VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

DATA ENGINEERING

Hadoop, Spark, Hive

Mentor: Dr.Phan Trọng Nhân

Group 6:

Phan Phước Minh - 2010418
Đinh Thanh Phong - 2270243
Trần Thọ Nhân - 1910405

March, 2023

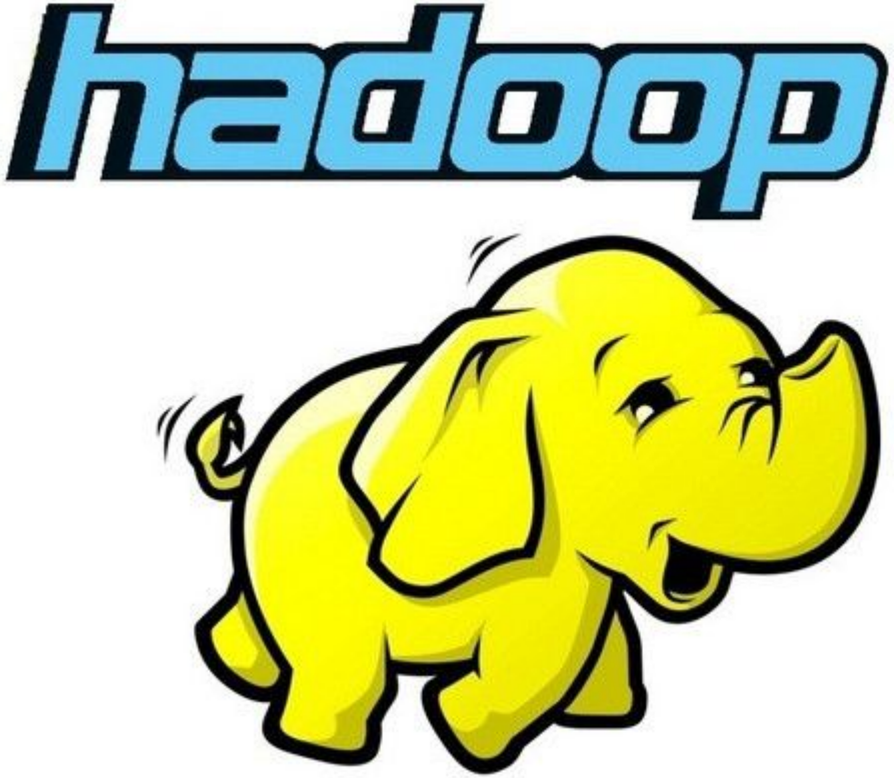
Agenda

1. Hadoop
2. Spark
3. Hive

Apache Hadoop

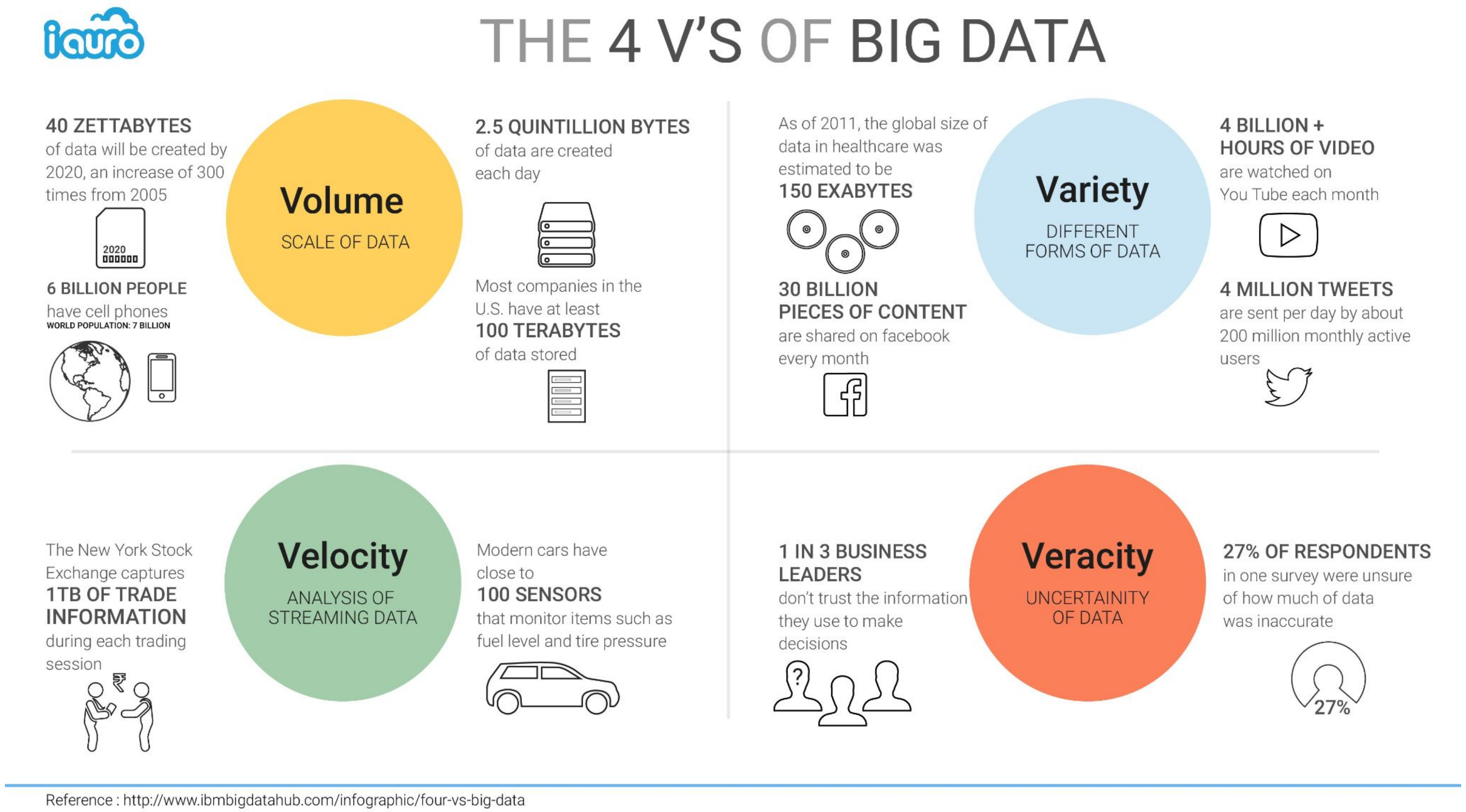
What is Hadoop?

Hadoop is an open-source framework used to process enormous data sets



Apache Hadoop

Why Hadoop?



iauro

THE 4 V'S OF BIG DATA

Volume
SCALE OF DATA

40 ZETTABYTES of data will be created by 2020, an increase of 300 times from 2005

6 BILLION PEOPLE have cell phones
WORLD POPULATION: 7 BILLION

2.5 QUINTILLION BYTES of data are created each day

Most companies in the U.S. have at least **100 TERABYTES** of data stored

Variety
DIFFERENT FORMS OF DATA

As of 2011, the global size of data in healthcare was estimated to be **150 EXABYTES**

30 BILLION PIECES OF CONTENT are shared on facebook every month

4 BILLION + HOURS OF VIDEO are watched on You Tube each month

4 MILLION TWEETS are sent per day by about 200 million monthly active users

Velocity
ANALYSIS OF STREAMING DATA

The New York Stock Exchange captures **1TB OF TRADE INFORMATION** during each trading session

Modern cars have close to **100 SENSORS** that monitor items such as fuel level and tire pressure

Veracity
UNCERTAINITY OF DATA

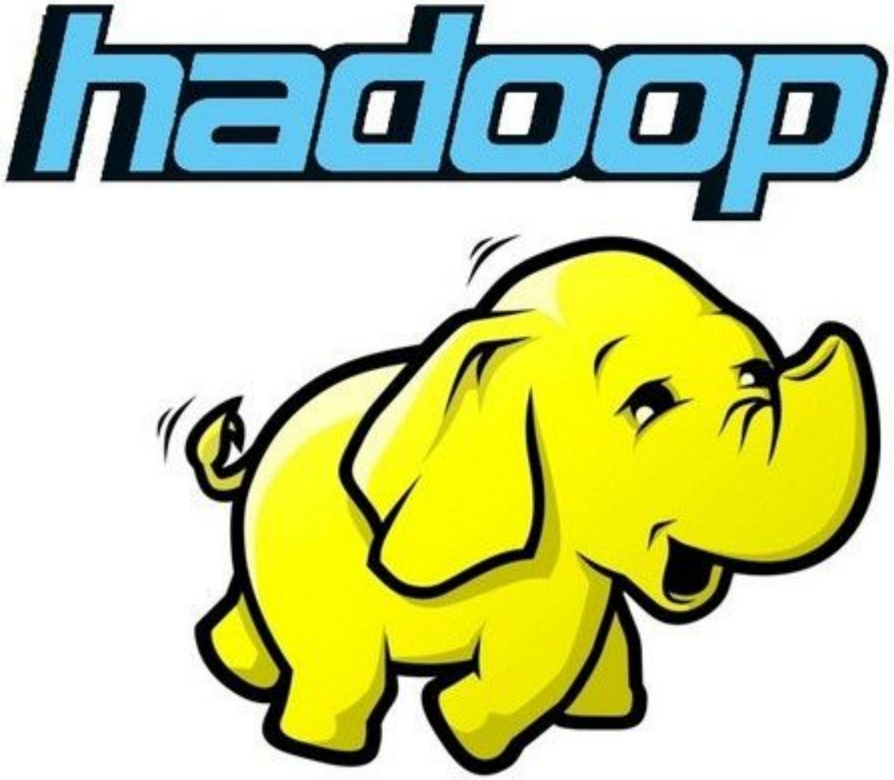
1 IN 3 BUSINESS LEADERS don't trust the information they use to make decisions

27% OF RESPONDENTS in one survey were unsure of how much of data was inaccurate

Reference : http://www.ibmbigdatahub.com/infographic/four-vs-big-data

Apache Hadoop

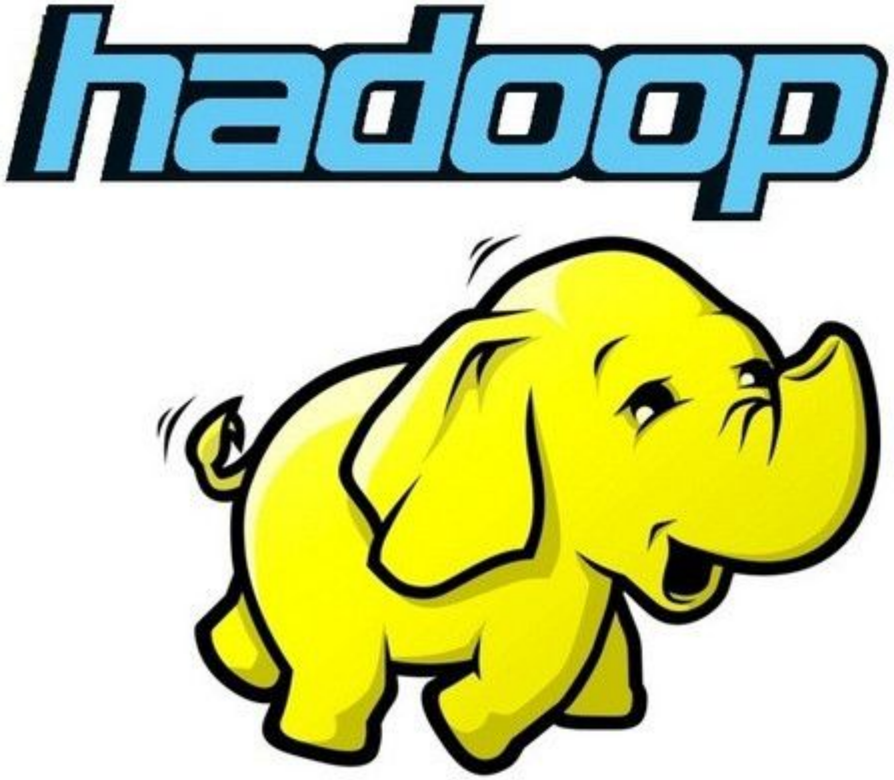
What is Hadoop?



