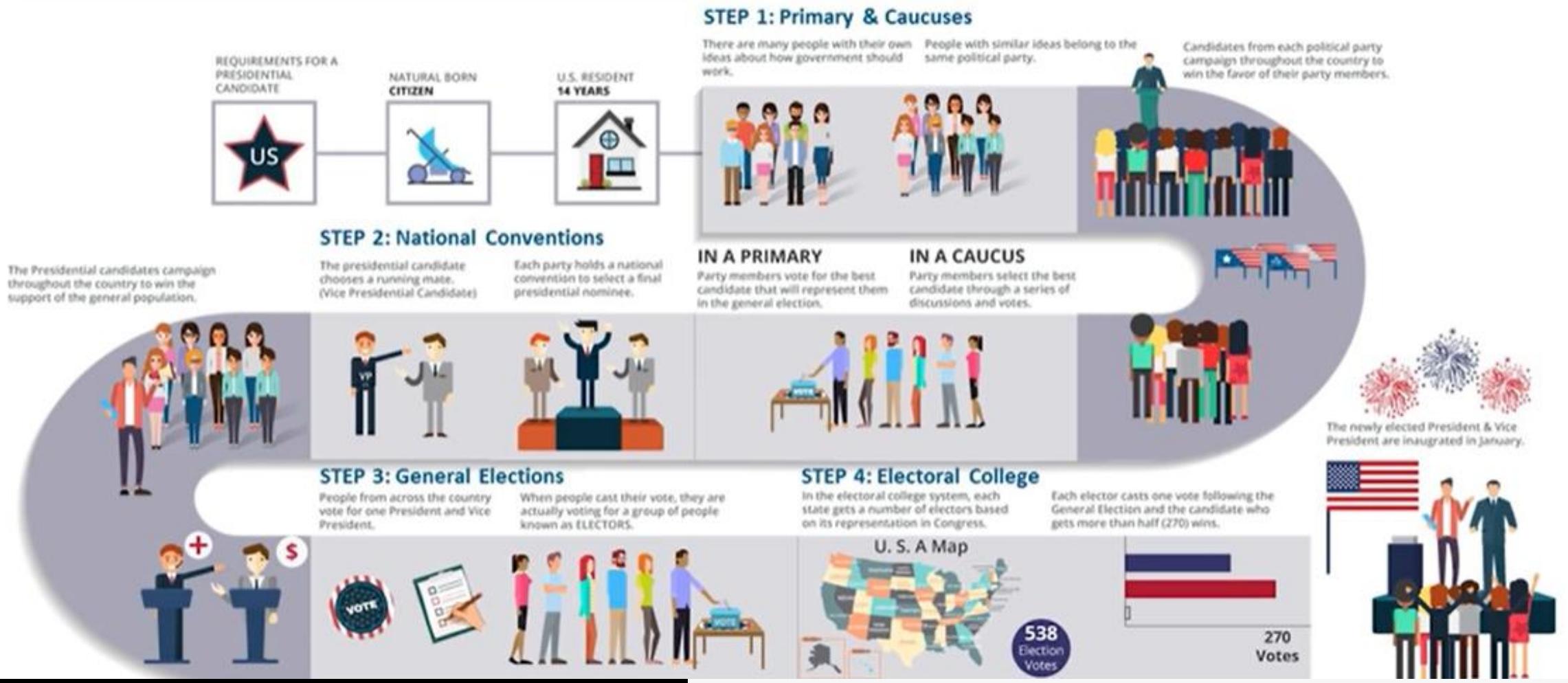
Hadoop Tutorial for Beginners

US Primary Election Analysis

1

2

US Election


REQUIREMENTS FOR A PRESIDENTIAL CANDIDATE
US
NATURAL BORN CITIZEN
U.S. RESIDENT 14 YEARS
STEP 1: Primary & Caucuses
There are many people with their own ideas about how government should work.
People with similar ideas belong to the same political party.
Candidates from each political party campaign throughout the country to win the favor of their party members.
IN A PRIMARY
Party members vote for the best candidate that will represent them in the general election.
IN A CAUCUS
Party members select the best candidate through a series of discussions and votes.
STEP 2: National Conventions
The presidential candidate chooses a running mate. (Vice Presidential Candidate)
Each party holds a national convention to select a final presidential nominee.
The Presidential candidates campaign throughout the country to win the support of the general population.
VP
VOTE
The newly elected President & Vice President are inaugrated in January.
STEP 3: General Elections
People from across the country vote for one President and Vice President.
When people cast their vote, they are actually voting for a group of people known as ELECTORS.
VOTE
STEP 4: Electoral College
In the electoral college system, each state gets a number of electors based on its representation in Congress.
Each elector casts one vote following the General Election and the candidate who gets more than half (270) wins.
U. S. A Map
538 Election Votes
270 Votes

US Primary Election Dataset

Now as a data analyst you have 2 datasets available :

US Primary Election Data Set

state	state_abbreviation	county	fips	party	candidate	votes	fraction_votes
Alabama	AL	Autauga	1001	Democrat	Bernie Sanders	544	0.182
Alabama	AL	Autauga	1001	Democrat	Hillary Clinton	2387	0.8
Alabama	AL	Baldwin	1003	Democrat	Bernie Sanders	2694	0.329
Alabama	AL	Baldwin	1003	Democrat	Hillary Clinton	5290	0.647
Alabama	AL	Barbour	1005	Democrat	Bernie Sanders	222	0.078
Alabama	AL	Barbour	1005	Democrat	Hillary Clinton	2567	0.906
Alabama	AL	Bibb	1007	Democrat	Bernie Sanders	246	0.197
Alabama	AL	Bibb	1007	Democrat	Hillary Clinton	942	0.755
Alabama	AL	Blount	1009	Democrat	Bernie Sanders	395	0.386
Alabama	AL	Blount	1009	Democrat	Hillary Clinton	564	0.551
Alabama	AL	Bullock	1011	Democrat	Bernie Sanders	178	0.066
Alabama	AL	Bullock	1011	Democrat	Hillary Clinton	2451	0.913
Alabama	AL	Butler	1013	Democrat	Bernie Sanders	156	0.065

US Demographic Features (County-wise) Data Set

fips	area_name	state_abbreviation	PST045214	PST040210	PST120214	POP010210	AGE135214	AGE295214	AGE775214	SEX255214
0	United States		318857056	308758105	3.3	308745538	6.2	23.1	14.5	50.8
1000	Alabama		4849377	4780127	1.4	4779736	6.1	22.8	15.3	51.5
1001	Autauga County	AL	55395	54571	1.5	54571	6	25.2	13.8	51.4
1003	Baldwin County	AL	200111	182265	9.8	182265	5.6	22.2	18.7	51.2
1005	Barbour County	AL	26887	27457	-2.1	27457	5.7	21.2	16.5	46.6
1007	Bibb County	AL	22506	22919	-1.8	22915	5.3	21	14.8	45.9
1009	Blount County	AL	57719	57322	0.7	57322	6.1	23.6	17	50.5
1011	Bullock County	AL	10764	10915	-1.4	10914	6.3	21.4	14.9	45.3
1013	Butler County	AL	20296	20946	-3.1	20947	6.1	23.6	18	53.6
1015	Calhoun County	AL	113916	118586	-2.3	118572	5.7	22.2	16	51.8
1017	Chambers County	AL	34076	34170	-0.3	34215	5.9	21.4	18.3	52.3
1019	Cherokee County	AL	26037	25986	0.2	25989	4.8	20.4	20.9	50.2
1021	Chilton County	AL	43931	43631	0.7	43643	6.4	24.2	15.2	50.8
1023	Choctaw County	AL	13323	13858	-3.9	13859	4.9	20.6	20.8	52.5

US Primary Election Dataset

state	state_abbreviation	county	fips	party	candidate	votes	fraction_votes
Alabama	AL	Autauga	1001	Democrat	Bernie Sanders	544	0.182
Alabama	AL	Autauga	1001	Democrat	Hillary Clinton	2387	0.8
Alabama	AL	Baldwin	1003	Democrat	Bernie Sanders	2694	0.329
Alabama	AL	Baldwin	1003	Democrat	Hillary Clinton	5290	0.647
Alabama	AL	Barbour	1005	Democrat	Bernie Sanders	222	0.078
Alabama	AL	Barbour	1005	Democrat	Hillary Clinton	2567	0.906
Alabama	AL	Bibb	1007	Democrat	Bernie Sanders	246	0.197
Alabama	AL	Bibb	1007	Democrat	Hillary Clinton	942	0.755
Alabama	AL	Blount	1009	Democrat	Bernie Sanders	395	0.386
Alabama	AL	Blount	1009	Democrat	Hillary Clinton	564	0.551
Alabama	AL	Bullock	1011	Democrat	Bernie Sanders	178	0.066
Alabama	AL	Bullock	1011	Democrat	Hillary Clinton	2451	0.913
Alabama	AL	Butler	1013	Democrat	Bernie Sanders	156	0.065

state: List of US states

state_abbreviation: Abbreviation of each US state

county: List of counties in each US states

fips: FIPS county code is a Federal Information Processing Standards (FIPS) code which uniquely identifies counties

party: Different parties in US (i.e. Republican & Democrat)

candidate: candidates in US primary election from different parties

votes: number of votes gained by a candidate

fraction_votes: total number of votes gained by a candidate/ total votes gained by the party

US County Demographic Features Dataset

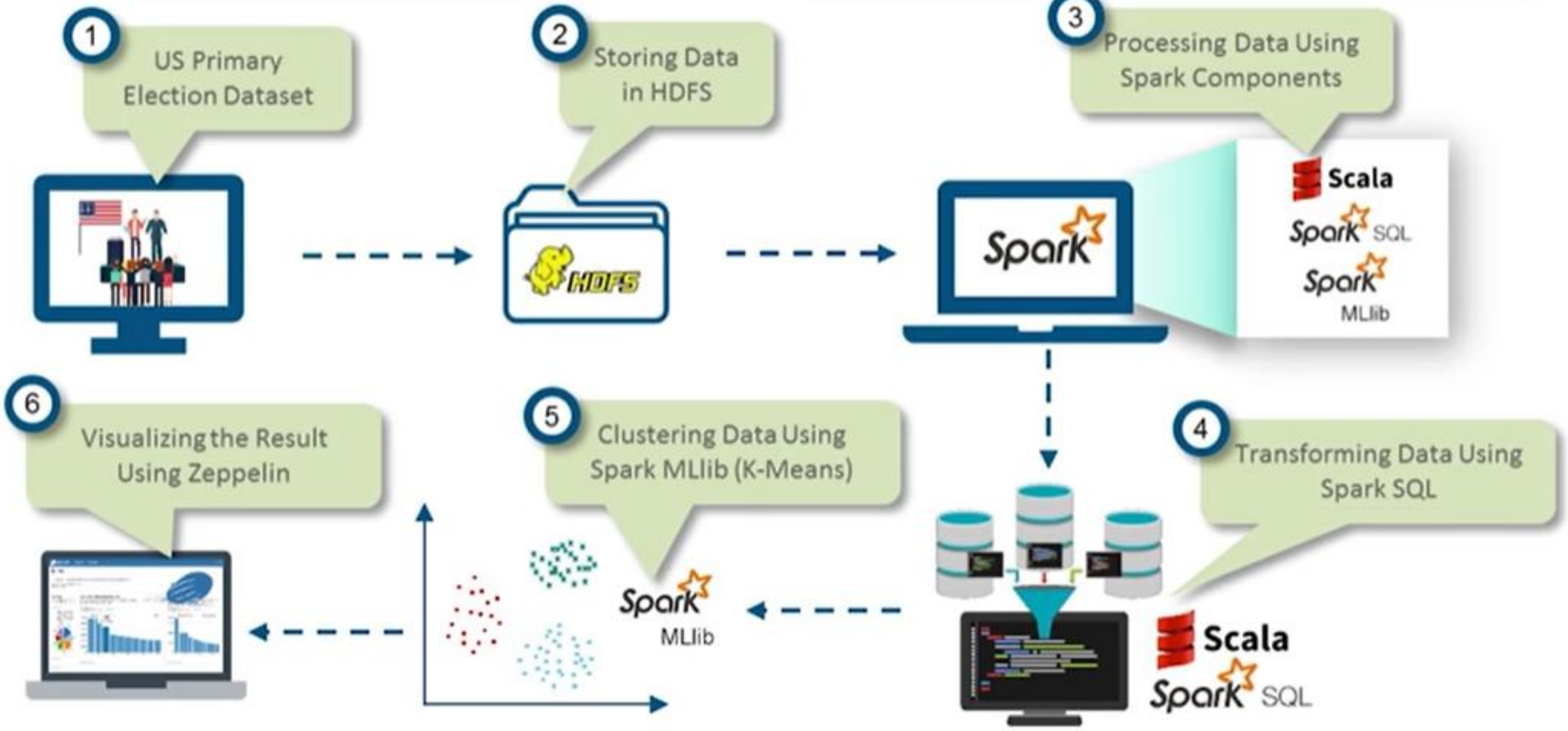
fips	area_name	state_abbreviation	PST045214	PST040210	PST120214	POP010210	AGE135214	AGE295214	AGE775214	SEX255214
0	United States		318857056	308758105	3.3	308745538	6.2	23.1	14.5	50.8
1000	Alabama		4849377	4780127	1.4	4779736	6.1	22.8	15.3	51.5
1001	Autauga County	AL	55395	54571	1.5	54571	6	25.2	13.8	51.4
1003	Baldwin County	AL	200111	182265	9.8	182265	5.6	22.2	18.7	51.2
1005	Barbour County	AL	26887	27457	-2.1	27457	5.7	21.2	16.5	46.6
1007	Bibb County	AL	22506	22919	-1.8	22915	5.3	21	14.8	45.9
1009	Blount County	AL	57719	57322	0.7	57322	6.1	23.6	17	50.5
1011	Bullock County	AL	10764	10915	-1.4	10914	6.3	21.4	14.9	45.3
1013	Butler County	AL	20296	20946	-3.1	20947	6.1	23.6	18	53.6
1015	Calhoun County	AL	115916	118586	-2.3	118572	5.7	22.2	16	51.8
1017	Chambers County	AL	34076	34170	-0.3	34215	5.9	21.4	18.3	52.3
1019	Cherokee County	AL	26037	25986	0.2	25989	4.8	20.4	20.9	50.2
1021	Chilton County	AL	43931	43631	0.7	43643	6.4	24.2	15.2	50.8
1023	Choctaw County	AL	13323	13858	-3.9	13859	4.9	20.6	20.8	52.5

DETAILS:

Population, 2014 estimate
Population, 2010 (April 1) estimates base
Population, percent change - April 1, 2010 to July 1, 2014
Population, 2010

Persons under 5 years, percent, 2014
Persons under 18 years, percent, 2014
Persons 65 years and over, percent, 2014
Female persons, percent, 2014
White alone, percent, 2014 ...

US Election Solution Strategy



Market Analysis for US Cab Start-Ups

PROBLEM STATEMENT:

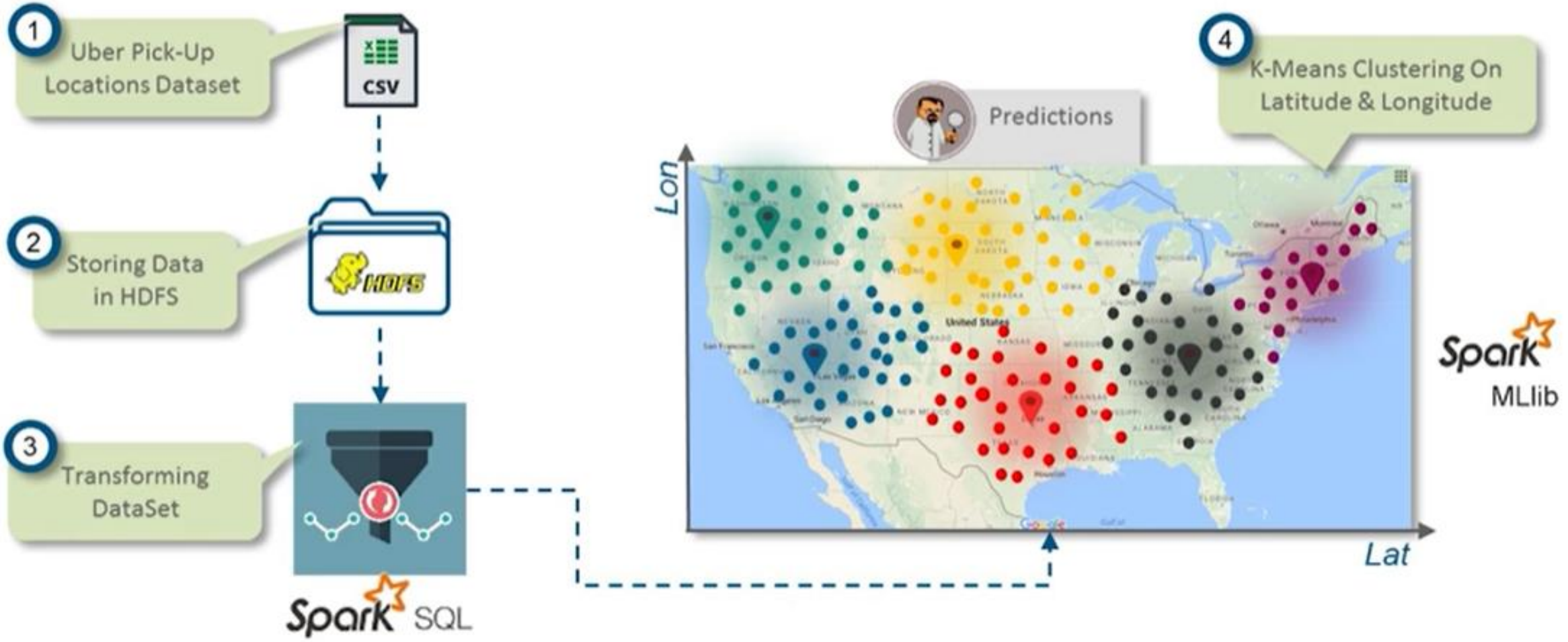
- A *US cab service start-up* wants to *meet the demands* in an optimum manner and *maximize the profit*.
- Thus, *they hired you as a data analyst* to *interpret the available Uber's data set* and *find out the beehive customer pick-up points & peak hours* for meeting the demand in a profitable manner.

