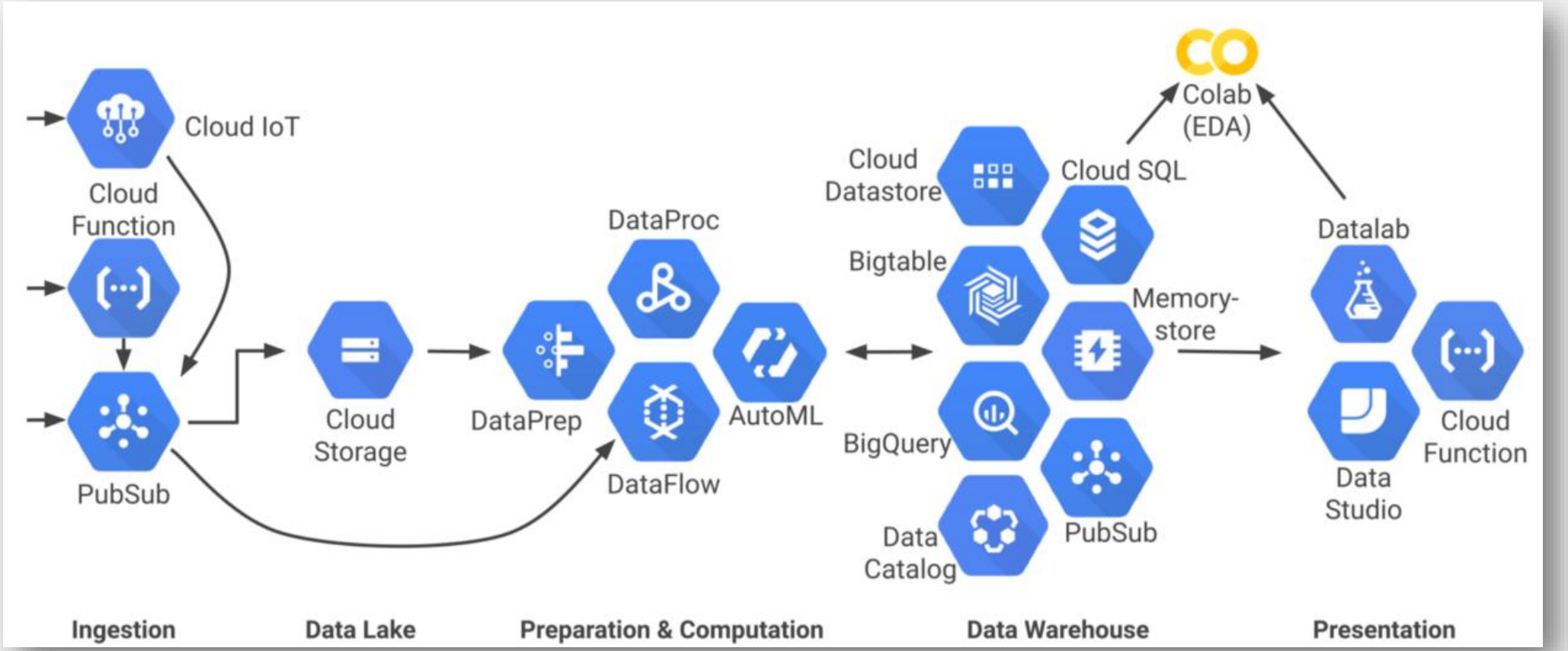
Big Data Architecture: Amazon Web Services



Big Data Architecture: Microsoft Azure



Big Data Architecture: Google Cloud Platform



Case Study on Big Data Architecture

Background

A multi-national company ABC provides financial services including investment services and insurance services using the online platform for financial products transactions. ABC has millions of clients worldwide to use their services and do online transactions. ABC has thousands of staffs to maintain the platform, serve the clients requests, collect and use the data to generate insights. However, there are no efficient systems to ingest the data and transform them into meaningful insights. The profit has dropped among this year and ABC wants to investigate the reasons from data point of view.

Case Study on Big Data Architecture

Needs and Requirements

ABC has requested to build a data warehouse and data streaming **pipeline** platform to utilize the data. In addition, ABC wants to use the data to analyze customer buying behavior trend and recommend them with personalized products.

Case Study on Big Data Architecture

Data Sources

After consolidating the needs and every kinds of data necessary, we have collected below three types of data:

1. Traditional enterprise data from operational systems related to customer touch points such as:

❑ Call centres.