- Set of open-source programs and procedures
- Used for processing large amount of data
- Servers run applications on cluster
- Handle parallel jobs or processes

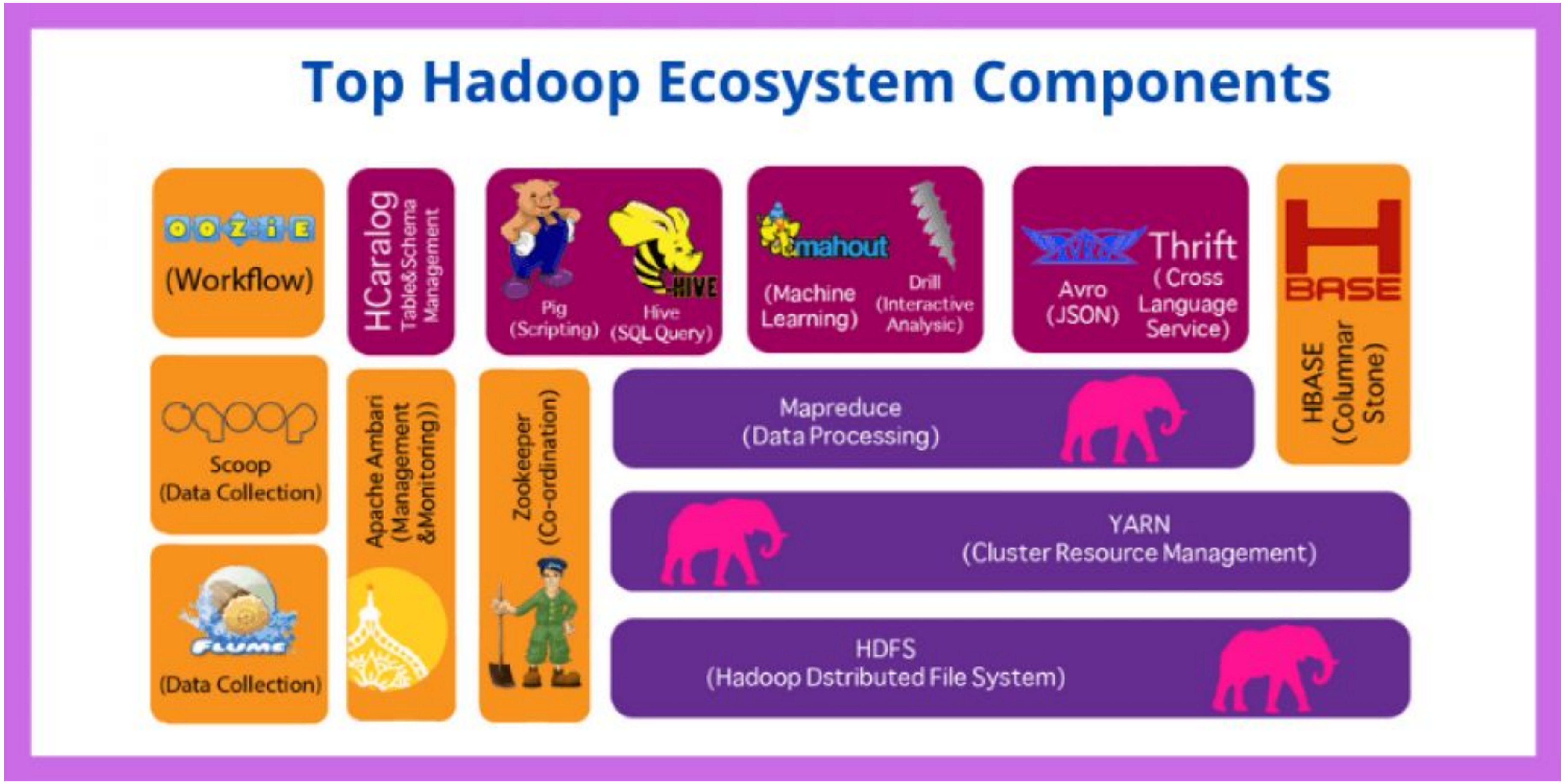
Apache Hadoop

Hadoop History



- 1999: Apache software foundation established
- 2002: Nutch web search engine created
- 2006: Nutch was divided into Web crawler, distributed systems
- 2008: Yahoo released Hadoop as an open-source project

Apache Hadoop



Hadoop Architecture

Apache Hadoop

Main component

MapReduce	HDFS	YARN
● Hadoop’s processing unit ● Processes Big Data by splitting the data into smaller units ● First method to query data stored in HDFS	● Hadoop Distributed File System ● Handles and stores large data ● Scales a single Hadoop cluster into as much as thousand cluster	● Yet Another Resource Negotiator acronym ● Prepare Hadoop for batch, stream, interactive and graph processing

Apache Hadoop

MapReduce Algorithm

- Programming model used Hadoop for Big Data processing
- Processing technique for distributed computing
- Consist of a Map task and a Reduce task
- Can be implemented in many languages: Java, Python, C/C++

Apache Hadoop

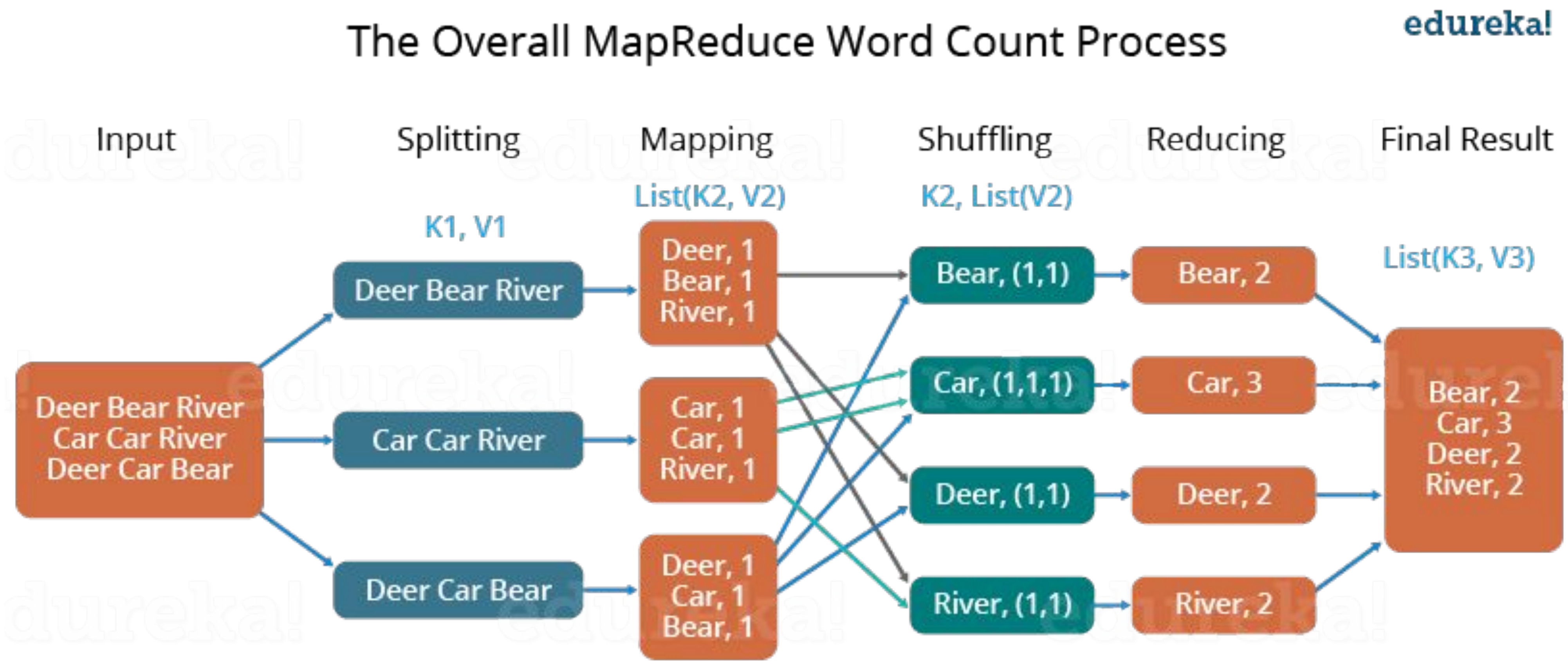
MapReduce Algorithm

- Specify two functions:
  map $(k_1, v_1) \rightarrow [\langle k_2, v_2 \rangle]$

  reduce $(k_2, [v_2]) \rightarrow [\langle k_3, v_3 \rangle]$
  (All the values with the same key are sent to same reducer)

- Execution framework handles everything else

Apache Hadoop



Apache Hadoop

MapReduce “runtime”

- Handles scheduling

  Assign workers to map and reduce tasks
- Handles “data distribution”

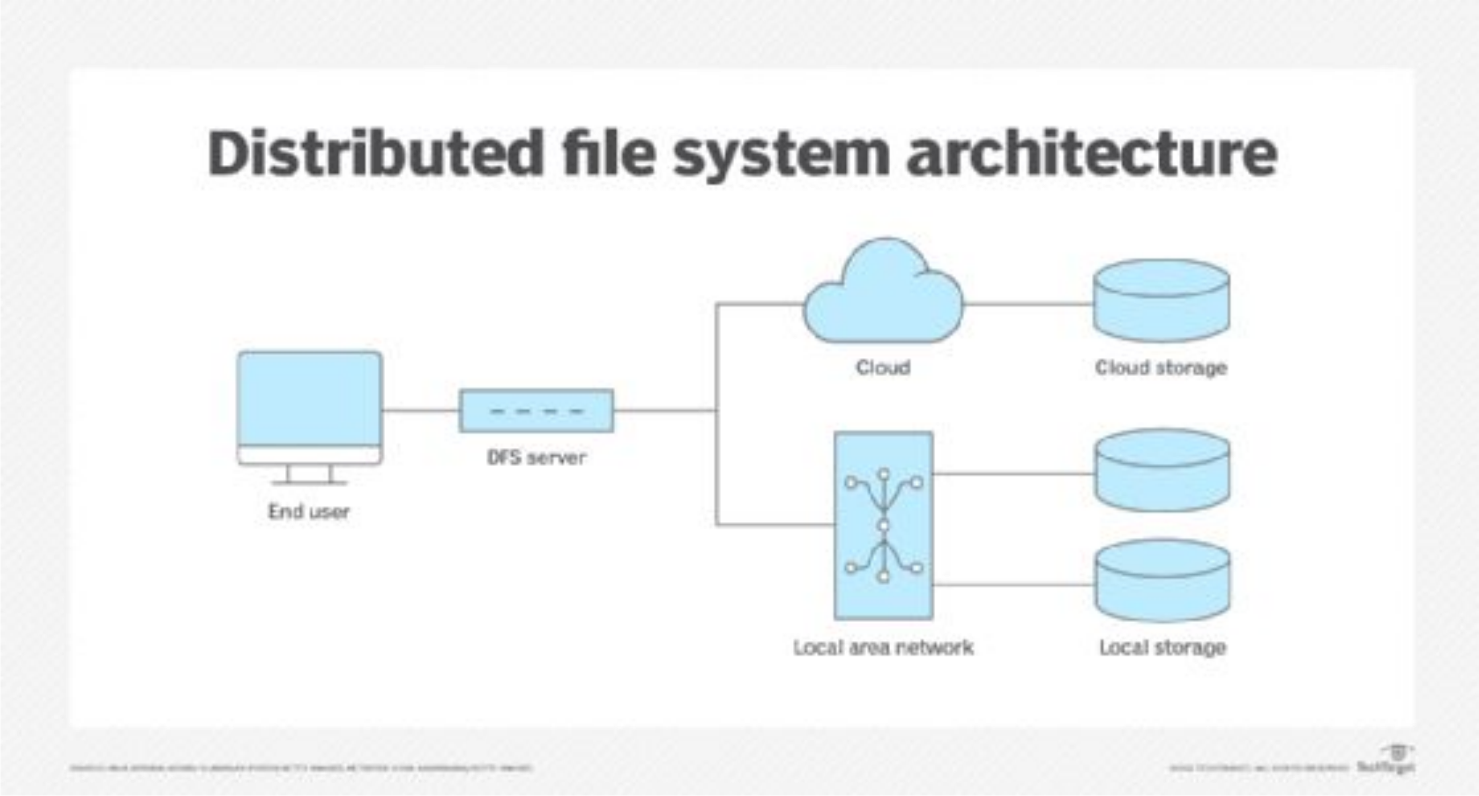
  Move processes to data
- Handles synchronization