Uber Dataset

Date/Time	Lat	Lon	Base
08-01-2014 00:03	40.7366	-73.9906	B02512
08-01-2014 00:09	40.726	-73.9918	B02512
08-01-2014 00:12	40.7209	-74.0507	B02512
08-01-2014 00:12	40.7387	-73.9856	B02512
08-01-2014 00:12	40.7323	-74.0077	B02512
08-01-2014 00:13	40.7349	-74.0033	B02512
08-01-2014 00:15	40.7279	-73.9542	B02512
08-01-2014 00:17	40.721	-73.9937	B02512
08-01-2014 00:19	40.7195	-74.006	B02512
08-01-2014 00:20	40.7448	-73.9799	B02512
08-01-2014 00:21	40.7399	-74.0057	B02512
08-01-2014 00:25	40.7651	-73.9683	B02512
08-01-2014 00:27	40.7354	-74.0081	B02512
08-01-2014 00:29	40.7339	-74.0028	B02512
08-01-2014 00:29	40.7364	-74.0301	B02512
08-01-2014 00:29	40.7364	-74.0301	B02512
08-01-2014 00:30	40.7252	-73.9516	B02512
08-01-2014 00:30	40.7433	-73.986	B02512
08-01-2014 00:34	40.7437	-73.9884	B02512

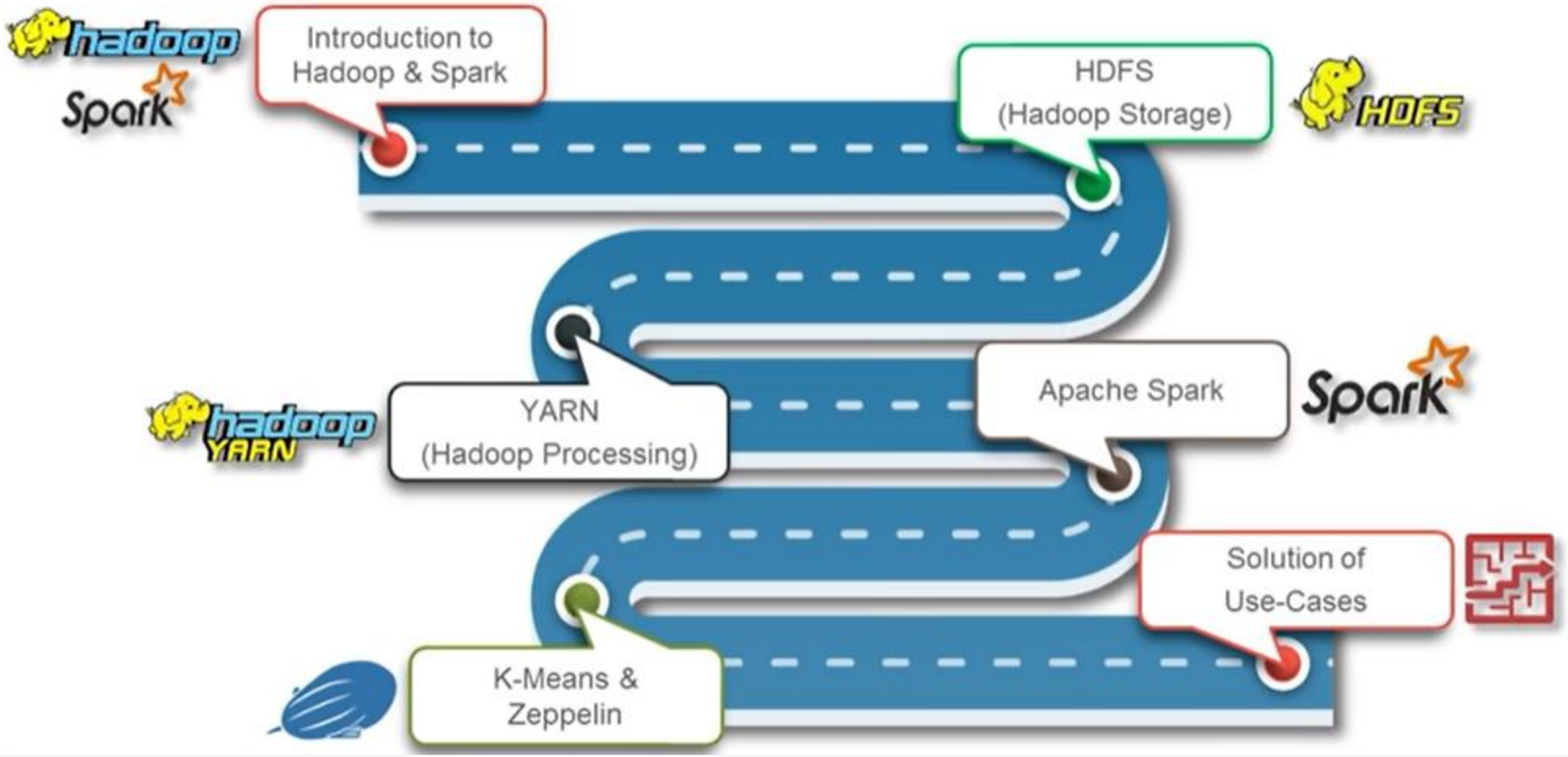
- **Date/Time** – Pickup Date & Time
- **Lat** – Latitude of Pickup
- **Lon** – Longitude of Pickup
- **Base** – TLC Base Code

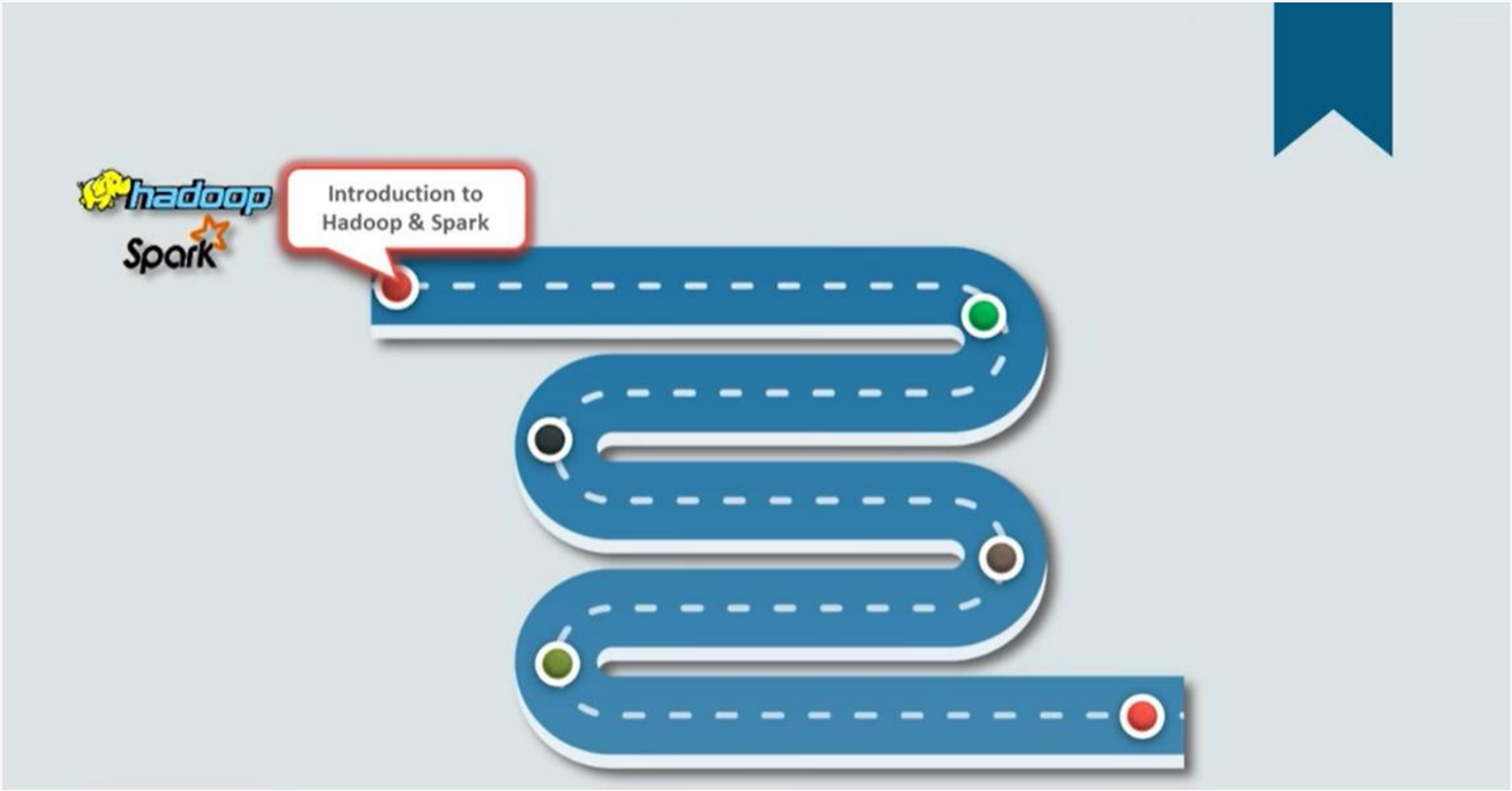
Market Analysis for US Cab Start-Ups Solution Strategy



Let Us Know What It Takes...

Fundamentals Road Map




hadoop
Spark
Introduction to
Hadoop & Spark

Introduction to Hadoop & Spark

Hadoop

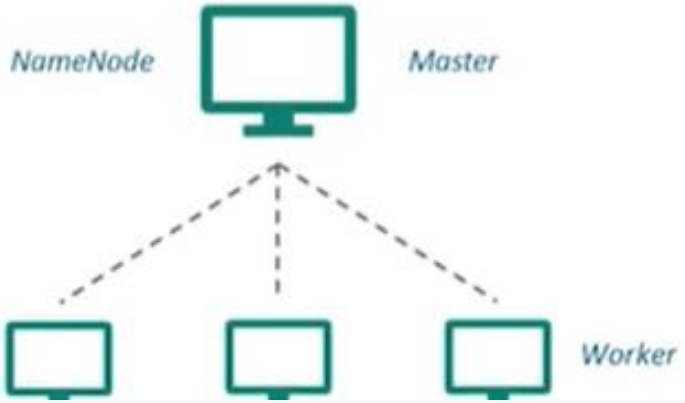
Hadoop is a framework that allows you to *store and process large data sets* in *parallel and distributed fashion.*

- Hadoop has two core components:
  - *HDFS*: Allows to dump any kind of data across the cluster
  - *YARN*: Allows parallel processing of the data stored in HDFS

Spark

Apache Spark is an open-source *cluster-computing* framework for *real time* processing

- Provides an interface for programming entire clusters with implicit *data parallelism* and *fault-tolerance*
- Built on top of *YARN* and it *extends the YARN model* to efficiently use more types of computations



Spark Complementing Hadoop

Spark & Hadoop

1. Spark processes data 100 times faster than MapReduce
2. Spark Applications can run on YARN leveraging Hadoop cluster
3. Apache Spark can use HDFS as its storage

Challenges Addressed :

- Faster Analytic

- Cost Optimization

- Avoid Duplication

Combining Spark's ability, i.e. high processing speed, advance analytics and multiple integration support with Hadoop's low cost operation on commodity hardware gives the best results

Big Data Use-Cases

- **Web and e-tailing**
  - Recommendation Engines
  - Ad Targeting
  - Search Quality
  - Abuse and Click Fraud Detection

- **Government**
  - Fraud Detection and Cyber Security
  - Welfare Schemes
  - Justice

- **Telecommunications**
  - Customer Churn Prevention
  - Network Performance Optimization
  - Calling Data Record (CDR) Analysis
  - Analysing Network to Predict Failure

- **Healthcare and Life Sciences**
  - Health Information Exchange
  - Gene Sequencing
  - Serialization
  - Healthcare Service Quality Improvements
  - Drug Safety

Big Data Use-Cases

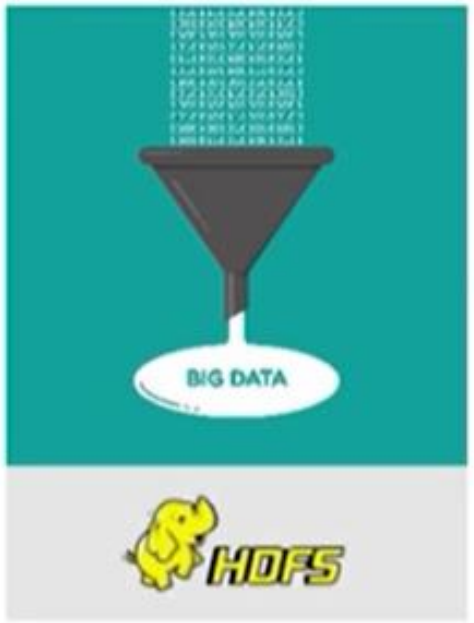
- **Banks and Financial services**
  - Modeling True Risk
  - Threat Analysis
  - Fraud Detection
  - Trade Surveillance
  - Credit Scoring and Analysis

- **Transportation Services**
  - Data from Location based social network
  - High speed data from telecom
  - Transport demand models
  - Route Planning

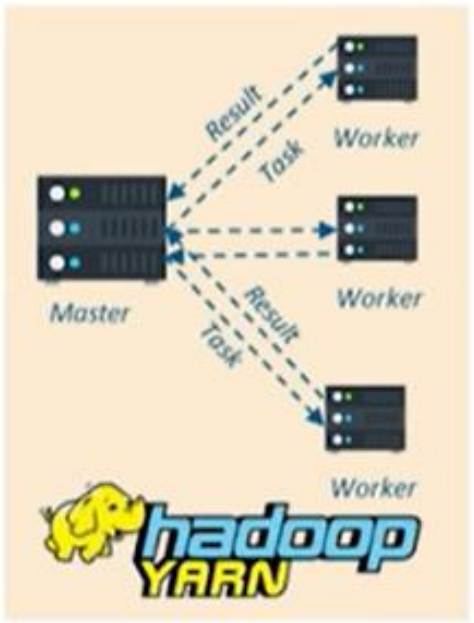
- **Retail**
  - Point of Sales Transaction Analysis
  - Customer Churn Analysis
  - Sentiment Analysis

- **Hotels and Food Delivery Services**
  - Customer Demands
  - Details of Customers
  - Availability and Seasonal Data Changes

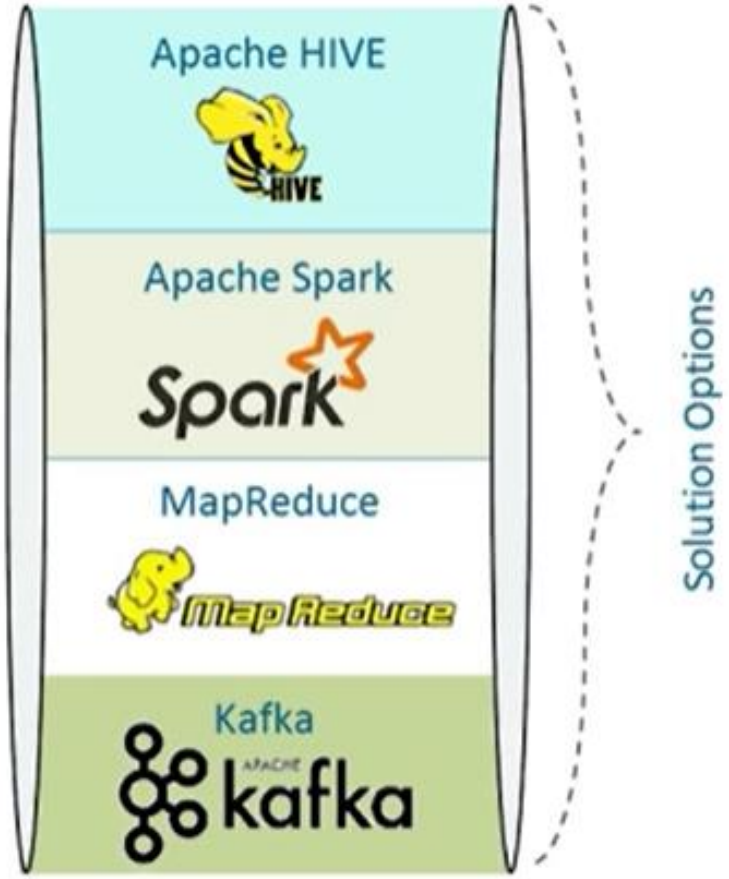
Big Data Use-Cases Solution Architecture



