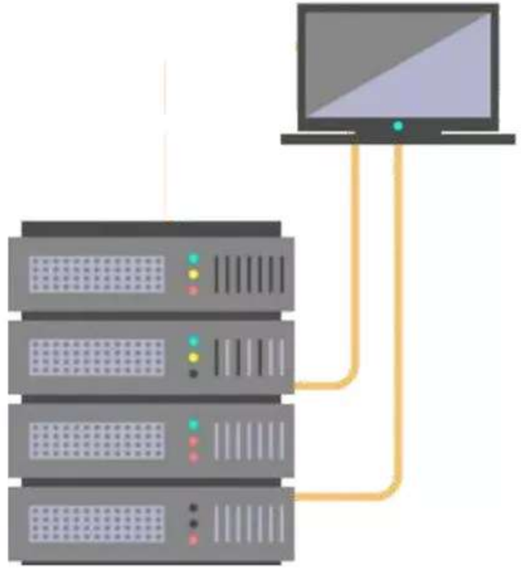
Hadoop Architecture

simplilearn

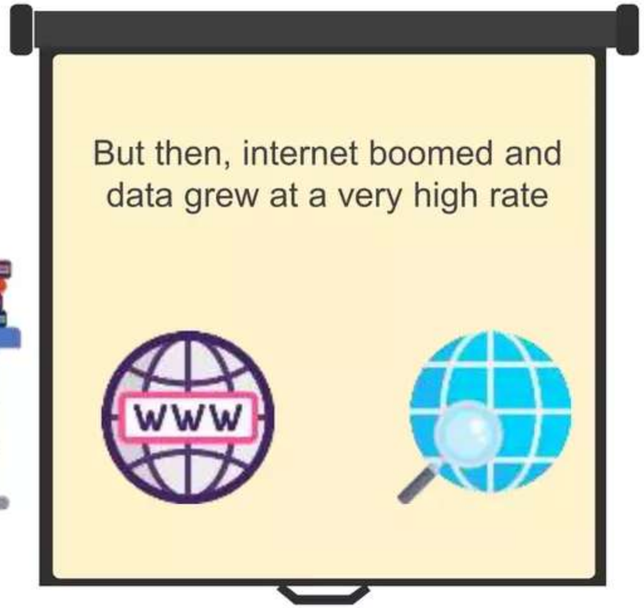
How Big Data evolved?

Back in the days when there was no internet, data used to be less and was often structured

This data was easily stored on a central sever storage

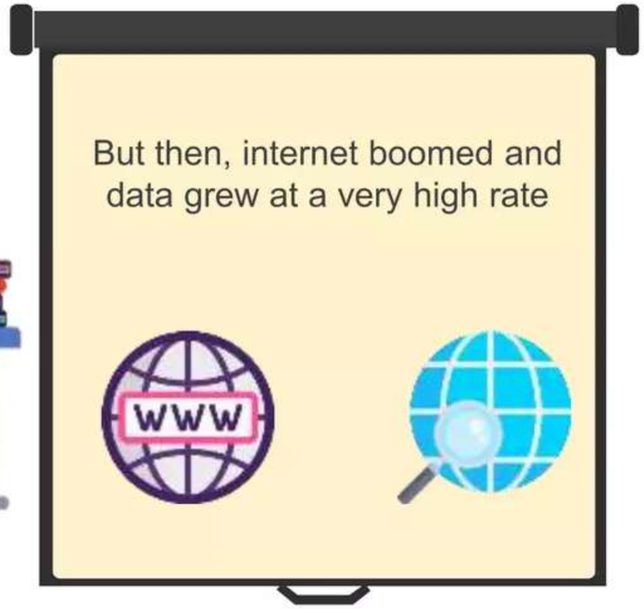


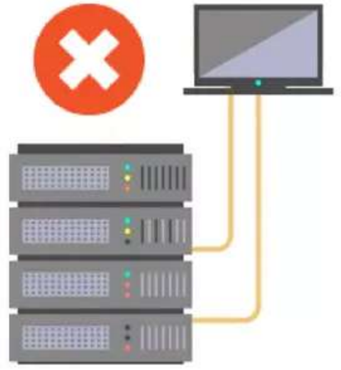
How Big Data evolved?



A lot of semi-structured and unstructured data was being generated

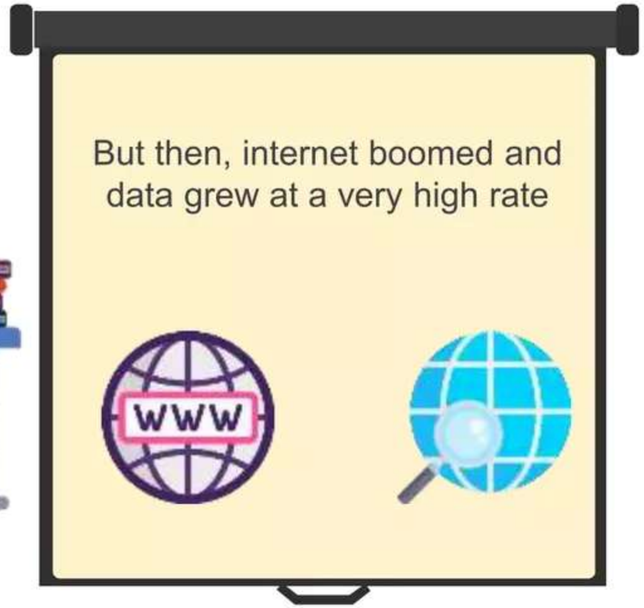
How Big Data evolved?





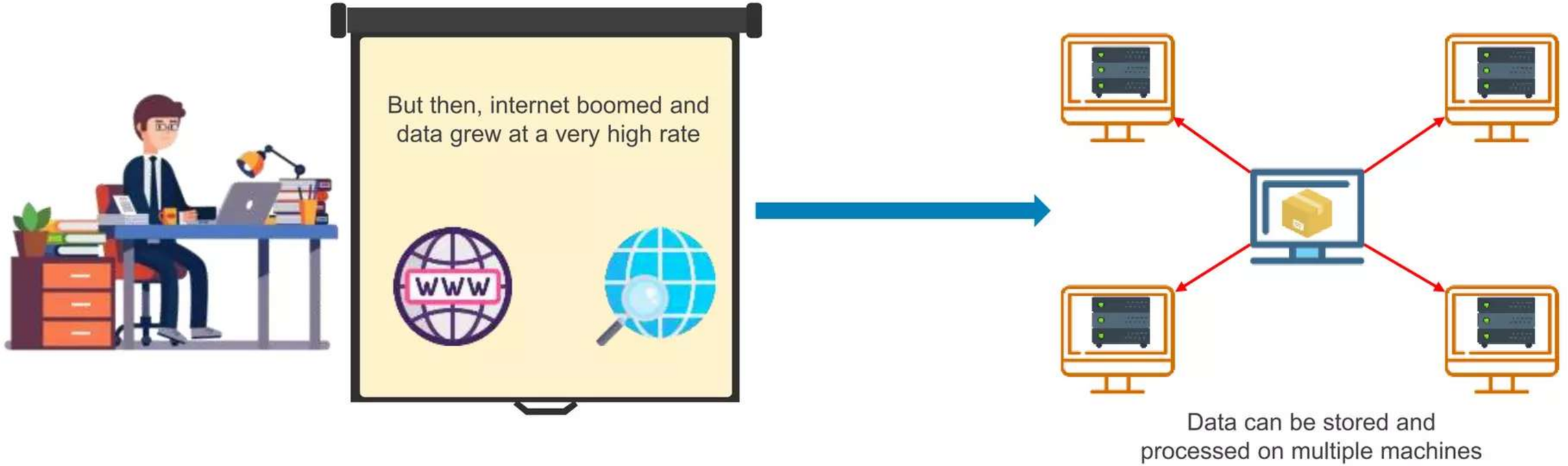
Storing such huge volumes of data on a single server was not an efficient way

How Big Data evolved?



There was a need for distributed storage machines where data could be stored and processed parallelly

How Big Data evolved?



How Big Data evolved?

But then, internet boomed and data grew at a very high rate

Solution

BIG DATA TECHNOLOGIES

Hadoop is a framework that allows distributed storage and parallel processing of big data

What's in it for you?

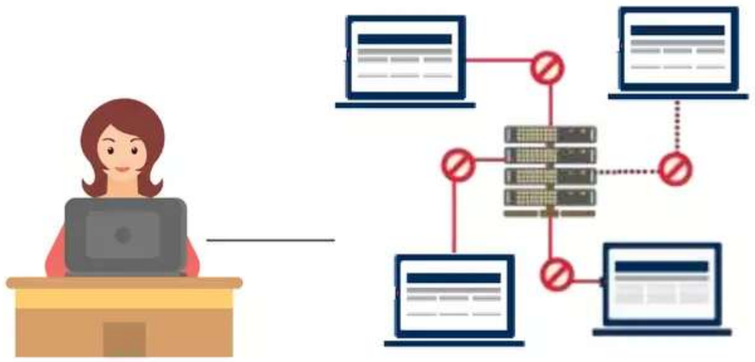
- What is Hadoop?
- Components of Hadoop
- What is HDFS?
- HDFS Architecture
- Hadoop MapReduce
- Hadoop MapReduce Example
- Hadoop YARN
- Demo on MapReduce

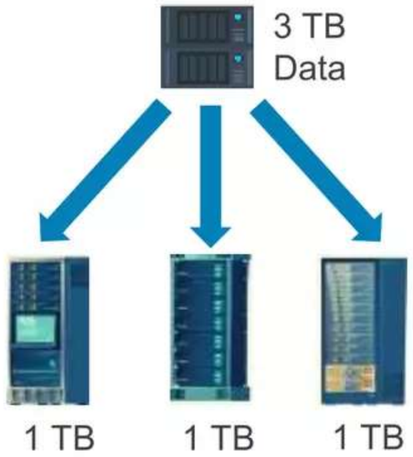

What is Hadoop?
simplilearn

What is Hadoop?

Hadoop is a framework that allows you to store large volumes of data on several node machines

It also helps in processing the data in a parallel manner

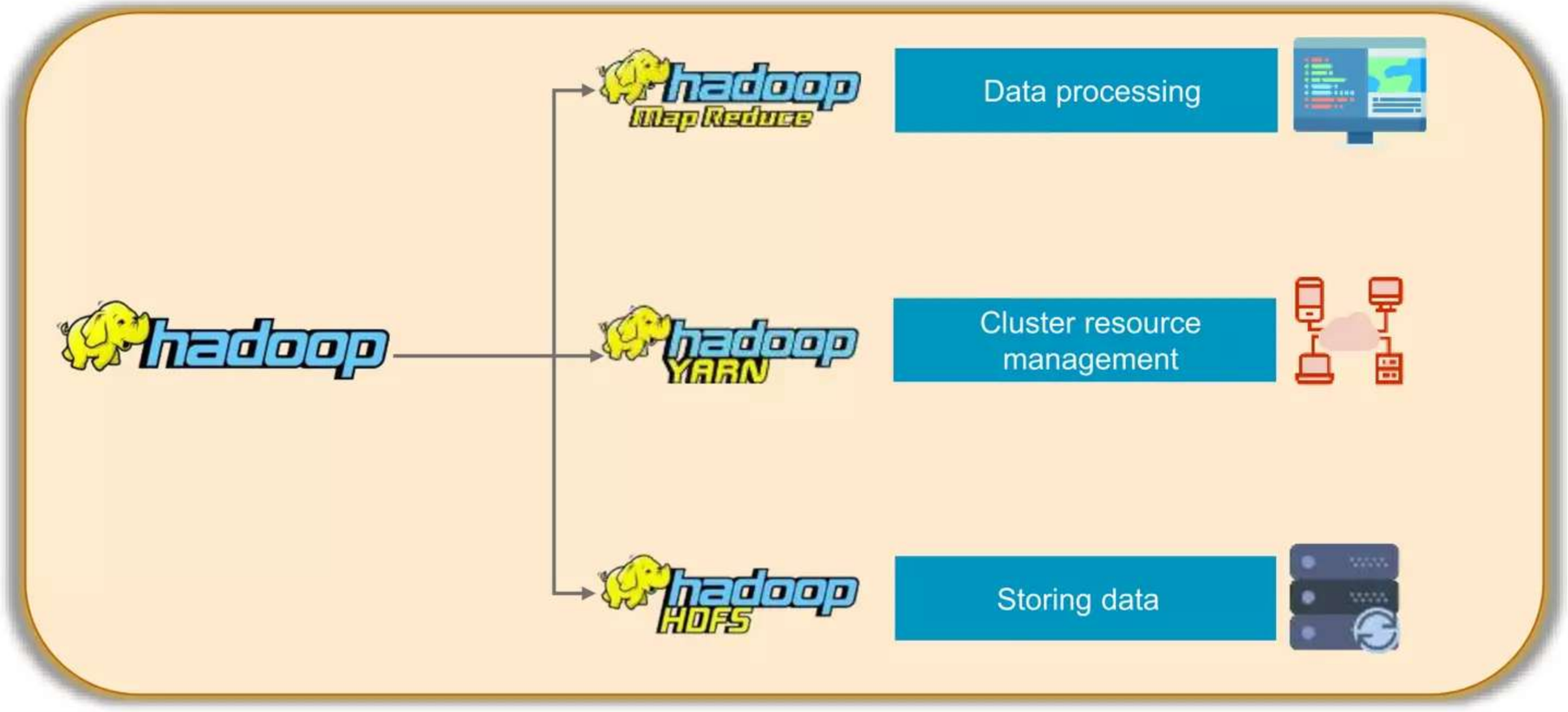




simplilearn


Components of Hadoop
simplilearn

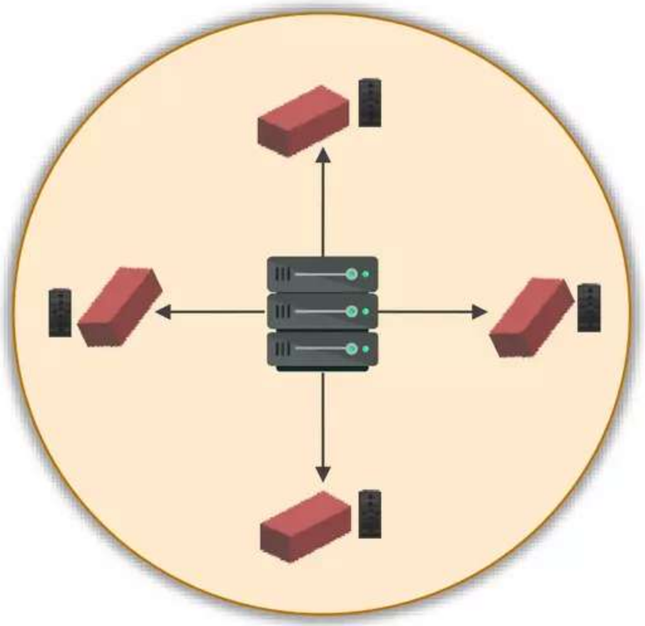
Components of Hadoop




What is HDFS?
simplilearn

What is HDFS?

Hadoop Distributed File System (HDFS) is the storage layer of Hadoop that stores data in multiple data servers



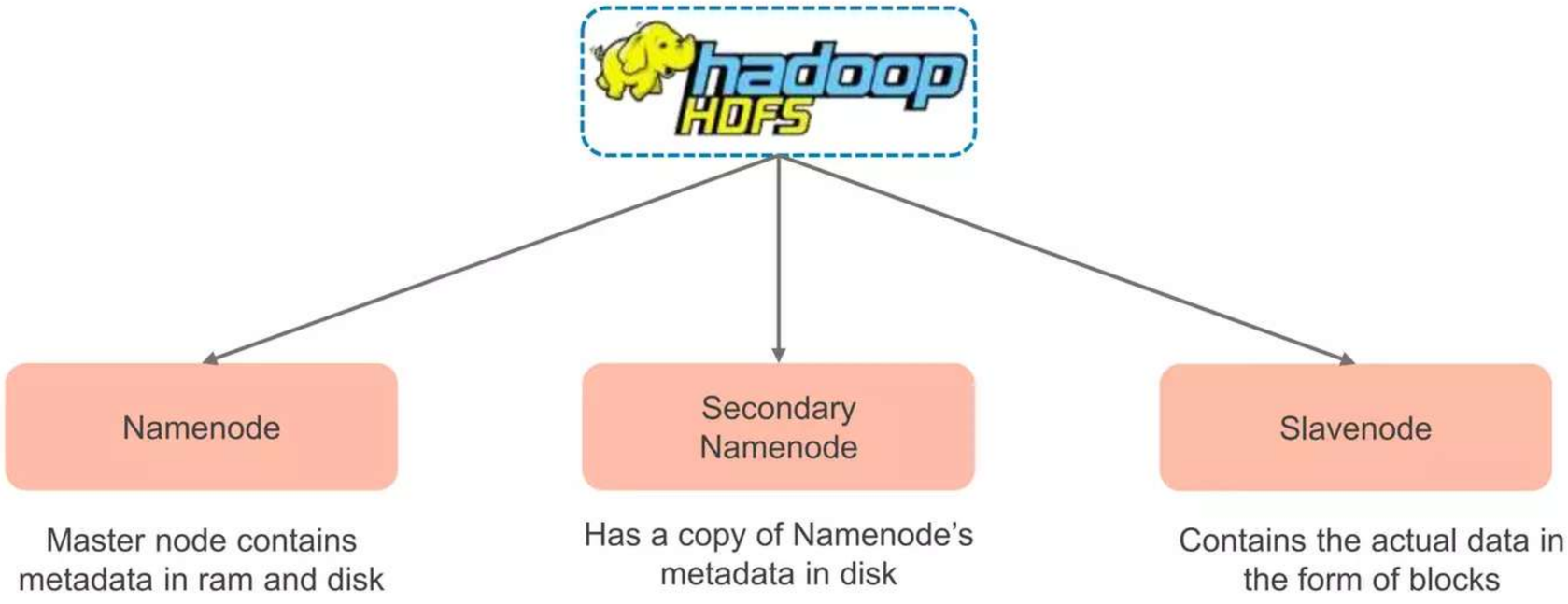
Data is divided into multiple blocks

Stores them over multiple nodes of the cluster

What is HDFS?