  Gathers, sorts, and shuffles intermediate data
- Handles errors and faults

  Detects workers failures and restart
- Happen on top of distributed file system

Apache Hadoop

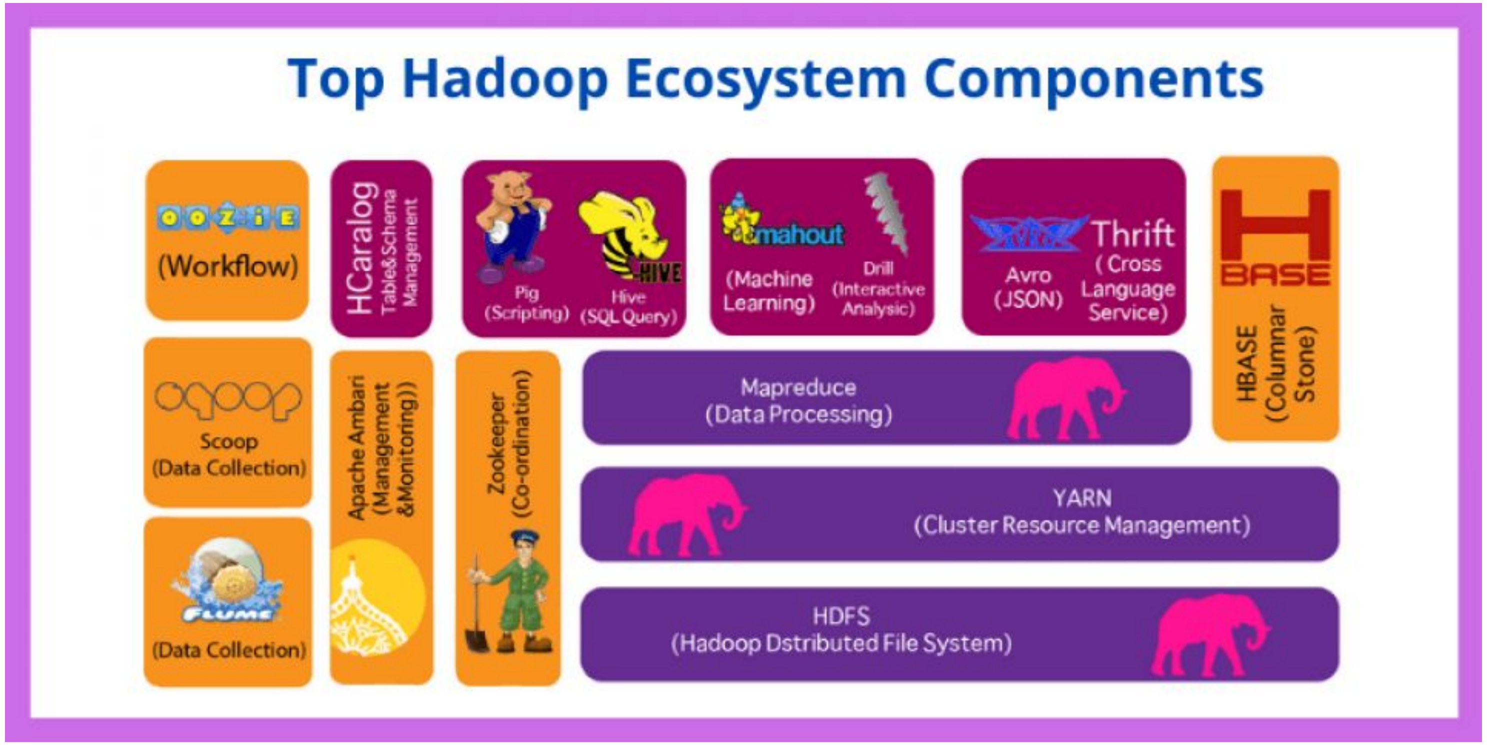
Hadoop Distributed File System



Apache Hadoop

Hadoop Distributed File System

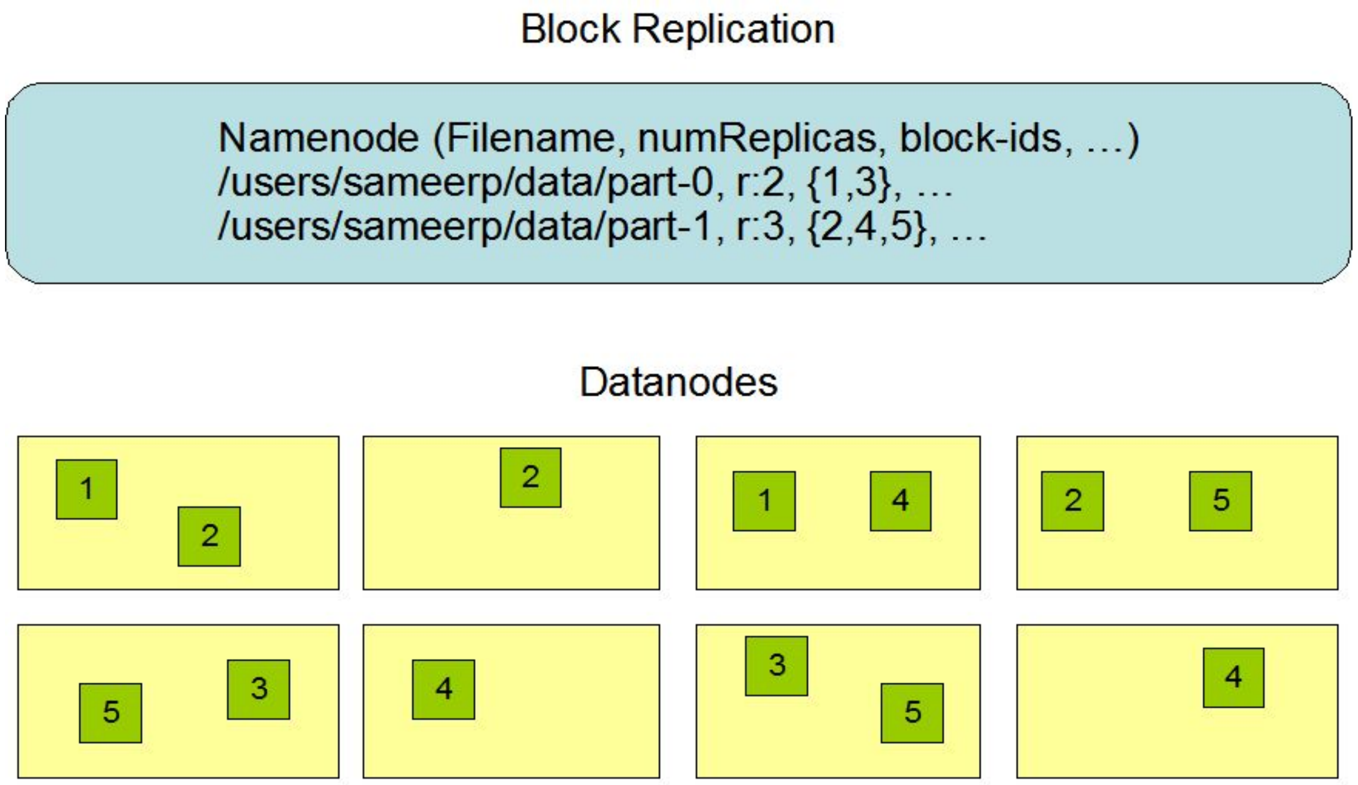
- Acronym for Hadoop Distributed File System
- Storage layer of Hadoop



Apache Hadoop

Hadoop Distributed File System

- Split the files into blocks, create replicas, and store on different machine



Block Replication

Namenode (Filename, numReplicas, block-ids, …)
/users/sameerp/data/part-0, r:2, {1,3}, …
/users/sameerp/data/part-1, r:3, {2,4,5}, …

Datanodes

Apache Hadoop

Hadoop Distributed File System

- Provides access to stream data
- HDFS using command line interface to interact with Hadoop

Apache Hadoop

HDFS key feature

- Cost efficient: hardware to storage is not expensive
- Large amount of data: up to petabytes data
- Replication: multiple copies of data
- Fault tolerant: work well when a node interrupt
- Scalable: easy to scale up
- Portable: easy to move across platforms

Apache Hadoop

HDFS nodes

- Node: single system responsible to store and process data
- Namenode/Primary node: Regulates files access to clients and maintains, manages, and assign task to secondary node
- Secondary namenode: Recovery metadata when namenode interrupt
- Datanode/Secondary node: Workers to handle task from namenode.
- Rack: collection of 40-50 datanodes using the same network switches.

Apache Hadoop

Rack awareness

- If a node interrupt, a datanode that closest to this in the same rack to handle jobs
- Improve performance and do replication
- Namenode keep rack ID information

Apache Hadoop

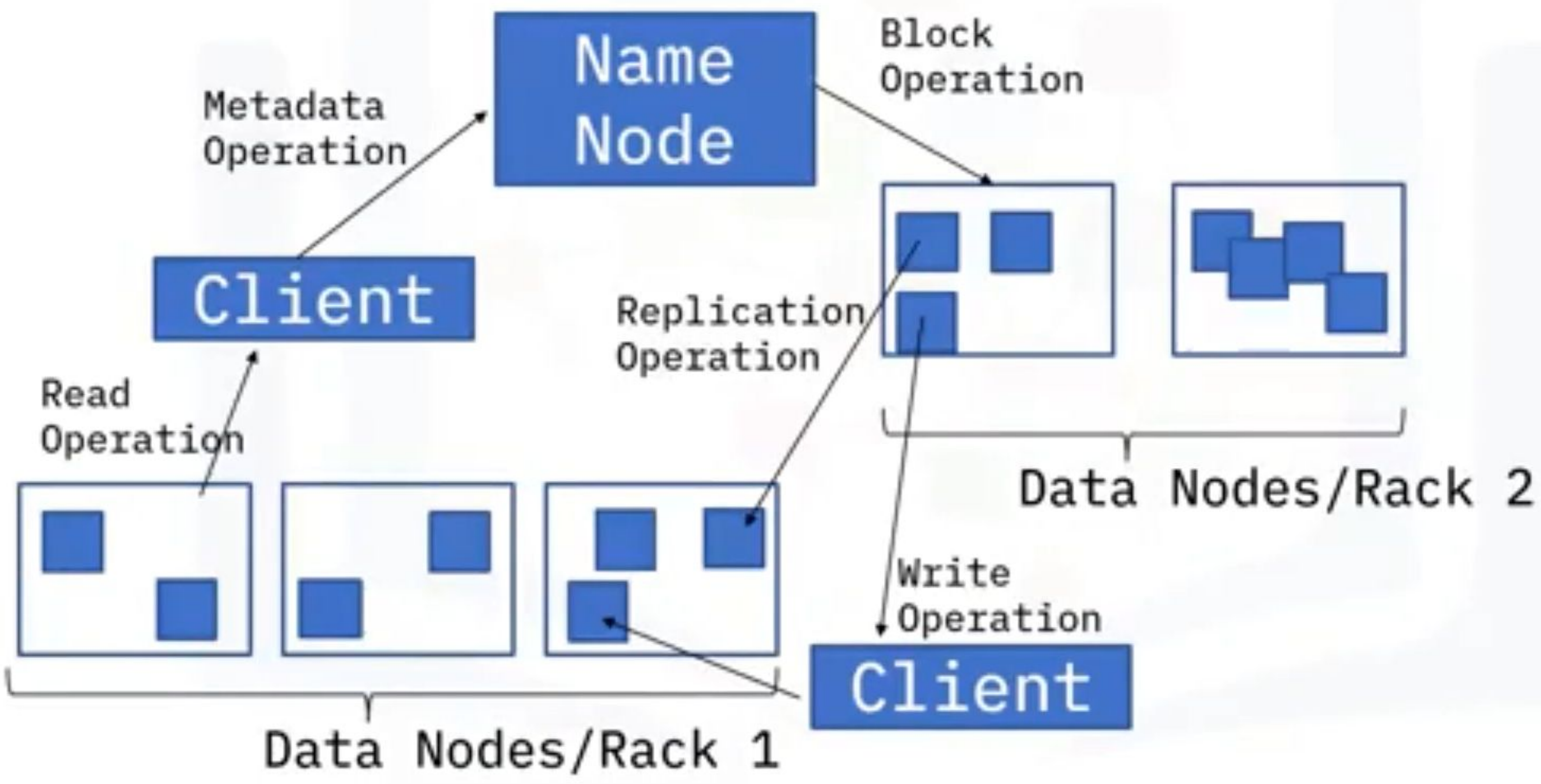
HDFS Read and Write

HDFS allow user write once read many operations

Read	Write
● Client send request to Namenode to get location of data nodes containing blocks ● Read the files closest to the datanode	● Namenode makes sure the file doesn’t exist ● If the files exists, IO exception messages sent ● If file doesn’t exist, give access to write file

Apache Hadoop

HDFS architecture



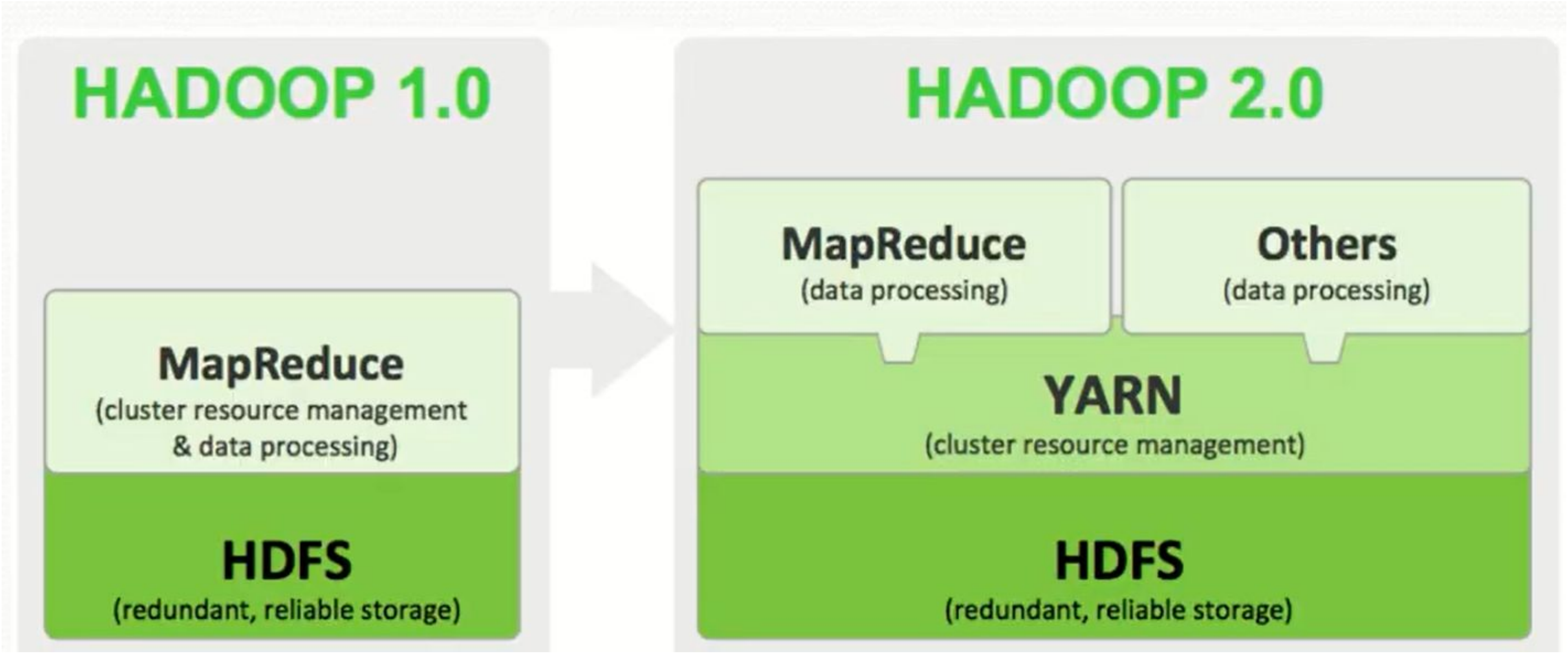
Apache Hadoop

YARN

- Framework support distributed system
- Roles:
  - Resource Management
  - Job Scheduler

Apache Hadoop

YARN - why is it?