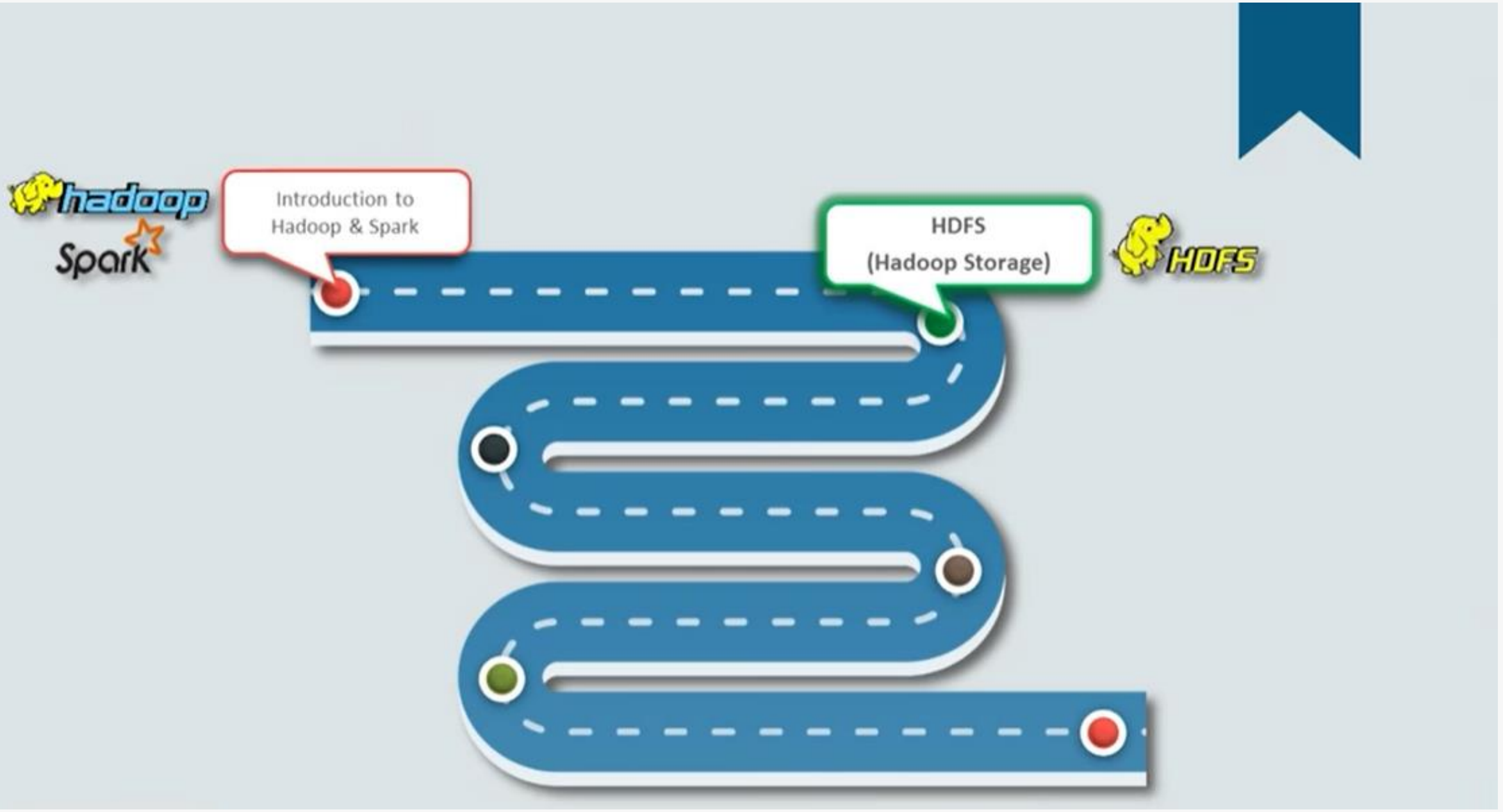
Storing Big Data on HDFS



Processing through YARN framework

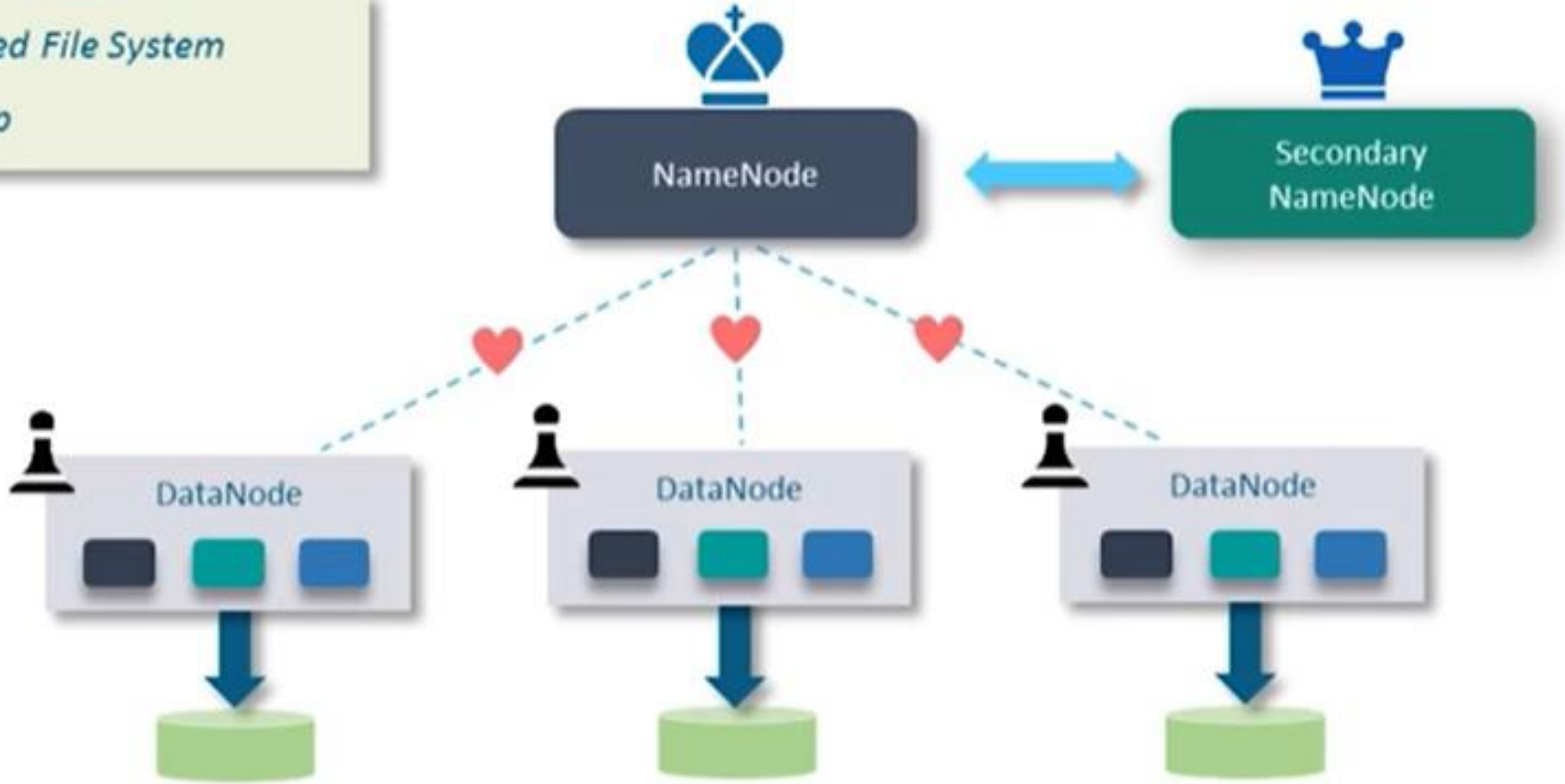


Tools used for processing


hadoop
Spark
Introduction to
Hadoop & Spark
HDFS
(Hadoop Storage)
HDFS

HDFS

- HDFS stands for *Hadoop Distributed File System*
- HDFS is the *storage unit of Hadoop*



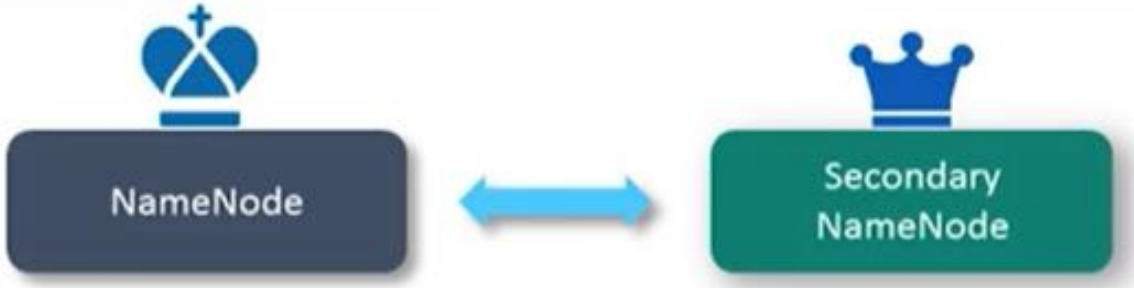
HDFS creates an *abstraction layer* over the distributed storage resources, from where we can see the *whole HDFS as a single unit*.

NameNode

NameNode

- Master daemon
- Maintains and Manages DataNodes
- Records metadata e.g. location of blocks stored, the size of the files, permissions, hierarchy, etc.
- Receives heartbeat and block report from all the DataNodes

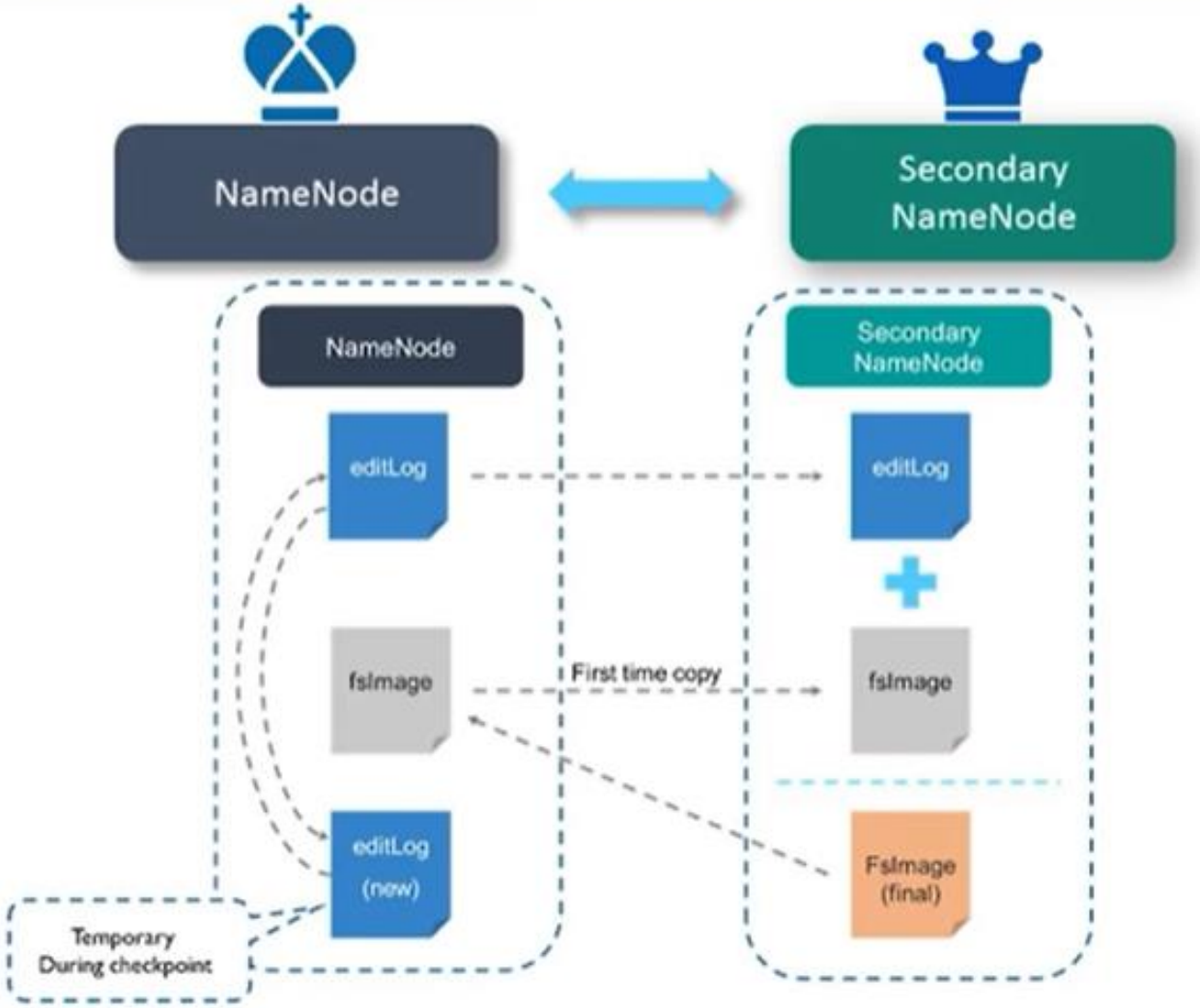
Secondary NameNode



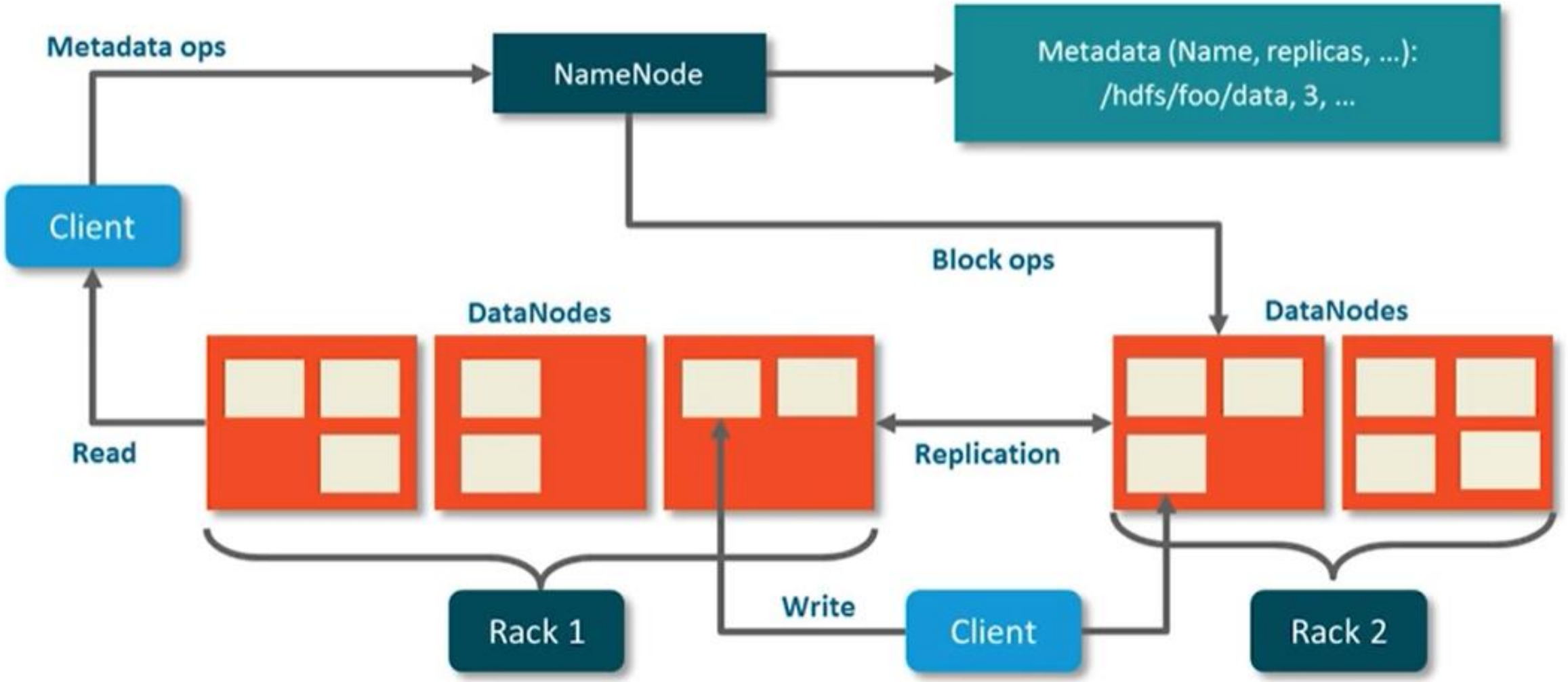
Secondary NameNode

- Checkpointing is a process of combining edit logs with FsImage
- Allows faster Failover as we have a back up of the metadata
- Checkpointing happens periodically (default: 1 hour)