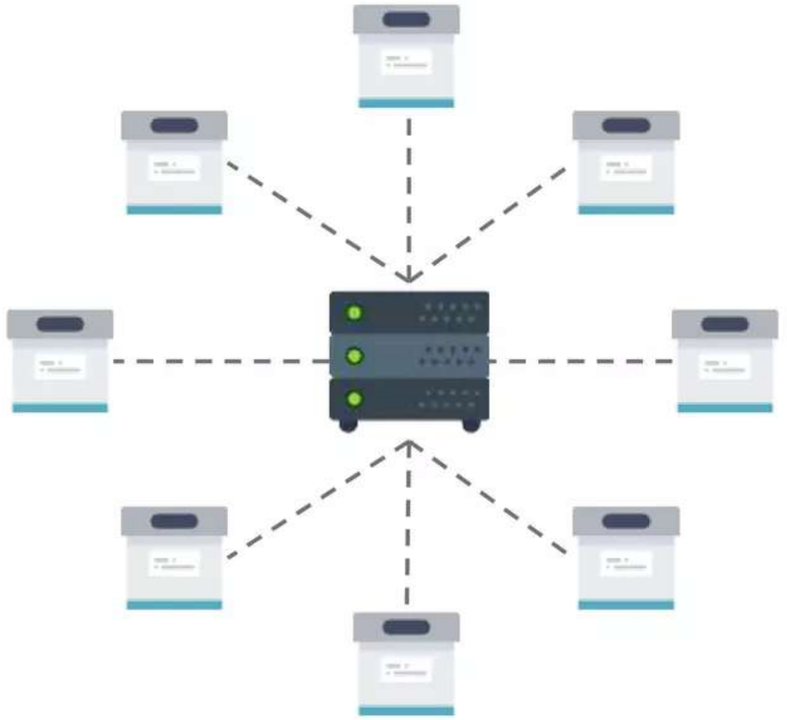
Hadoop Distributed File System (HDFS) is the storage layer of Hadoop that stores data in multiple data servers

3 core components

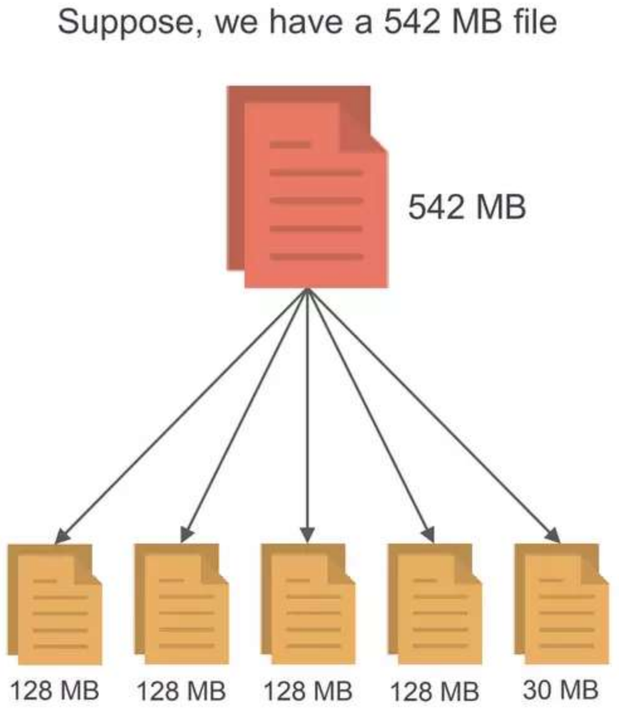



HDFS Blocks
simplilearn

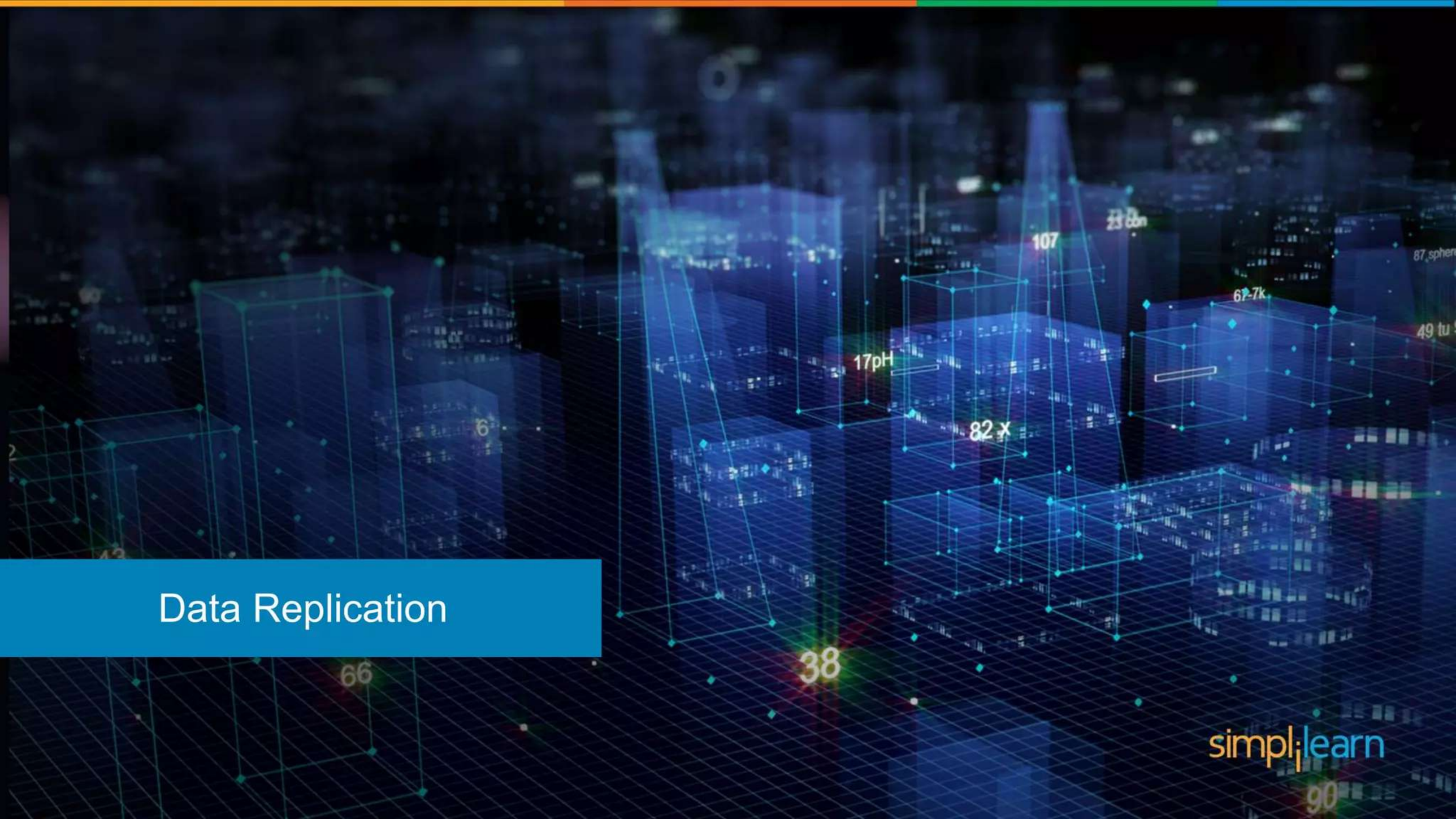
HDFS Blocks



HDFS divides large data into different blocks



Each block by default has 128 MB's of data


Data Replication
simplilearn

Data Replication in HDFS

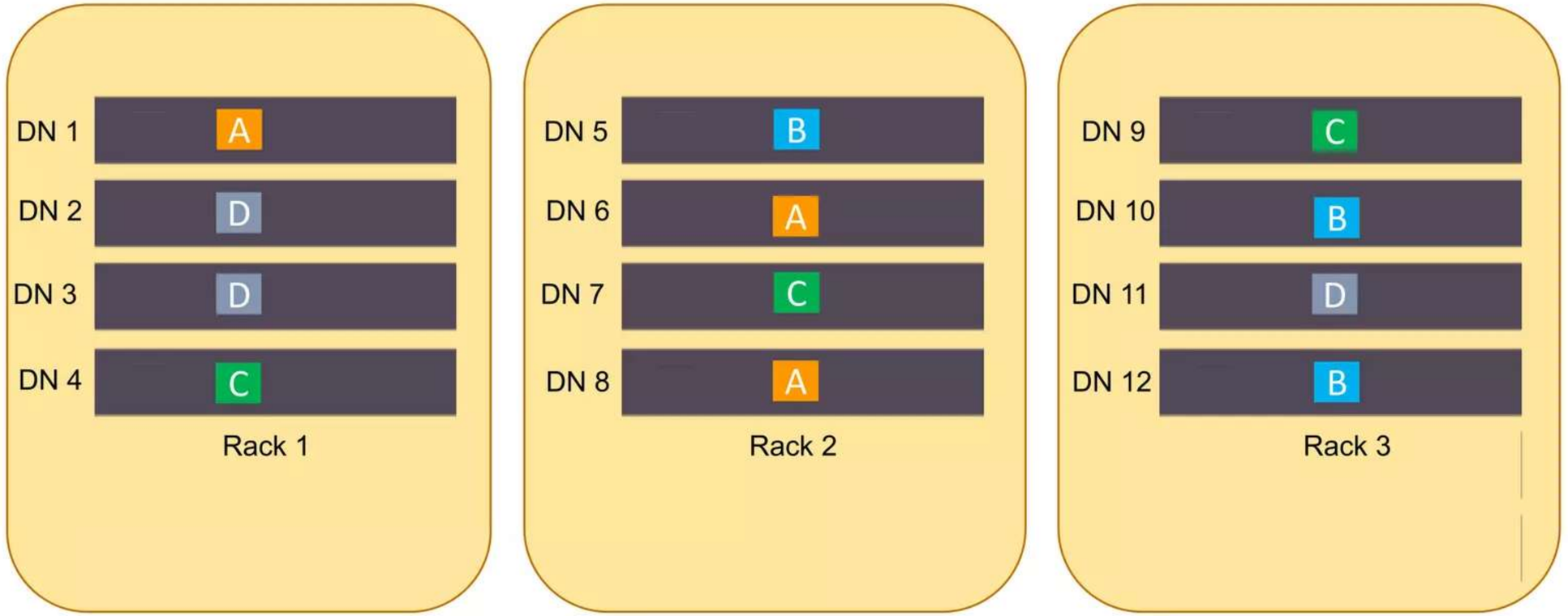
DN -------> Datanode

A -------> Block A

B -------> Block B

C -------> Block C

D -------> Block D



Do you understand what's happening here?

Each block of data is being replicated thrice on different datanodes present in different racks

simplilearn

Data Replication in HDFS

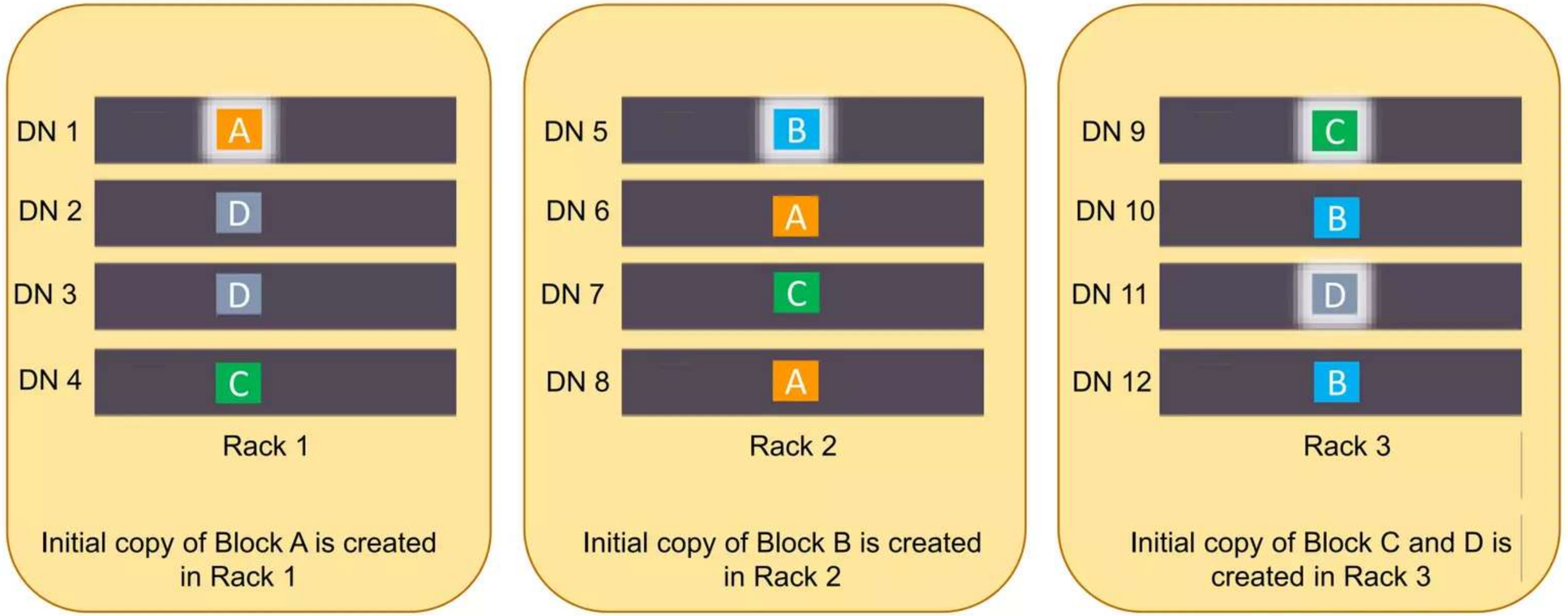
DN -------> Datanode

A -------> Block A

B -------> Block B

C -------> Block C

D -------> Block D



Two identical blocks cannot be placed on the same datanode

simplilearn

Data Replication in HDFS

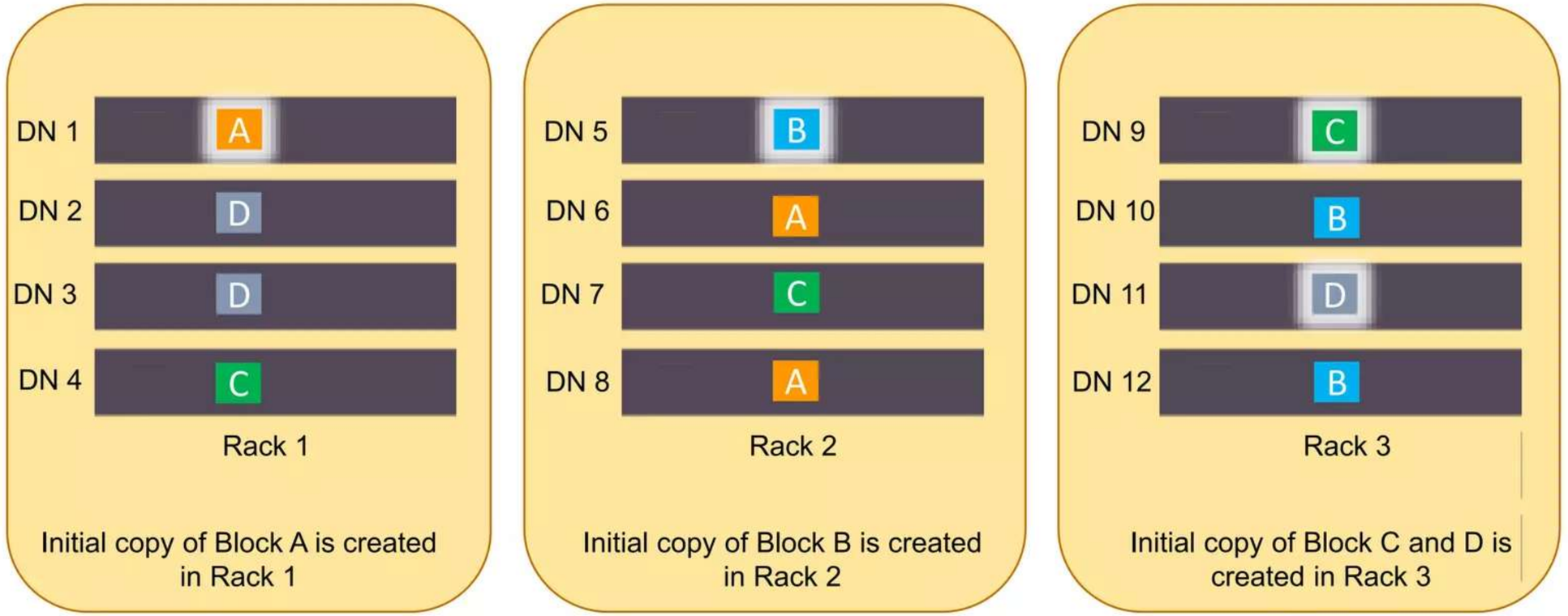
DN -------> Datanode

A -------> Block A

B -------> Block B

C -------> Block C

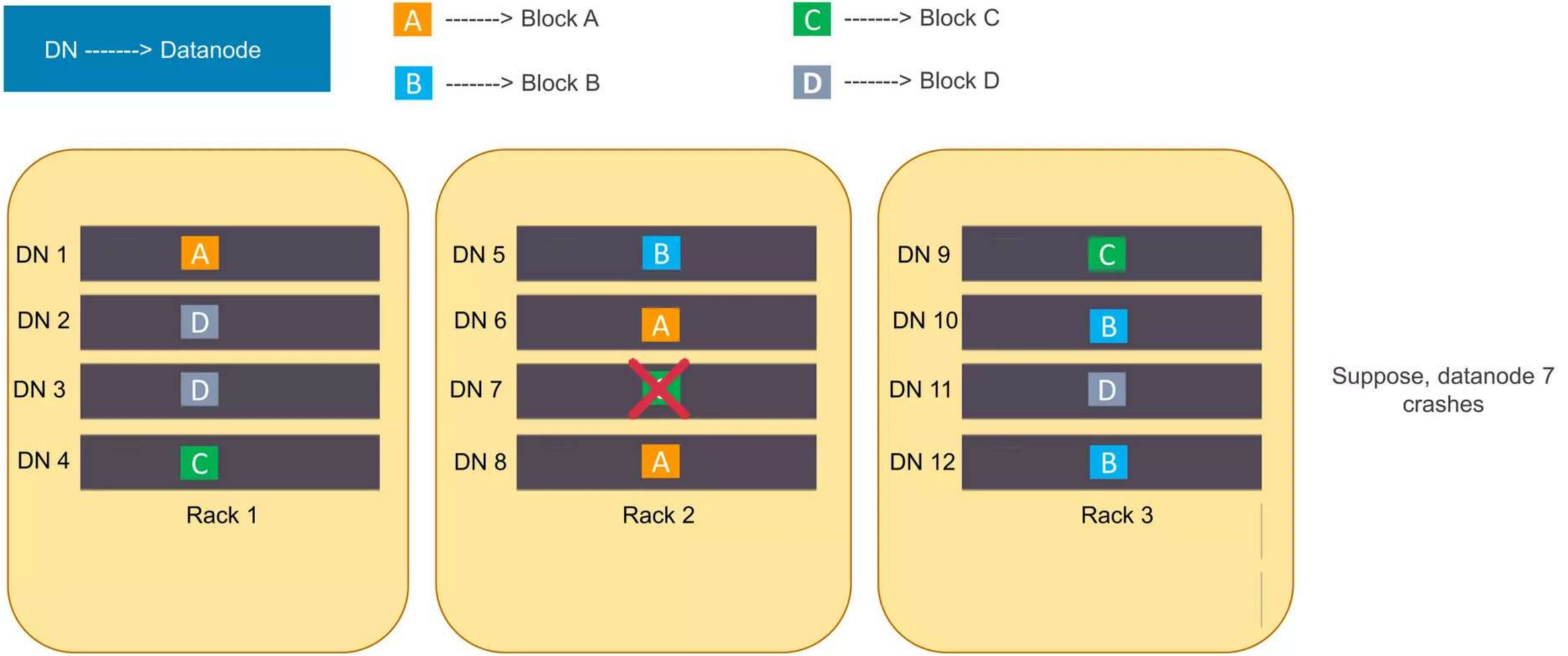
D -------> Block D



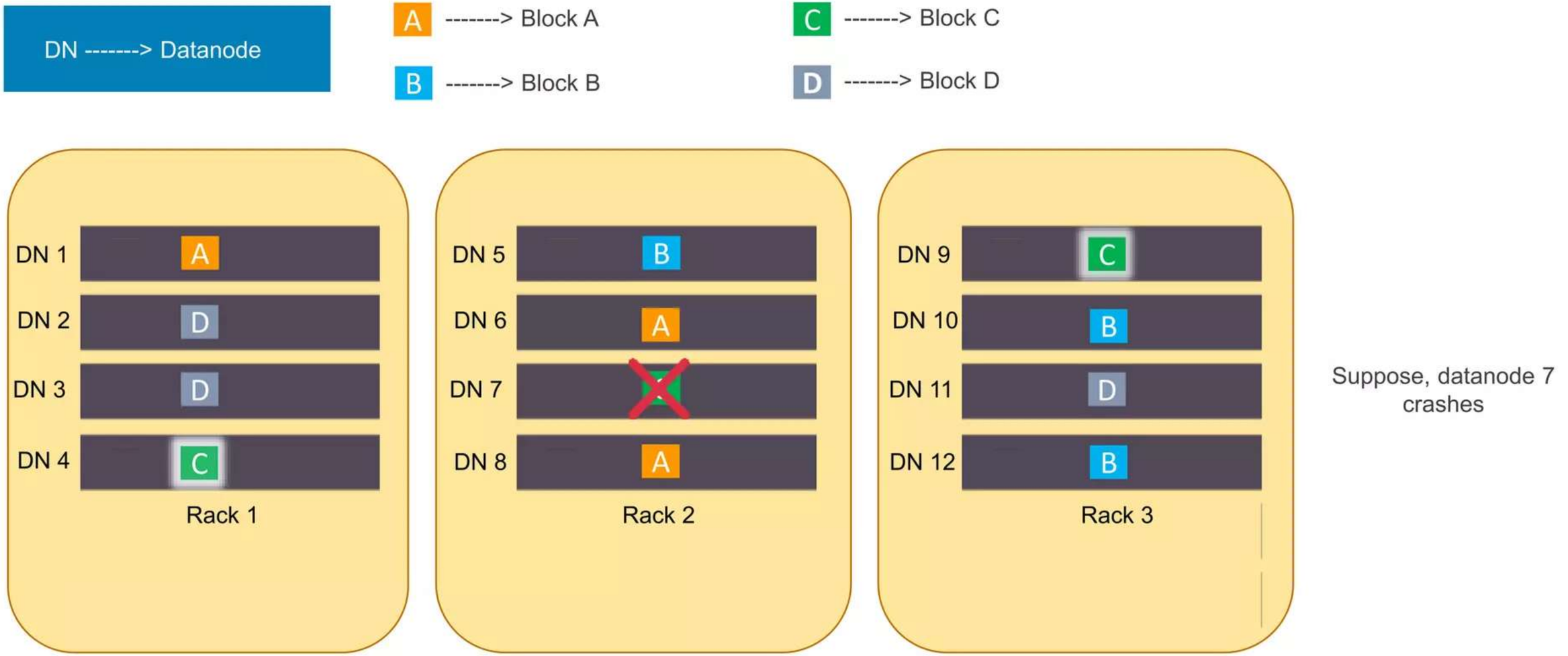
When cluster is rack aware, all the replicas of a block will not be placed on the same rack