❑ Branches/Brokerage units.

❑ Credit cards.

❑ Volatility measures that impact the clients' portfolios.

Case Study on Big Data Architecture

Data Sources

2. Financial business forecasts from various sources such as:

❑ Industry data.

❑ Trading data.

❑ Alerts about events (news, blogs, Twitter and other messaging feeds).

Case Study on Big Data Architecture

Data Sources

3. Other sources of data such as:

❑ Advertising response data.

❑ Social media data.

As far as we know, the data is a category of **structured**, **unstructured** and **semi-structured** data which increased the difficulty of building the big data system.

Case Study on Big Data Architecture

Key Questions To Think Through

1. What is the business challenge?
2. What is the need for data transactions: real-time vs. batch processing vs. transaction processing?
3. What is the data ingestion / storage challenge?
4. Among big data architecture, what tools and languages will be used?

Let’s keep these questions in mind and go through them one by one.

Case Study on Big Data Architecture

What is the business challenge?

ABC is facing the challenge of connecting efficiently with their customers without the prescriptive analytics platform to visualize their data and provide customer insights in terms of customer buying behaviors and customer engagement.

ABC also doesn't have a good data ingestion / storage platform and architecture design to make use of their data. The inability of data modeling / analytics / architecture leads to the potential loss of customer satisfaction and further loss of their customers from business perspective.

Case Study on Big Data Architecture

What is the business challenge?

In summary, there are three main challenges:

1. Understand the transactions of customer data, financial data and other kinds of data across platforms, design the big data storage techniques (data formats, data store types and storage tools)

2. Determine data pipeline architecture in terms of real-time vs batch processing, tools and language selection

    i. Real-time vs batch processing

    ii. Tools and language selection

3. Build a machine learning platform to predict:

    i. How location impacts customer behavior in order to provide accurate financial products offering

    ii. Customer engagement based on their preferences to recommend more attractive business selling

Case Study on Big Data Architecture

What is the need for data transaction: real-time vs. batch processing vs. transaction processing?

To determine which method is the best, we have to understand what each means and best used case.

- ✓ **Real-time** processing, or streaming processing, means the data can be processed in milliseconds.
- ✓ **Batch processing** is collecting all data at specific time and process all at once in a regular manner.
- ✓ **Transactional processing**, or data store processing, means once data has been processed by streaming or batch computations, it needs to be stored in a way that can be quickly accessed by a data scientist.

Case Study on Big Data Architecture

What is the need for data transaction: real-time vs. batch processing vs. transaction processing?

Scenario: Given that Twitter data can be used to find reasons on customer dissatisfaction, the DS team wants to build a pipeline for sentiment analytics.

What type of processing should they choose?

Case Study on Big Data Architecture

What is the data ingestion / storage challenge?

The data described above comes from different sources, e.g., from databases, log files, online financial applications, social media networks and offline operational systems. These data are collected in databases, local disks, and cloud file systems (such as AWS S3) in structured and unstructured formats. We need to collect these data and integrate it into a data lake, transform and deliver it into a data warehouse.

What storage type to choose?

Case Study on Big Data Architecture

What is the data ingestion / storage challenge?

Scenario 1: User profiles storage

Every user has unique profile data such as user id, name, gender, and other identification attributes, as well as preferences such as language, time zone, which products the user has access to, and so on. In this case, we assume that each user has a unique key, and assume that some profiles data are optional to fill in, such as user age.

The optimal storage will be **key-value store** or column family store, providing the capability of in-memory processing. However, if the data is not growing, RDBMS could be an option but not optimal.

Case Study on Big Data Architecture

Among big data architecture, what tools and languages will be used?

There are thousands of tools in the market to evaluate and compare. Can you make a Reference Architecture with Open Source Ecosystem components?

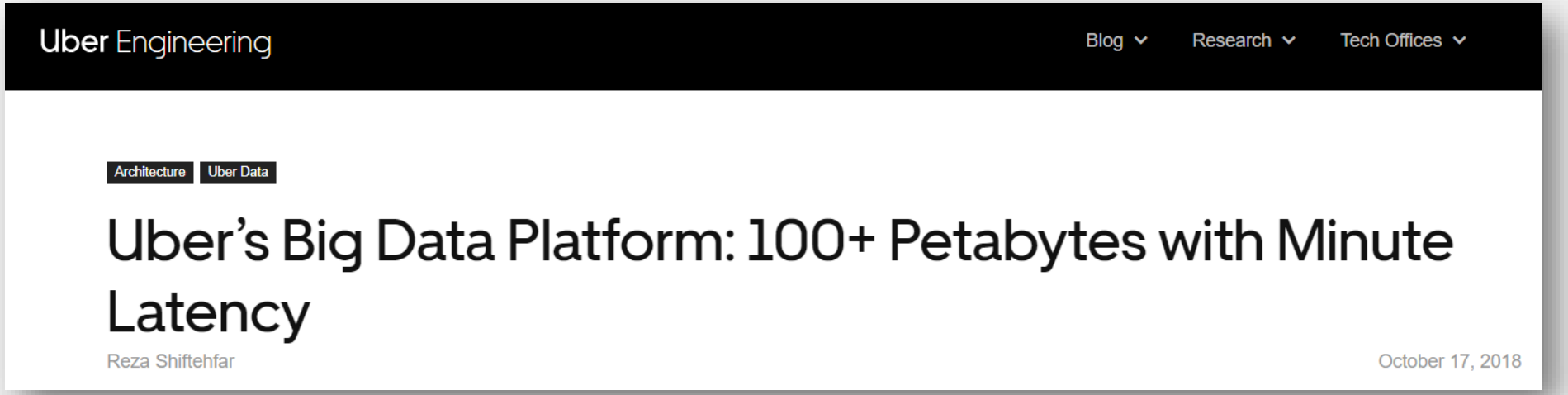
Just have a look how many ecosystem components are there?

http://bigdata.andreamostosi.name/

Case Study (Uber)

Transformation Journey towards Big Data Platform.

Please do read this article.



https://eng.uber.com/uber-big-data-platform/

Installation of Hadoop Distribution

Modes in which Hadoop run

1) Standalone Mode

By default, Hadoop is configured to run in a Standalone Mode, non-distributed mode, as a single Java process. This is useful for debugging. This does not offer you a true distributed environment. The usage of this mode is very limited and it can be only used for experimentation.

2) Pseudo-Distributed Mode

Hadoop is run on a single node in a pseudo(false) distributed mode, Just like the Standalone mode. The difference is that each Hadoop daemon runs in a separate Java process in Pseudo-Distributed Mode. Whereas in Local mode each Hadoop daemon runs as a single Java process. Again the usage of this mode is very limited and it can be only used for experimentation.

3) Fully-Distributed Mode

In the fully-distributed mode, all daemons are executed in separate nodes forming a multi-node cluster. This setup offers true distributed computing capability and offers built-in reliability, scalability and Fault Tolerance.

Installation of Hadoop Distribution

- HDP (Hortonworks Data Platform) on your Machine (Using Virtual Box Sandbox)
- CM (Cloudera Manager) on Cloud in Multi-node Cluster (Using GCP)
- Pseudo Distribution Mode

Ref: http://arif.works/bdahive

What is Hive

Hive

- ✓ SQL like Querying tool to query the data stored in HDFS and Filesytems that integrate with Hadoop
- ✓ Developed by Facebook and later on taken by Apache Foundation
- ✓ It process the structured data that can be stored into Tables
- ✓ Efficient for Batch processing
- ✓ Hive is a lens between MapReduce and HDFS
- ✓ It provide is various storage file formats like Parquet, Sequence file, ORC file, text file with significant compression

What Hive is not

- ✓ Hive is not a Database. It just points to the Data lying in HDFS
- ✓ Hive is not a tool for OLTP. It is closer being as OLAP tool
- ✓ It doesn't provide row level Insert, Update and Delete Operations
- ✓ Not used where fast response time is required as in RDMS. Rather used where high latency is acceptable with Batch processing
- ✓ Does not support Unstructured data like Audio, Video and Pictures

Hive vs SQL

Hive

- ✓ Hive is not a Database. It does not store any physical data
- ✓ Built on write-once and read-many concept
- ✓ Best suited for OLAP system
- ✓ Easily and cost effective scalable

SQL

- ✓ It is a pure Database. It does store physical data
- ✓ Built on write-many and read-many concept
- ✓ Best suited for OLTP system
- ✓ Not Easily scalable

Hive Architecture