Secondary NameNode



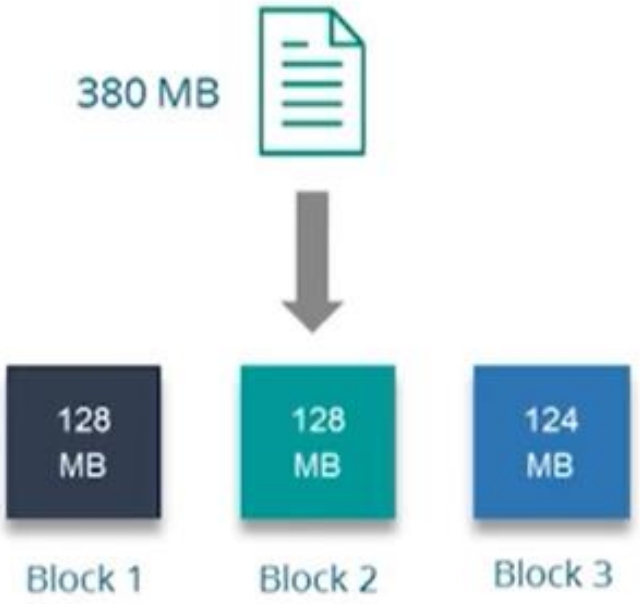
HDFS Architecture in Detail



HDFS Block & Replication

HDFS Data Block

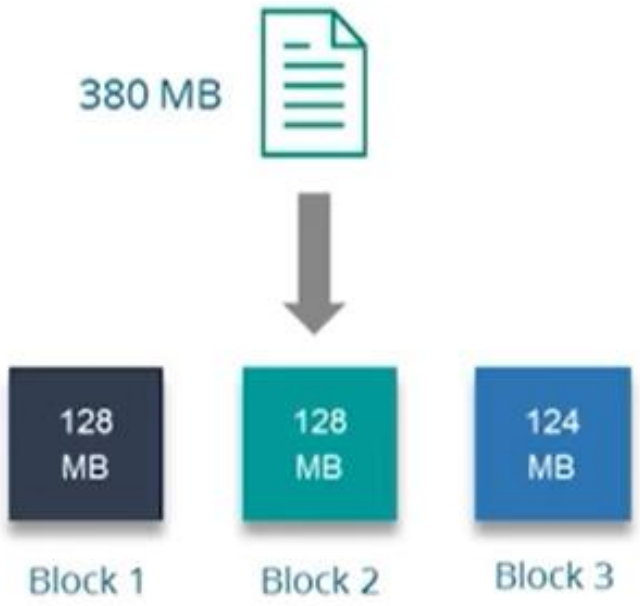
- Each file is stored on HDFS as block
- The default size of each block is 128 MB
- Let us say, I have a file example.txt of size 380 MB:



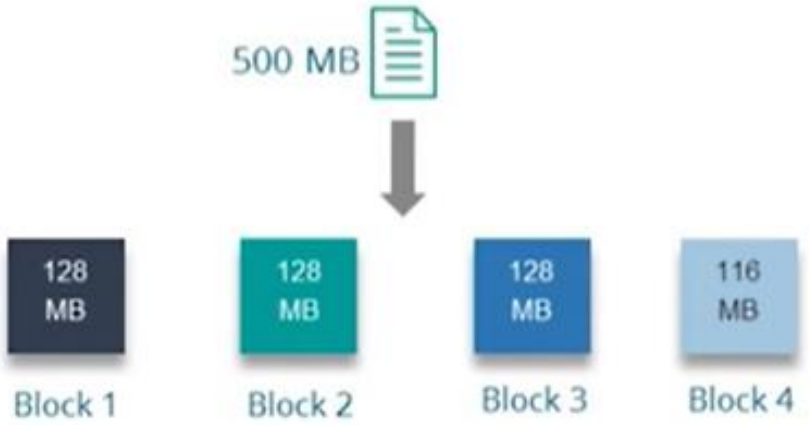
How many blocks will be created if a file of size 500 MB is copied to HDFS?

HDFS Data Block

- Each file is stored on HDFS as block
- The default size of each block is 128 MB
- Let us say, I have a file example.txt of size 500 MB:



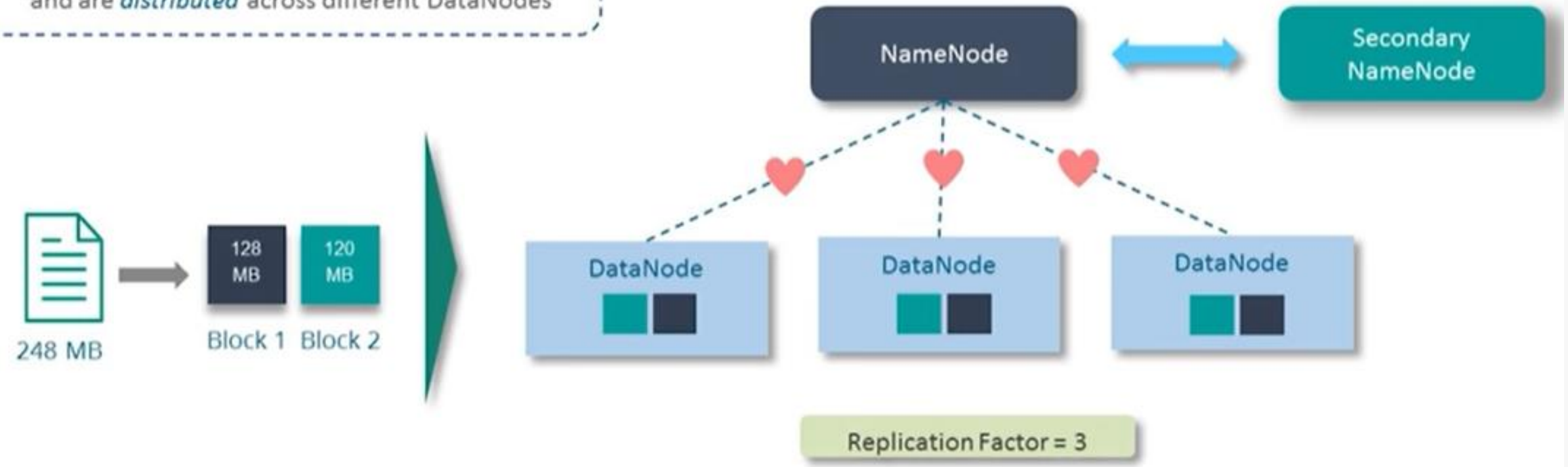
How many blocks will be created if a file of size 500 MB is copied to HDFS?



HDFS Block Replication

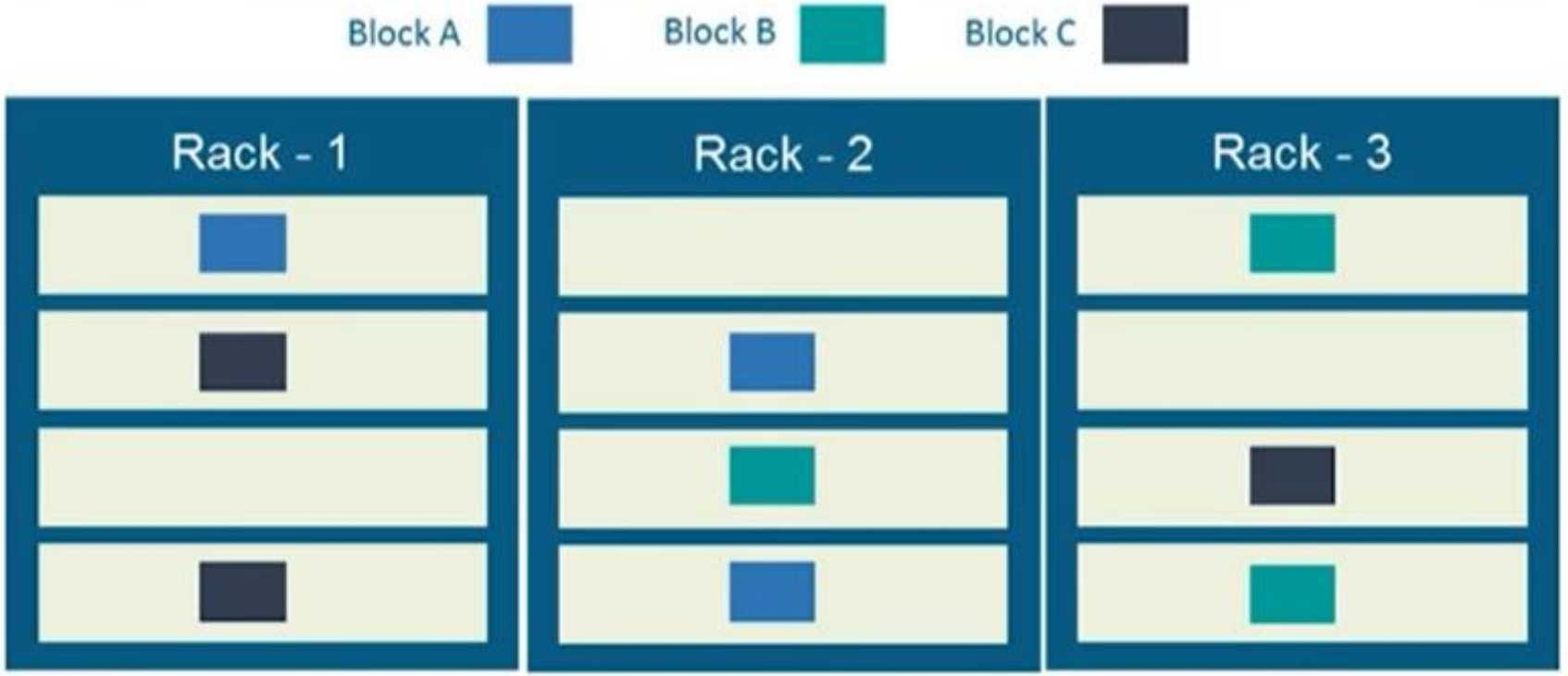
Each *data blocks are replicated* (*thrice by default*)

and are *distributed* across different DataNodes

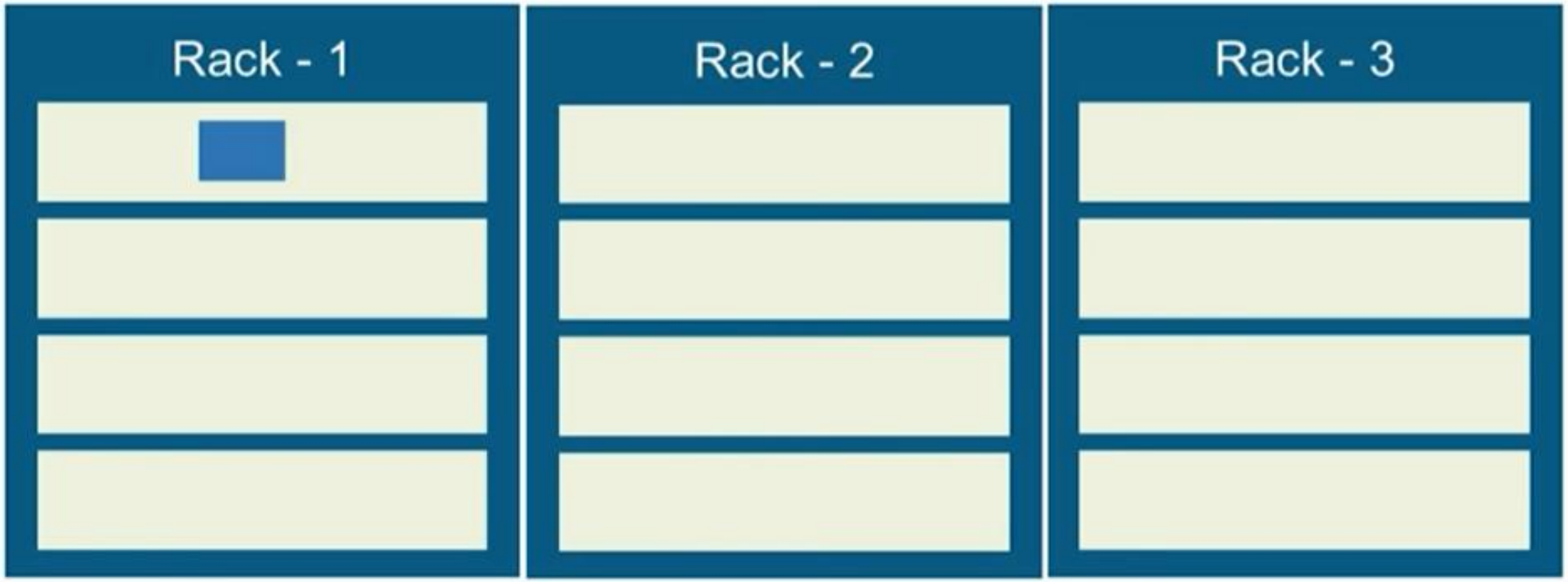


Rack Awareness

- *Rack Awareness Algorithm reduces latency* as well as *provide fault tolerance* by replicating data block
- Rack Awareness Algorithm says that the *first replica of a block will be stored on a local rack & the next two replicas will be stored on a different (remote) rack*



Rack Awareness



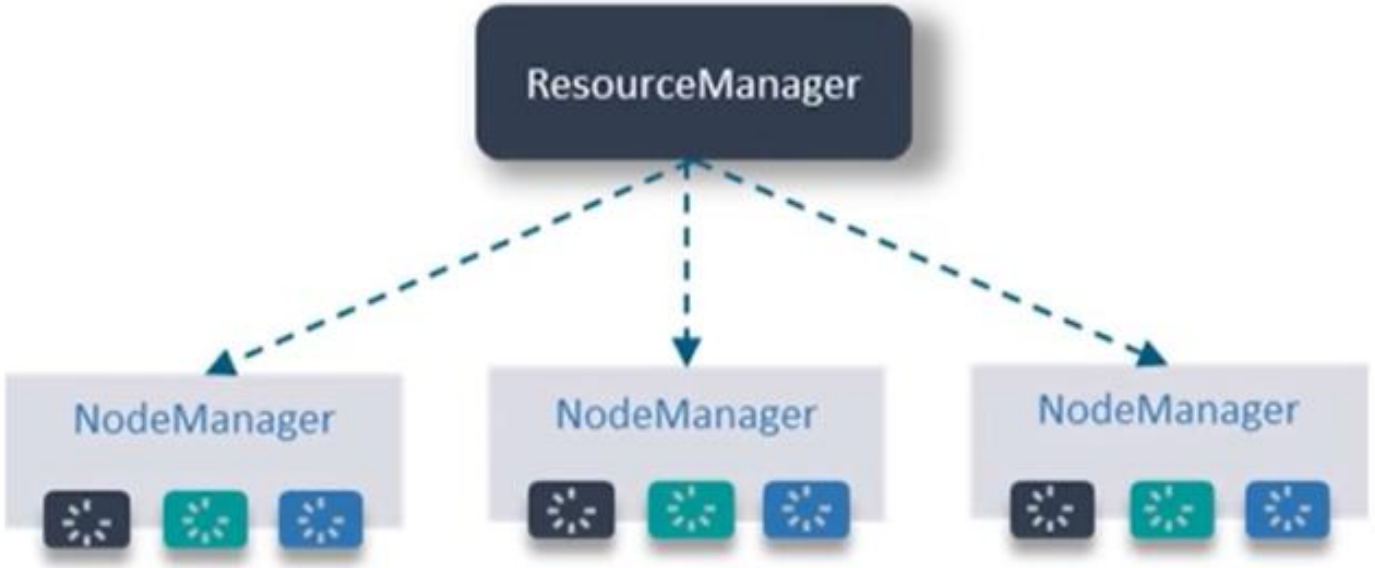
Start Hadoop Daemons

1	./sbin/start-all.sh	Starts all the Hadoop daemons(HDFS & YARN)
2	./sbin/stop-all.sh	Stops all the Hadoop daemons
3	jps	Checks all the daemons running on you machines

Writing & Deleting a File in Hadoop

What is YARN ?