simplilearn

Data Replication in HDFS



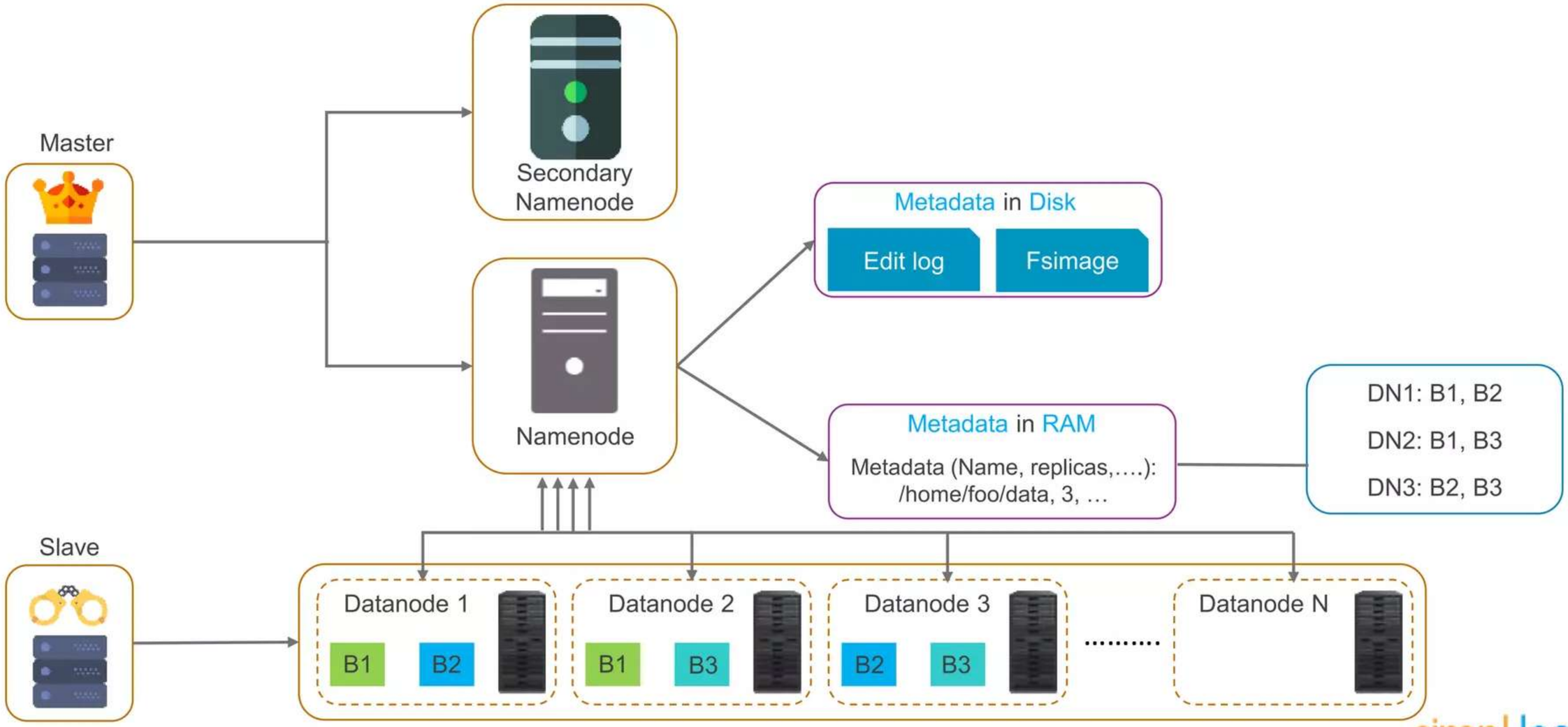
Data Replication in HDFS



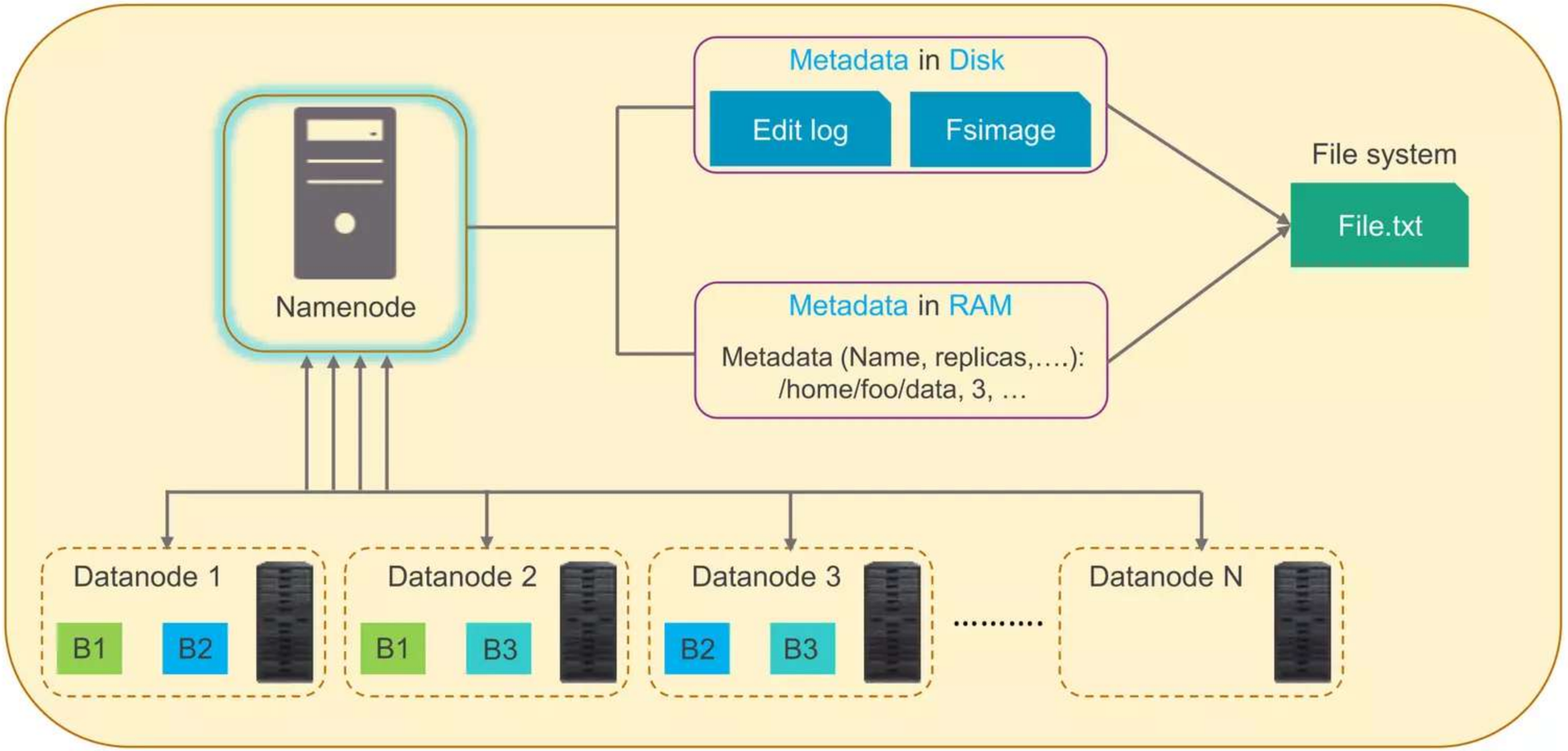
We will still have 2 copies of Block C data on DN 4 of Rack 1 and DN 9 of Rack 3


HDFS Architecture
simplilearn

HDFS Architecture

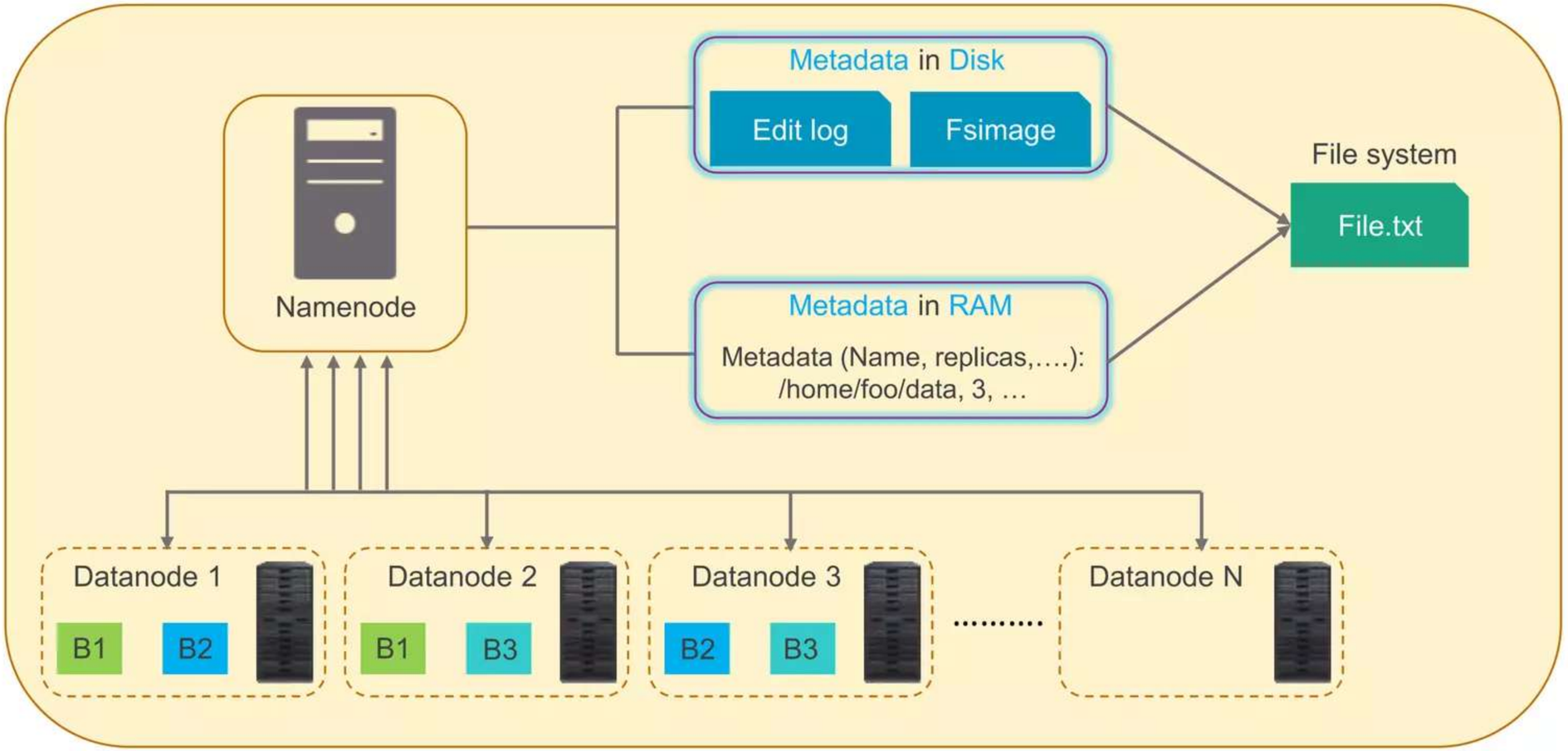


HDFS - Namenode



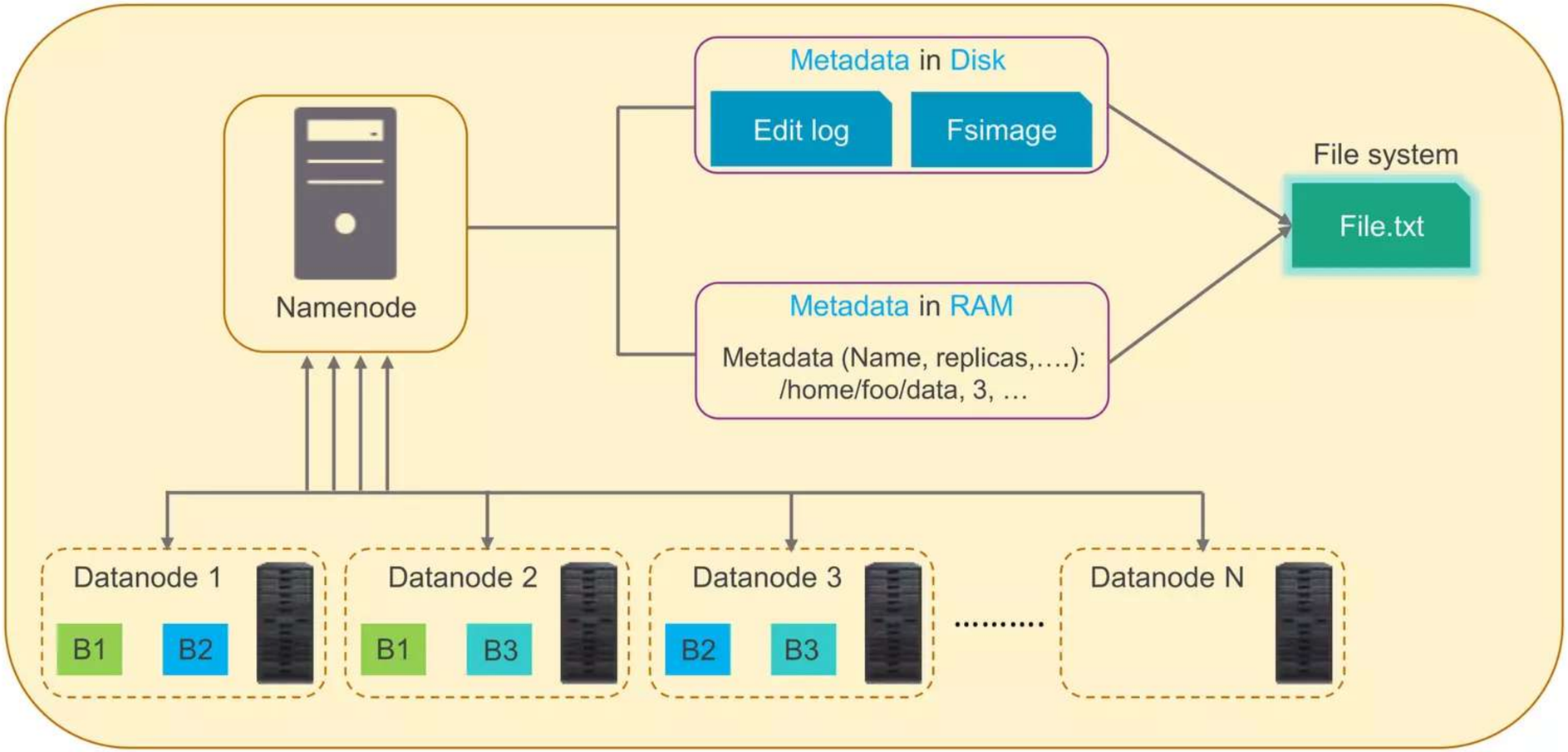
Namenode is the master server. In a non high availability cluster, there can be only one Namenode. In a Hadoop cluster, 2 Namenodes are possible

HDFS - Namenode



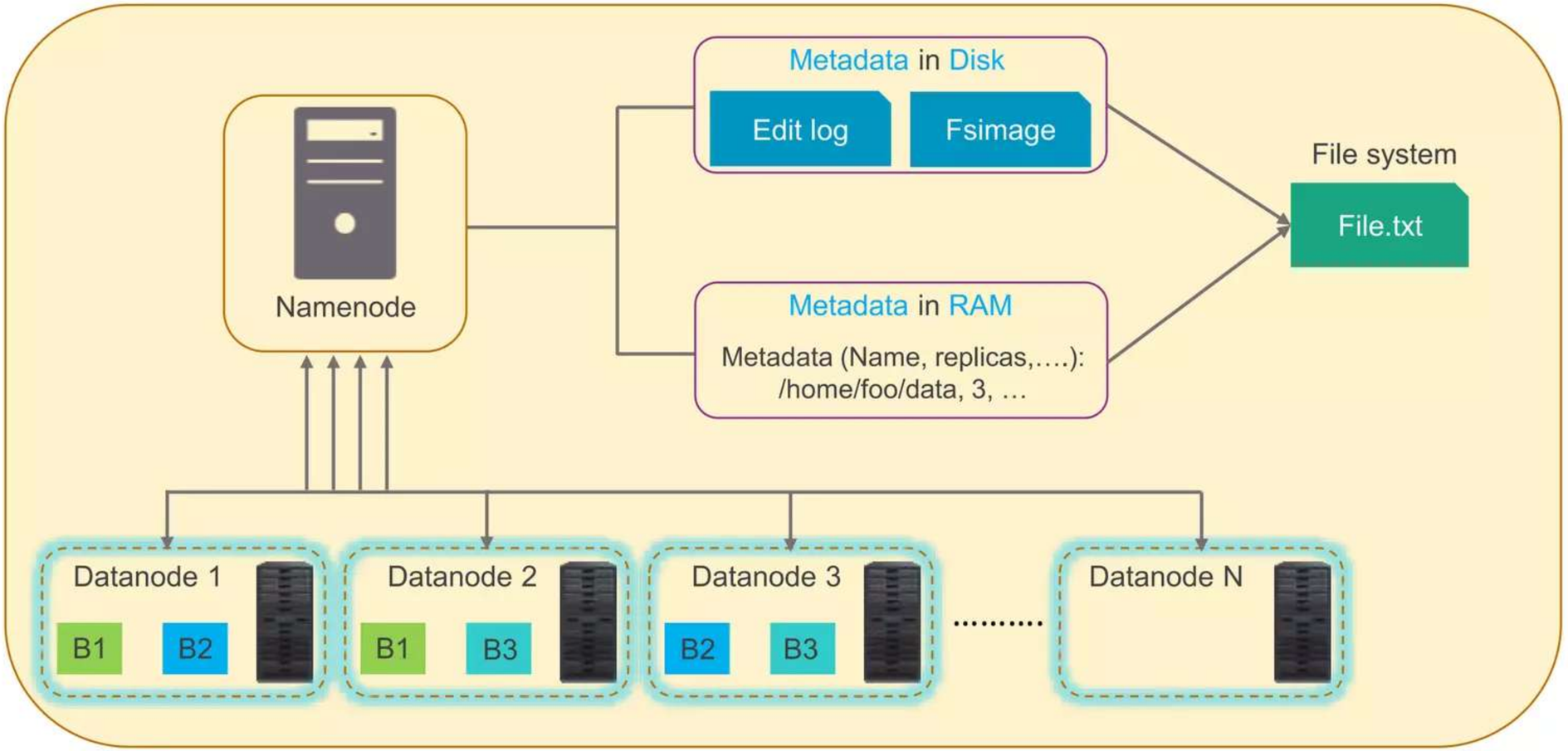
Namenode holds metadata information about the various Datanodes, their location, the size of each block, etc.