HADOOP 1.0
MapReduce
(cluster resource management & data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MapReduce
(data processing)
Others
(data processing)
YARN
(cluster resource management)
HDFS
(redundant, reliable storage)

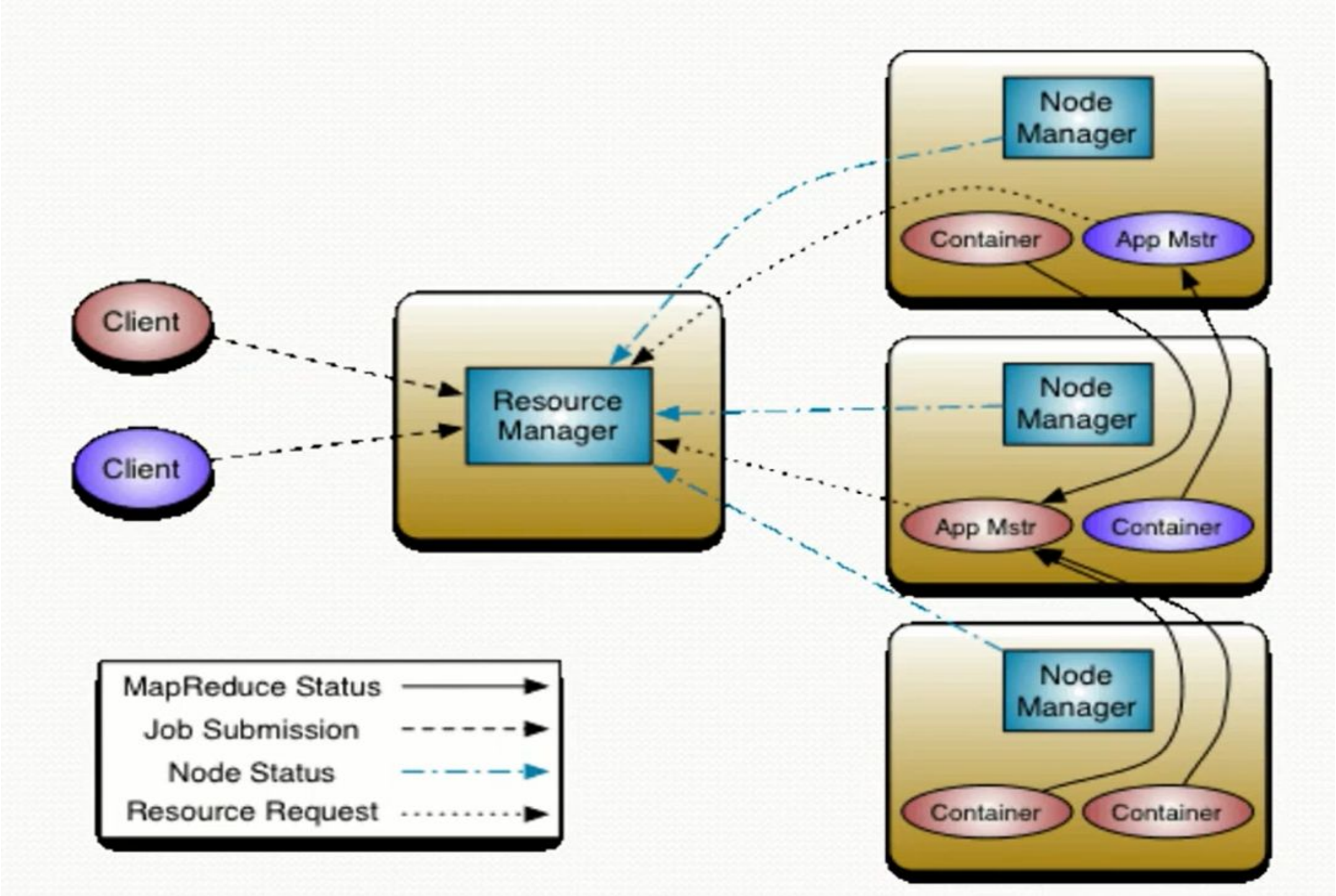
Apache Hadoop

YARN component

- Resource Manager (RM): Manage all resource for clusters
- Node manager (NN): Manage resources for Node, running job on container of Node, init container
- Container: Handle logic and compute task
- Application Master: Receive task to manage jobs

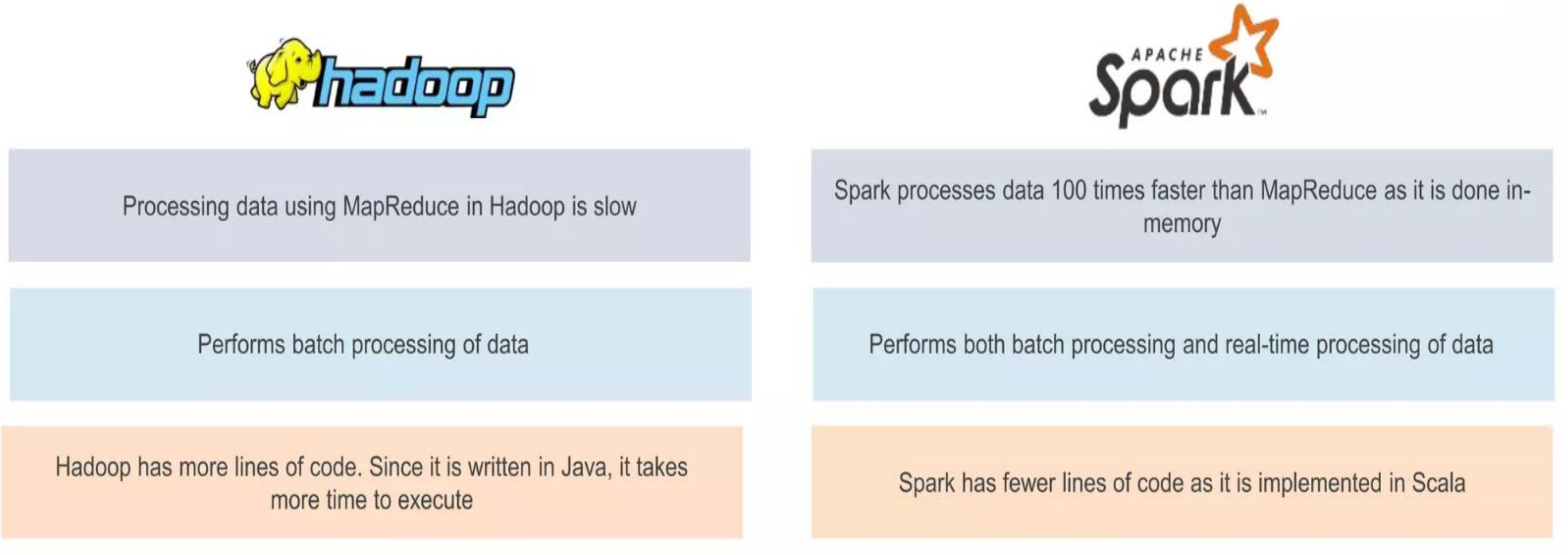
Apache Hadoop

YARN component



Apache Spark

Why do we need Spark?



hadoop	APACHE Spark
Processing data using MapReduce in Hadoop is slow	Spark processes data 100 times faster than MapReduce as it is done in-memory
Performs batch processing of data	Performs both batch processing and real-time processing of data
Hadoop has more lines of code. Since it is written in Java, it takes more time to execute	Spark has fewer lines of code as it is implemented in Scala

Apache Spark

What is Spark?

- Spark is a fast and general processing engine.
- It can run in Hadoop clusters through YARN or Spark's standalone mode.
- Spark can process data in HDFS, HBase, Cassandra, Hive, and any Hadoop InputFormat.

Apache Spark

What is Spark?

Apache Spark is an open-source data processing engine to store and process data in real-time across various clusters of computers using simple programming constructs

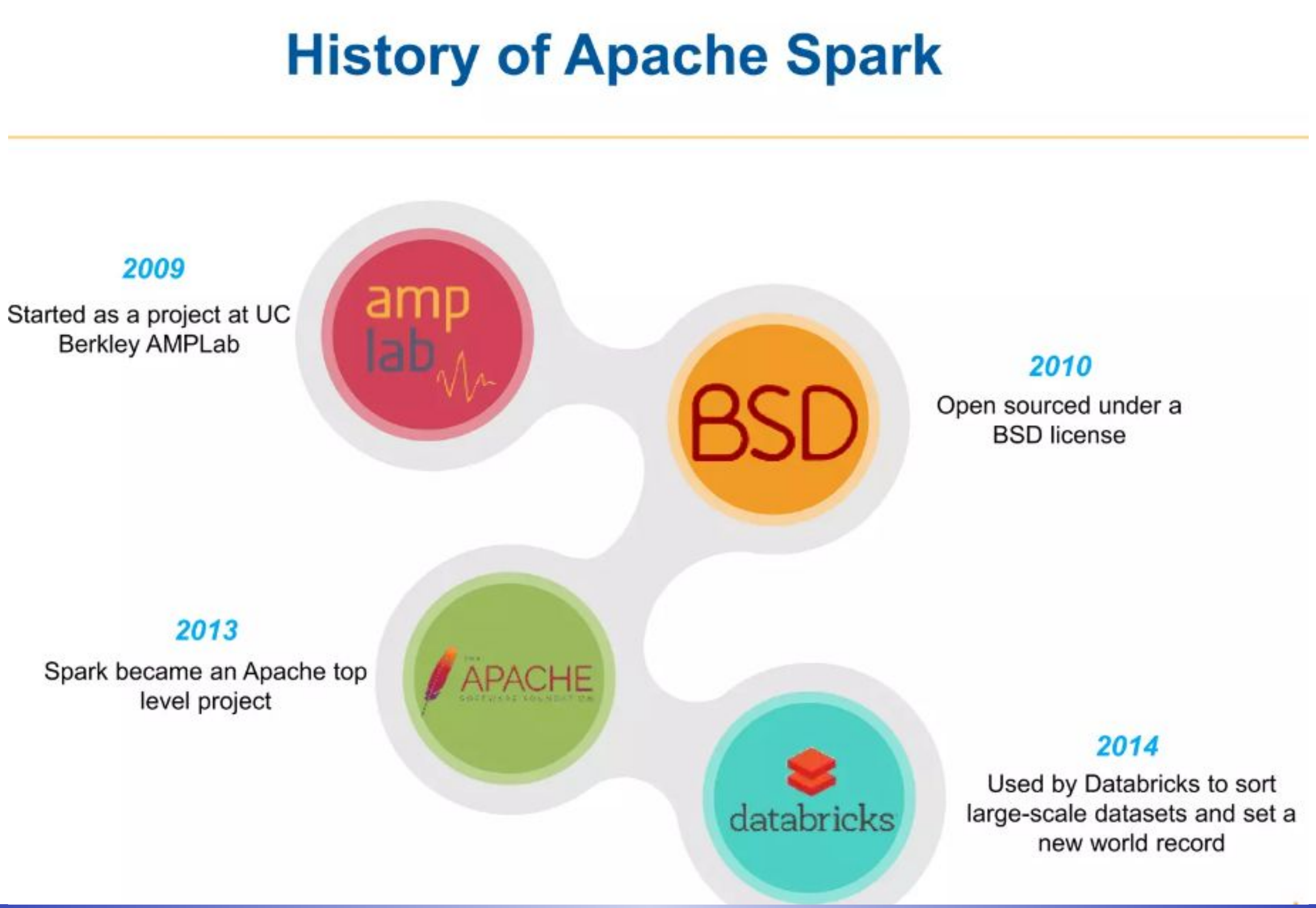
Support various programming languages

R
python
Java
Scala

Developers and data scientists incorporate Spark into their applications to rapidly query, analyze, and transform data at scale

Query
Analyze
Transform

Apache Spark



History of Apache Spark

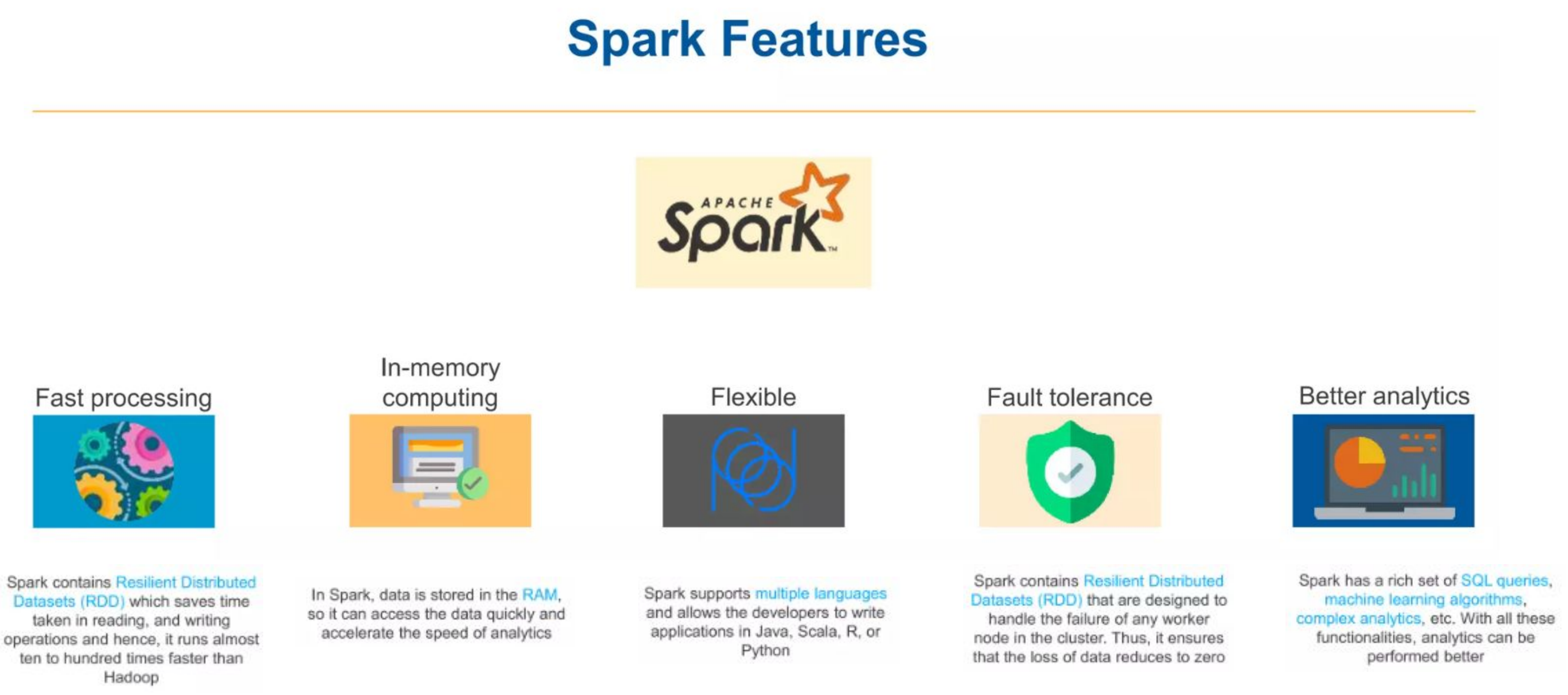
2009
Started as a project at UC Berkley AMPLab

2010
Open sourced under a BSD license

2013
Spark became an Apache top level project

2014
Used by Databricks to sort large-scale datasets and set a new world record

Apache Spark



Spark Features

Fast processing

Spark contains Resilient Distributed Datasets (RDD) which saves time taken in reading, and writing operations and hence, it runs almost ten to hundred times faster than Hadoop

In-memory computing

In Spark, data is stored in the RAM, so it can access the data quickly and accelerate the speed of analytics

Flexible

Spark supports multiple languages and allows the developers to write applications in Java, Scala, R, or Python

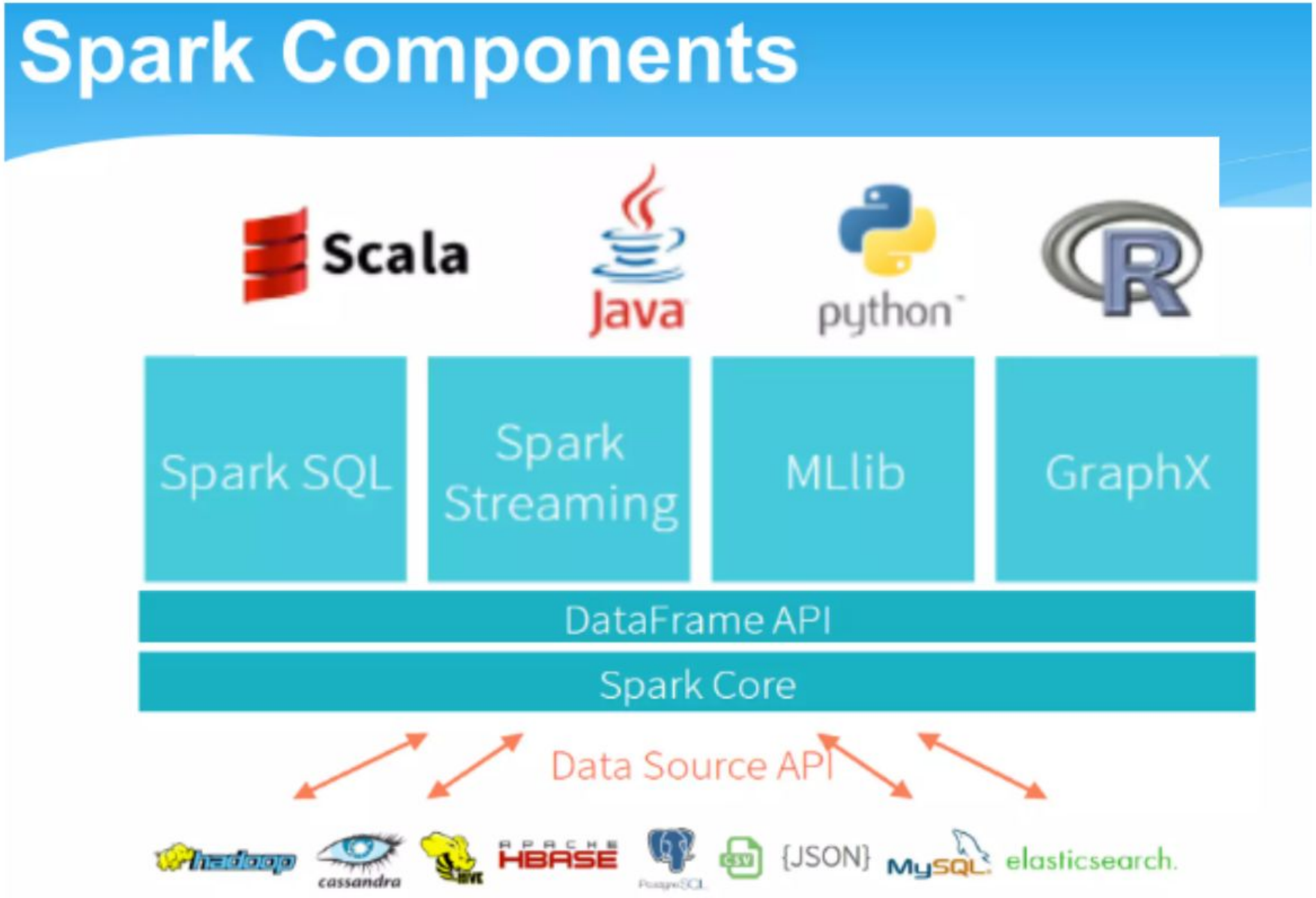
Fault tolerance

Spark contains Resilient Distributed Datasets (RDD) that are designed to handle the failure of any worker node in the cluster. Thus, it ensures that the loss of data reduces to zero

Better analytics

Spark has a rich set of SQL queries, machine learning algorithms, complex analytics, etc. With all these functionalities, analytics can be performed better

Apache Spark



Spark Components

Spark SQL

Spark Streaming

MLlib

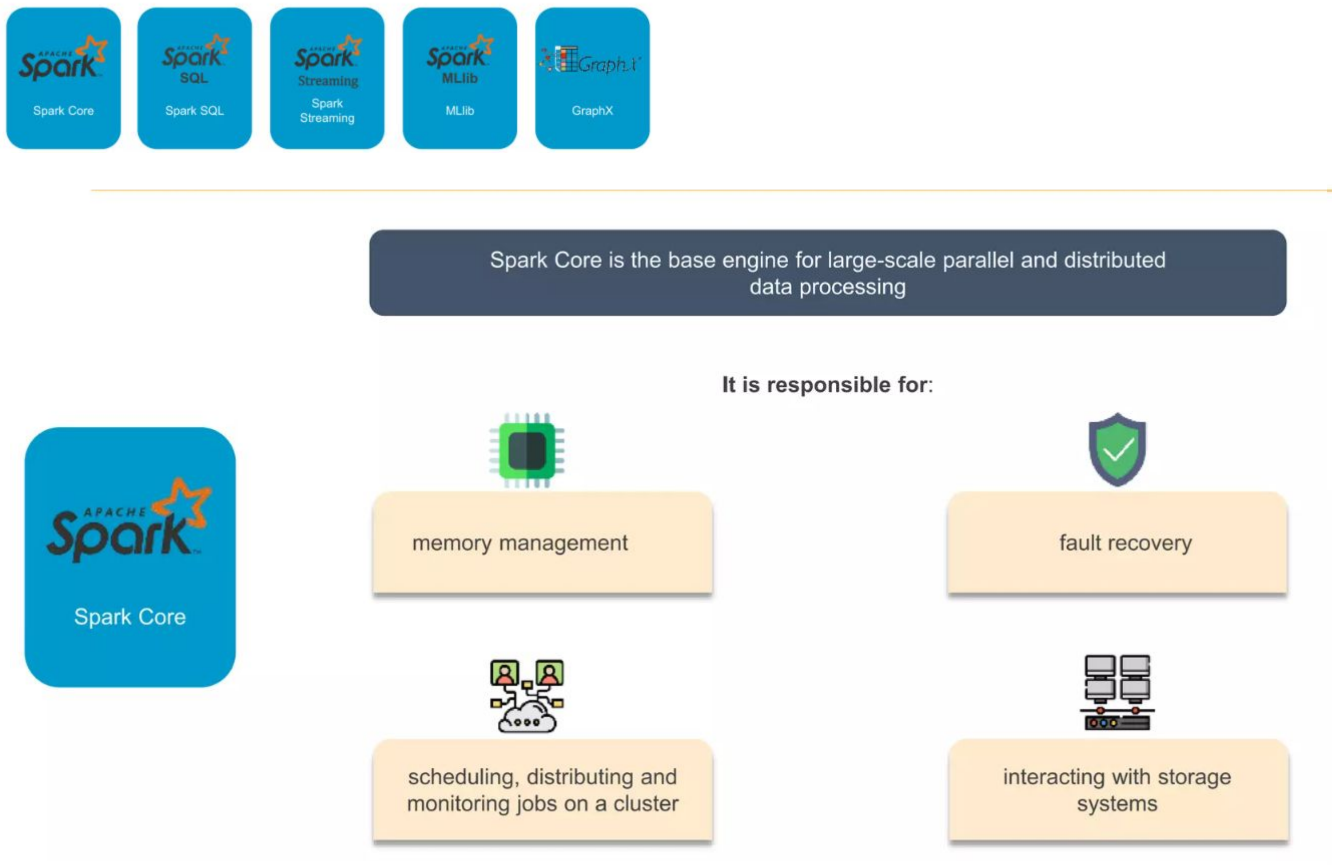
GraphX

DataFrame API

Spark Core

Data Source API

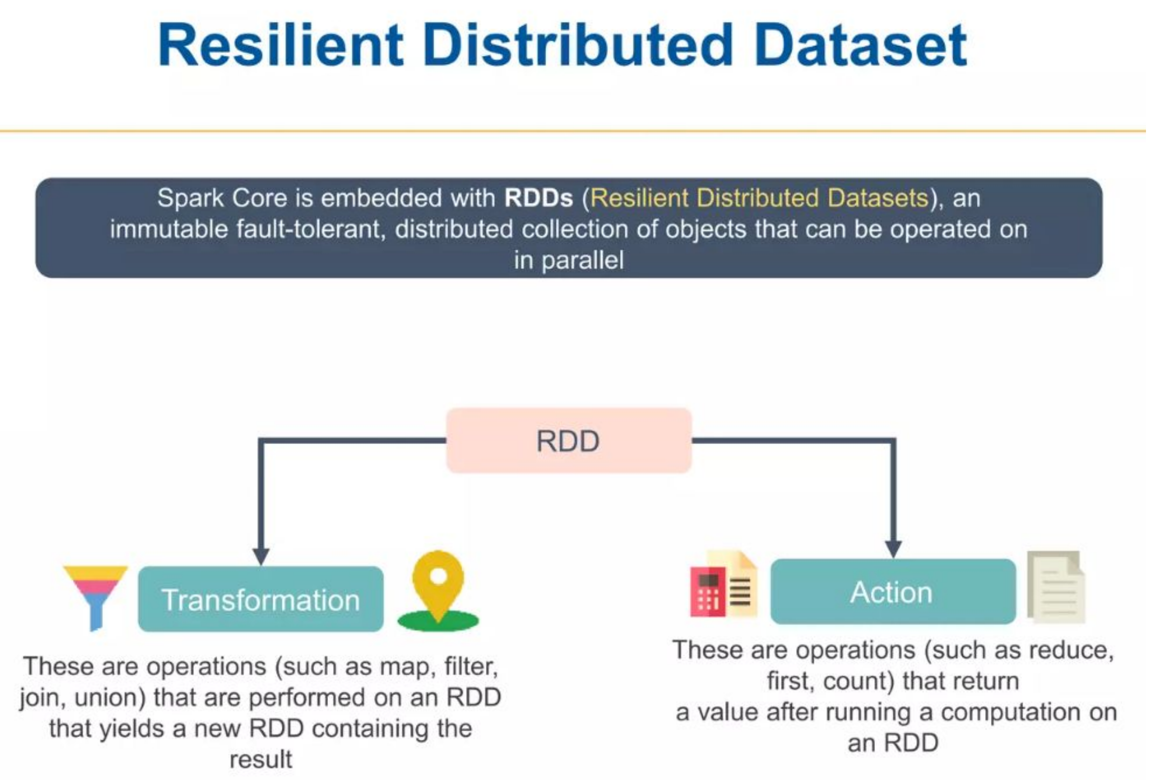
Apache Spark



Spark Core is the base engine for large-scale parallel and distributed data processing

It is responsible for:

- memory management
- fault recovery
- scheduling, distributing and monitoring jobs on a cluster
- interacting with storage systems



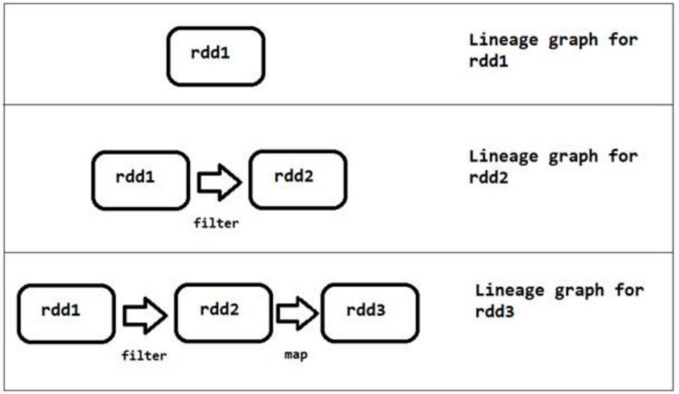
Resilient Distributed Dataset

Spark Core is embedded with **RDDs** (Resilient Distributed Datasets), an immutable fault-tolerant, distributed collection of objects that can be operated on in parallel

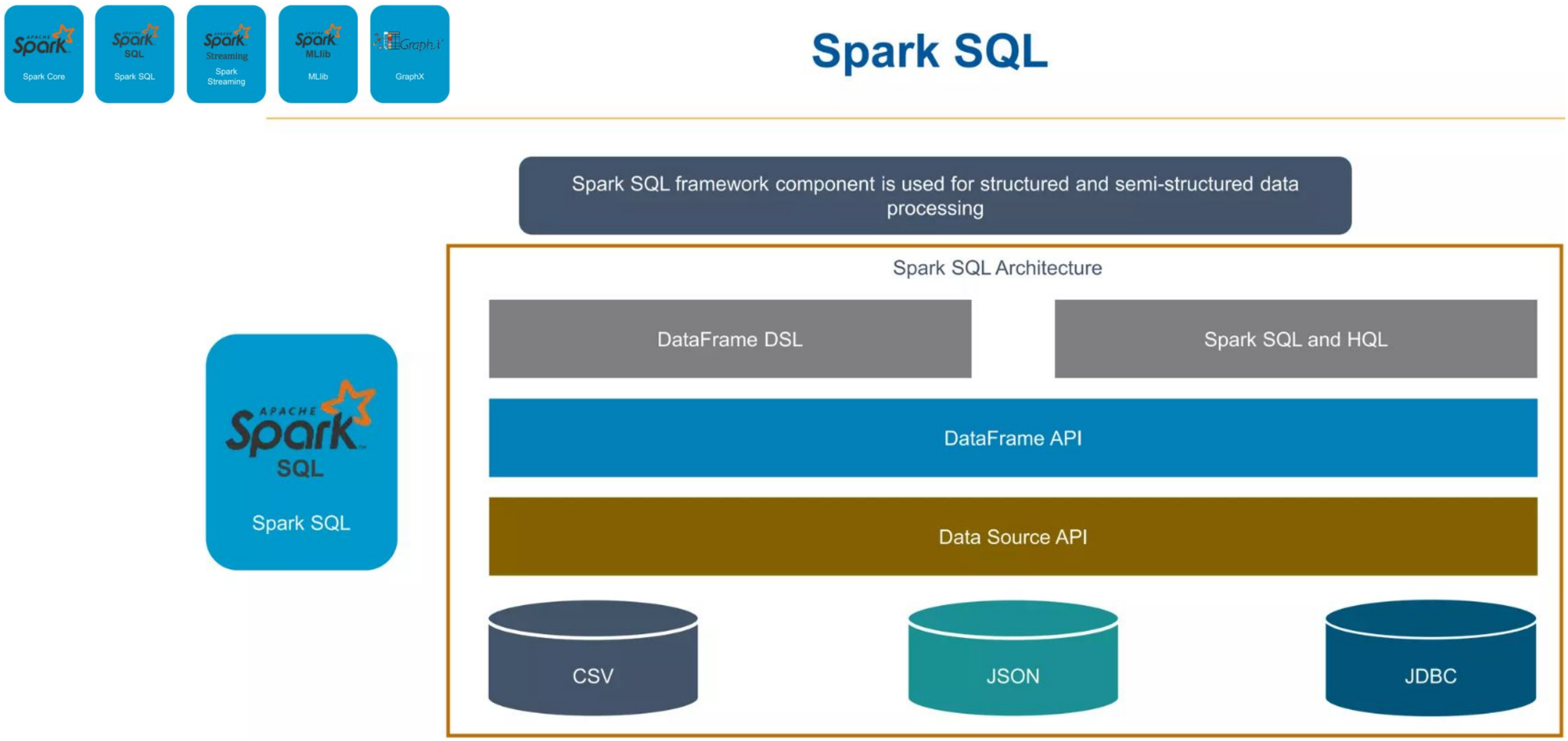
These are operations (such as map, filter, join, union) that are performed on an RDD that yields a new RDD containing the result

These are operations (such as reduce, first, count) that return a value after running a computation on an RDD

Lazy evaluation



Apache Spark

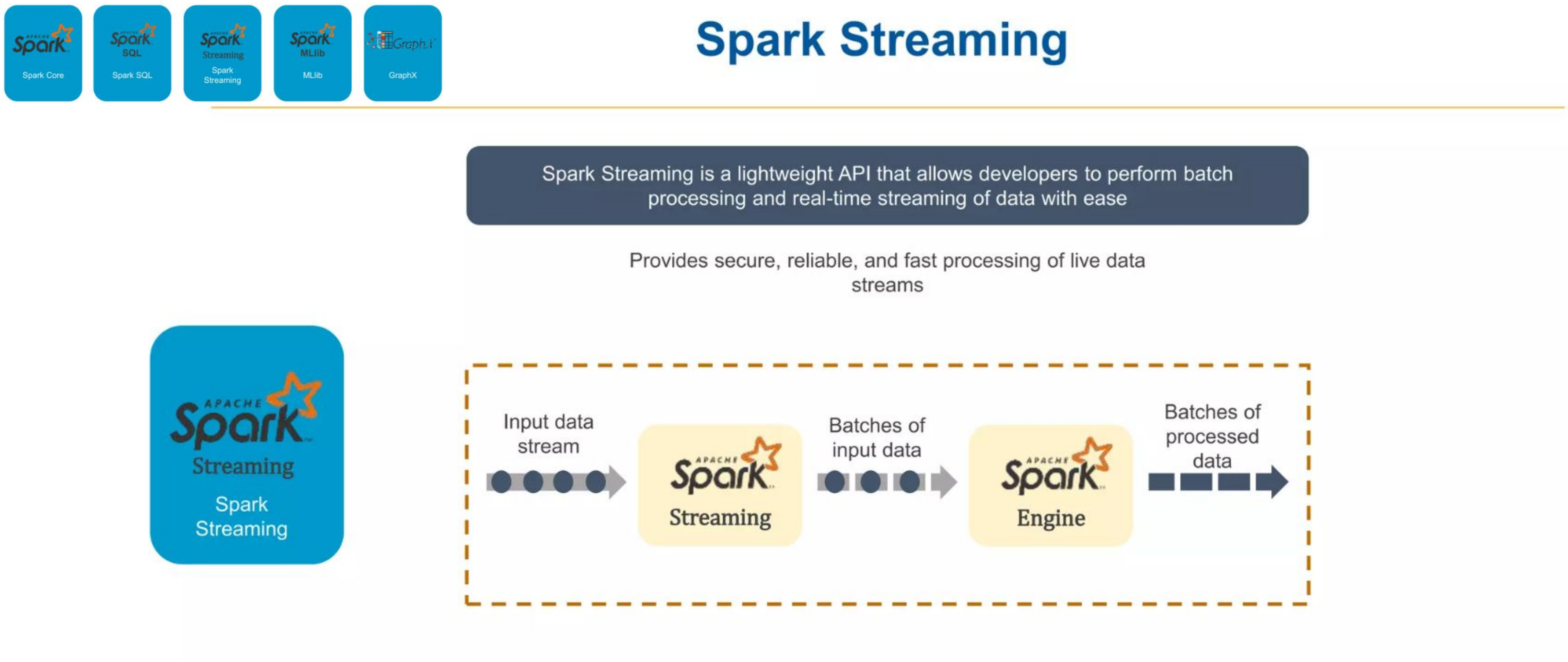


Spark SQL

Spark SQL framework component is used for structured and semi-structured data processing

Spark SQL Architecture

Apache Spark

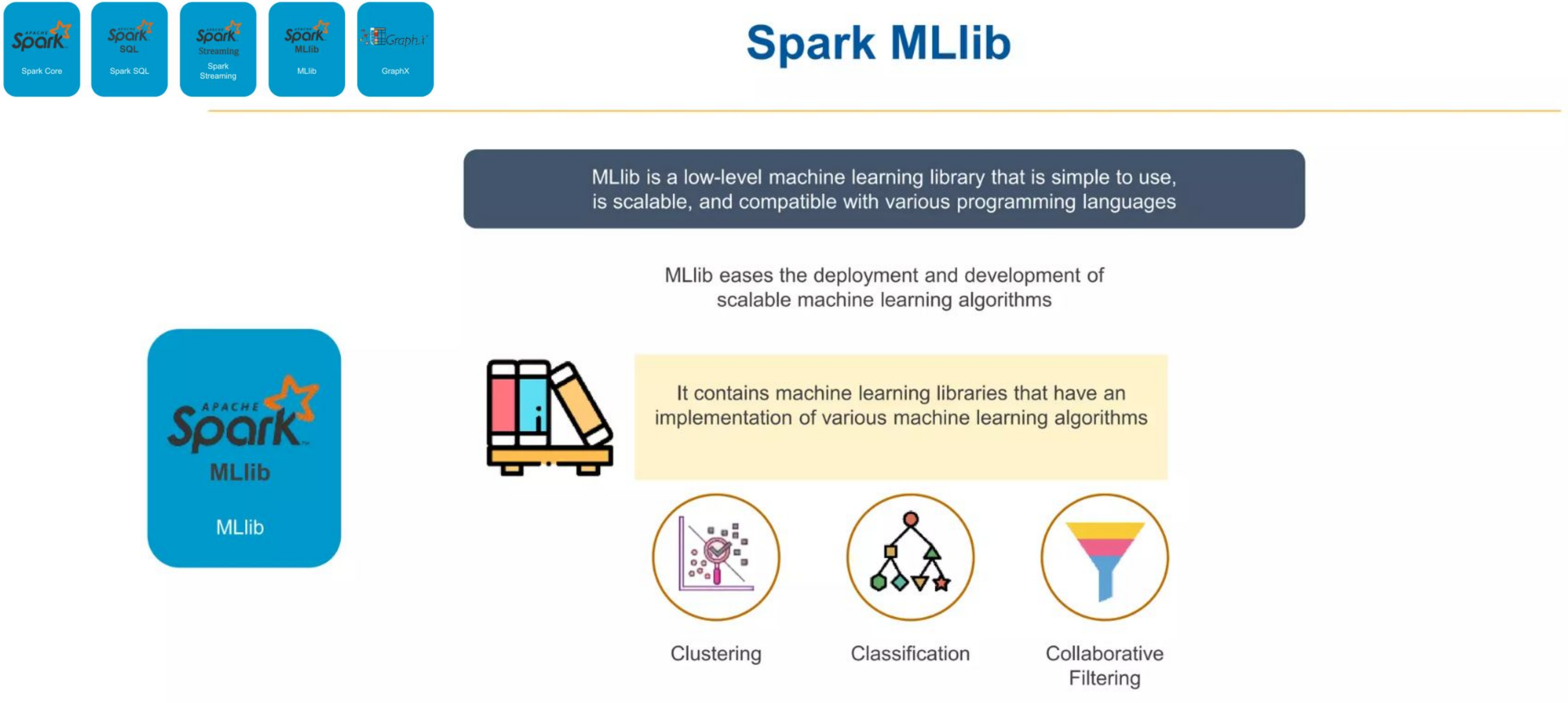


Spark Streaming

Spark Streaming is a lightweight API that allows developers to perform batch processing and real-time streaming of data with ease

Provides secure, reliable, and fast processing of live data streams

Apache Spark



Spark MLlib

MLlib is a low-level machine learning library that is simple to use, is scalable, and compatible with various programming languages

MLlib eases the deployment and development of scalable machine learning algorithms

It contains machine learning libraries that have an implementation of various machine learning algorithms

Clustering

Classification

Collaborative Filtering

Apache Spark

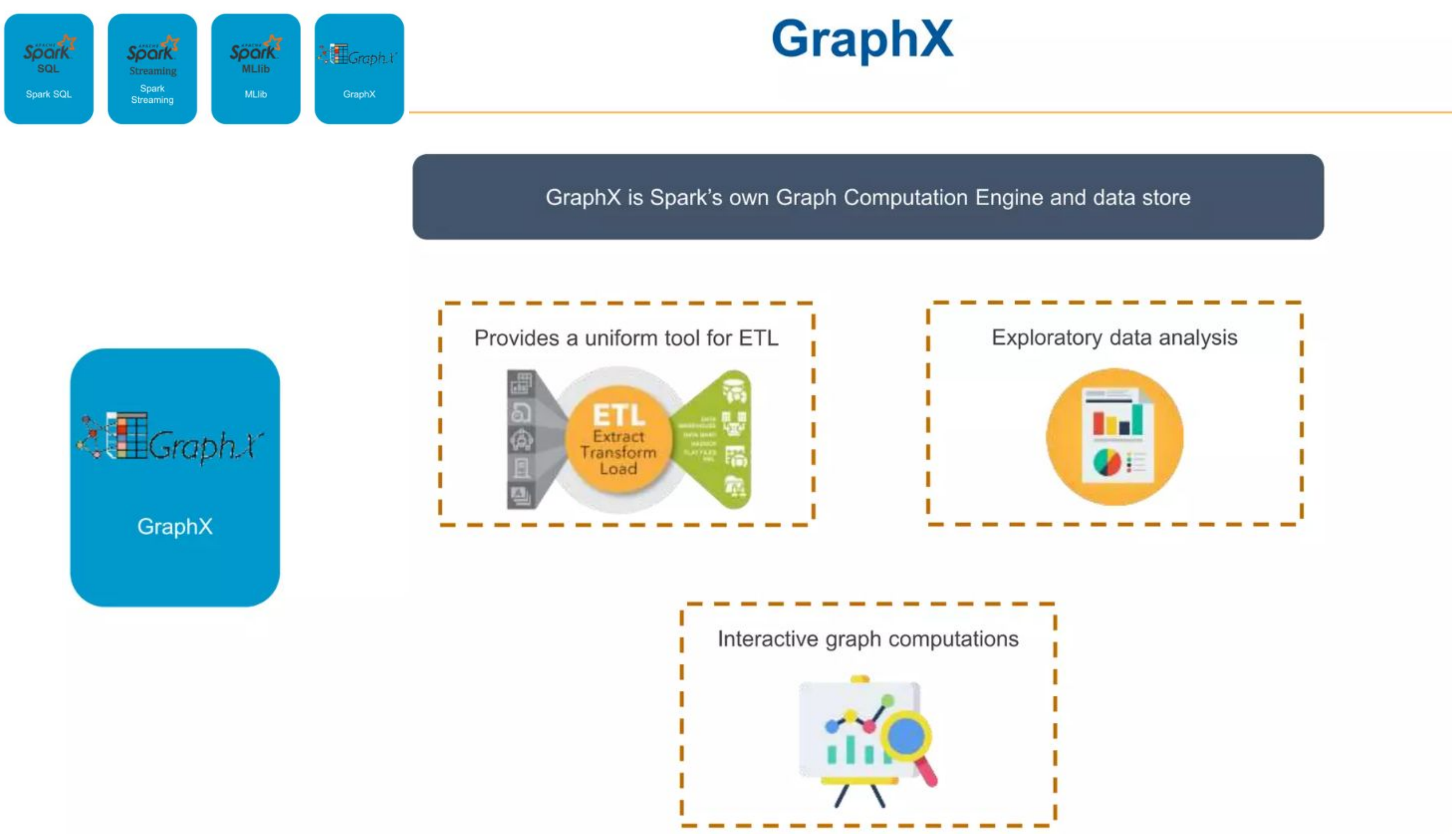
Spark Core

Spark SQL

Spark Streaming

MLlib

GraphX



GraphX

GraphX

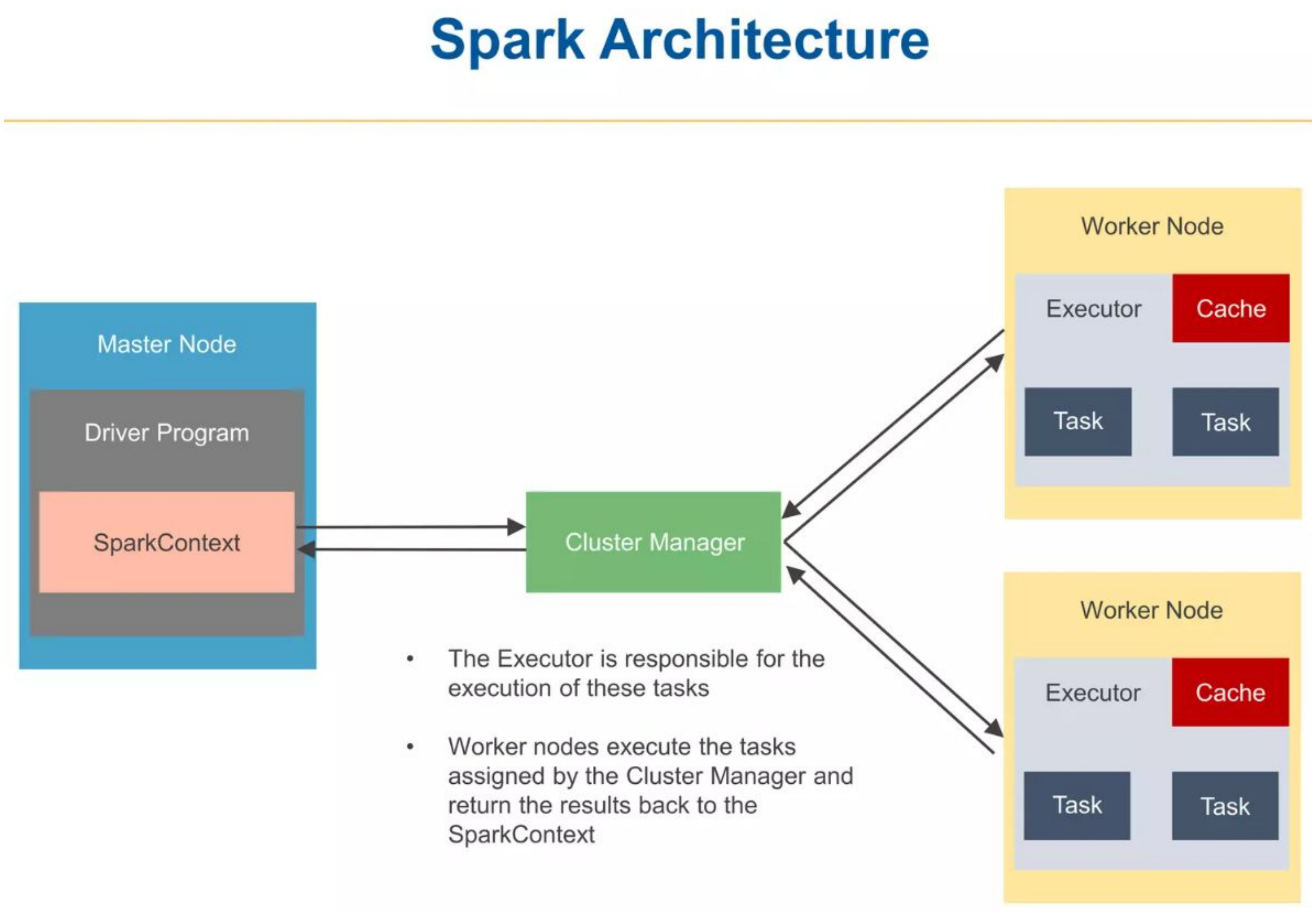
GraphX is Spark's own Graph Computation Engine and data store

Provides a uniform tool for ETL

Exploratory data analysis

Interactive graph computations

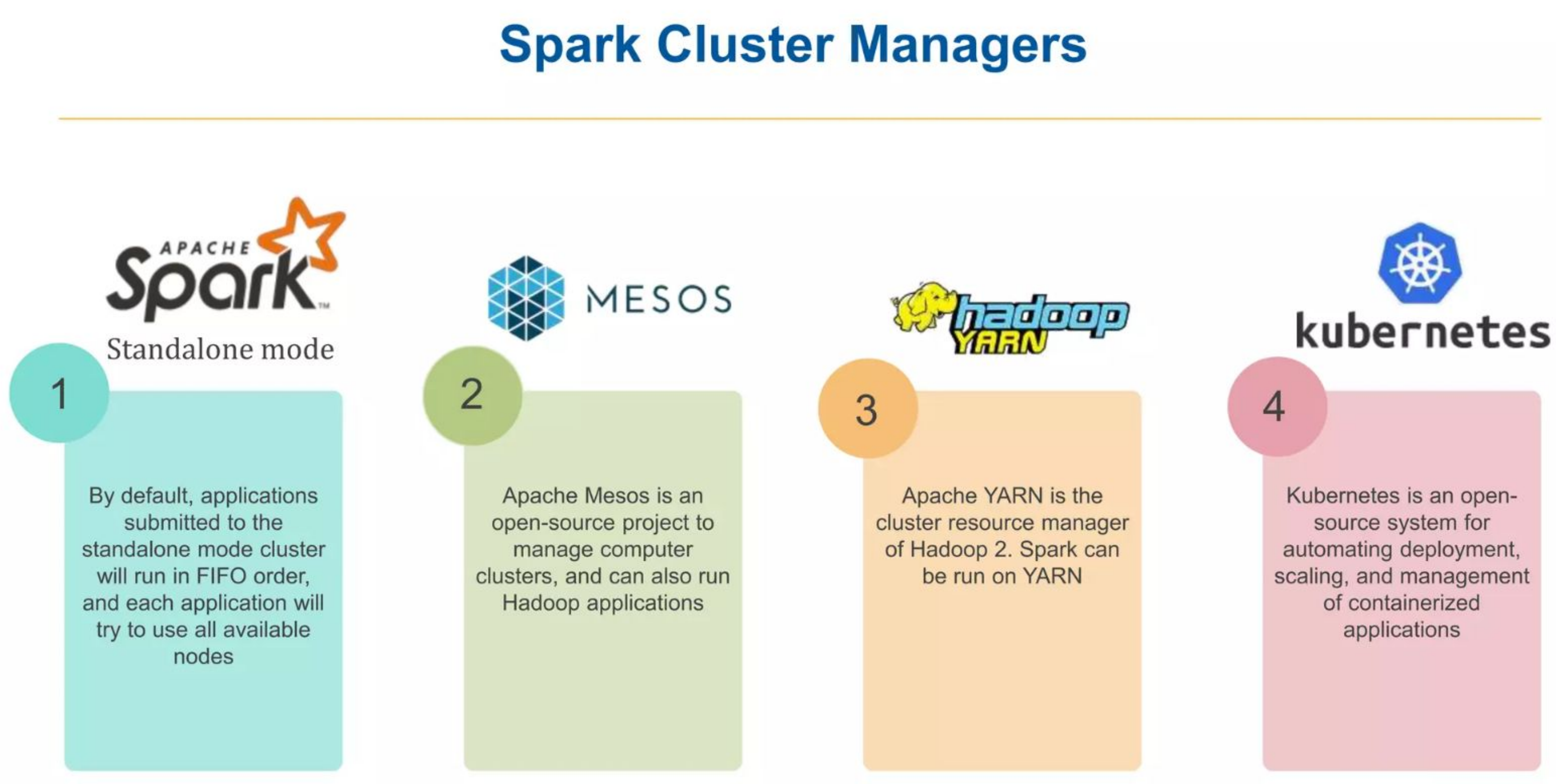
Apache Spark



Spark Architecture

- The Executor is responsible for the execution of these tasks
- Worker nodes execute the tasks assigned by the Cluster Manager and return the results back to the SparkContext

Apache Spark



Spark Cluster Managers

Standalone mode

1

By default, applications submitted to the standalone mode cluster will run in FIFO order, and each application will try to use all available nodes

2

Apache Mesos is an open-source project to manage computer clusters, and can also run Hadoop applications

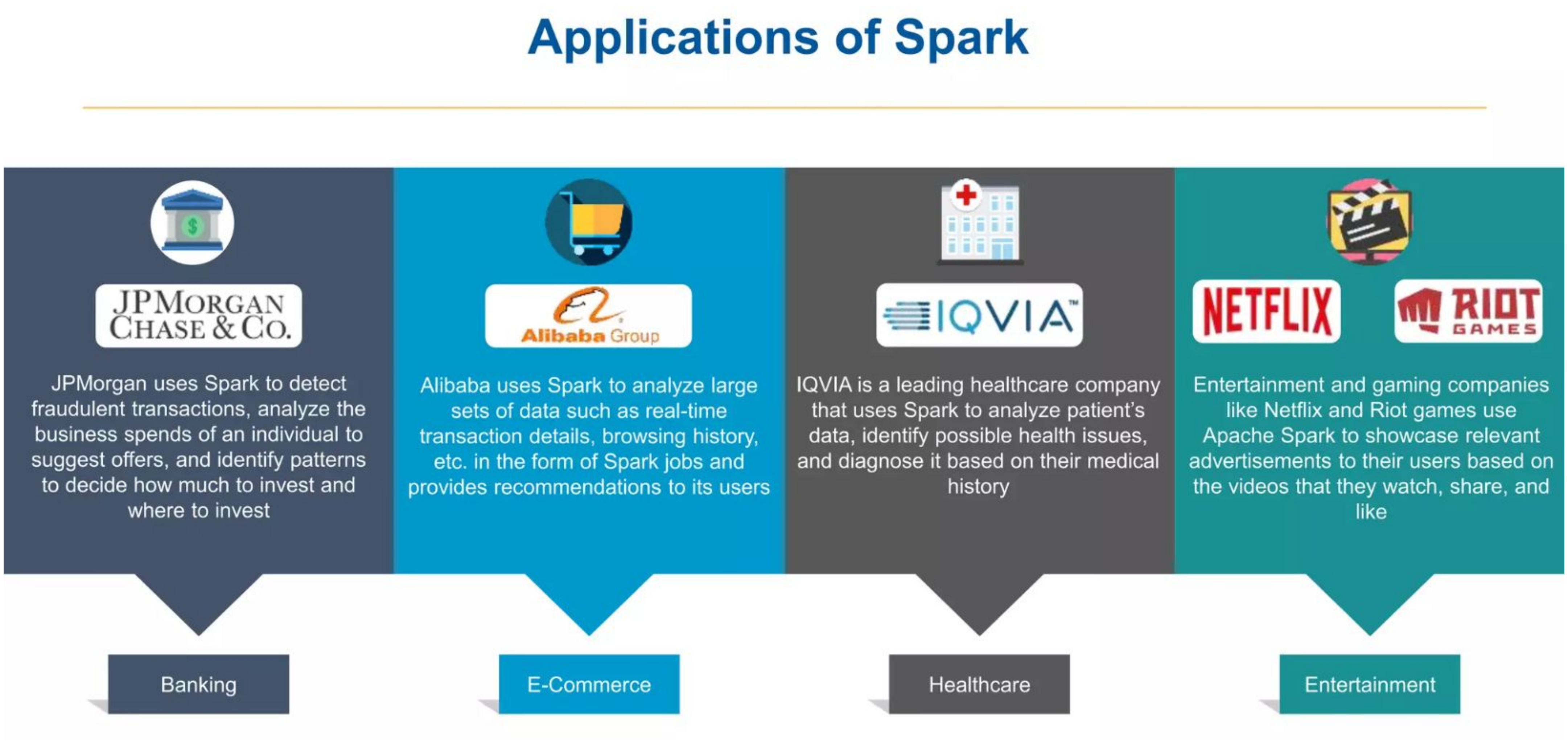
3

Apache YARN is the cluster resource manager of Hadoop 2. Spark can be run on YARN

4

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications

Apache Spark



Applications of Spark

JPMORGAN CHASE & CO.

JPMorgan uses Spark to detect fraudulent transactions, analyze the business spends of an individual to suggest offers, and identify patterns to decide how much to invest and where to invest

Banking

Alibaba Group

Alibaba uses Spark to analyze large sets of data such as real-time transaction details, browsing history, etc. in the form of Spark jobs and provides recommendations to its users

E-Commerce

IQVIA™

IQVIA is a leading healthcare company that uses Spark to analyze patient's data, identify possible health issues, and diagnose it based on their medical history

Healthcare

NETFLIX RIOT GAMES

Entertainment and gaming companies like Netflix and Riot games use Apache Spark to showcase relevant advertisements to their users based on the videos that they watch, share, and like

Entertainment

Apache Hive

What is Hive?

Apache Hive is an open source data warehouse system built on top of Hadoop Haused for querying and analyzing large datasets stored in Hadoop files.

Hive use language called HiveQL (HQL), which is similar to SQL. HiveQL automatically translates SQL-like queries into MapReduce jobs.

Apache Hive

Why to Use Hive?

- Hive provides summarization, analysis, and query of data.
- Hive is very fast and scalable.
- Hive reduces the complexity of MapReduce

Apache Hive

Where to Use Hive?

Apache Hive can be used in the following places:

- Data Mining
- Log Processing
- Document Indexing
- Customer Facing Business Intelligence
- Predictive Modelling
- Hypothesis Testing

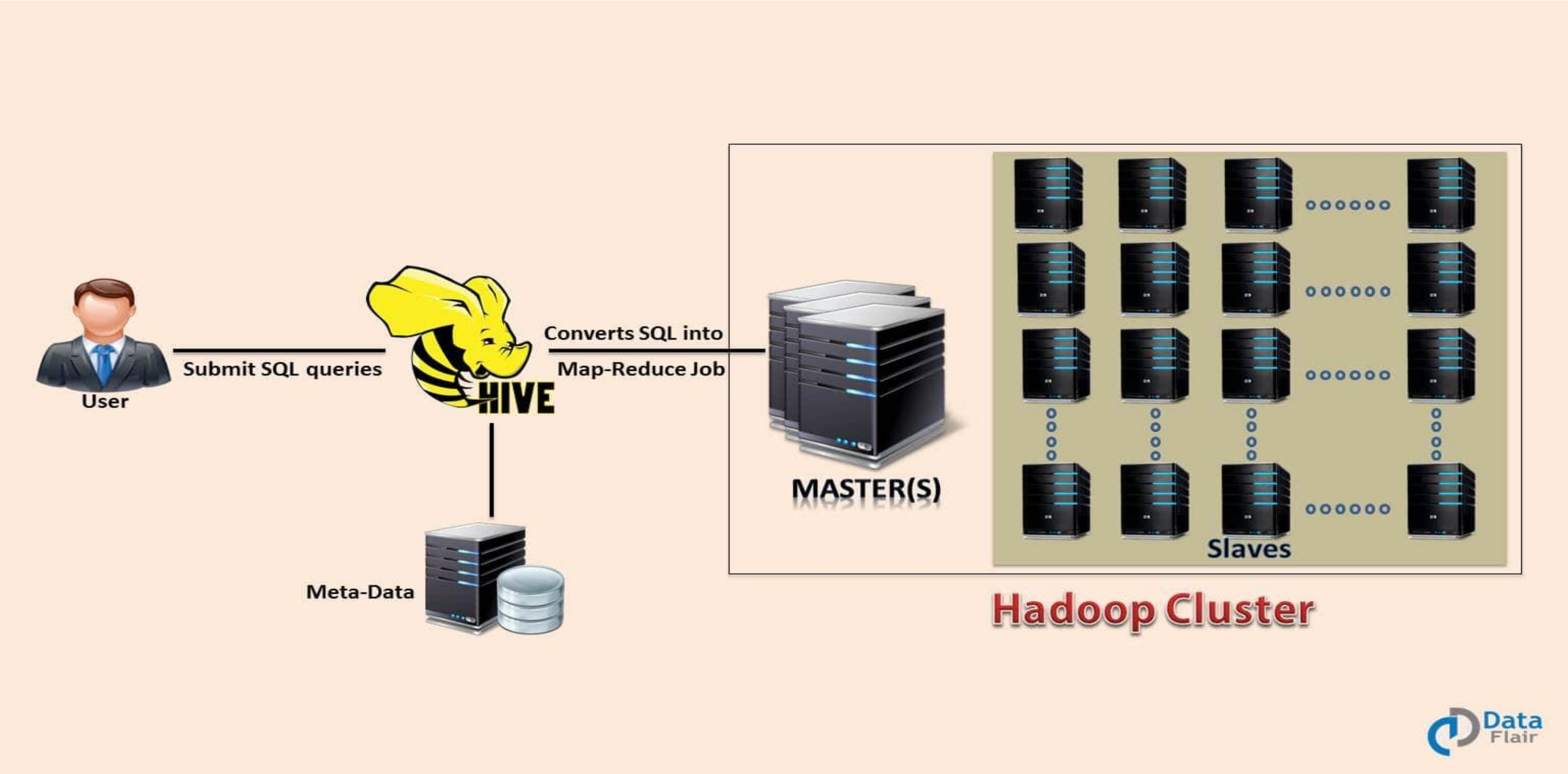
Apache Hive

History of Hive

- Data Infrastructure Team at Facebook developed Hive.
- Hive started as a subproject of Apache Hadoop, but has graduated to become a top-level project of its own
- Now it is being used and developed by a number of companies like Amazon, IBM, Yahoo, Netflix, Financial Industry Regulatory Authority (FINRA) and many others.

Apache Hive

Hive Architecture:



Apache Hive

Hive components:

- Metastore
- Driver
- Compiler
- Optimizer
- Executor
- CLI, UI, and Thrift Server

Apache Hive

Features of Hive:

- Hive provides data summarization, query, and analysis in much easier manner.
- Hive supports external tables which make it possible to process data without actually storing in HDFS.
- Apache Hive fits the low-level interface requirement of Hadoop perfectly.
- It also supports partitioning of data at the level of tables to improve performance.
- Hive has a rule based optimizer for optimizing logical plans.

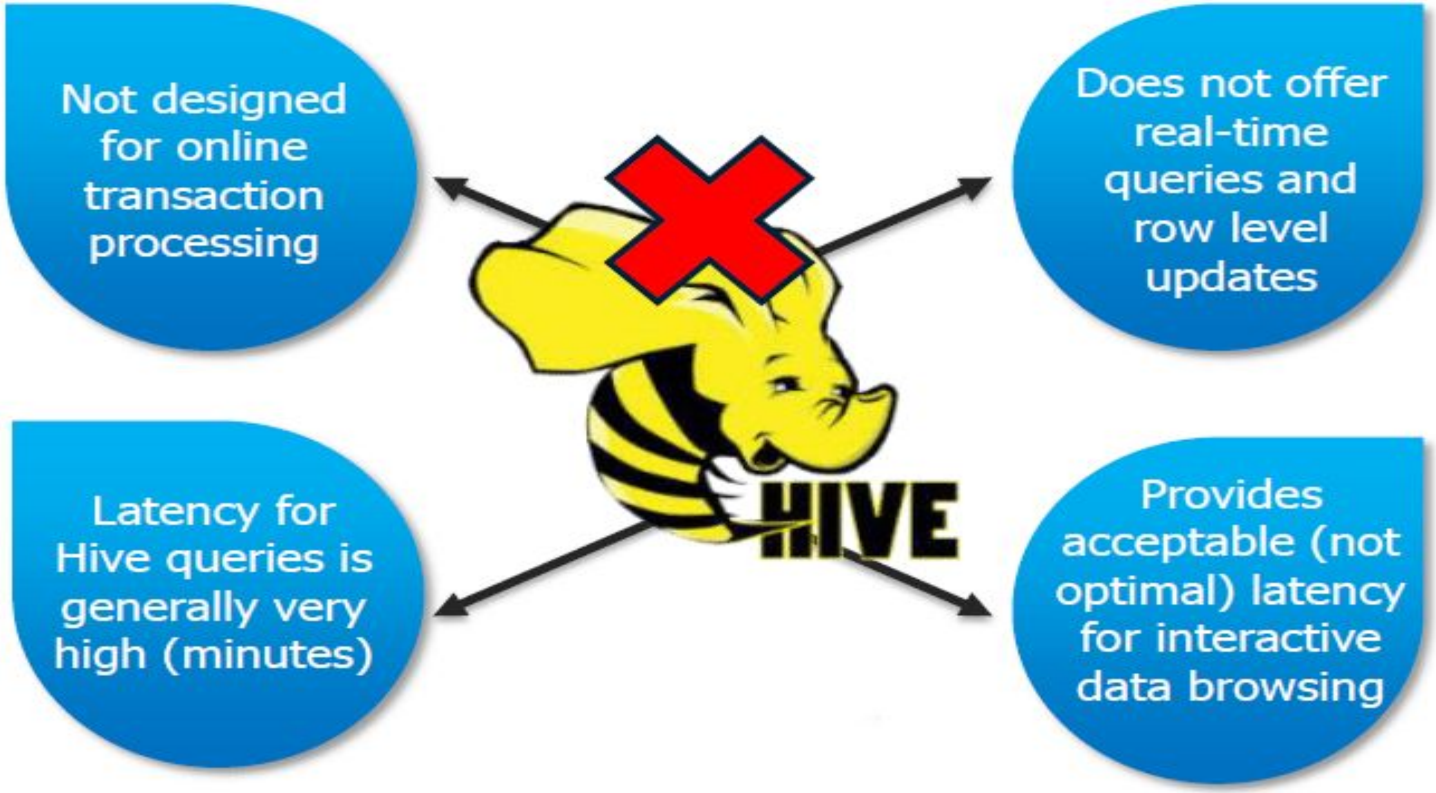
Apache Hive

Features of Hive:

- It is scalable, familiar, and extensible.
- Using HiveQL doesn’t require any knowledge of programming language, Knowledge of basic SQL query is enough.
- We can easily process structured data in Hadoop using Hive.
- Querying in Hive is very simple as it is similar to SQL.
- We can also run Ad-hoc queries for the data analysis using Hive.

Apache Hive

Limitations of Hive:



Not designed for online transaction processing

Does not offer real-time queries and row level updates

Latency for Hive queries is generally very high (minutes)

Provides acceptable (not optimal) latency for interactive data browsing

Thank You For Your Attention