- Hadoop 2.0 came up with new framework *YARN (Yet Another Resource Negotiator)*, which provides ability to run Non-MapReduce application.
- It provides a paradigm for parallel processing over Hadoop.
- YARN framework is responsible for *integration of different tools* with Hadoop like Spark, Hive, Pig.



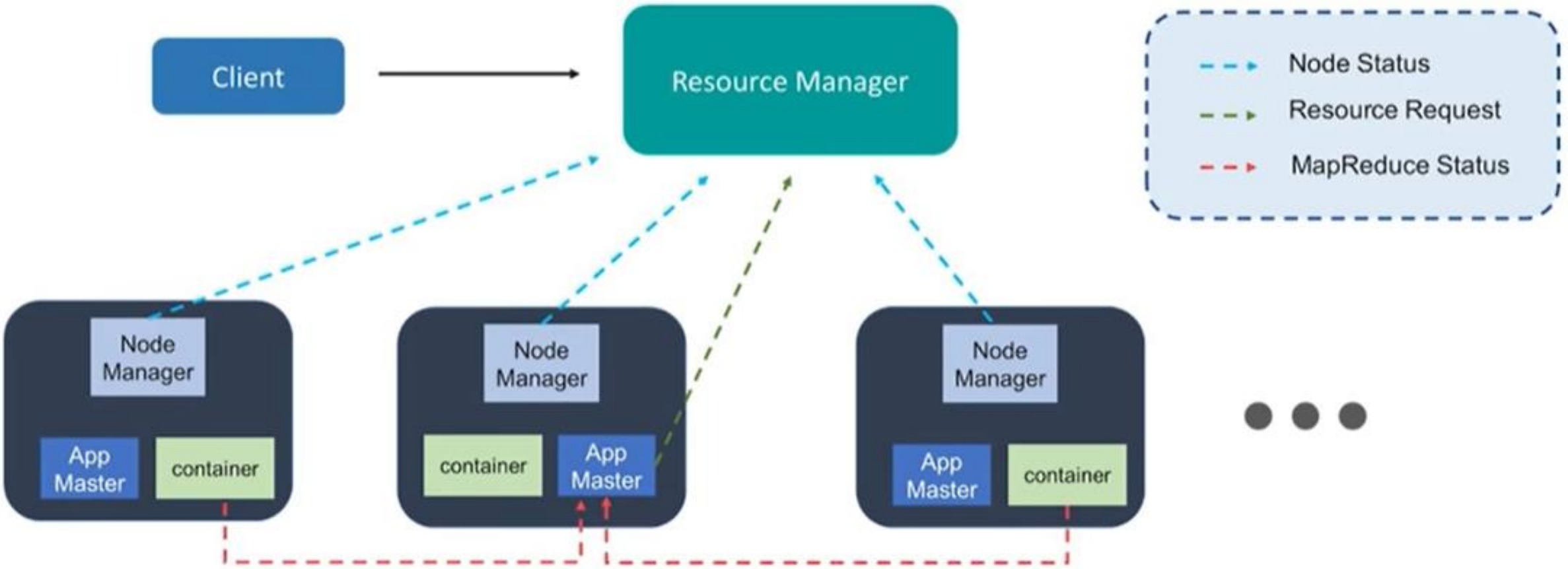
ResourceManager

NodeManager

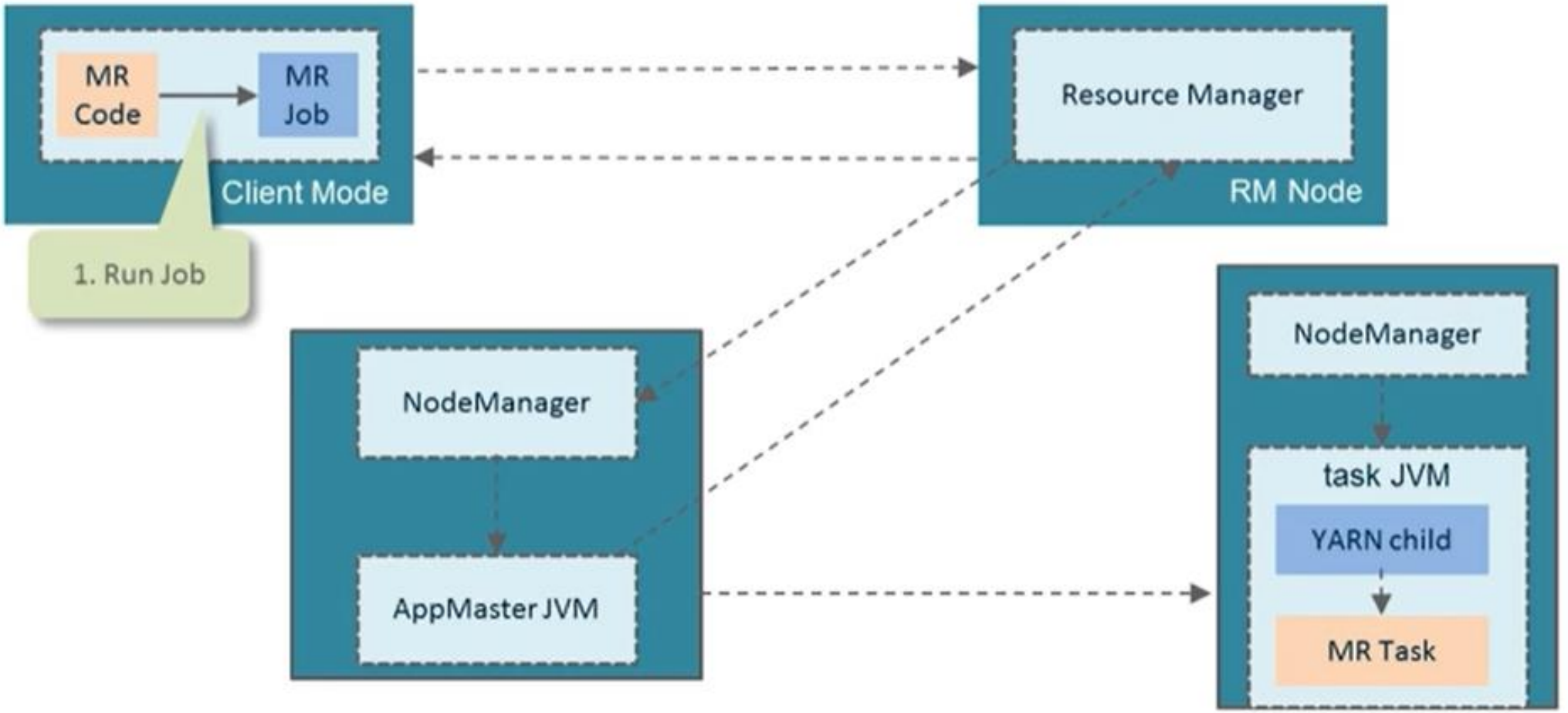
NodeManager

- Installed on every DataNode
- Responsible for execution of task on every single DataNode