HDFS - Namenode



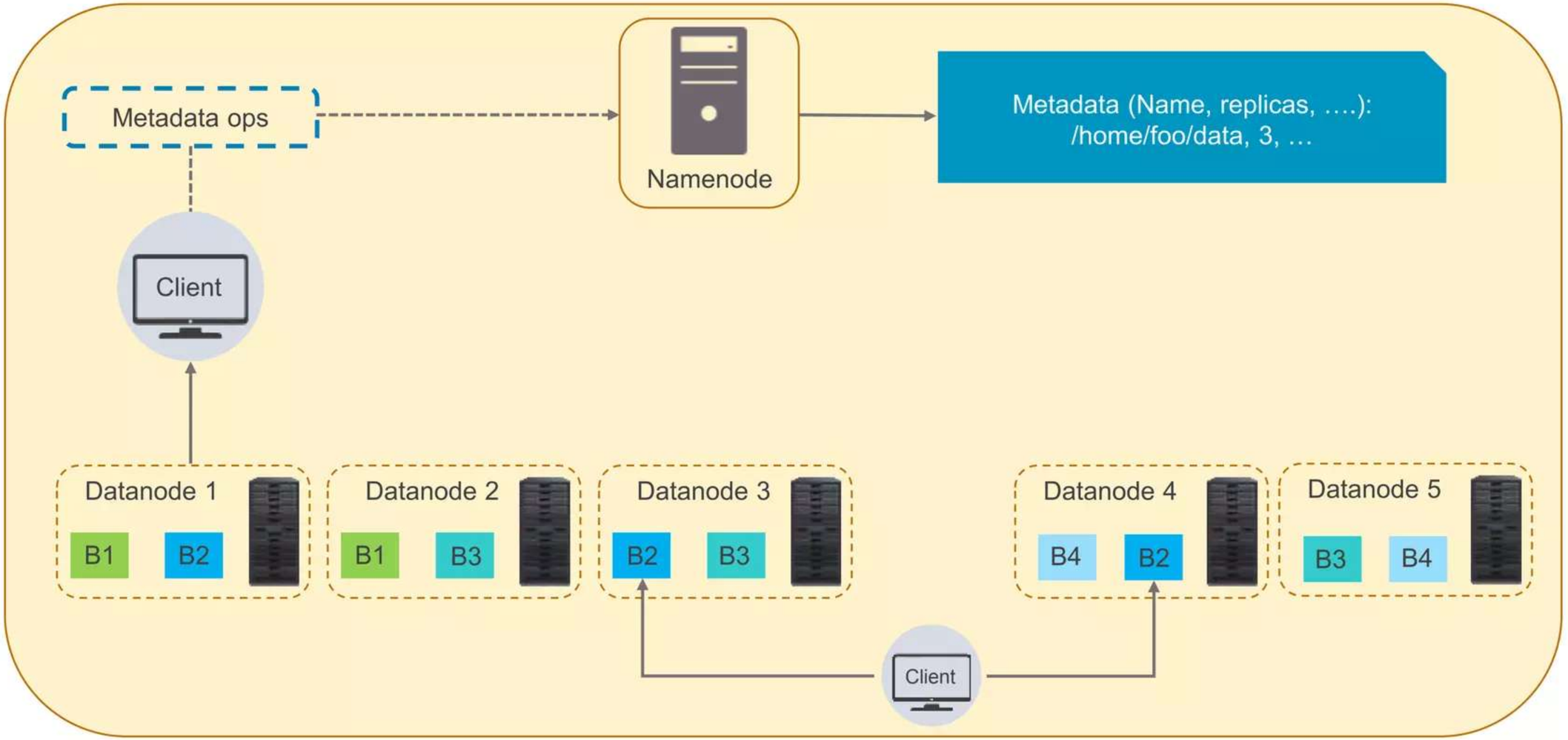
Helps to execute file system namespace operations – opening, closing, renaming files and directories

HDFS - Namenode



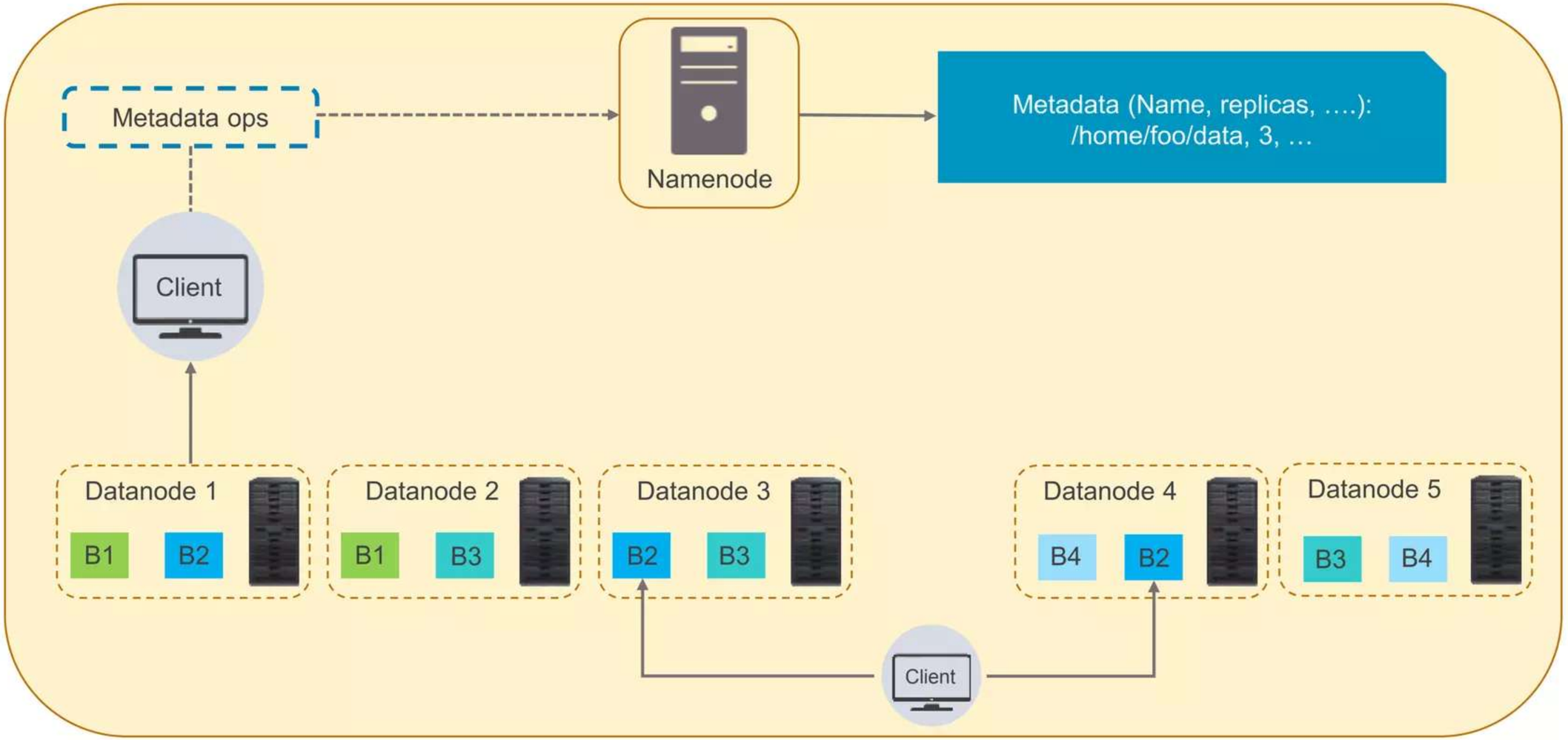
Datanodes send block reports to Namenode every 10 seconds

HDFS - Datanode



Datanode is a multiple instance server. There can be N number of Datanode servers