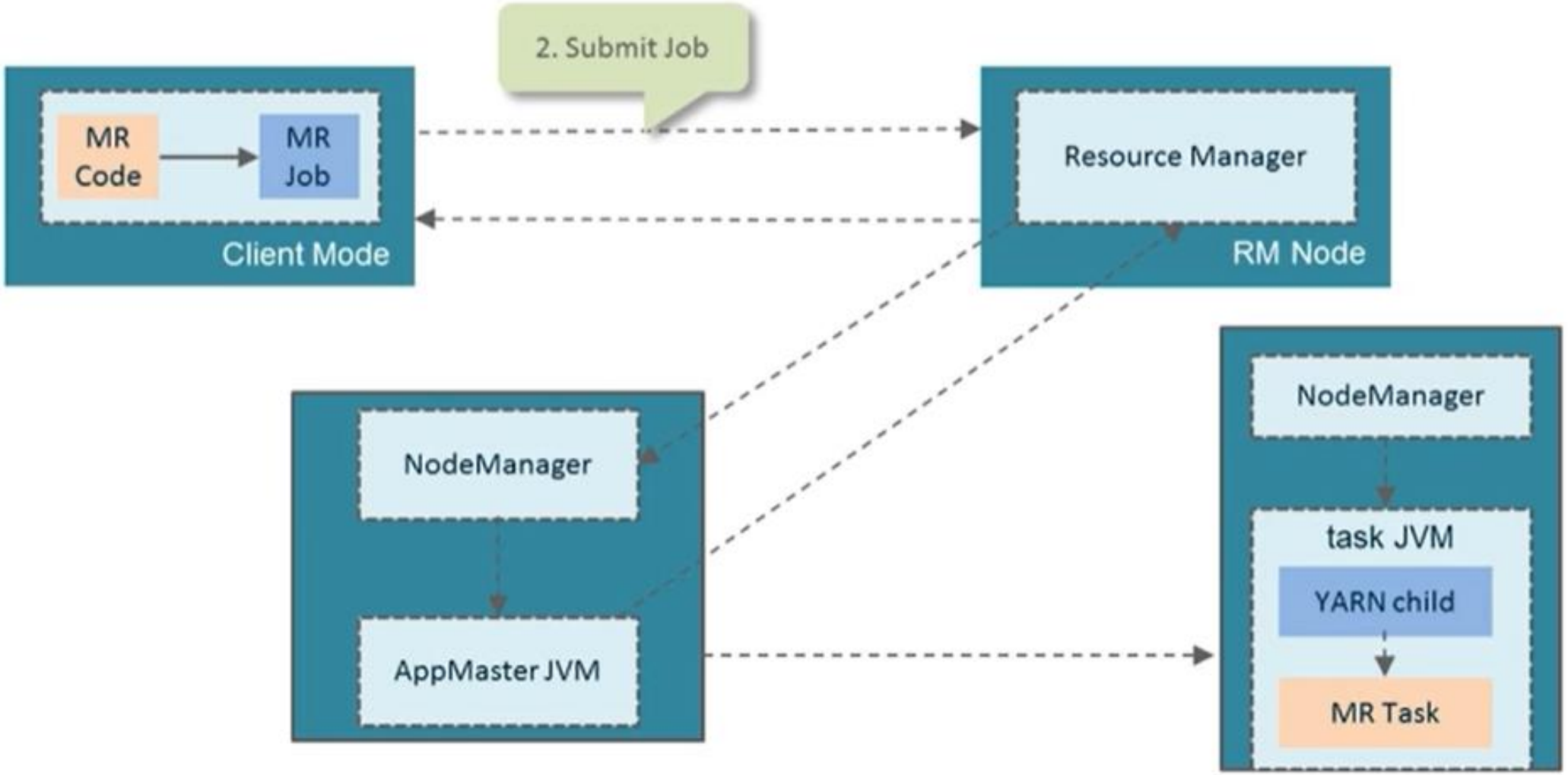
YARN Architecture in Detail



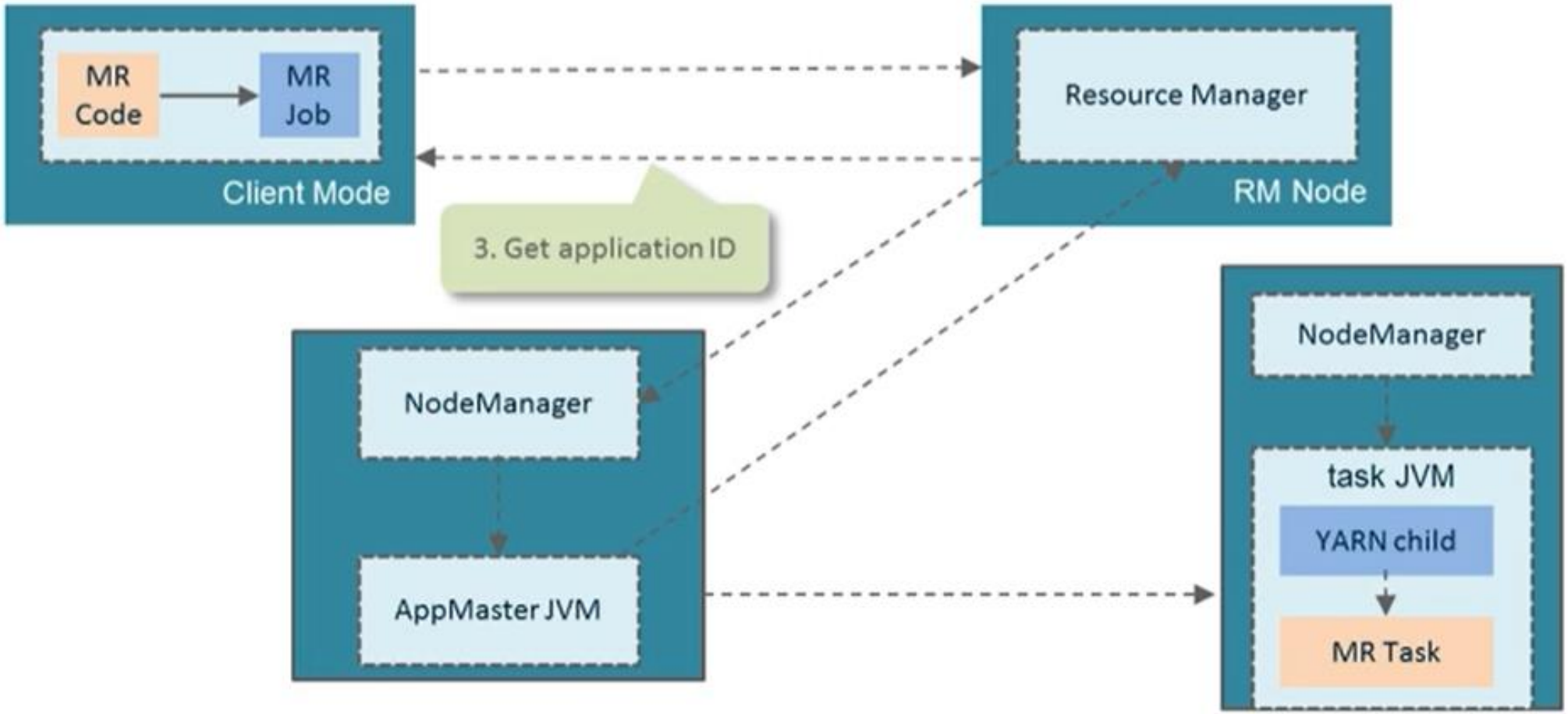
Application Submission in YARN



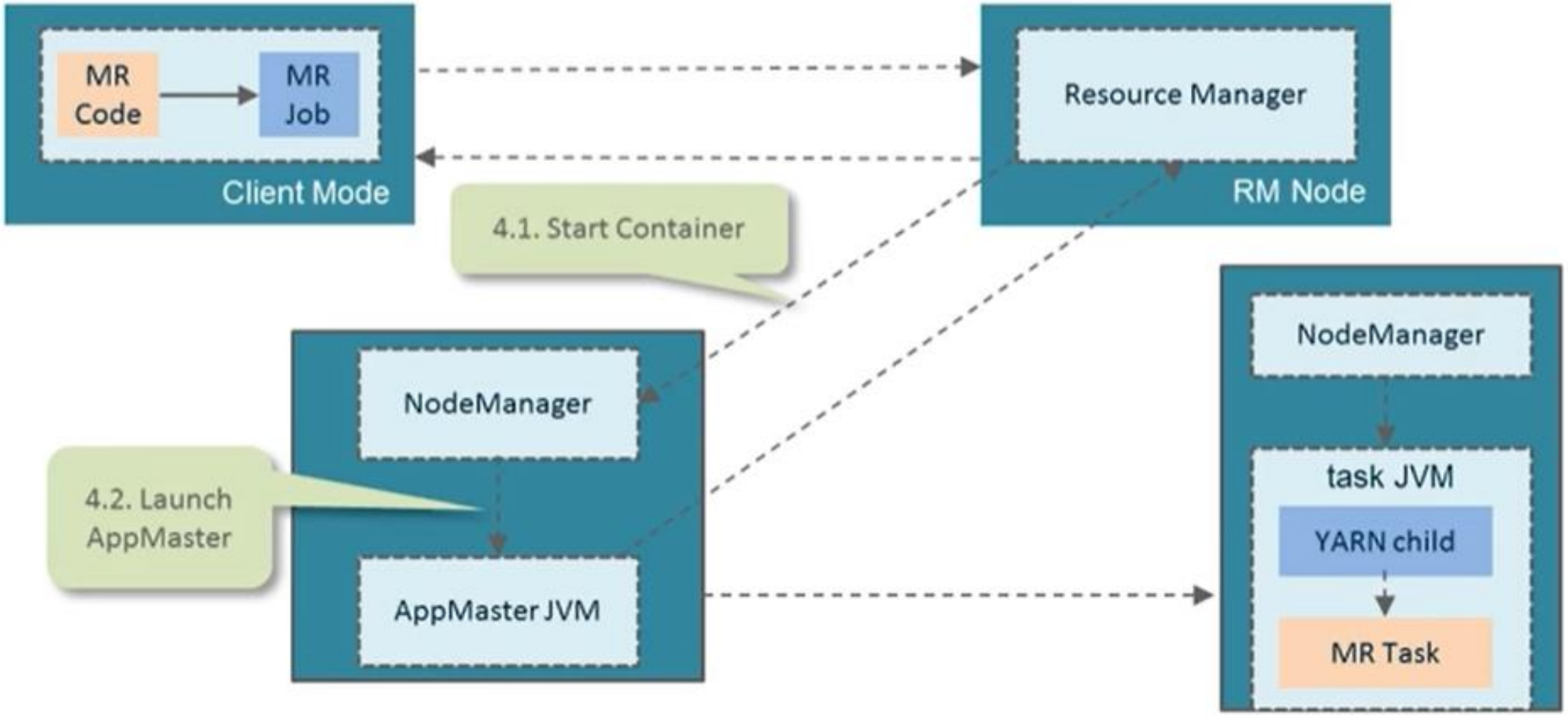
Application Submission in YARN



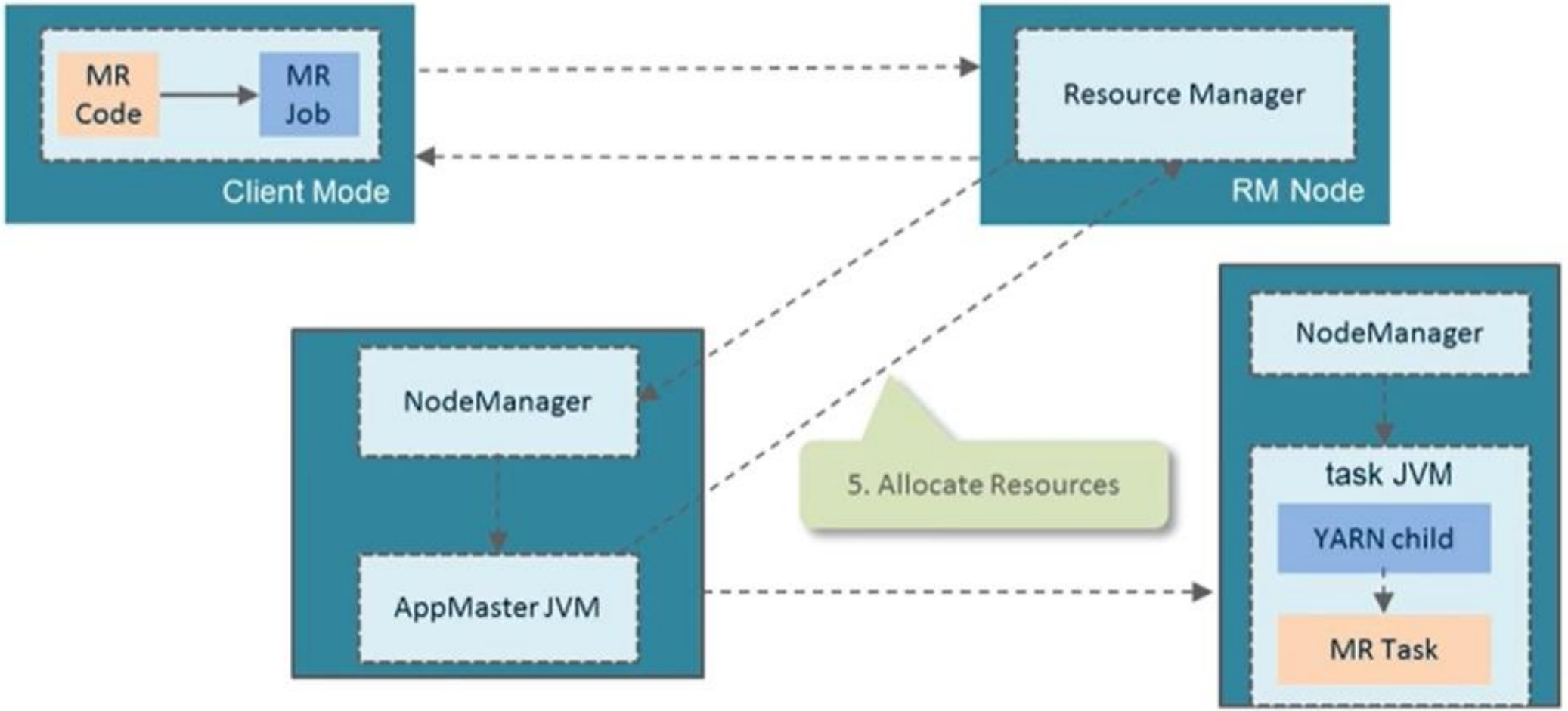
Application Submission in YARN



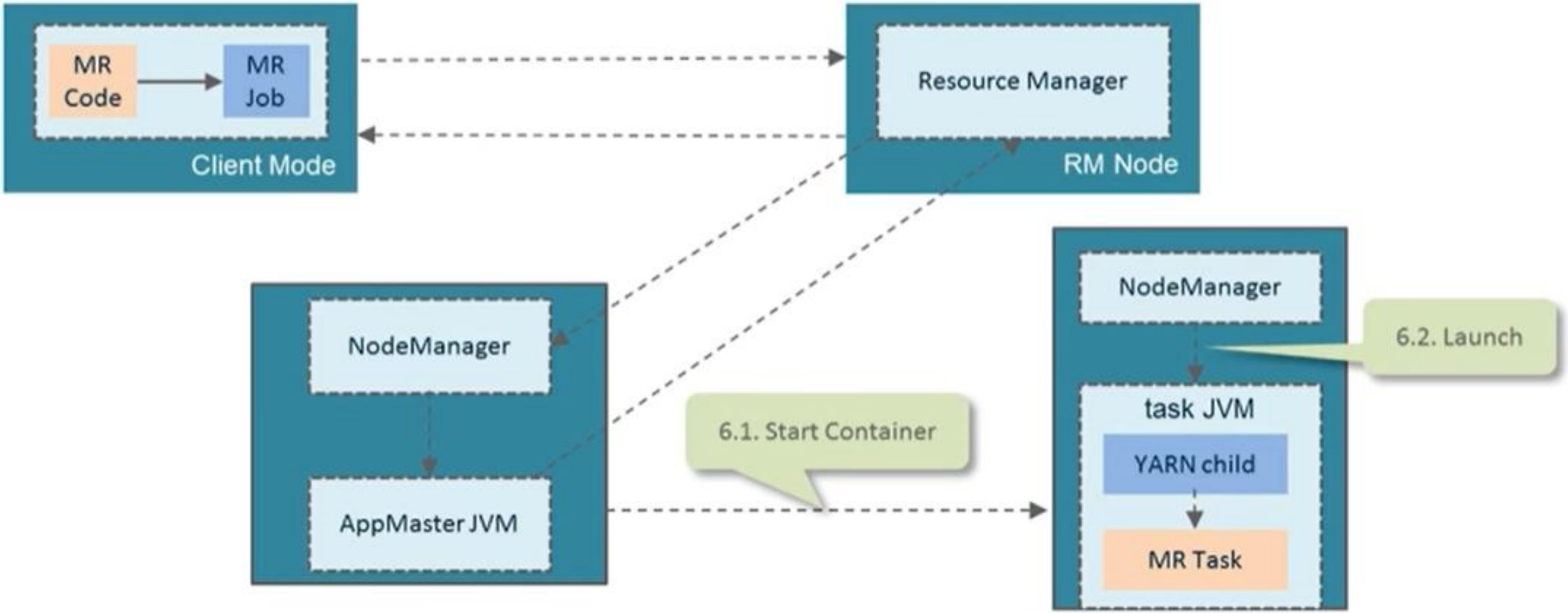
Application Submission in YARN



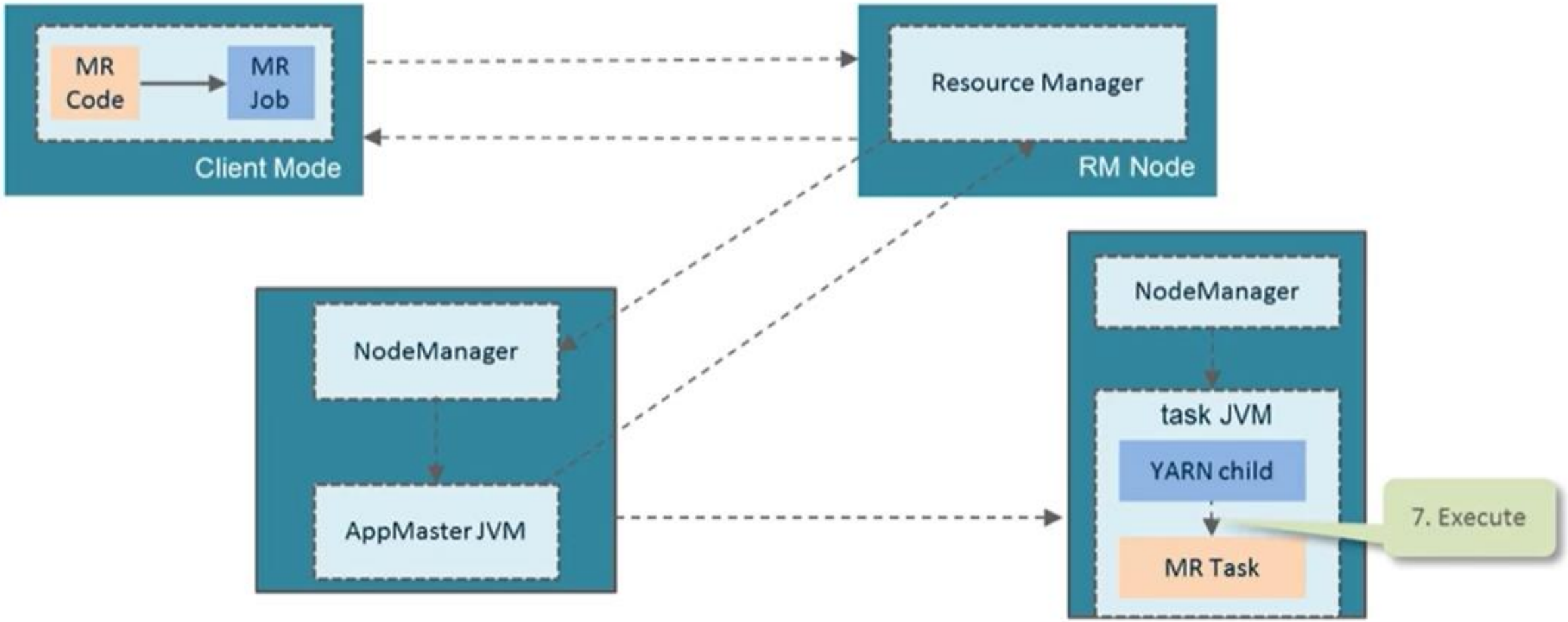
Application Submission in YARN



Application Submission in YARN

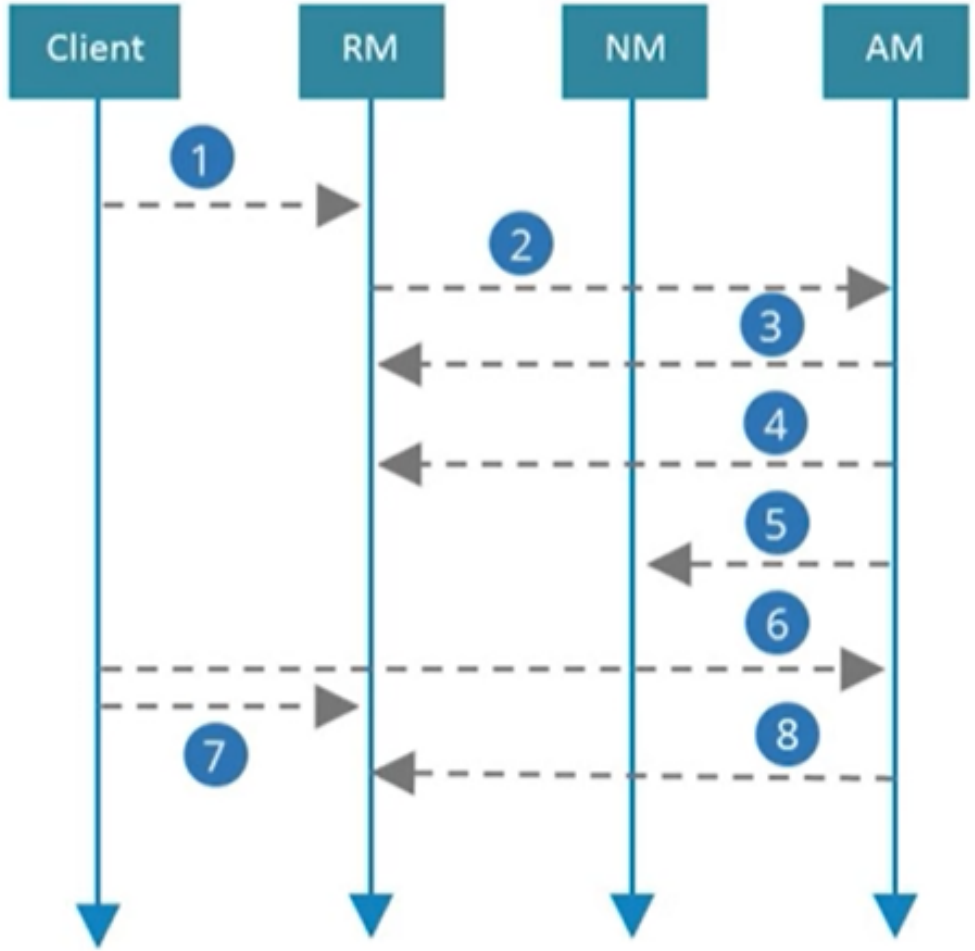


Application Submission in YARN



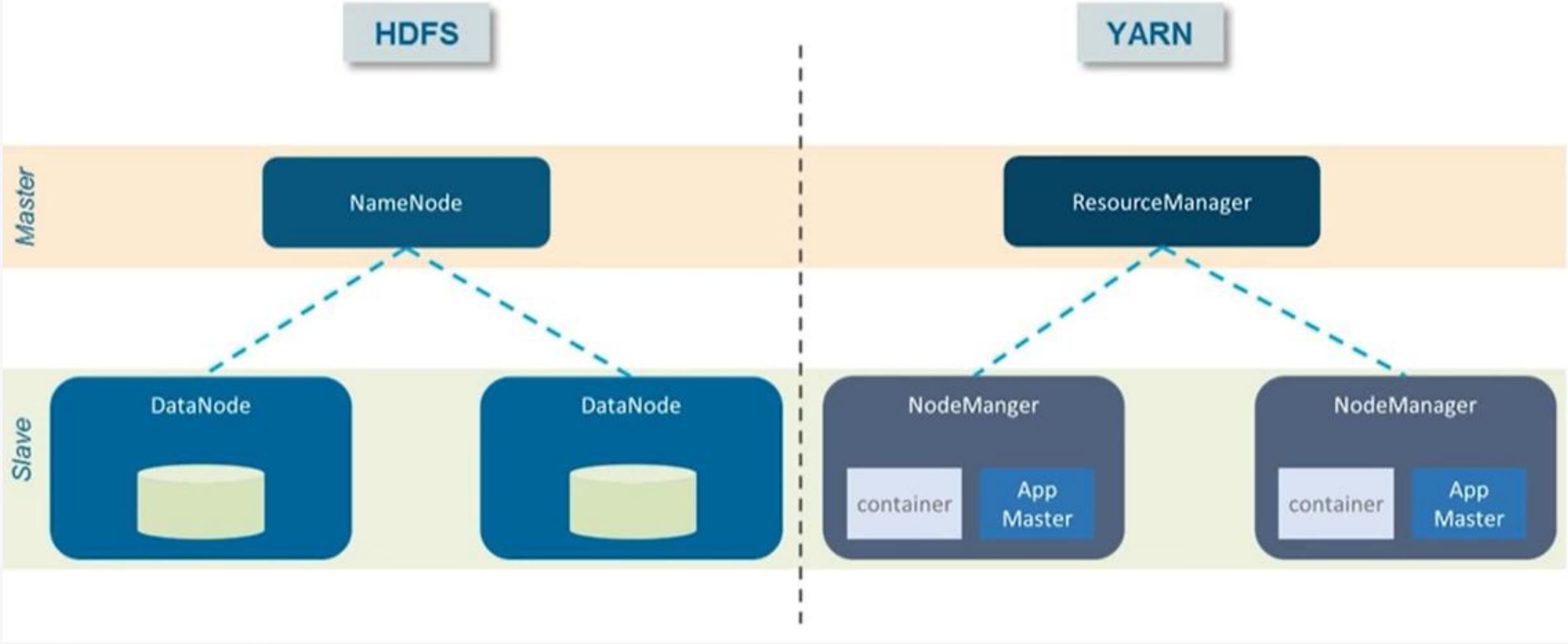
YARN Application Workflow

1. Client submits an application
2. RM allocates a container to start AM
3. AM registers with RM
4. AM asks containers from RM
5. AM notifies NM to launch containers
6. Application code is executed in container
7. Client contacts RM/AM to monitor application's status
8. AM unregisters with RM



Hadoop Cluster Architecture = HDFS + YARN

Hadoop Cluster Architecture



Hadoop Cluster Hardware Specification

NameNode

- **RAM:** 64 GB,
- **Hard disk:** 1 TB
- **Processor:** Xenon with 8 Cores
- **Ethernet:** 3 x 10 GB/s
- **OS:** 64-bit CentOS / Linux
- **Power:** Redundant Power Supply

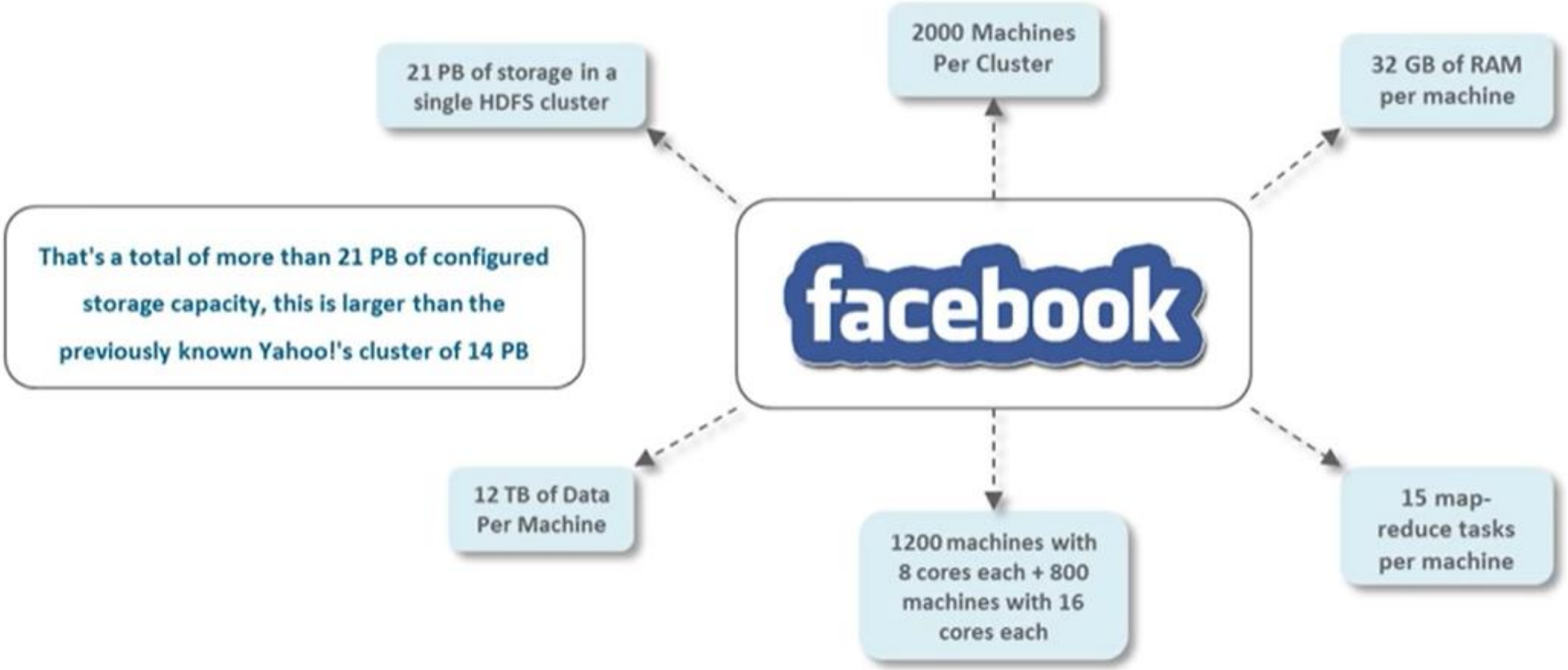
Secondary NameNode

- **RAM:** 32 GB
- **Hard disk:** 1 TB
- **Processor:** Xenon with 4 Cores
- **Ethernet:** 3 x 10 GB/s
- **OS:** 64-bit CentOS / Linux
- **Power:** Redundant Power Supply

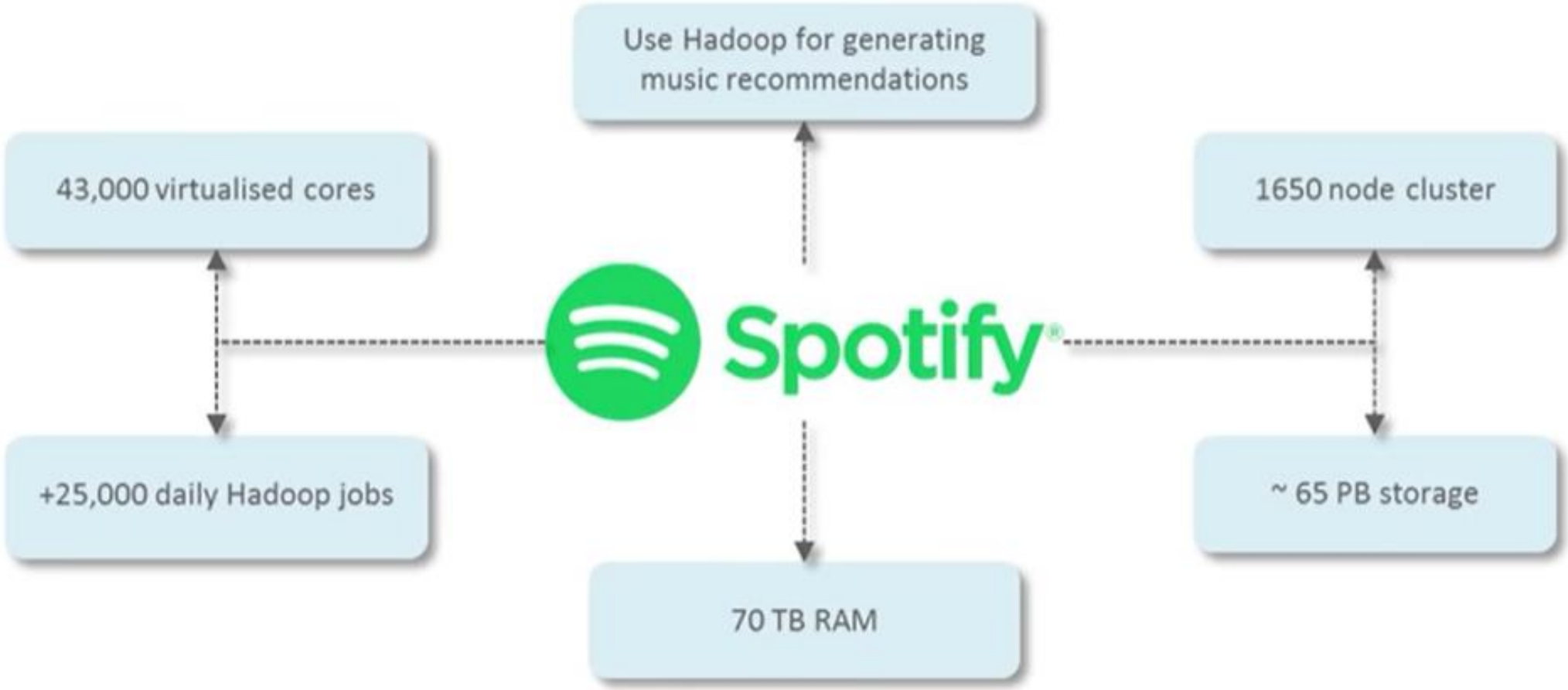
DataNode

- **RAM:** 16GB
- **Hard disk:** 6 x 2TB
- **Processor:** Xenon with 2 cores
- **Ethernet:** 3 x 10 GB/s
- **OS:** 64-bit CentOS / Linux

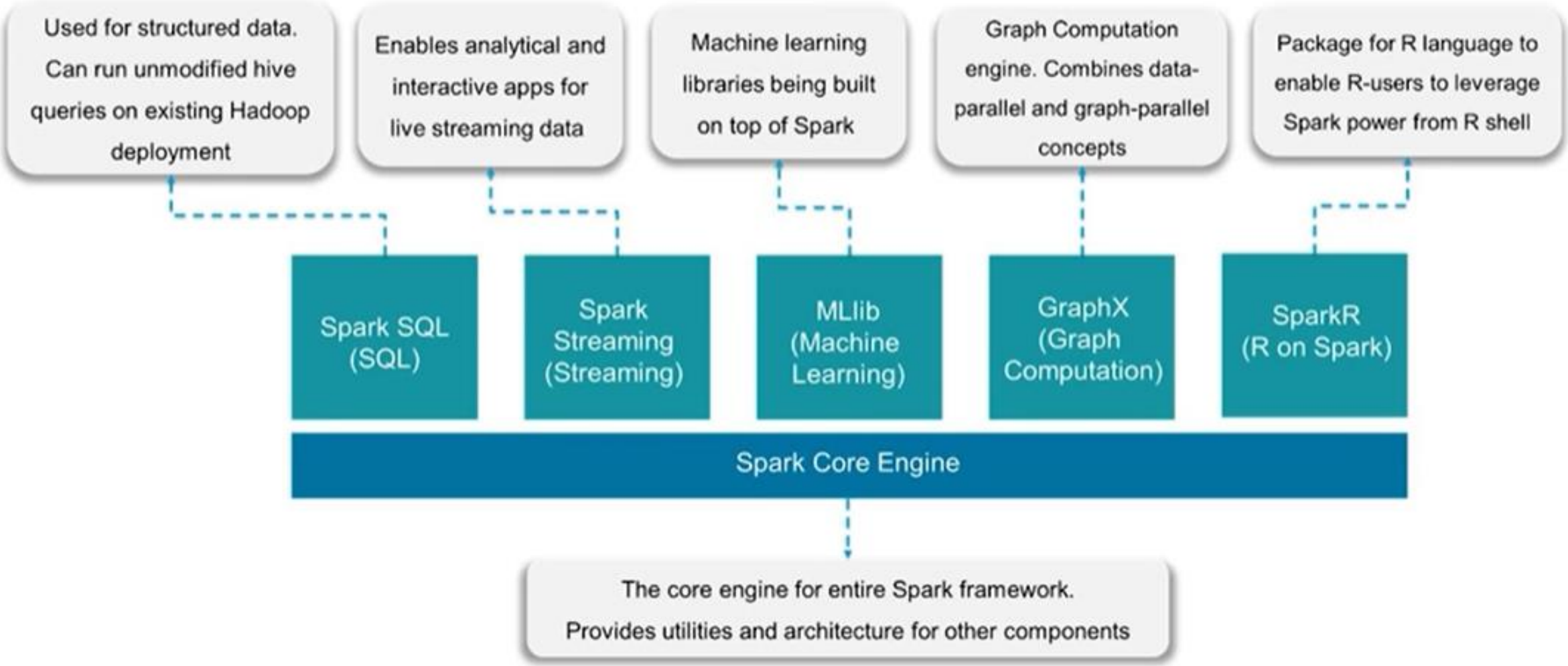
Hadoop Cluster : Facebook Use Case



Hadoop Cluster : Spotify Use Case



Spark Core Components

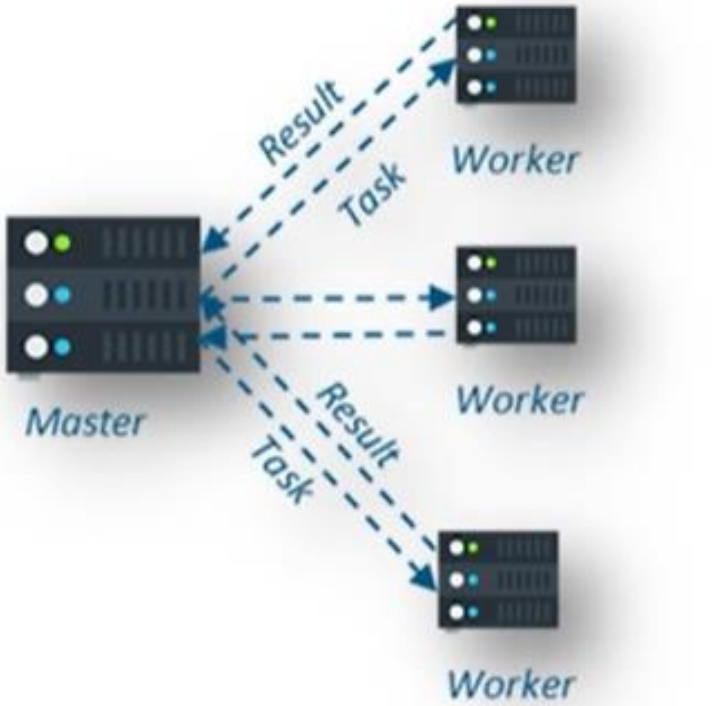


Spark Core

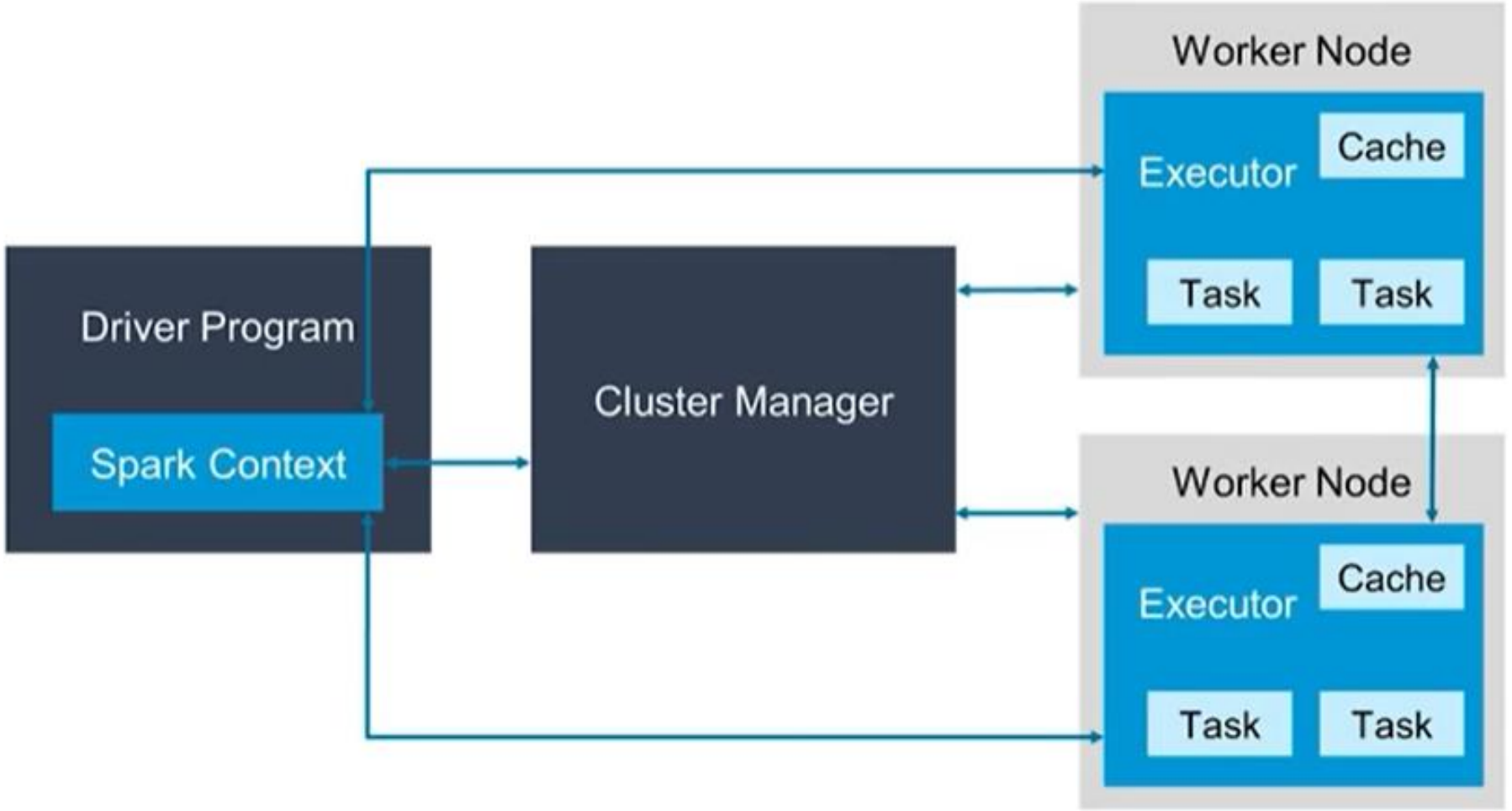
Spark Core is the base engine for large-scale parallel and distributed data processing