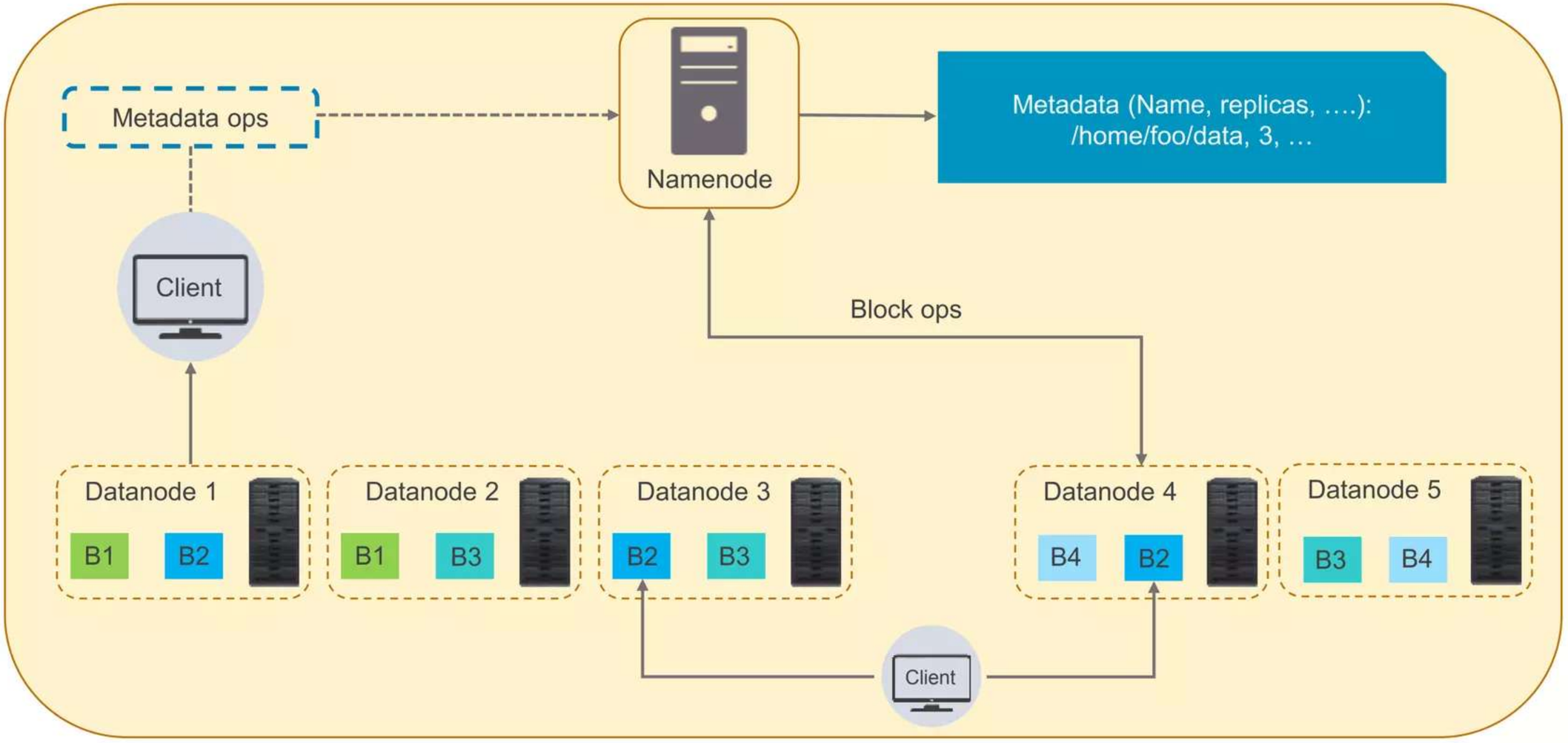
HDFS - Datanode



Datanode stores and maintains the data blocks

simplilearn

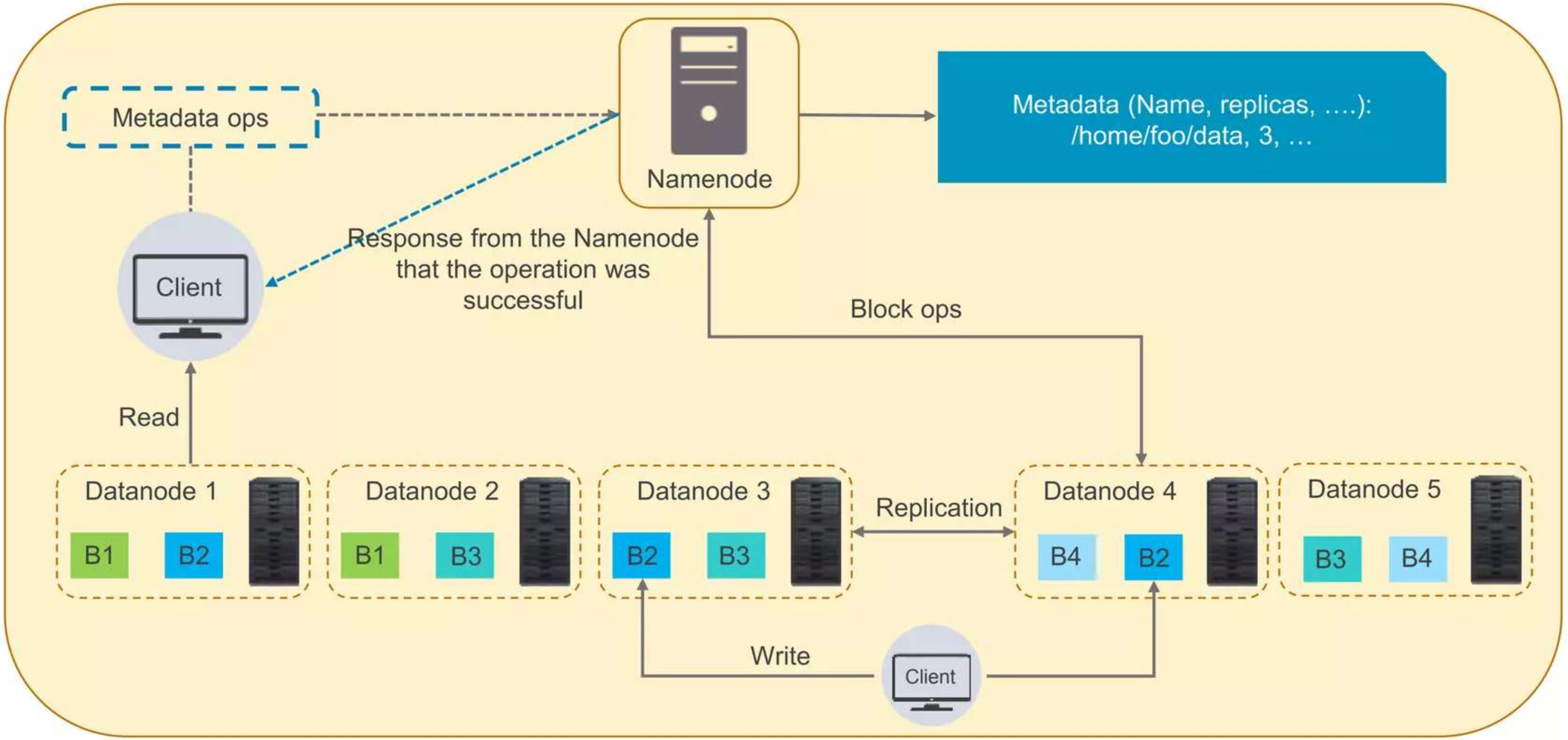
HDFS - Datanode



Datanode stores and retrieves the blocks when asked by the Namenode

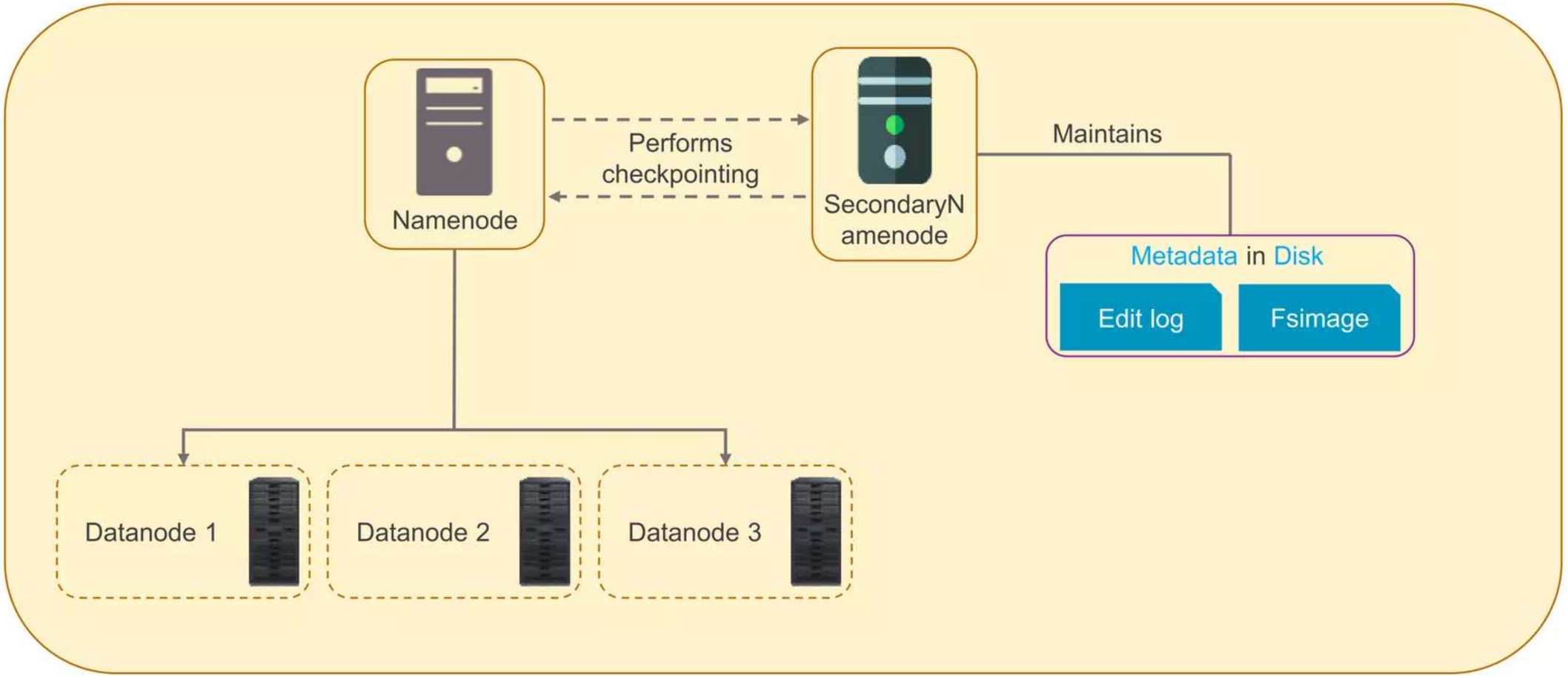
simplilearn

HDFS - Datanode



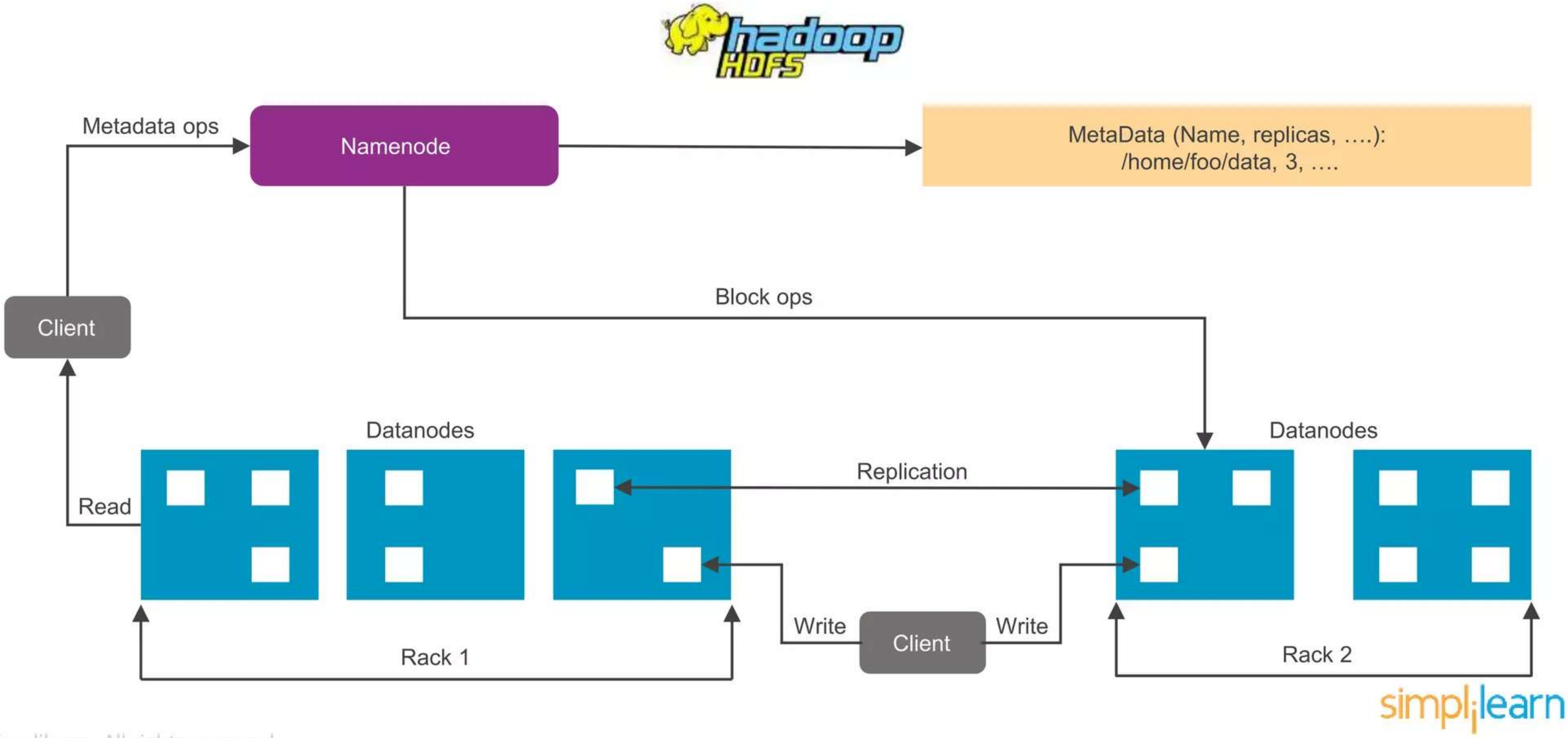
It reads and writes client's request and performs block creation, deletion and replication on instruction from the Namenode

HDFS – Secondary Namenode

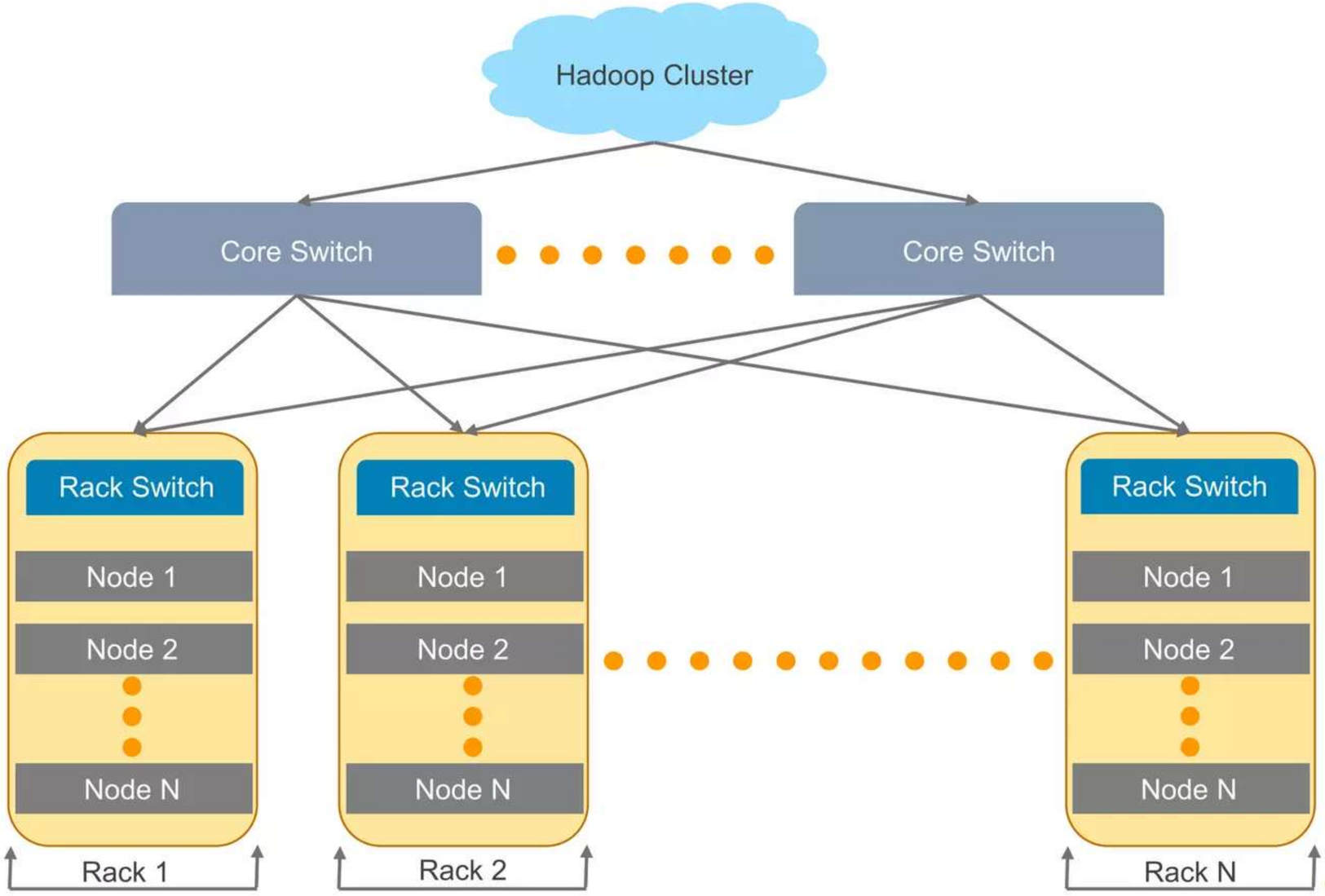


Secondary Namenode server is responsible for maintaining a copy of Metadata in disk

HDFS Architecture

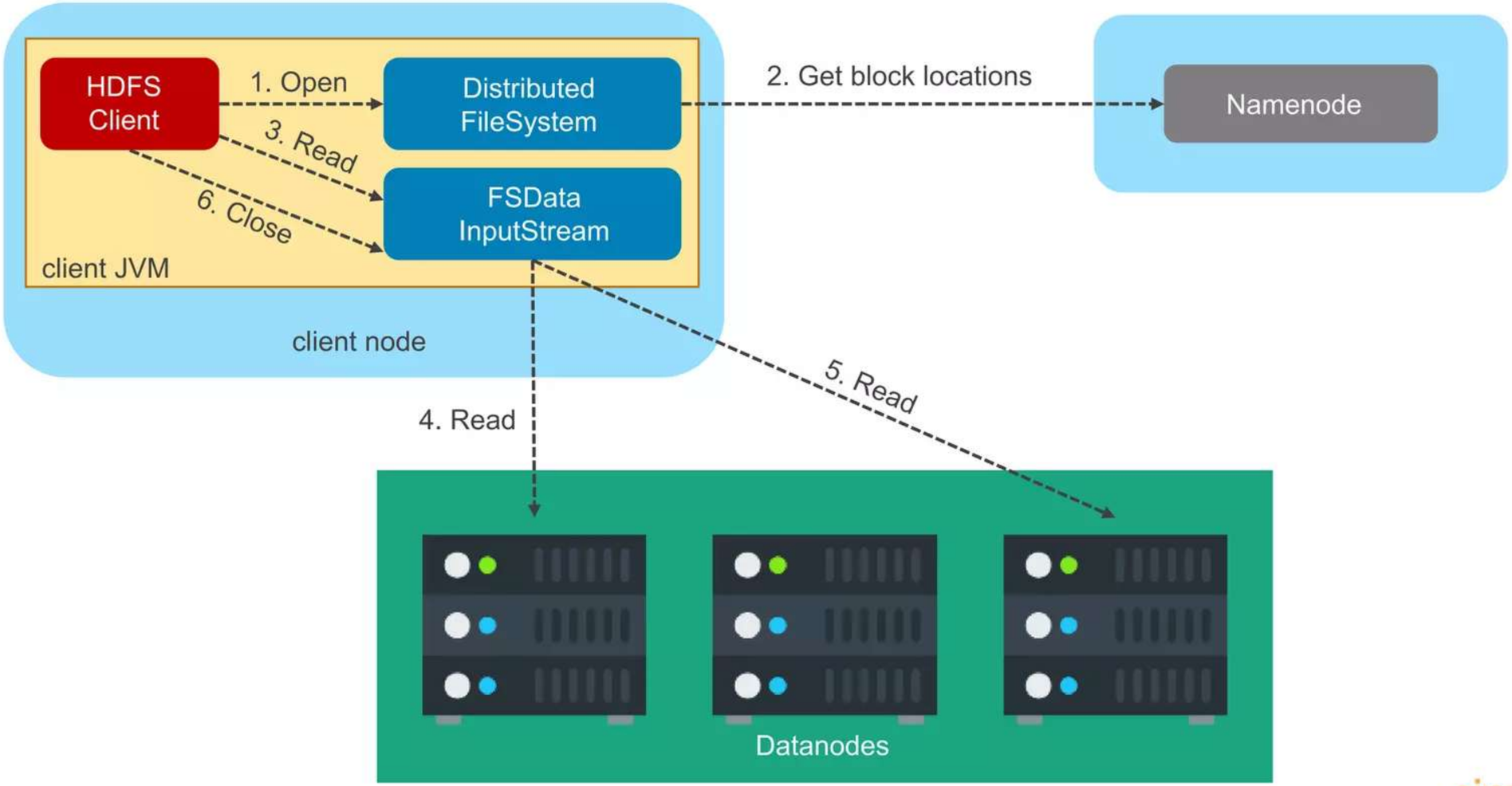


Hadoop Cluster – Rack Based Architecture

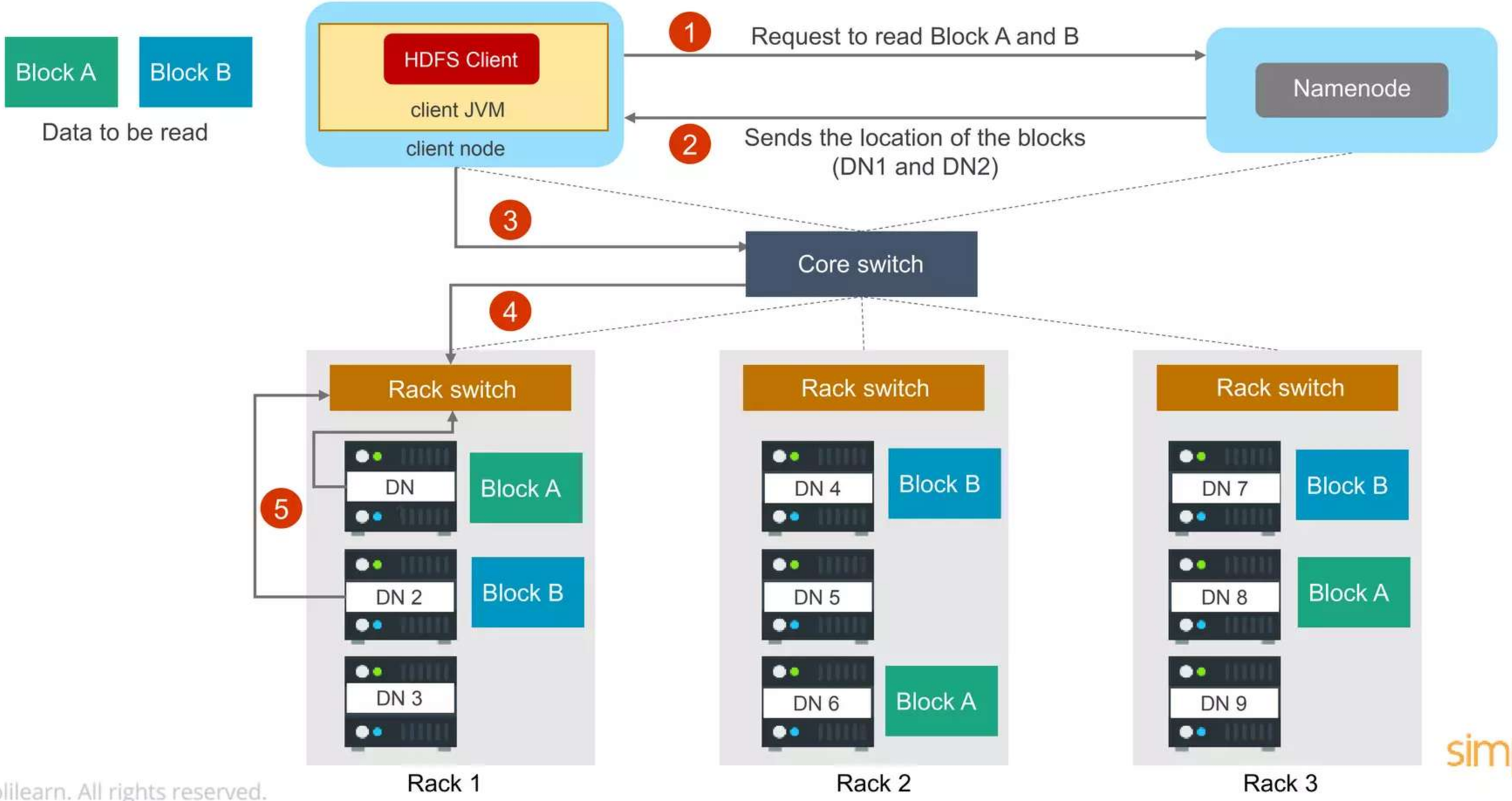



HDFS Read Mechanism
simplilearn

HDFS Read Mechanism

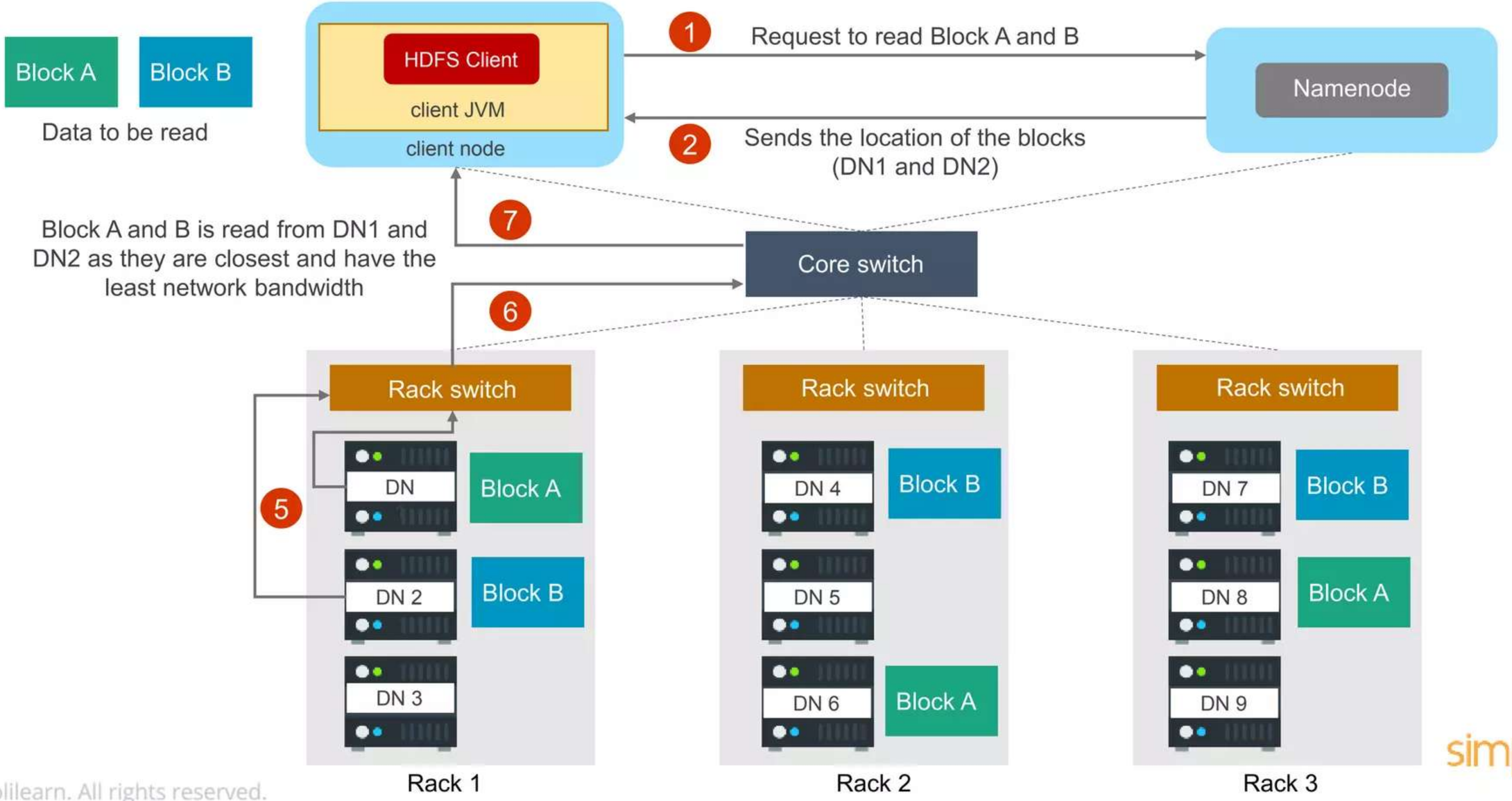


HDFS Read Mechanism



simplilearn

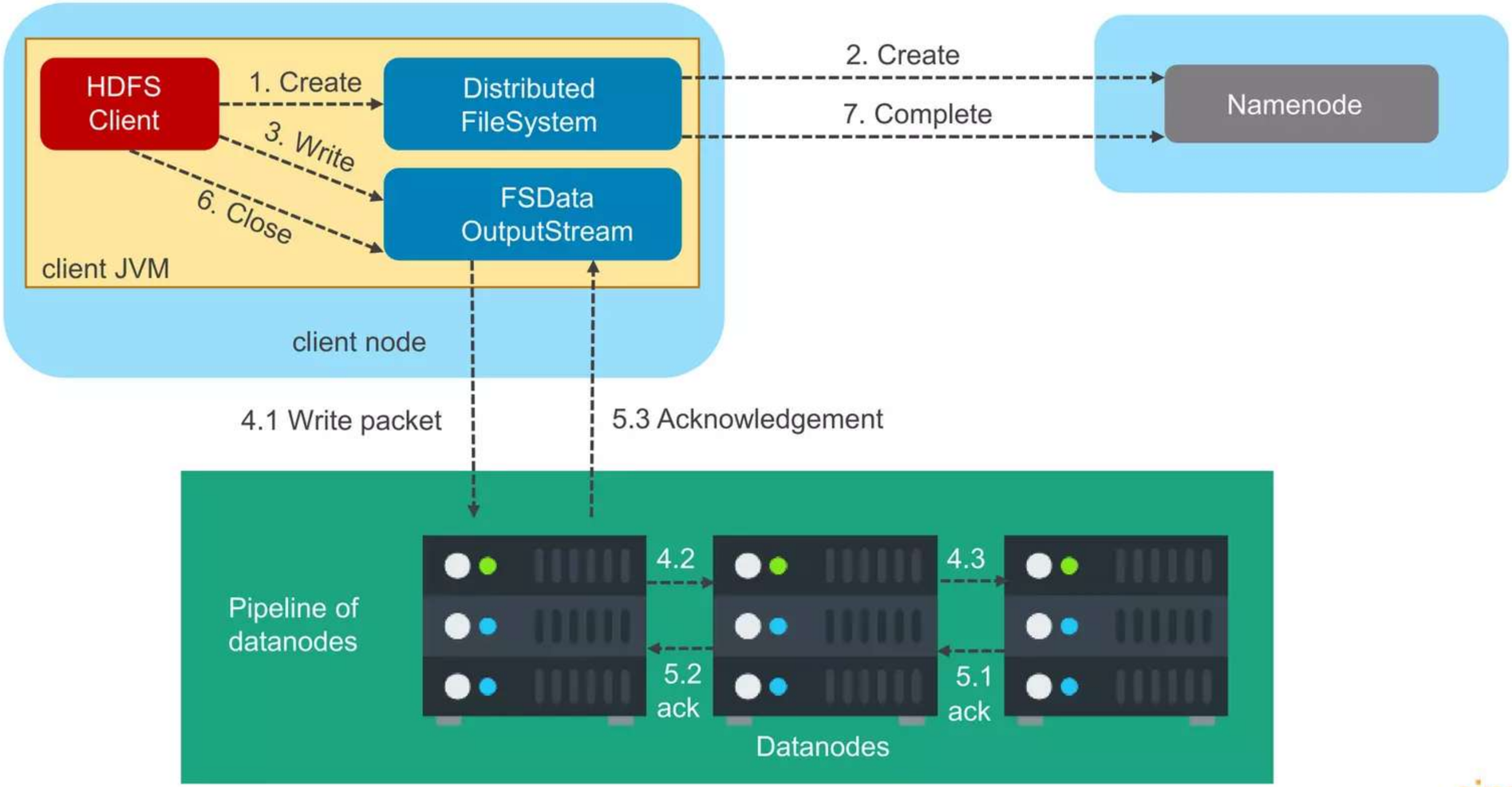
HDFS Read Mechanism



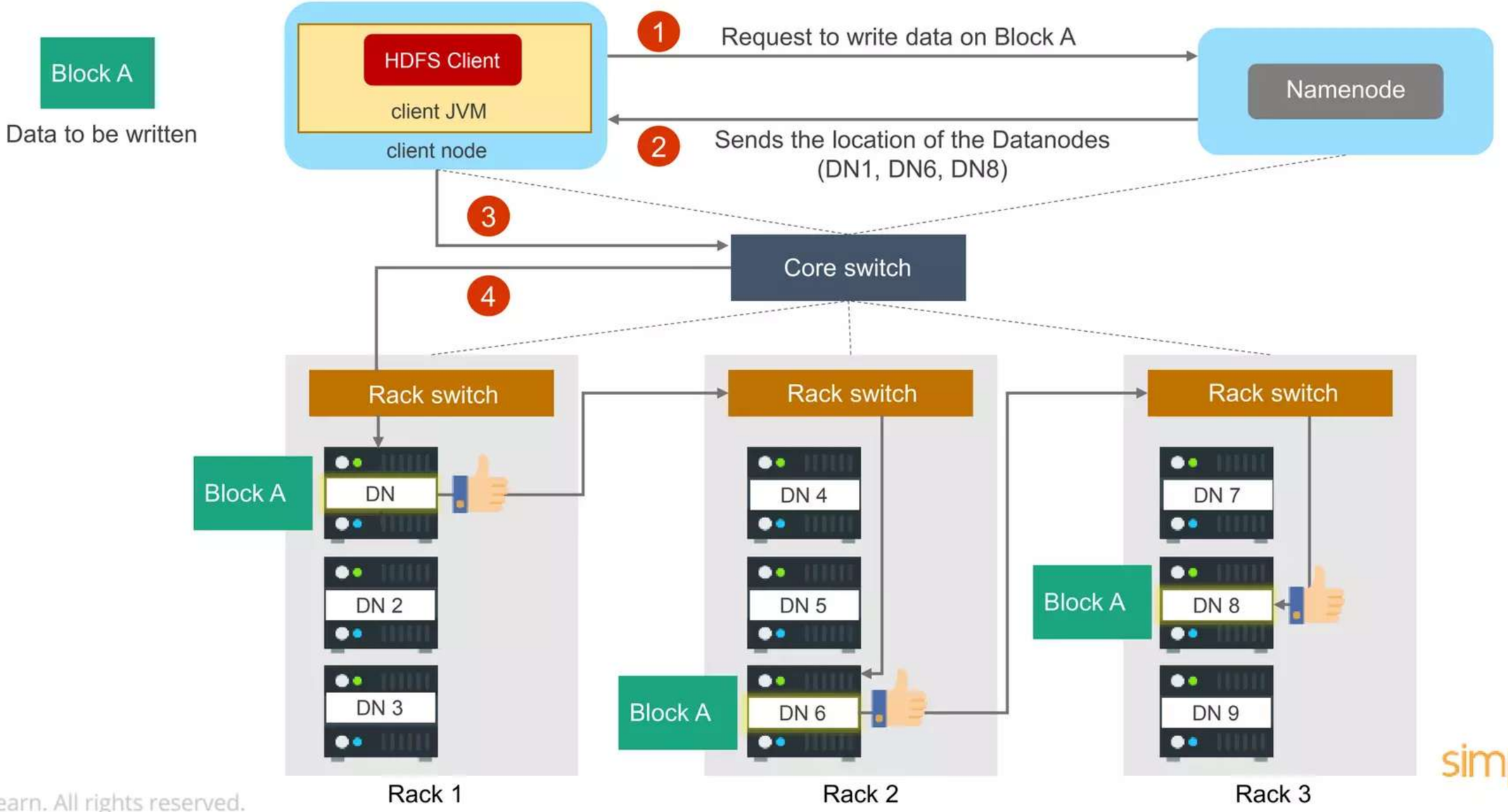
simplilearn


HDFS Write Mechanism
simplilearn

HDFS Write Mechanism

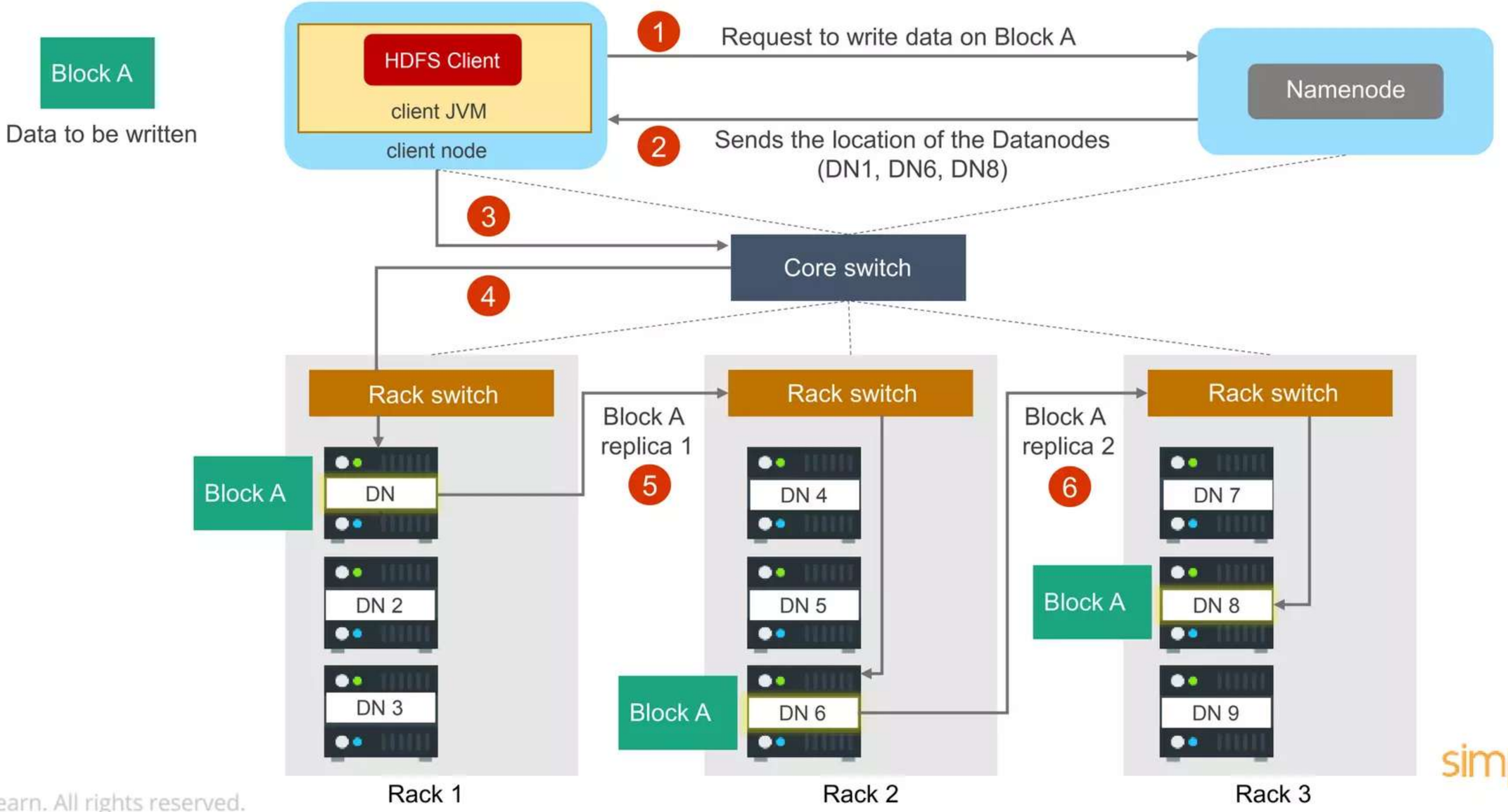


HDFS Write Mechanism



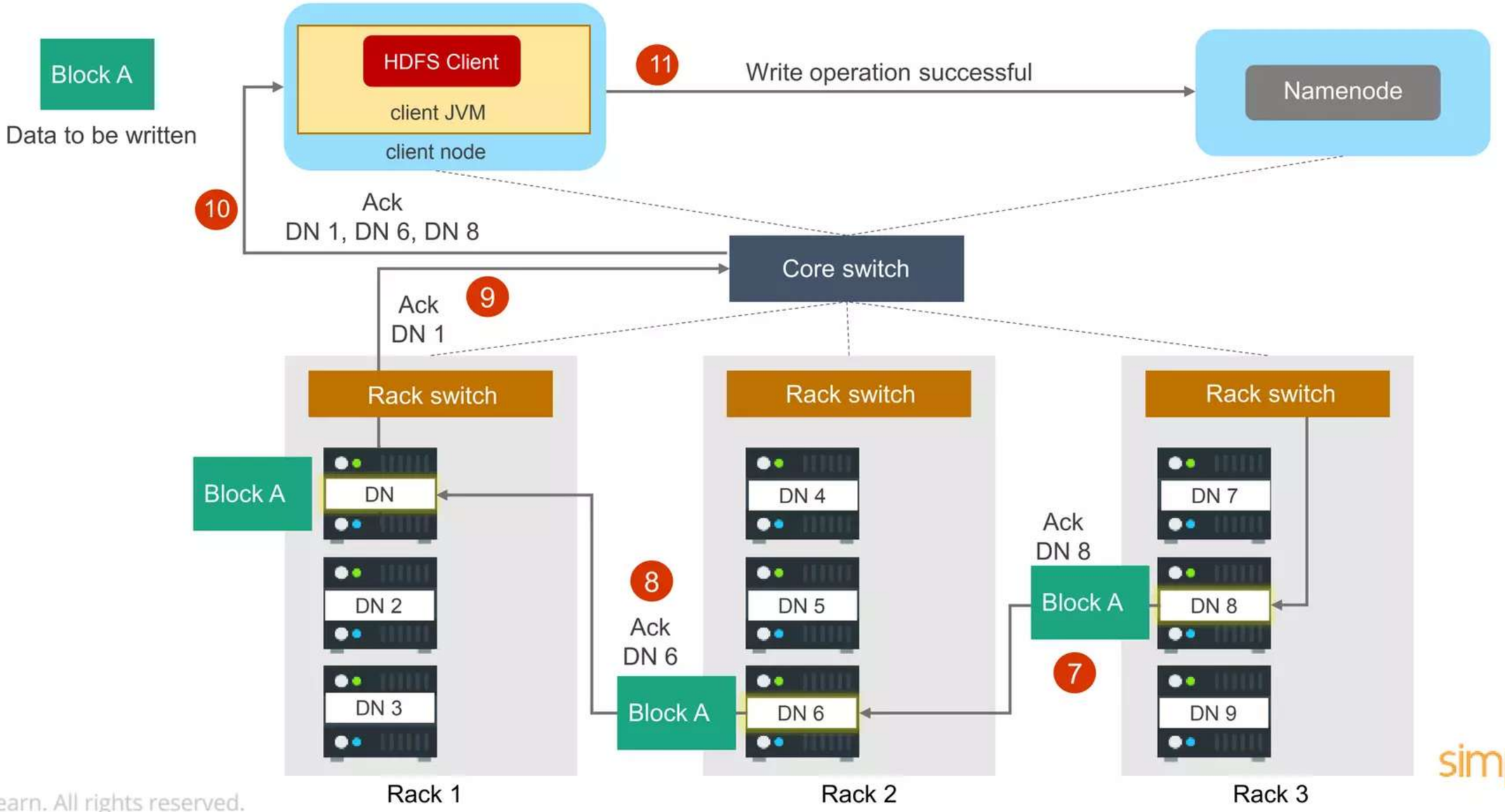
simplilearn

HDFS Write Mechanism



simplilearn

HDFS Write Mechanism

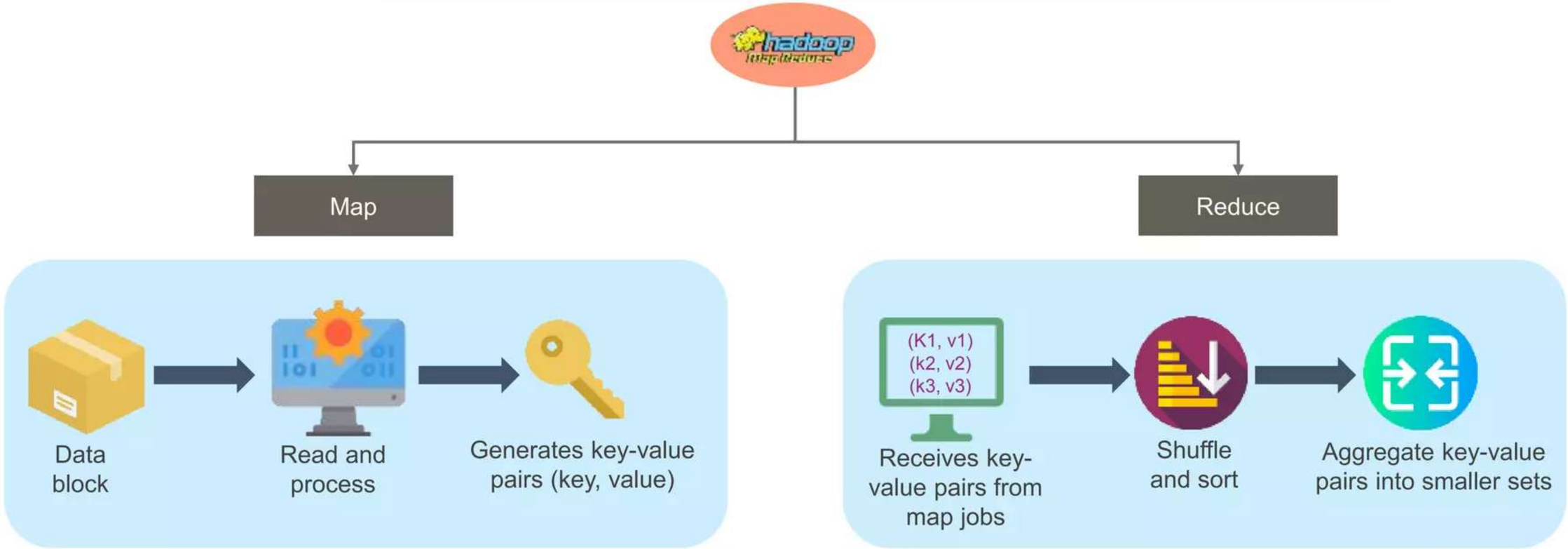


simplilearn


Hadoop MapReduce
simplilearn

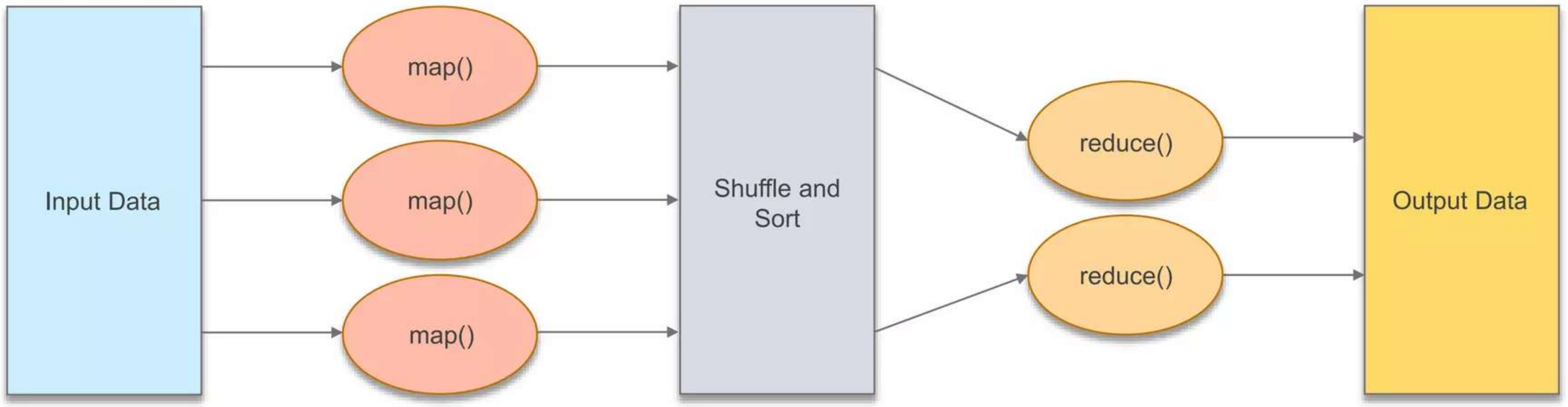
Hadoop MapReduce

MapReduce is a framework that performs distributed and parallel processing of large volumes of data

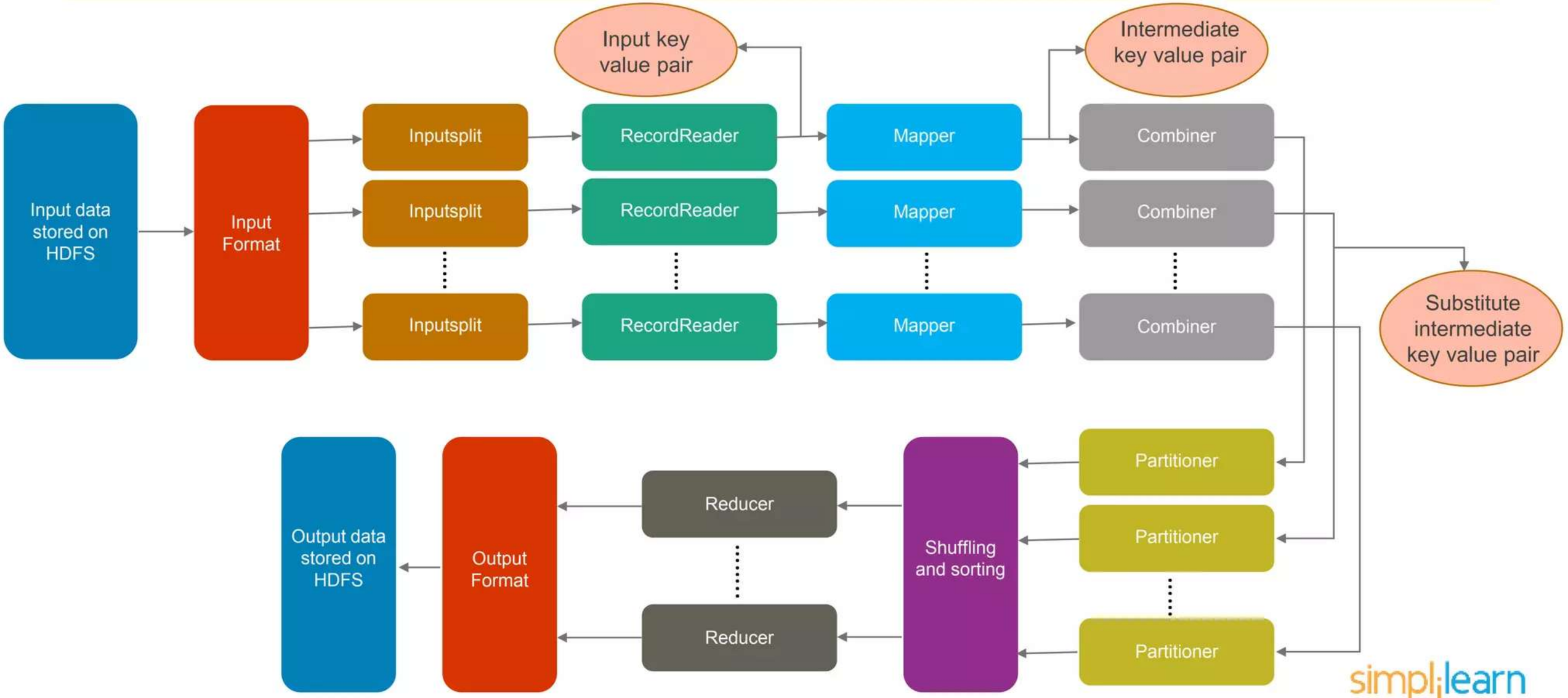


Hadoop MapReduce

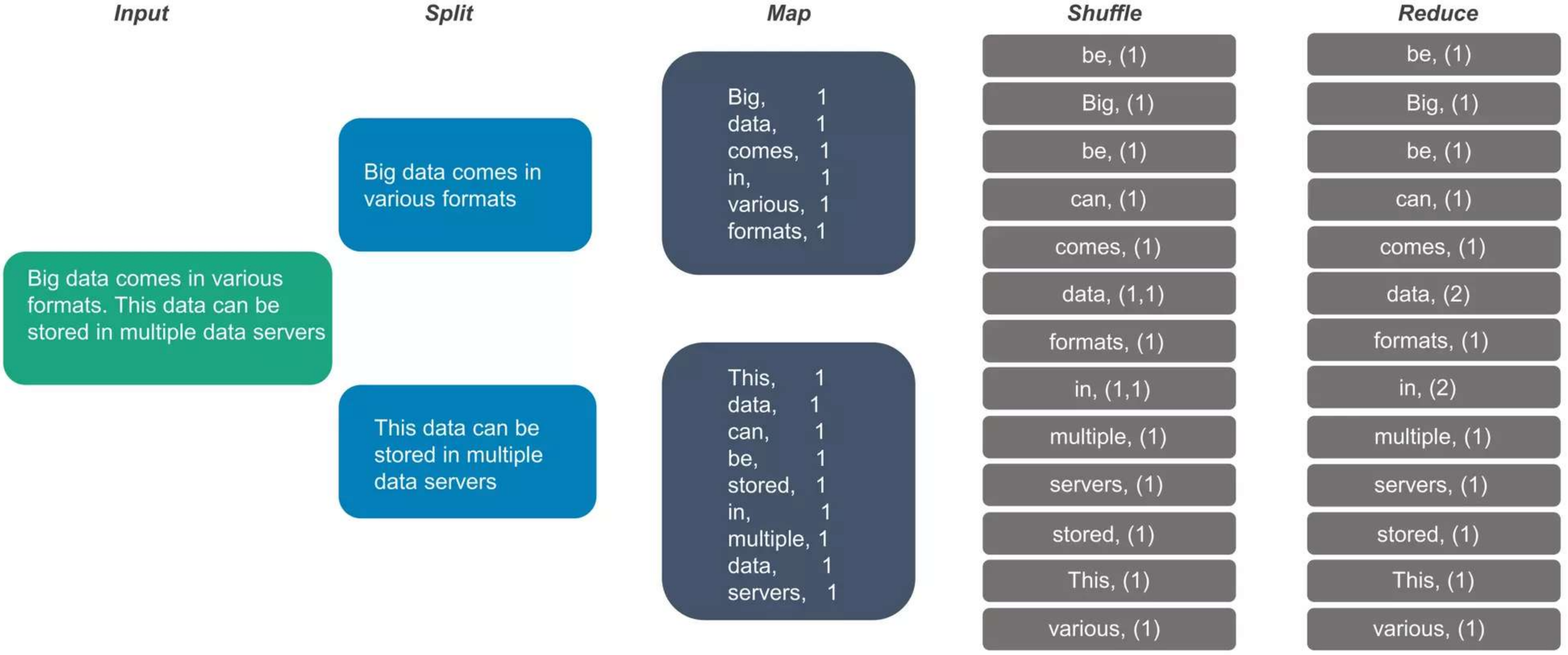
MapReduce is a framework that performs distributed and parallel processing of large volumes of data



MapReduce Job Execution



MapReduce Example



simplilearn

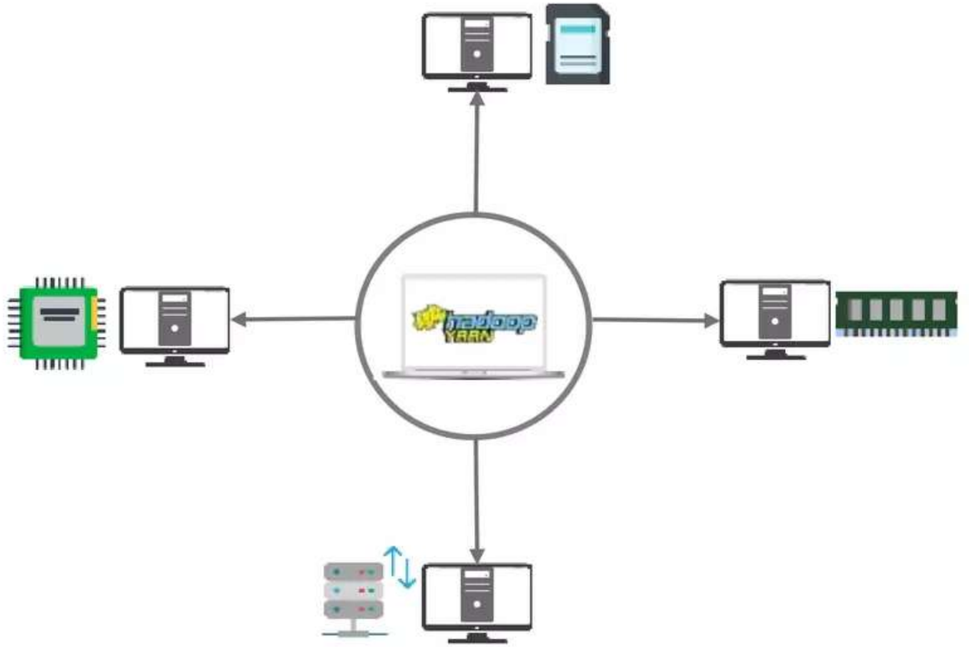
Hadoop YARN

Hadoop YARN

YARN ---------> Yet Another Resource Negotiator

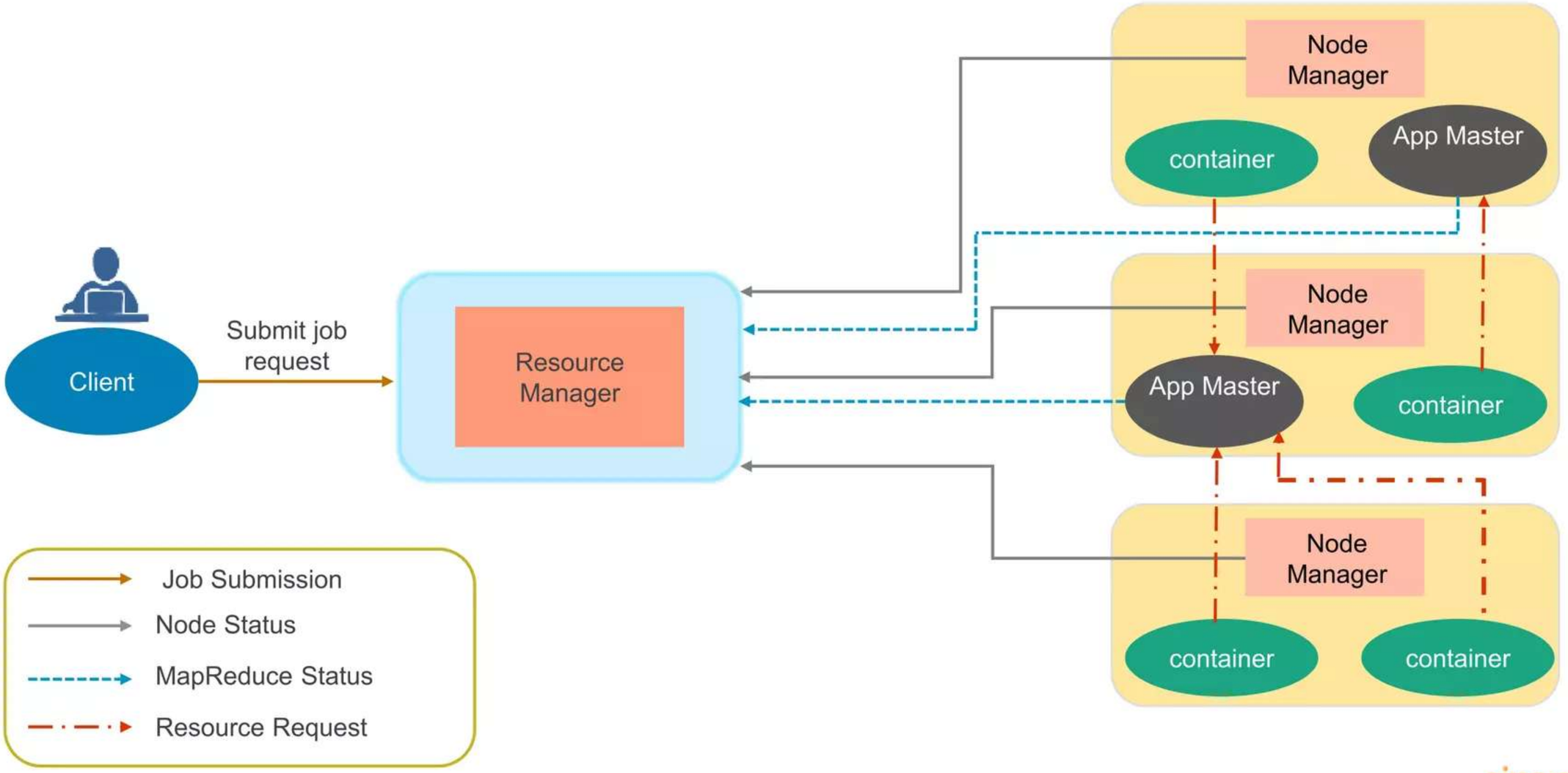
Introduced in Hadoop 2.0 version

It is the middle layer between HDFS and MapReduce

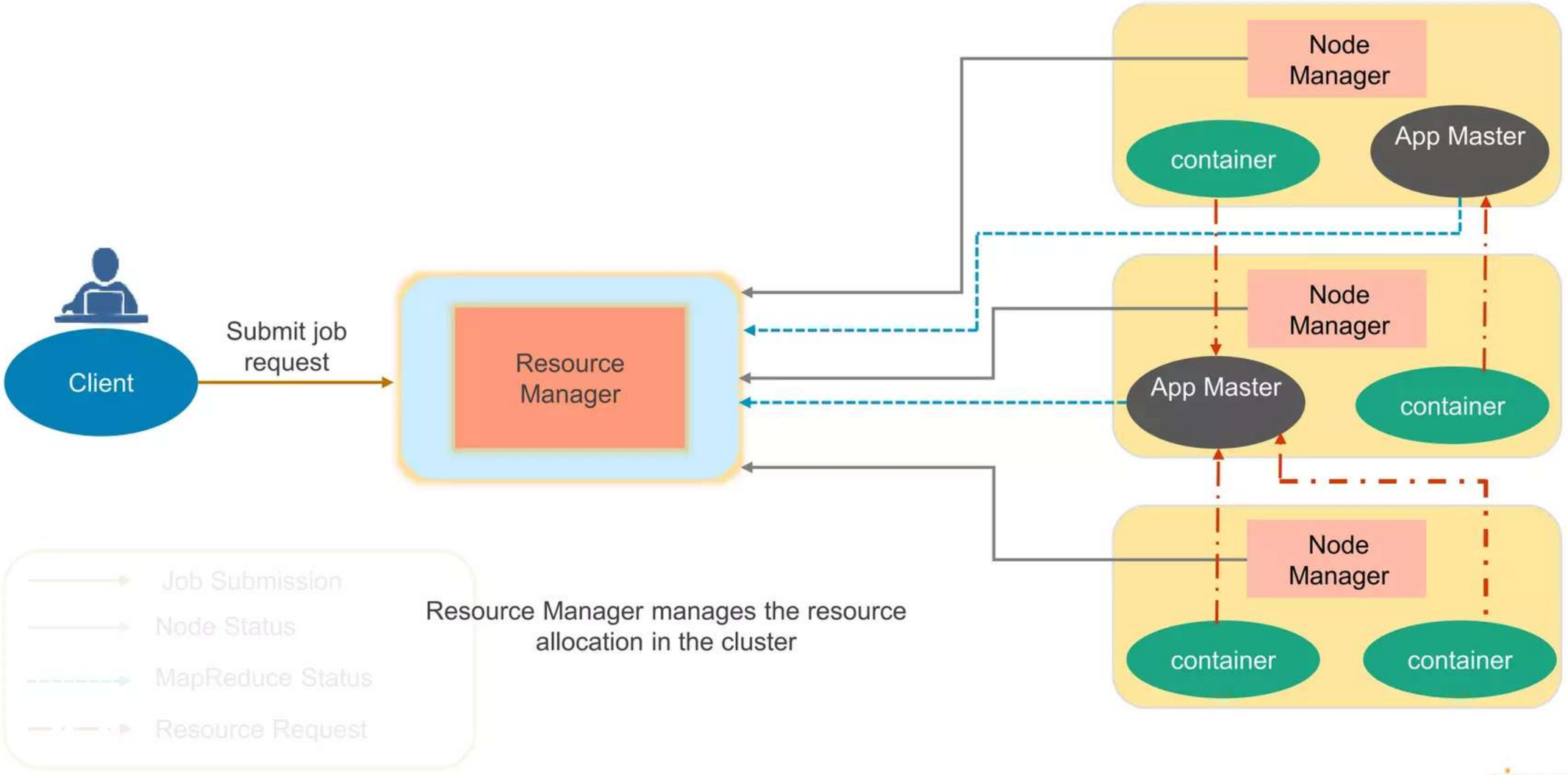


Manages cluster resources (memory, network bandwidth, disk IO, CPU)

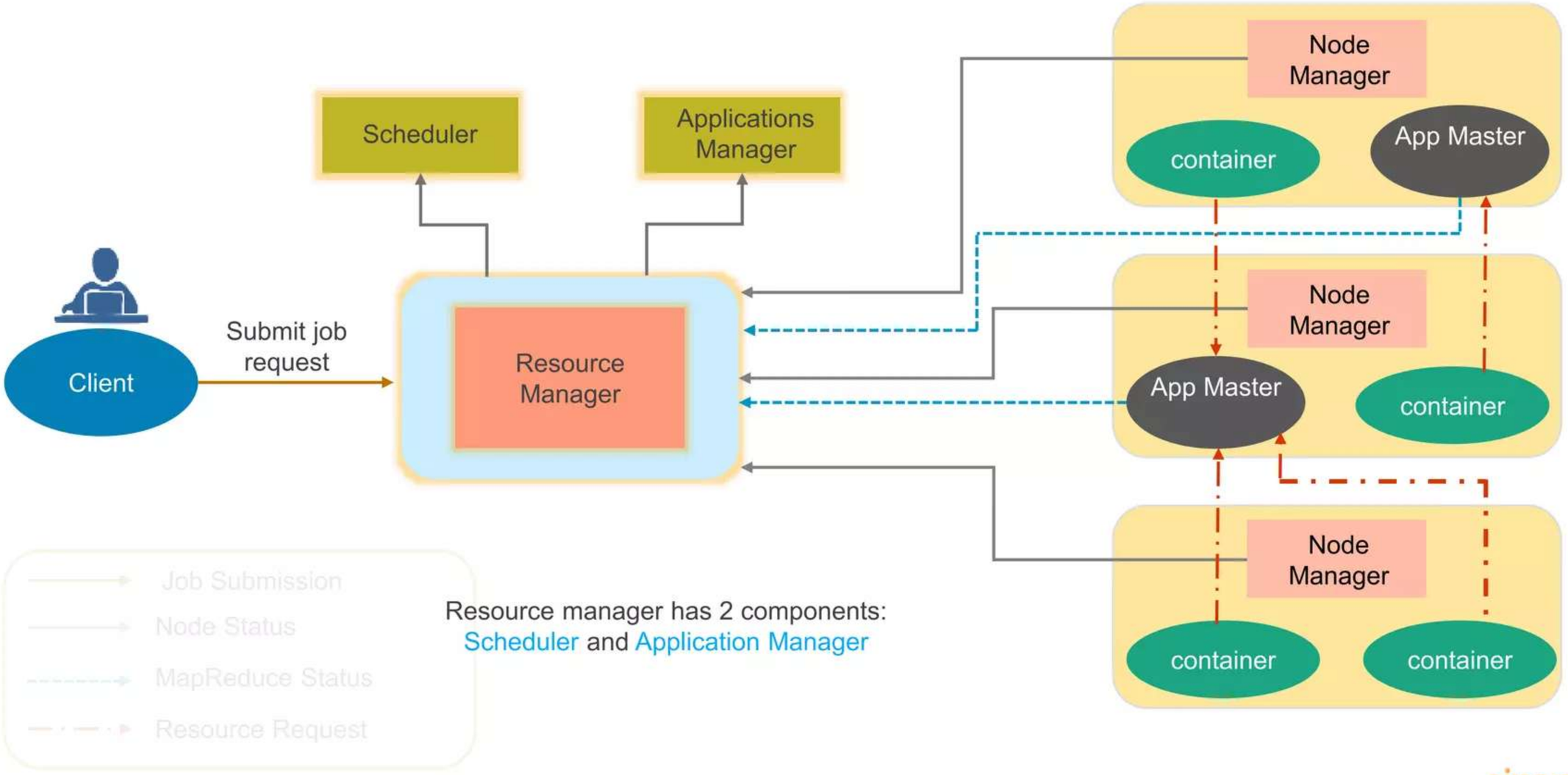
YARN Architecture



YARN Architecture – Resource Manager



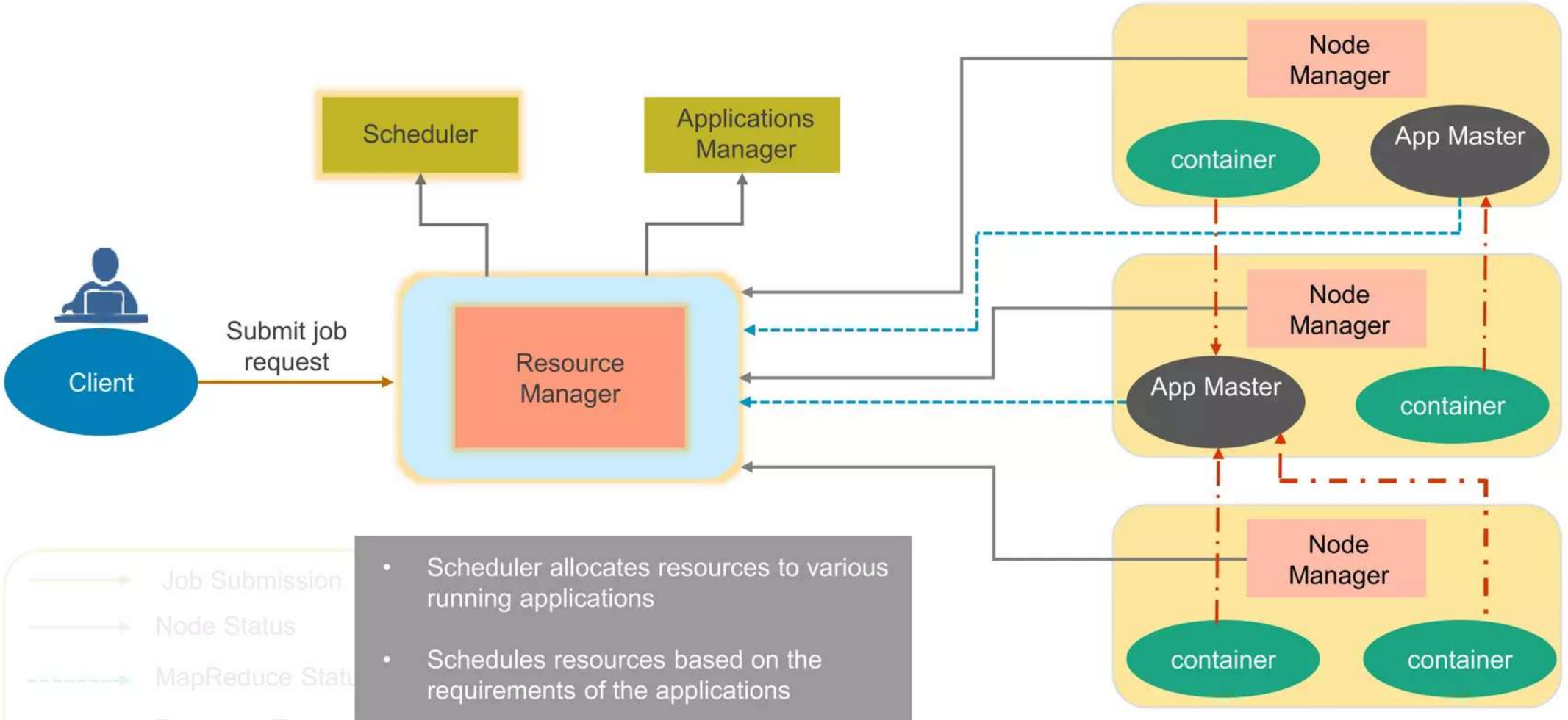
YARN Architecture – Resource Manager



Resource manager has 2 components:
Scheduler and Application Manager

simplilearn

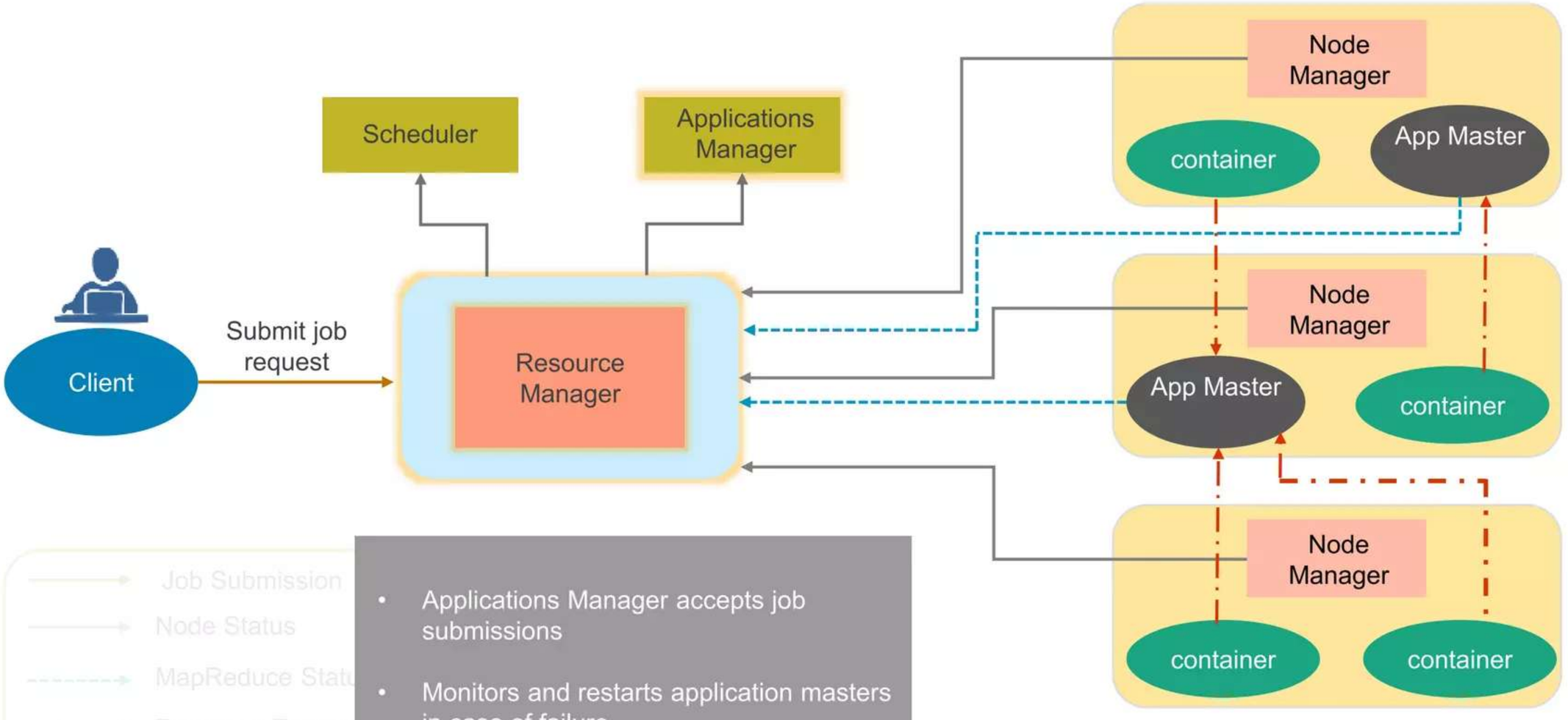
YARN Architecture – Scheduler



- Scheduler allocates resources to various running applications
- Schedules resources based on the requirements of the applications
- Does not monitor or track the status of the applications

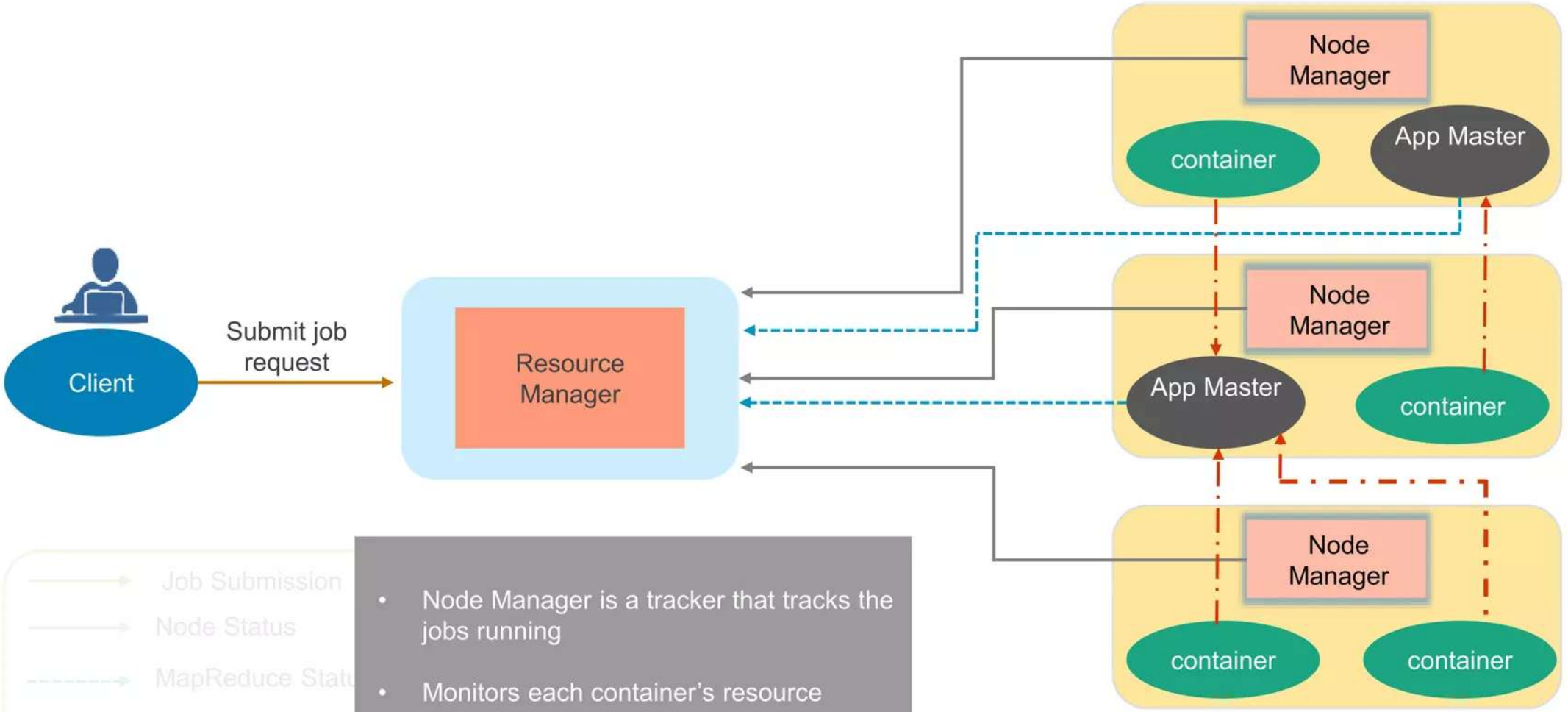
simplilearn

YARN Architecture – Applications Manager



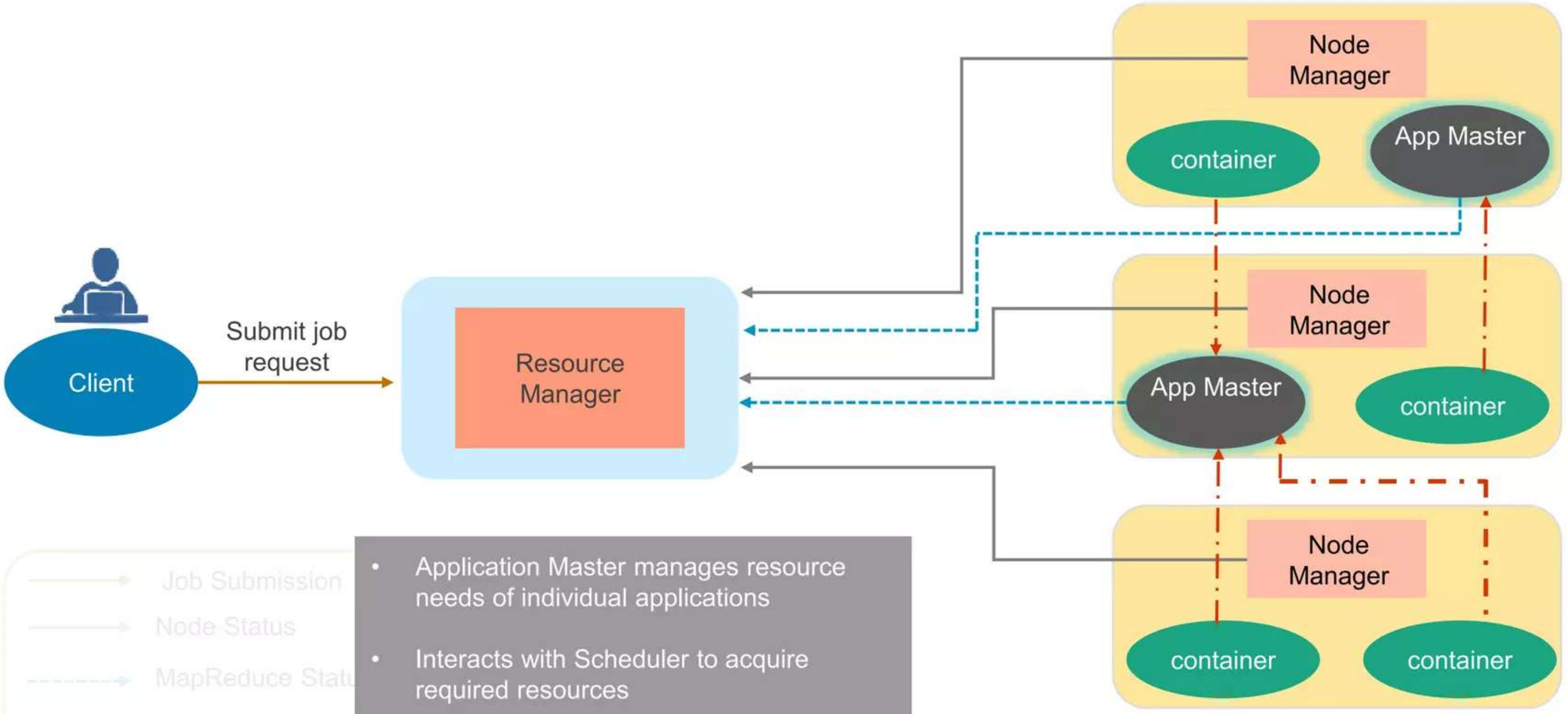
- Applications Manager accepts job submissions
- Monitors and restarts application masters in case of failure

YARN Architecture – Node Manager



- Node Manager is a tracker that tracks the jobs running
- Monitors each container's resource utilization

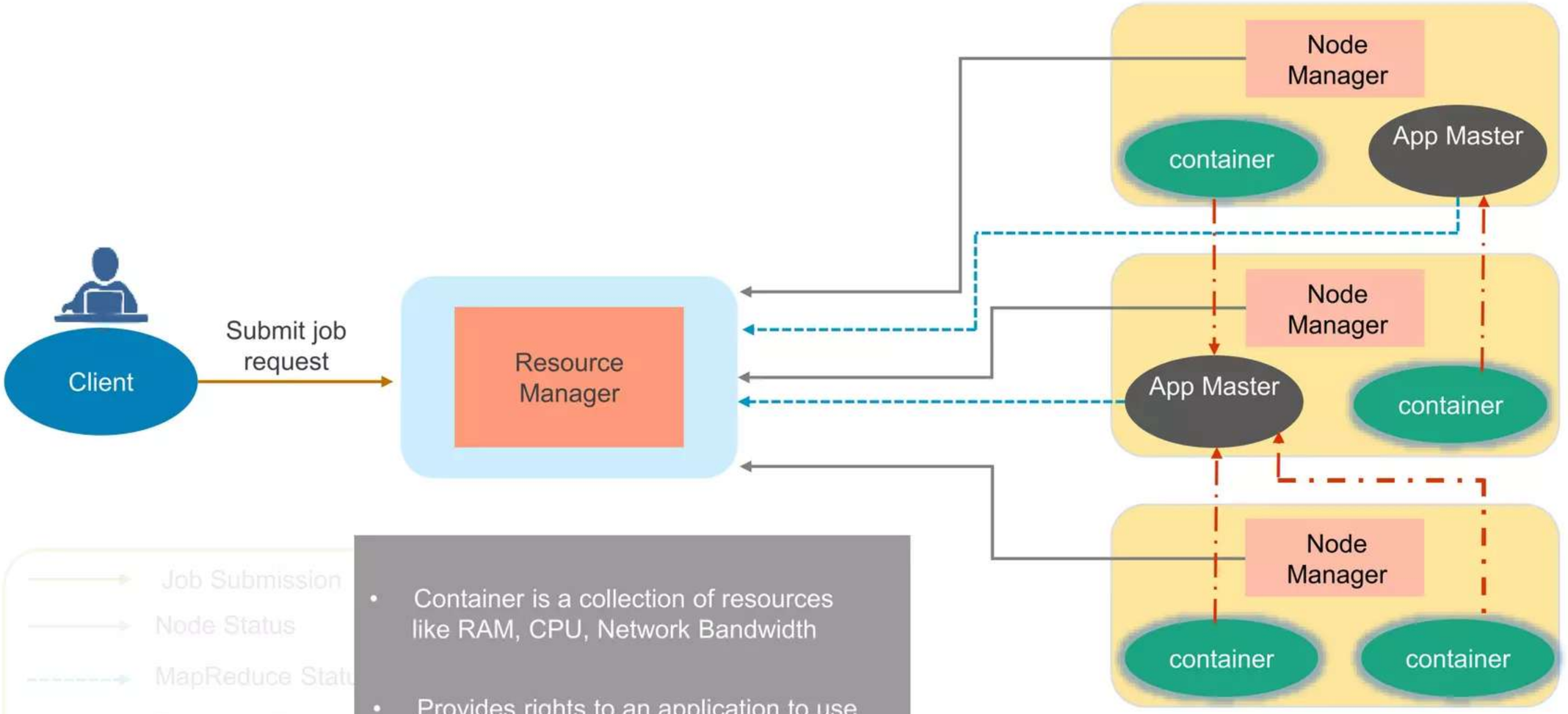
YARN Architecture – App Master



- Application Master manages resource needs of individual applications
- Interacts with Scheduler to acquire required resources
- Interacts with Node Manager to execute and monitor tasks

simplilearn

YARN Architecture - Container



- Container is a collection of resources like RAM, CPU, Network Bandwidth
- Provides rights to an application to use specific amount of resources

Use case – Word Count using MapReduce


THANK YOU
For more information, visit
www.simplilearn.com
simplilearn