It is responsible for:

- Memory management and fault recovery
- Scheduling, distributing and monitoring jobs on a cluster
- Interacting with storage systems

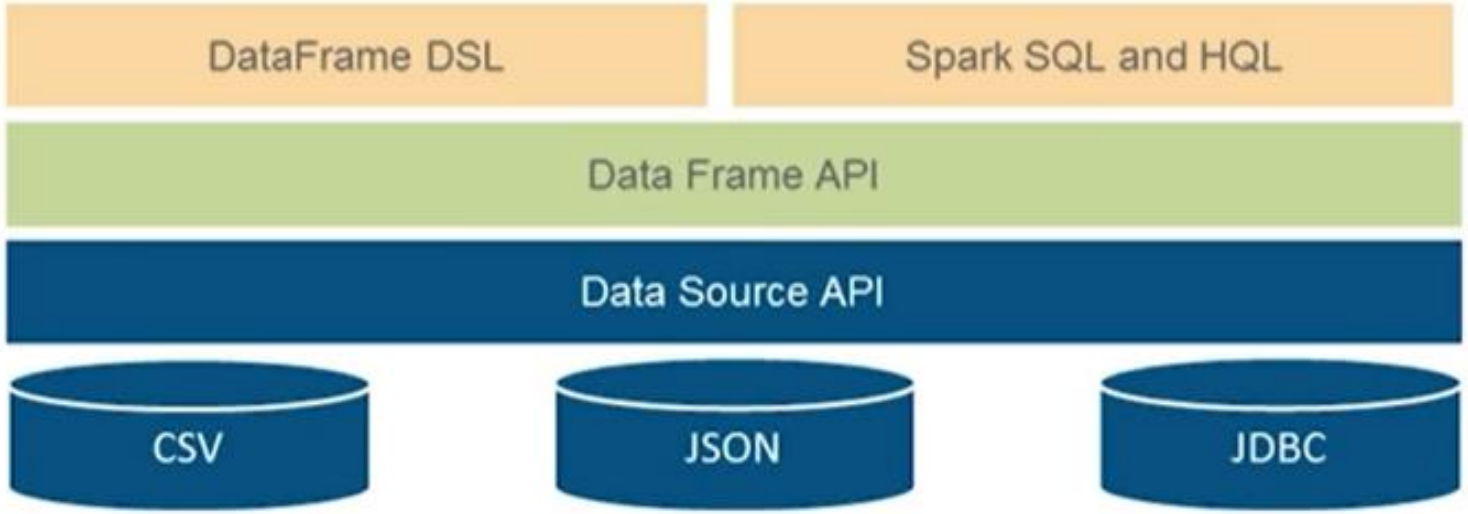


Spark Architecture



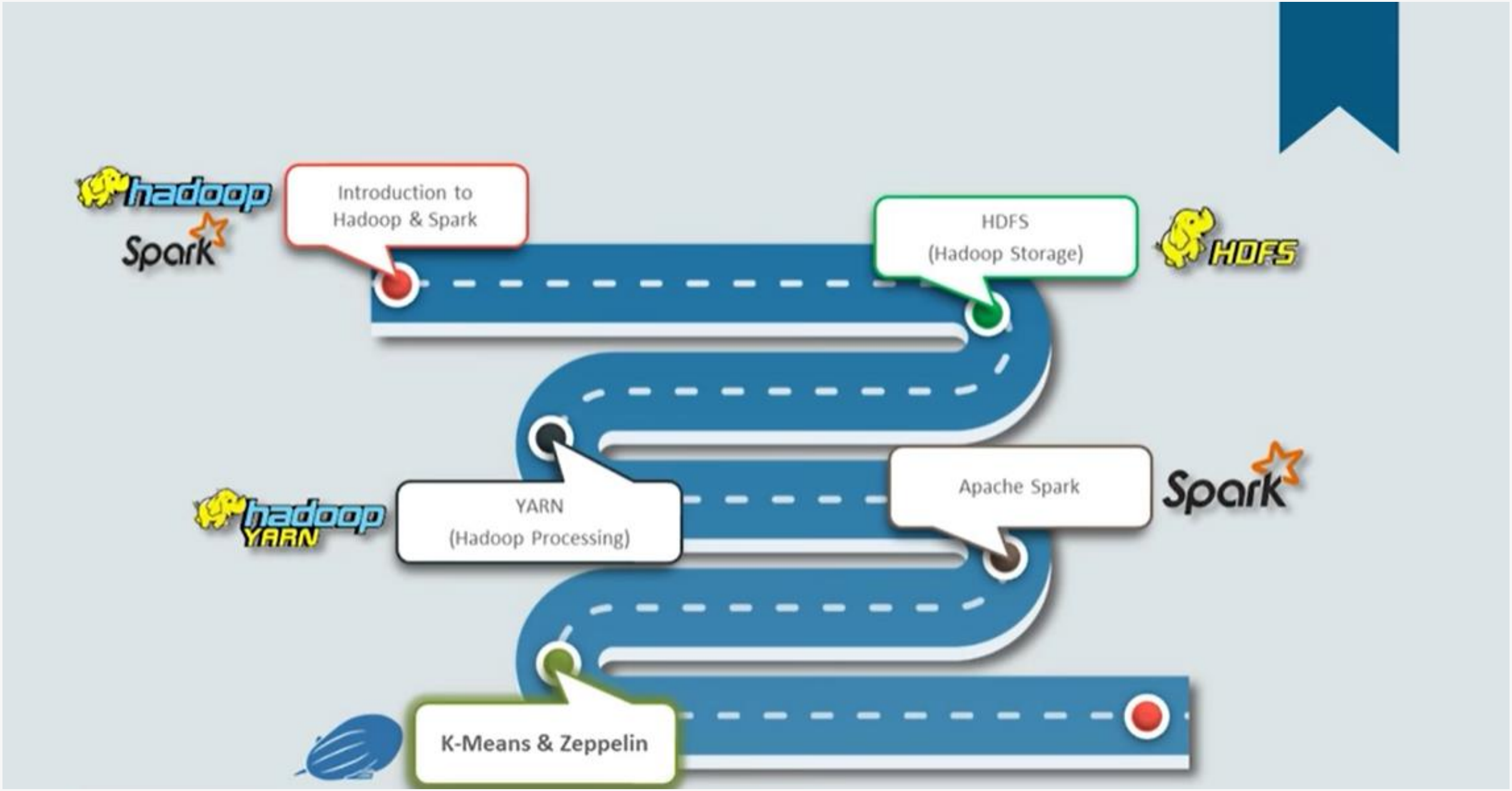
Spark SQL

- Spark SQL integrates relational processing with Spark's functional programming
- Provides support for various data sources and makes it possible to weave SQL queries with code transformations



Start Spark Daemons

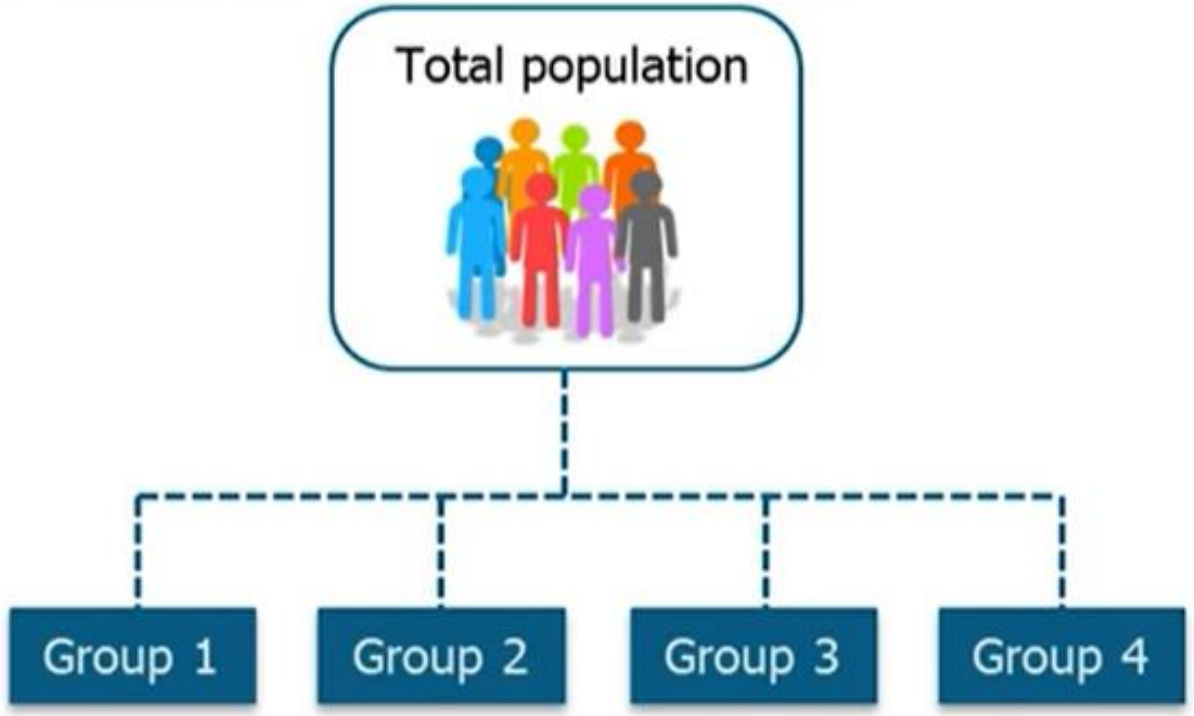
1	./sbin/start-all.sh	Starts all the Spark daemons(Master & Worker)
2	jps	Checks all the daemons running on you machines
3	./bin/spark-shell	Starts the Spark Shell


Introduction to
Hadoop & Spark
HDFS
(Hadoop Storage)
YARN
(Hadoop Processing)
Apache Spark
K-Means & Zeppelin

K-Means Clustering

The process by which *objects are classified into a predefined number of groups* so that they are as much dissimilar as possible from one group to another group, but as much similar as possible within each group

- The objects in group 1 should be as similar as possible
- But there should be much difference between an object in group 1 and group 2
- The attributes of the objects are allowed to determine which objects should be grouped together

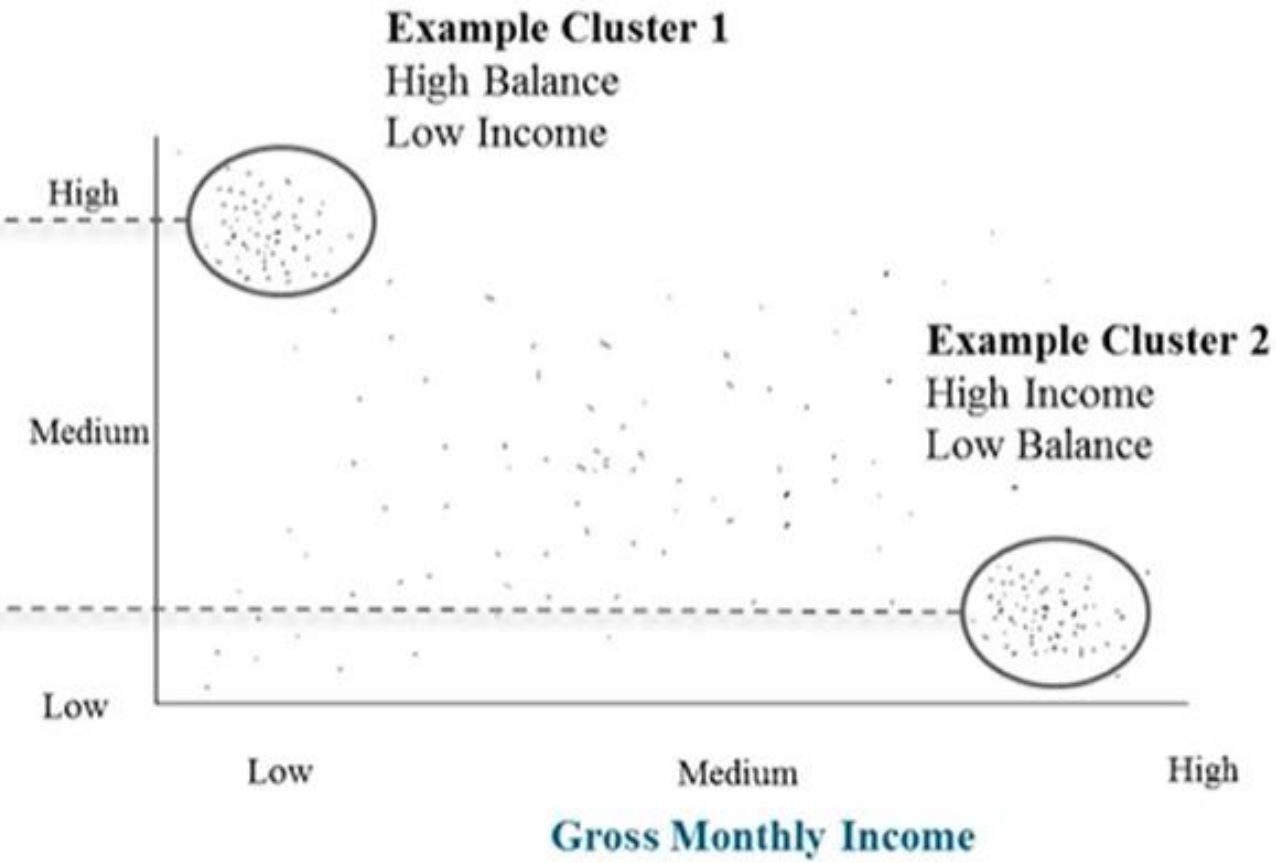


K-Means Clustering

- Consider a comparison on Income & Balance:

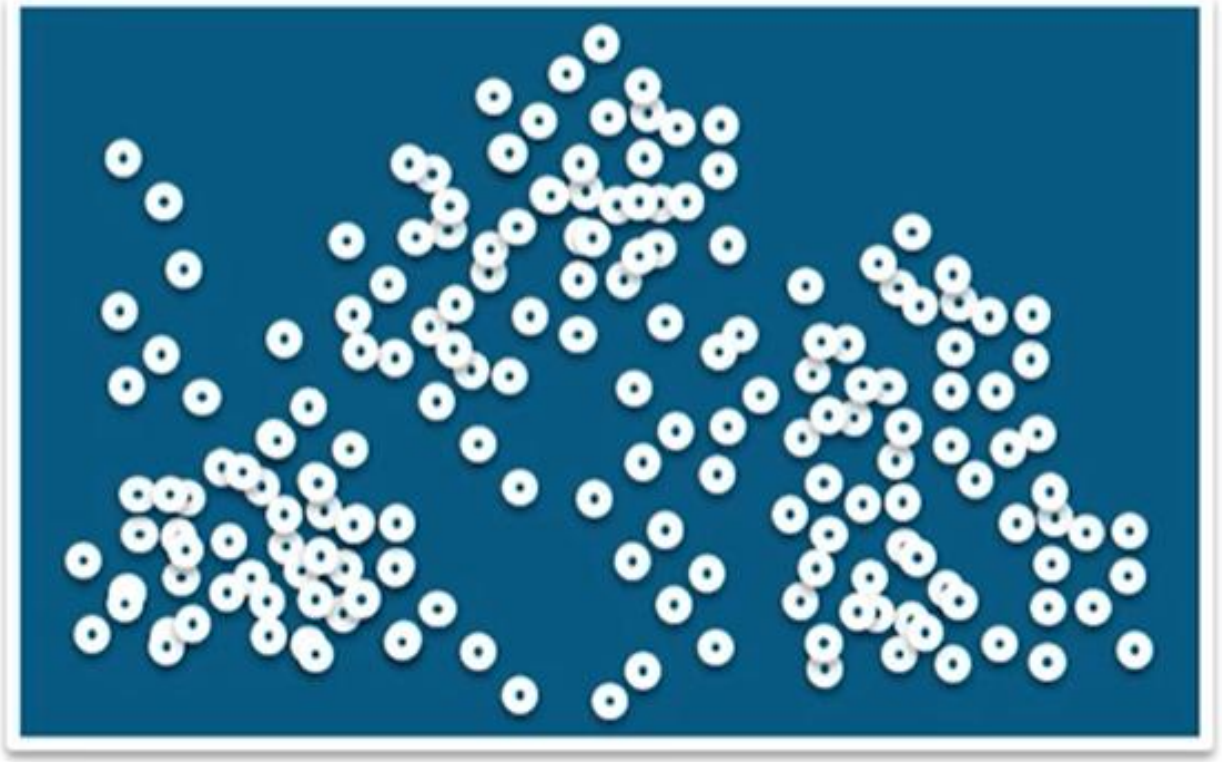
The objects in Cluster 1 have similar characteristics (High Income and Low balance)

Also the objects in Cluster 2 have the same characteristic (High Balance and Low Income)



Example

- The plot of students in an area is as given below



Example

- Using k-means clustering we got output as: