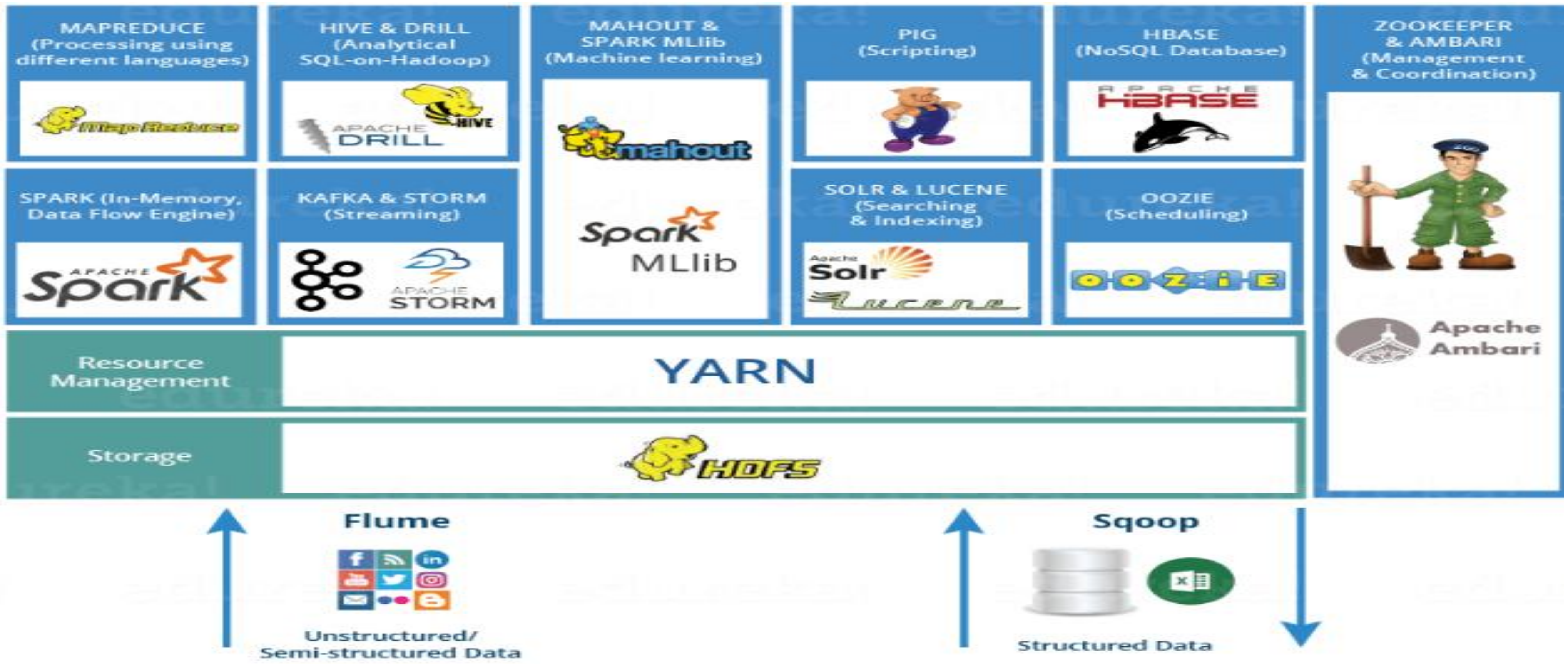
<u>Hadoop</u>

Hadoop ecosystem


MAPREDUCE (Processing using different languages)
MapReduce
HIVE & DRILL (Analytical SQL-on-Hadoop)
APACHE DRILL
HIVE
MAHOUT & SPARK MLlib (Machine learning)
mahout
Spark MLlib
PIG (Scripting)
HBASE (NoSQL Database)
APACHE HBASE
ZOOKEEPER & AMBARI (Management & Coordination)
SPARK (In-Memory, Data Flow Engine)
APACHE Spark
KAFKA & STORM (Streaming)
APACHE STORM
SOLR & LUCENE (Searching & Indexing)
Apache Solr
Lucene
OOZIE (Scheduling)
OOZIE
Apache Ambari
Resource Management
YARN
Storage
HDFS
Flume
Unstructured/ Semi-structured Data
Sqoop
Structured Data

- Hadoop ecosystem is the various tools and technologies provided by Hadoop collectively termed as Hadoop ecosystem to enable development and deployment of bigdata solutions in a cost effective manner.
- Core components of Hadoop ecosystem HDFS and map reduce.
- However these two are not sufficient.
- All these enable user to process large data sets in real time and provide tools to support various types of Hadoop projects, schedule jobs and manage cluster resources.

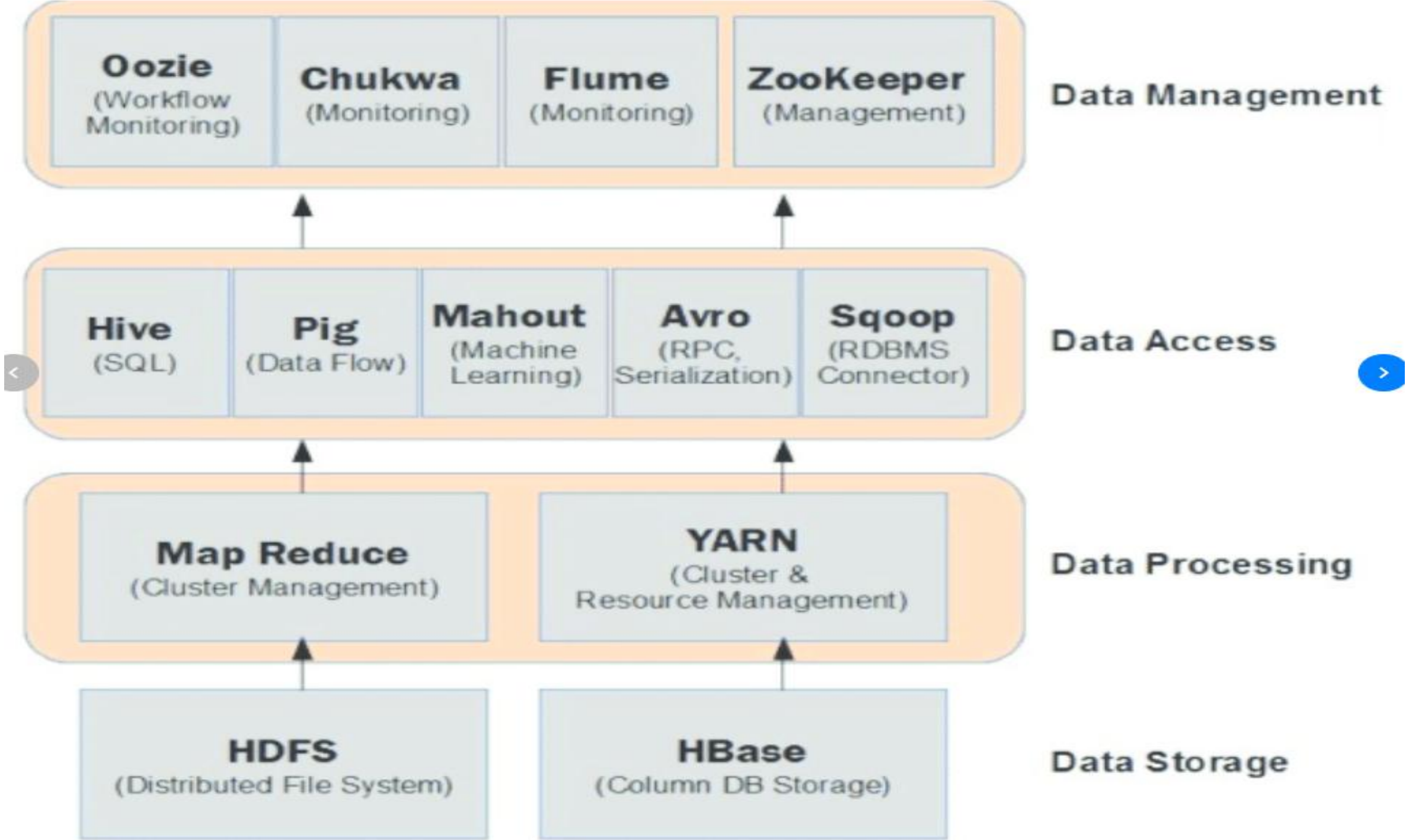
Features of Hadoop:-

- Hadoop uses googles file system and map reduces its foundation.
- Optimized to handle massive quantities of varying data using commodity hardware.
- It has shared nothing architecture.
- Data replication across multiple nodes.
- Not good if work cannot be parallelized.
- Best suited for huge files and huge data sets.

Advantages of hadoop

- Stores data in native format:- HDFS stores data in native format, no structure imposed while storing data.
- Scalable:- Hadoop can store and distribute large data sets across commodity hardware.
- Cost effective:-scale ou architecture, reduced cost per terabyte of storage.
- Flexibility:- ability to work with all kinds of data.
- Fast:- fast becoz follows move code to data

Elements at various stages of data processing



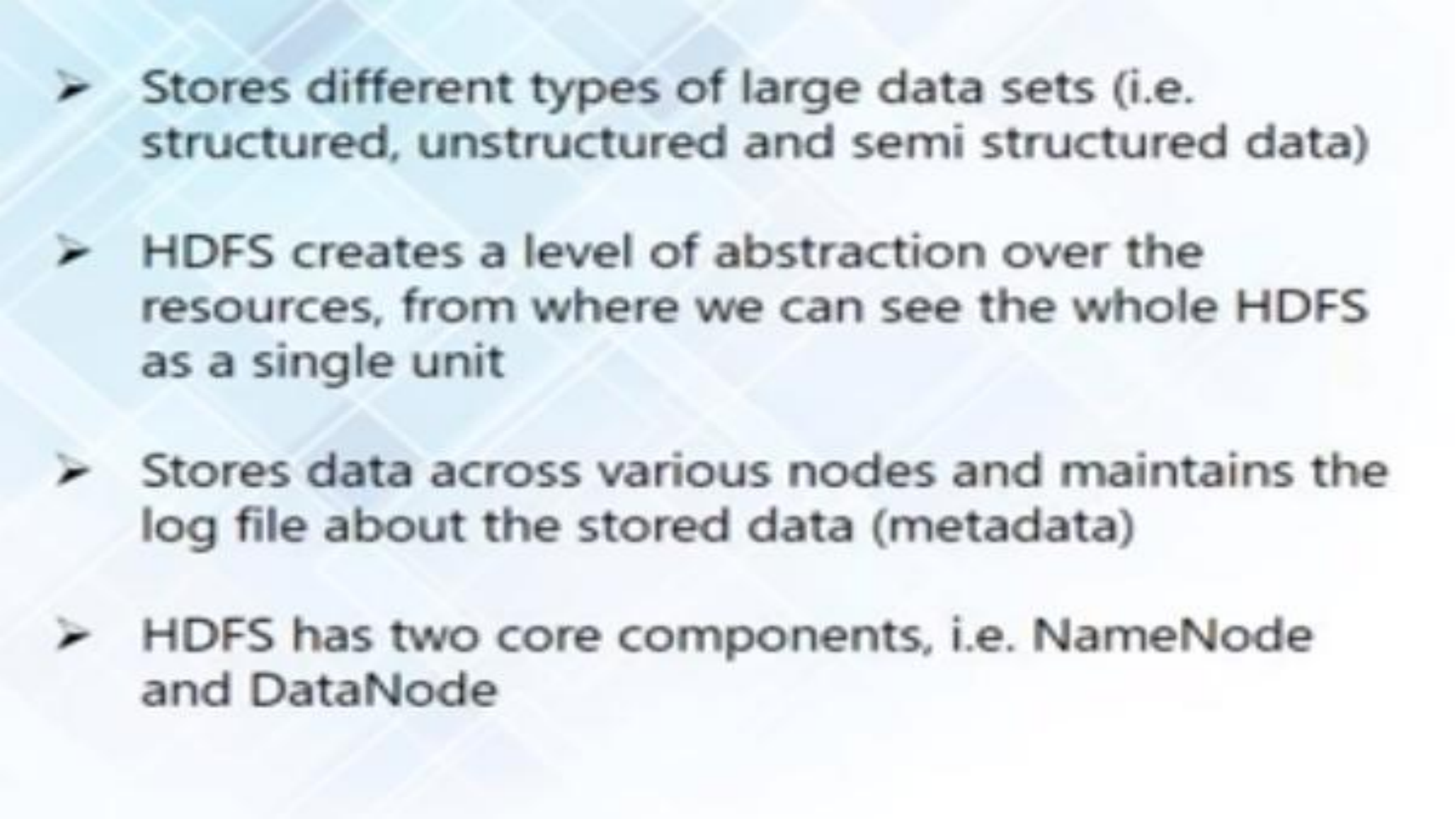
Oozie (Workflow Monitoring)
Chukwa (Monitoring)
Flume (Monitoring)
ZooKeeper (Management)
Data Management

Hive (SQL)
Pig (Data Flow)
Mahout (Machine Learning)
Avro (RPC, Serialization)
Sqoop (RDBMS Connector)
Data Access

Map Reduce (Cluster Management)
YARN (Cluster & Resource Management)
Data Processing

HDFS (Distributed File System)
HBase (Column DB Storage)
Data Storage

HDFS:-



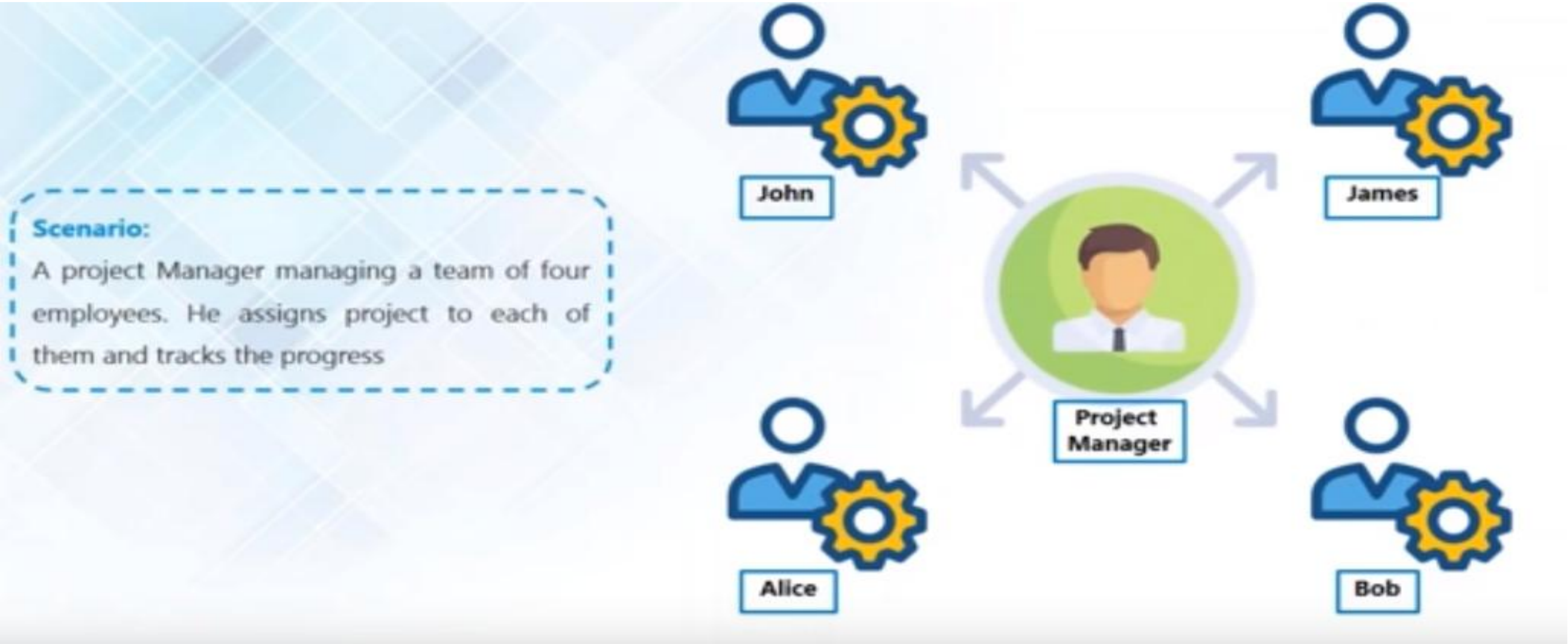
- Stores different types of large data sets (i.e. structured, unstructured and semi structured data)
- HDFS creates a level of abstraction over the resources, from where we can see the whole HDFS as a single unit
- Stores data across various nodes and maintains the log file about the stored data (metadata)
- HDFS has two core components, i.e. NameNode and DataNode

- Hdfs is effective, fault tolerant and distributed approach for storing and managing huge volumes of data.
- Data collected in Hadoop cluster is first broken into smaller chunks called blocks and distributed across multiple nodes.
- These smaller subsets of data are then operated upon by map and reduce functions.
- Result from each of these operations is combines together to provide aggregate outcome called big data solution.

- Hdfs keeps tracks of distributed pieces of data using filesystem metadata.
- Metadata “data about data”
- Acts as a template that provides the following information.

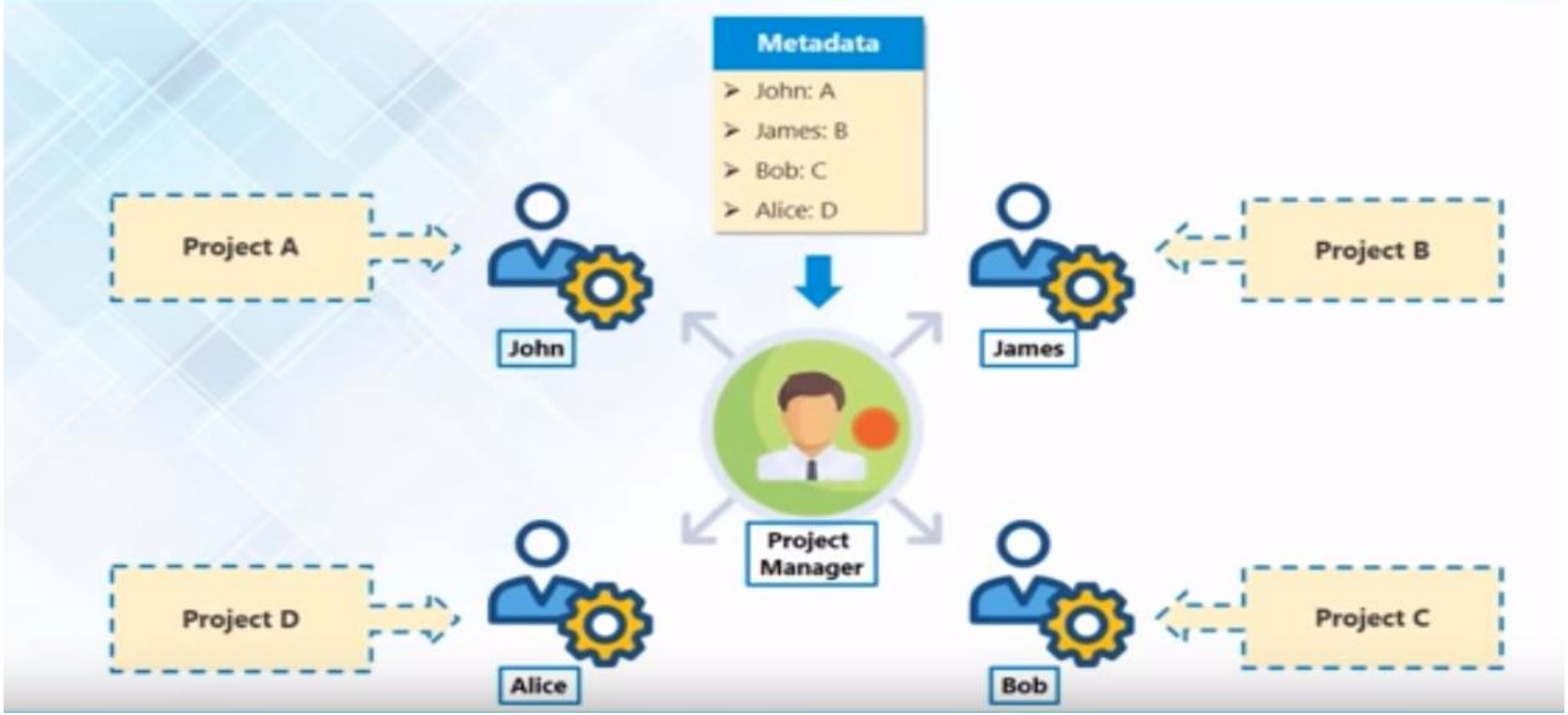
1. The time of creation, las access, modifications and deletion of file.
2. The storage location of blocks of file on a cluser.
3. Access permissions to view and modify a file.
4. Number of files stored on a cluster
5. Number of data nodes connected to form a cluster.
6. Location of transaction log on the cluster.

Hadoop(master slave architecture)



Scenario:

A project Manager managing a team of four employees. He assigns project to each of them and tracks the progress



Metadata

- John: A
- James: B
- Bob: C
- Alice: D

Project A

John

Project B

James

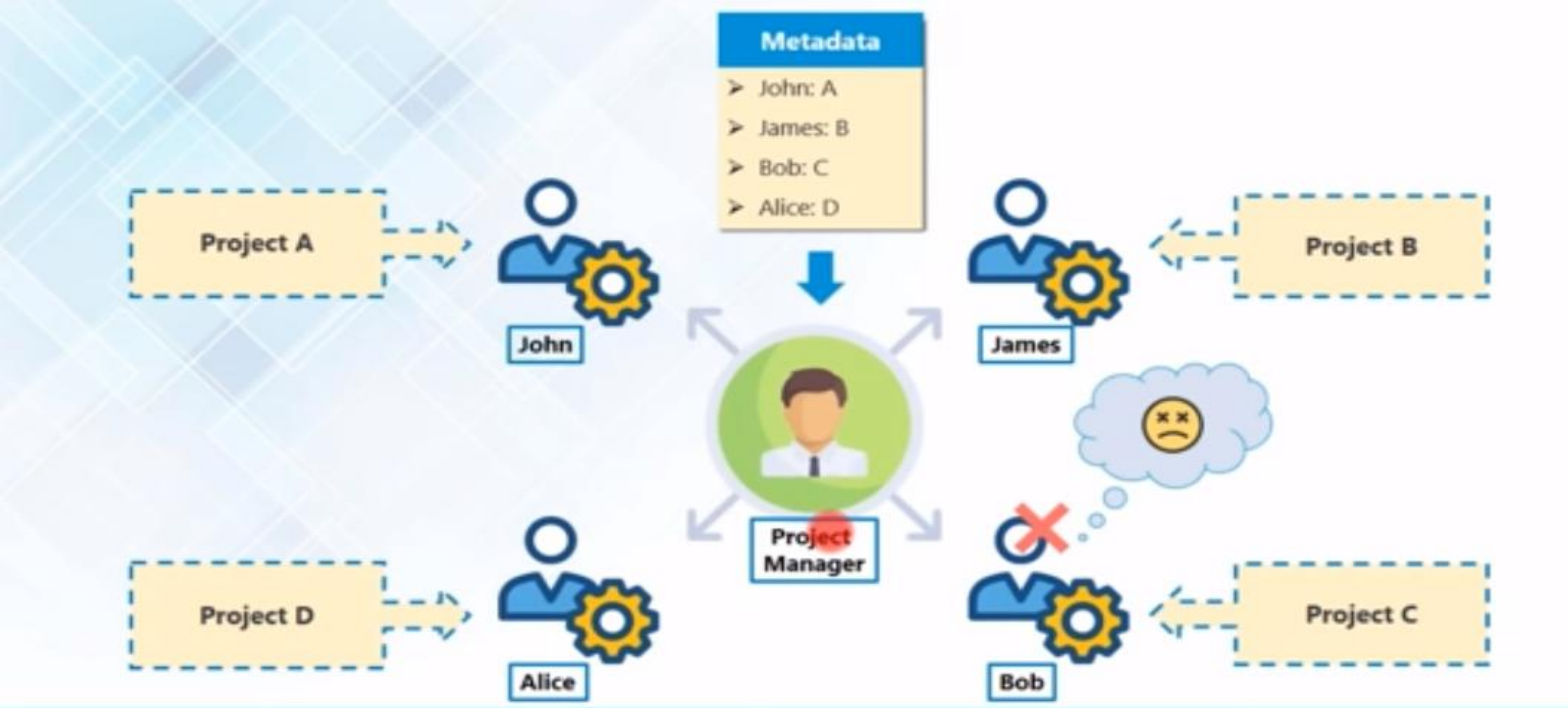
Project Manager

Project D

Alice

Project C

Bob



Metadata

- John: A
- James: B
- Bob: C
- Alice: D

Project A

John

Project B

James

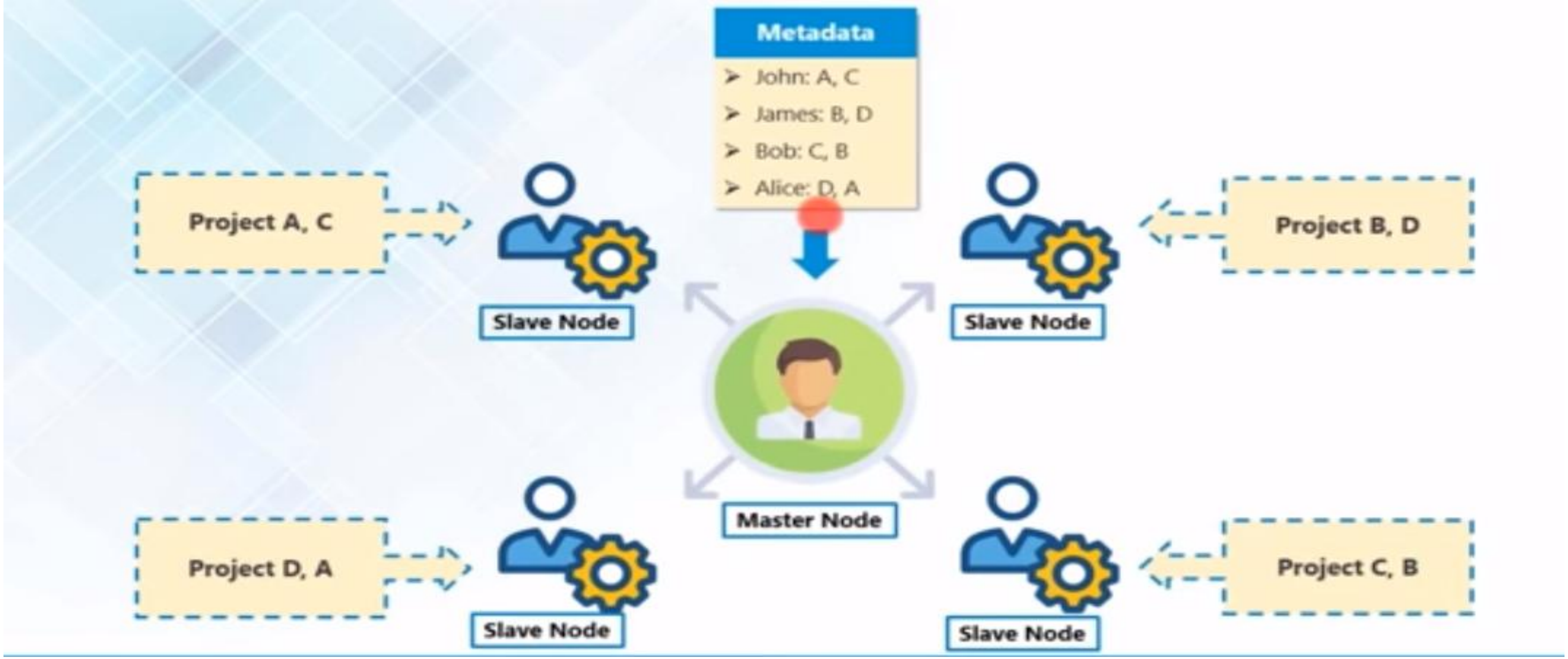
Project Manager

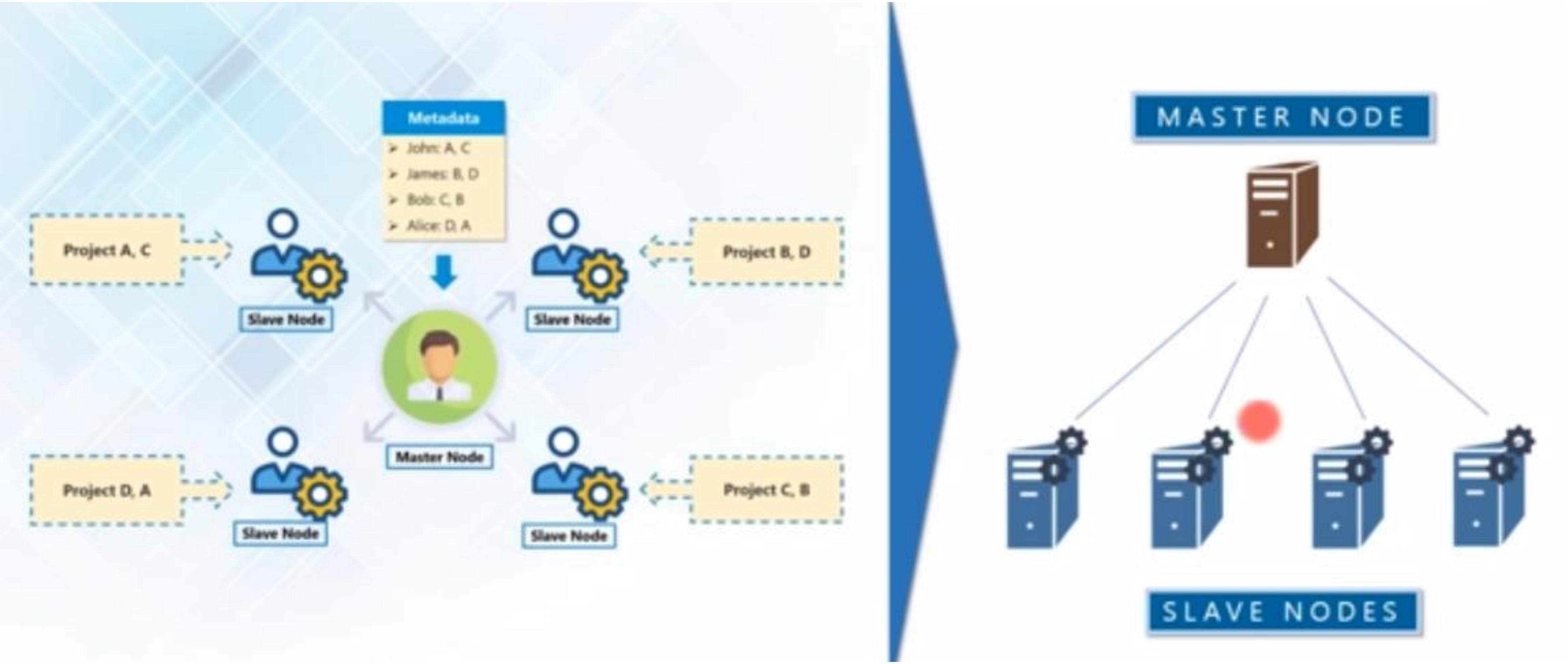
Project D

Alice

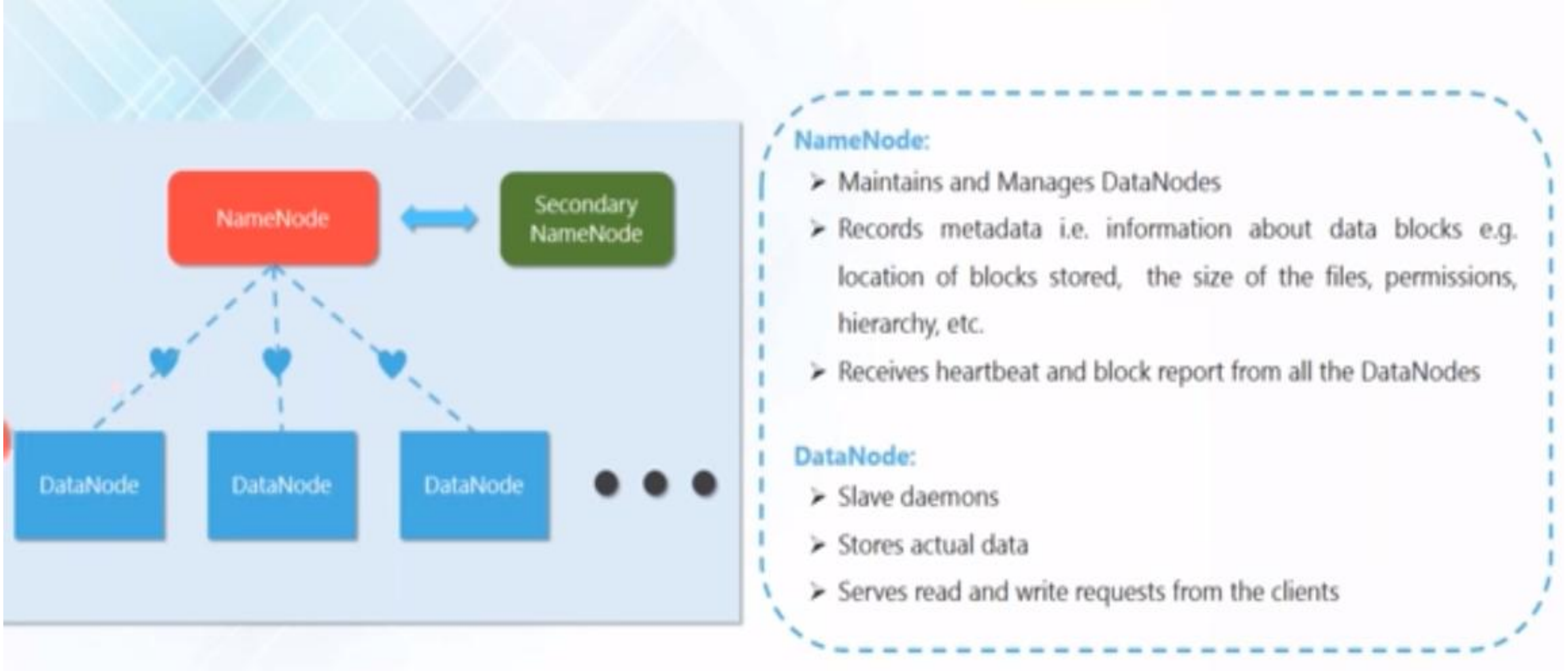
Project C

Bob


Metadata
John: A, C
James: B, D
Bob: C, B
Alice: D, A
Project A, C
Slave Node
Project B, D
Slave Node
Master Node
Project D, A
Slave Node
Project C, B
Slave Node


Metadata
John: A, C
James: B, D
Bob: C, B
Alice: D, A
Project A, C
Slave Node
Slave Node
Project B, D
Master Node
Project D, A
Slave Node
Slave Node
Project C, B
MASTER NODE
SLAVE NODES

Name node and data node?



NameNode:

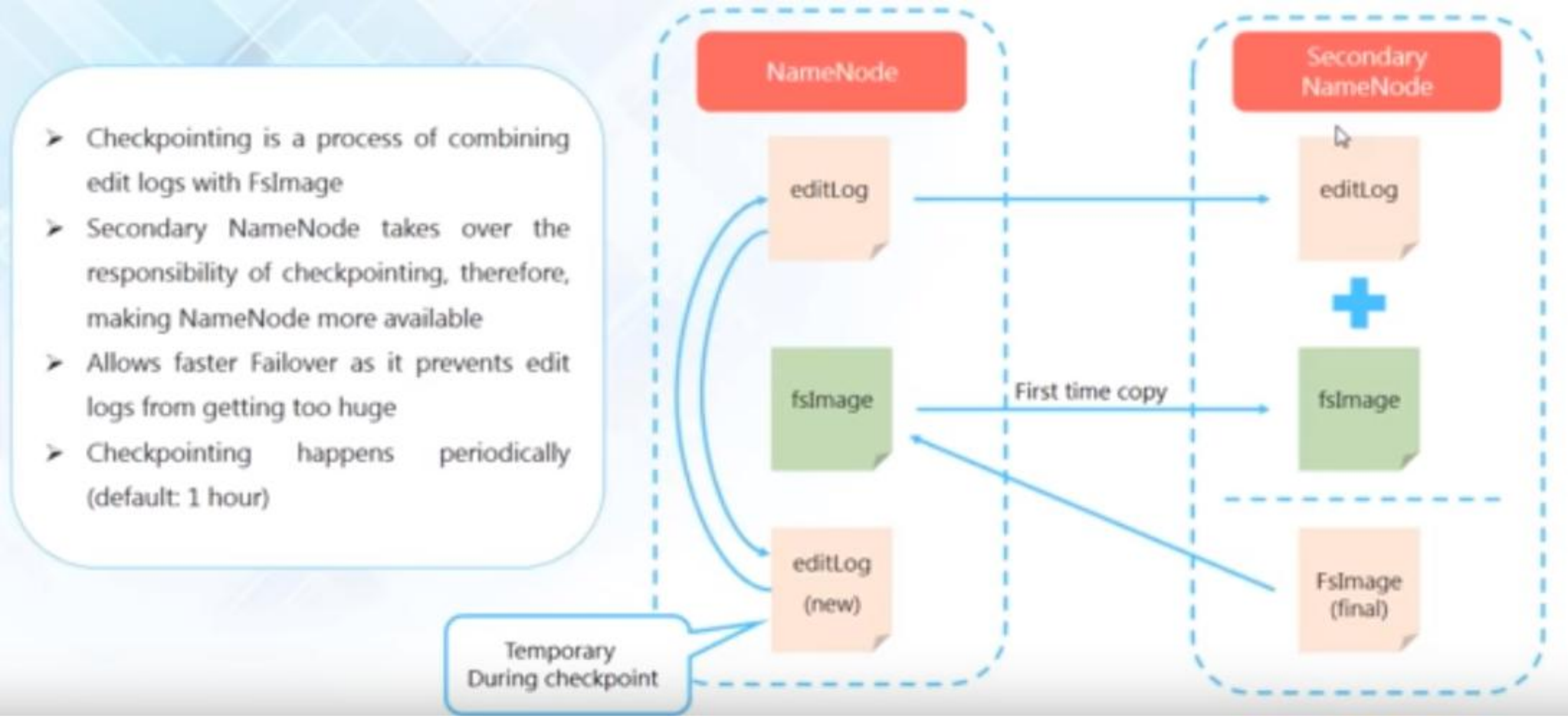
- Maintains and Manages DataNodes
- Records metadata i.e. information about data blocks e.g. location of blocks stored, the size of the files, permissions, hierarchy, etc.
- Receives heartbeat and block report from all the DataNodes

DataNode:

- Slave daemons
- Stores actual data
- Serves read and write requests from the clients

- Data nodes ensures connectivity with the name node by sending heartbeat messages.
- Whenever the name node ceases to receive a heartbeat message from data node it unmaps the data node from the cluster.
- When a heartbeat message re appears or a new heartbeat message is received, respective data node is added to the cluster.

Secondary node and checkpointing?



- ➢ Checkpointing is a process of combining edit logs with FsImage
- ➢ Secondary NameNode takes over the responsibility of checkpointing, therefore, making NameNode more available
- ➢ Allows faster Failover as it prevents edit logs from getting too huge
- ➢ Checkpointing happens periodically (default: 1 hour)

NameNode

editLog

fsImage

editLog
(new)

Temporary
During checkpoint

First time copy

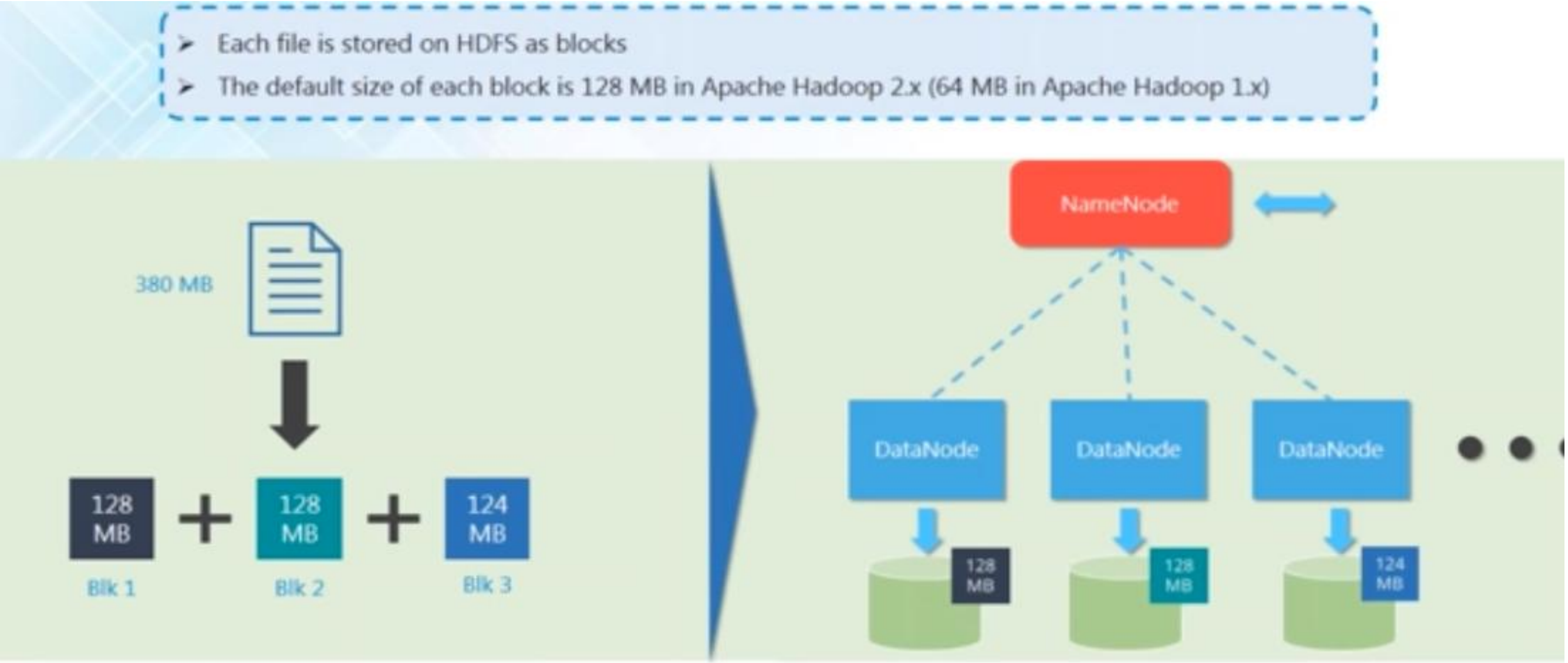
Secondary
NameNode

editLog

fsImage

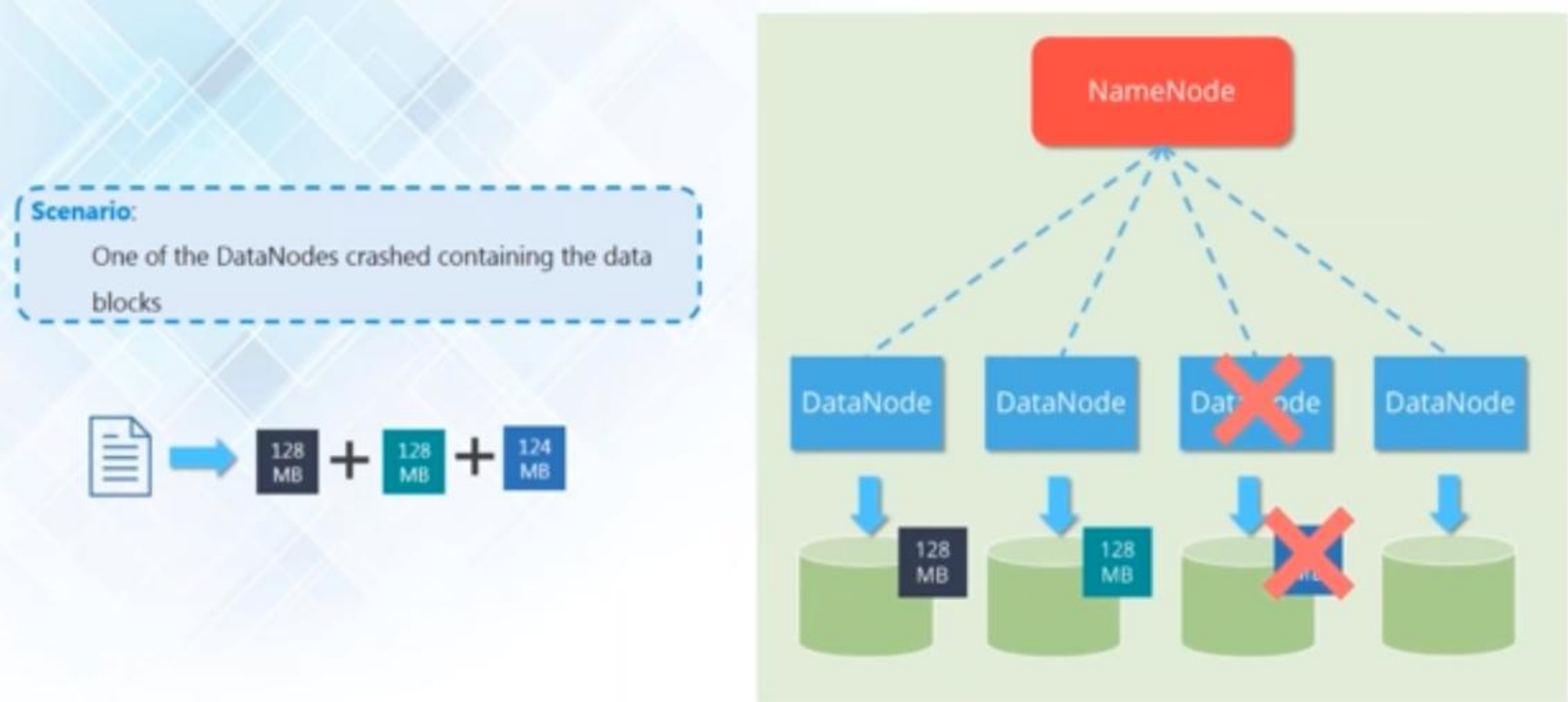
FsImage
(final)

HDFS data blocks?



- Each file is stored on HDFS as blocks
- The default size of each block is 128 MB in Apache Hadoop 2.x (64 MB in Apache Hadoop 1.x)

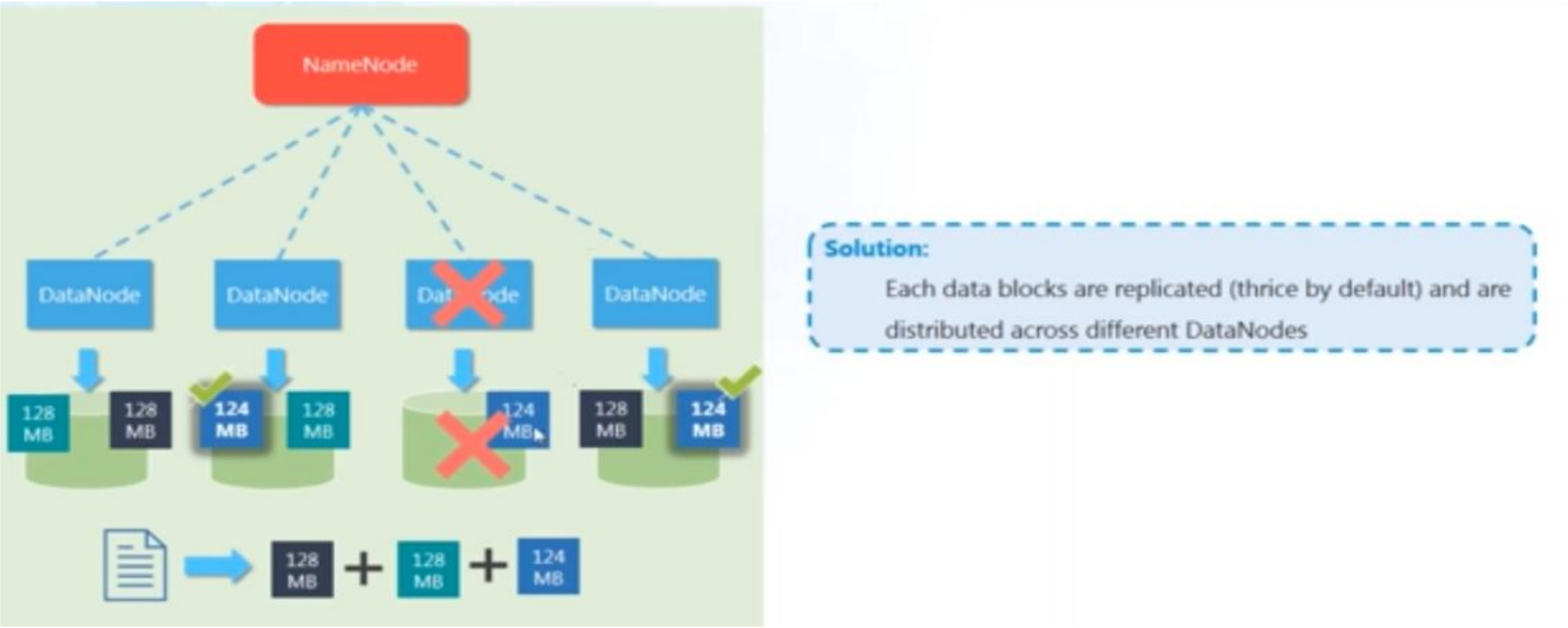
Failure of node??



Scenario:

One of the DataNodes crashed containing the data blocks

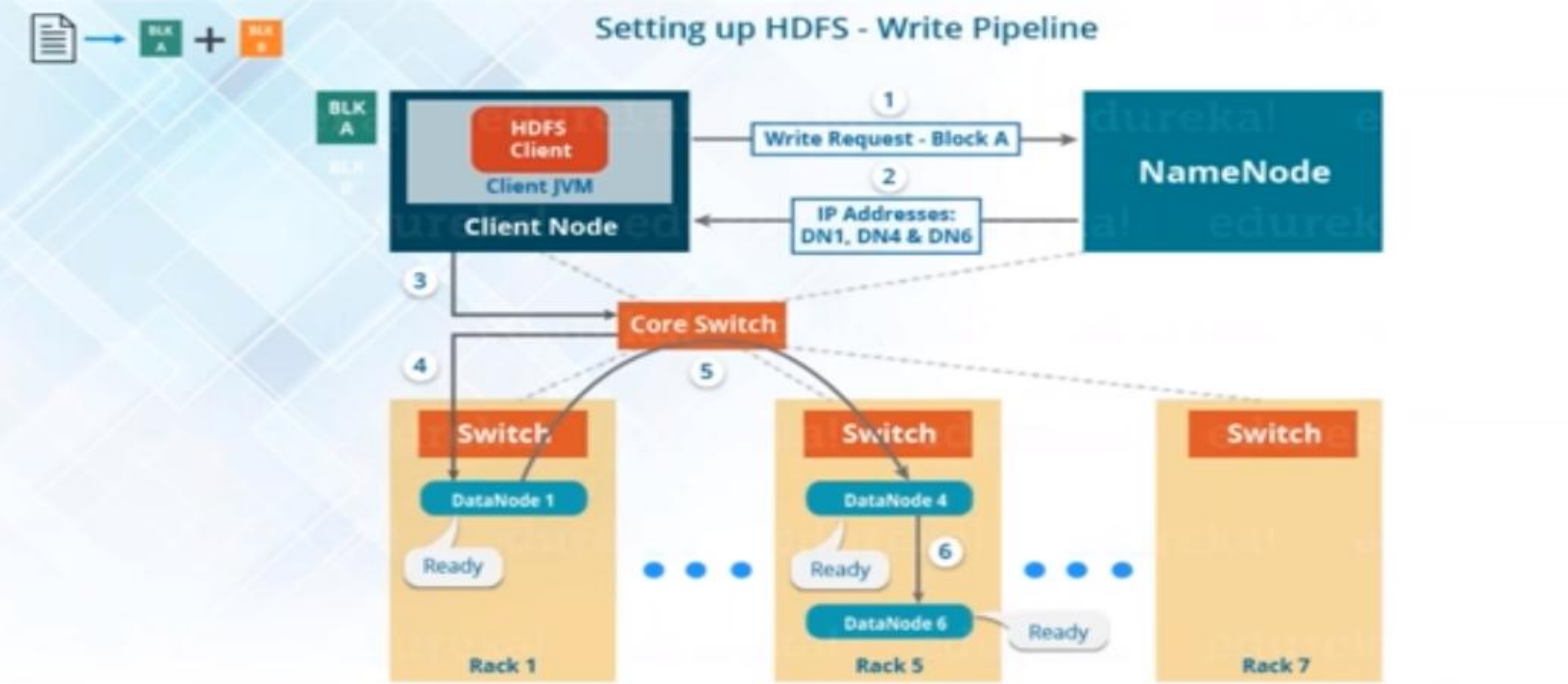
Replication factor.



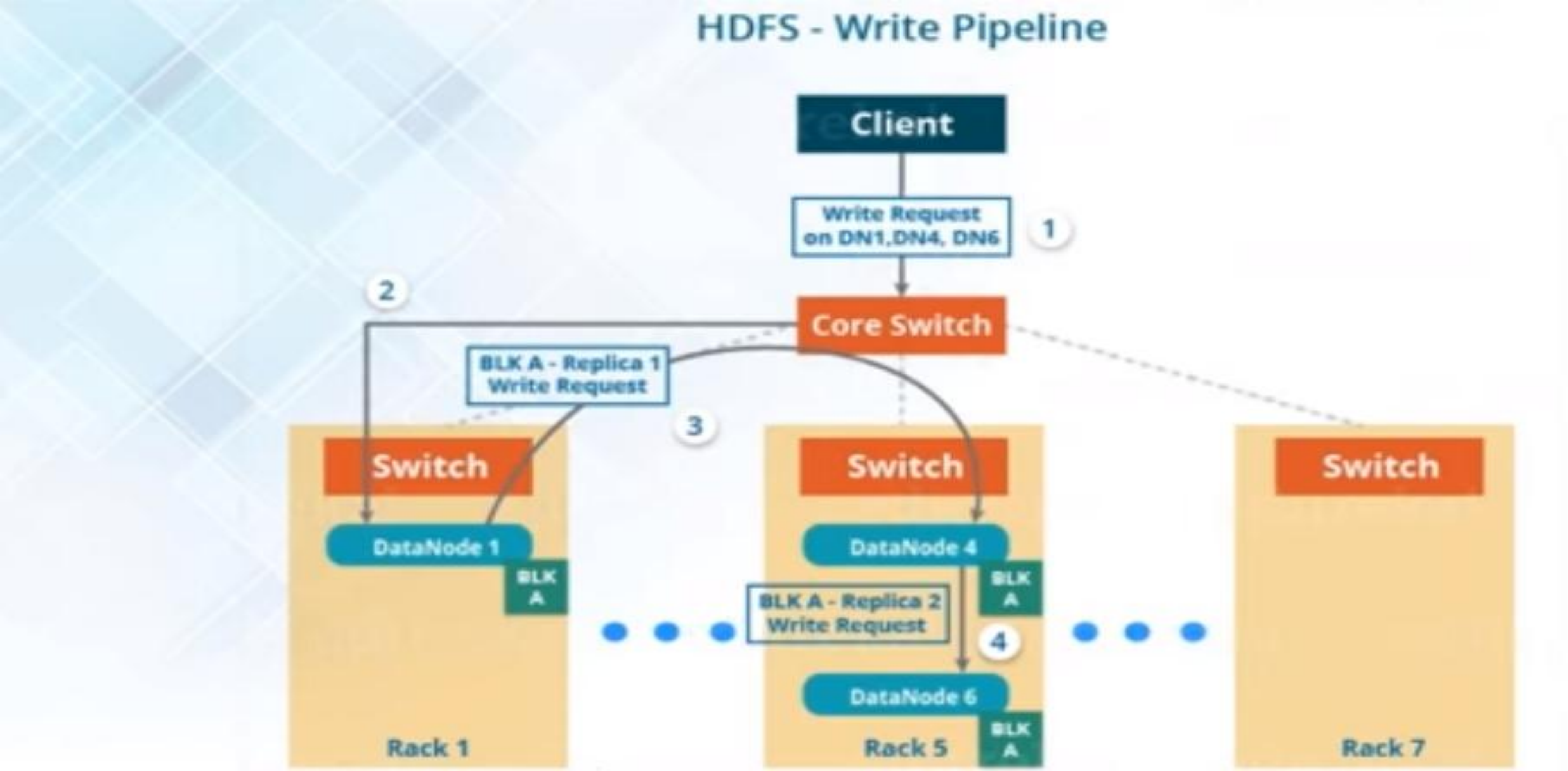
Solution:

Each data blocks are replicated (thrice by default) and are distributed across different DataNodes

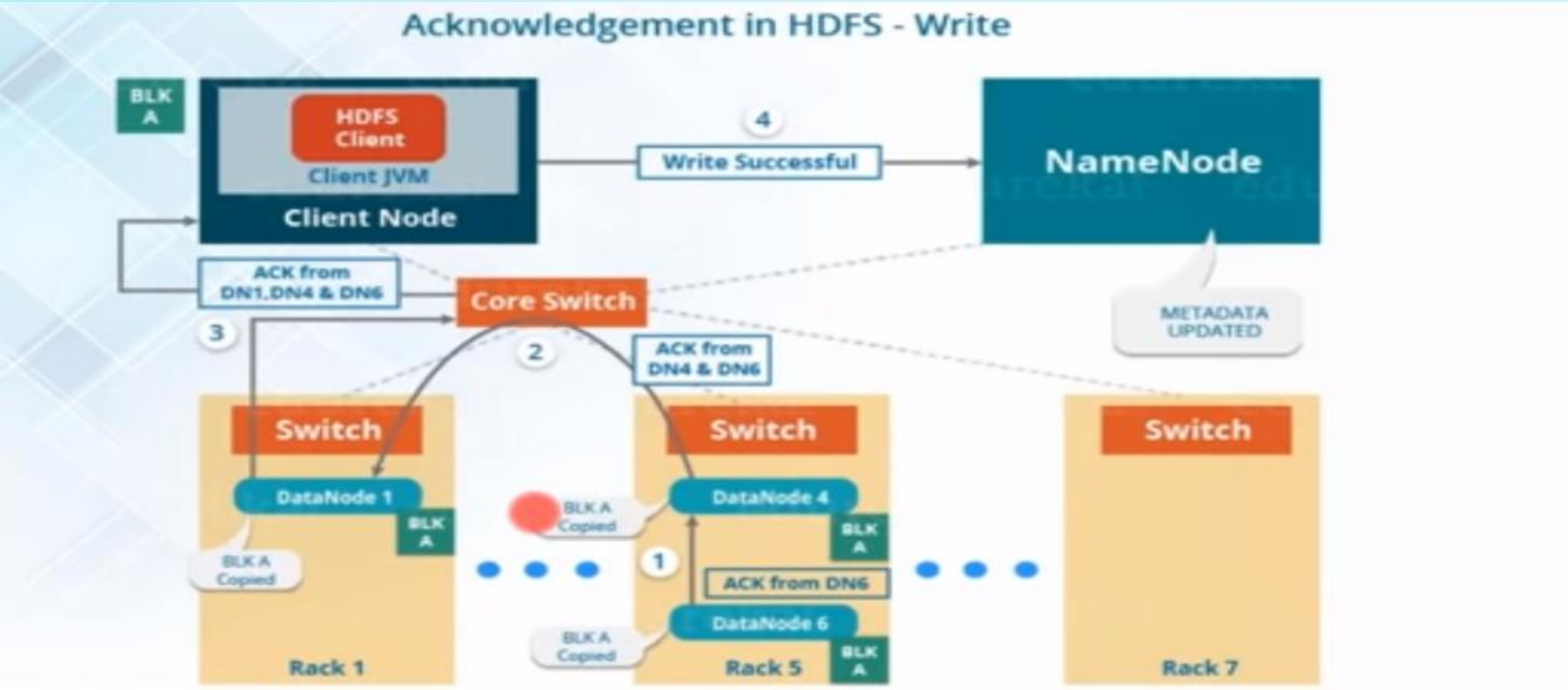
HDFS Write Mechanism(Pipeline setup)



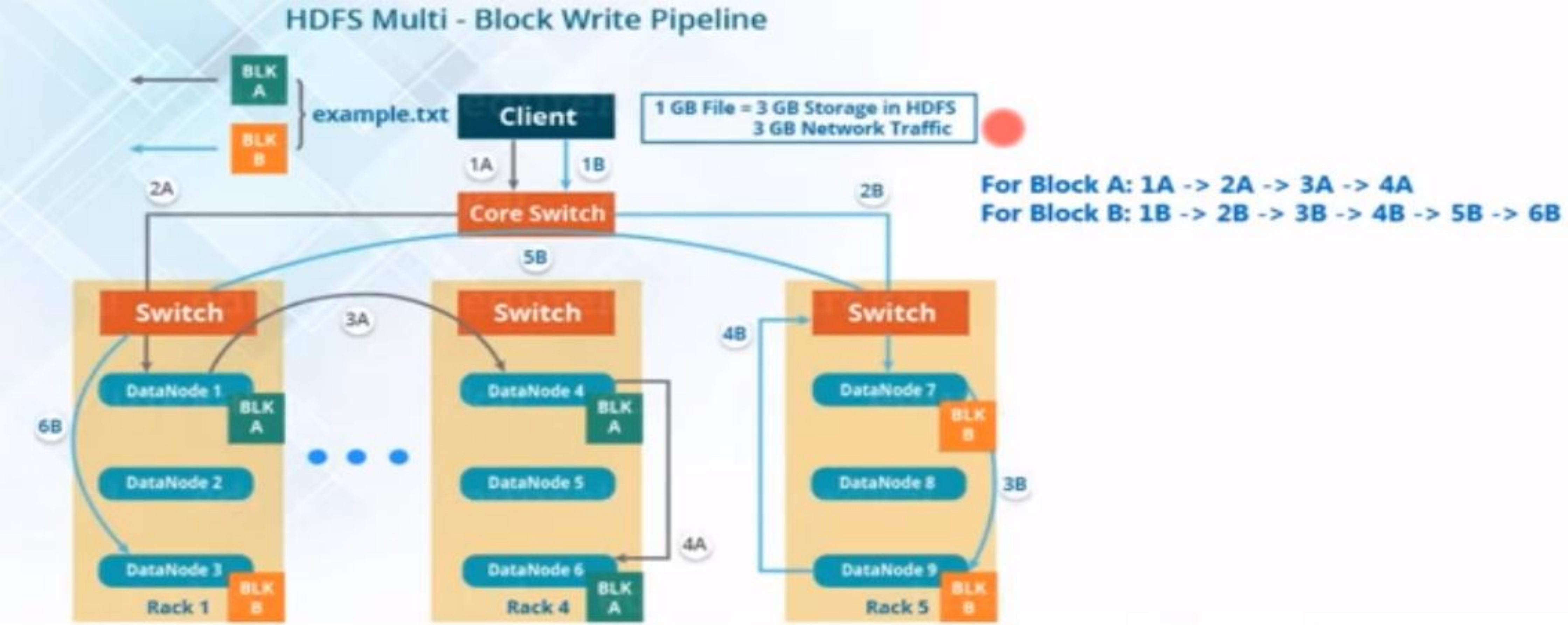
HDFS Write Mechanism(Writing a block)



HDFS write mechanism(Acknowledgement)



HDFS Multi block write mechanism

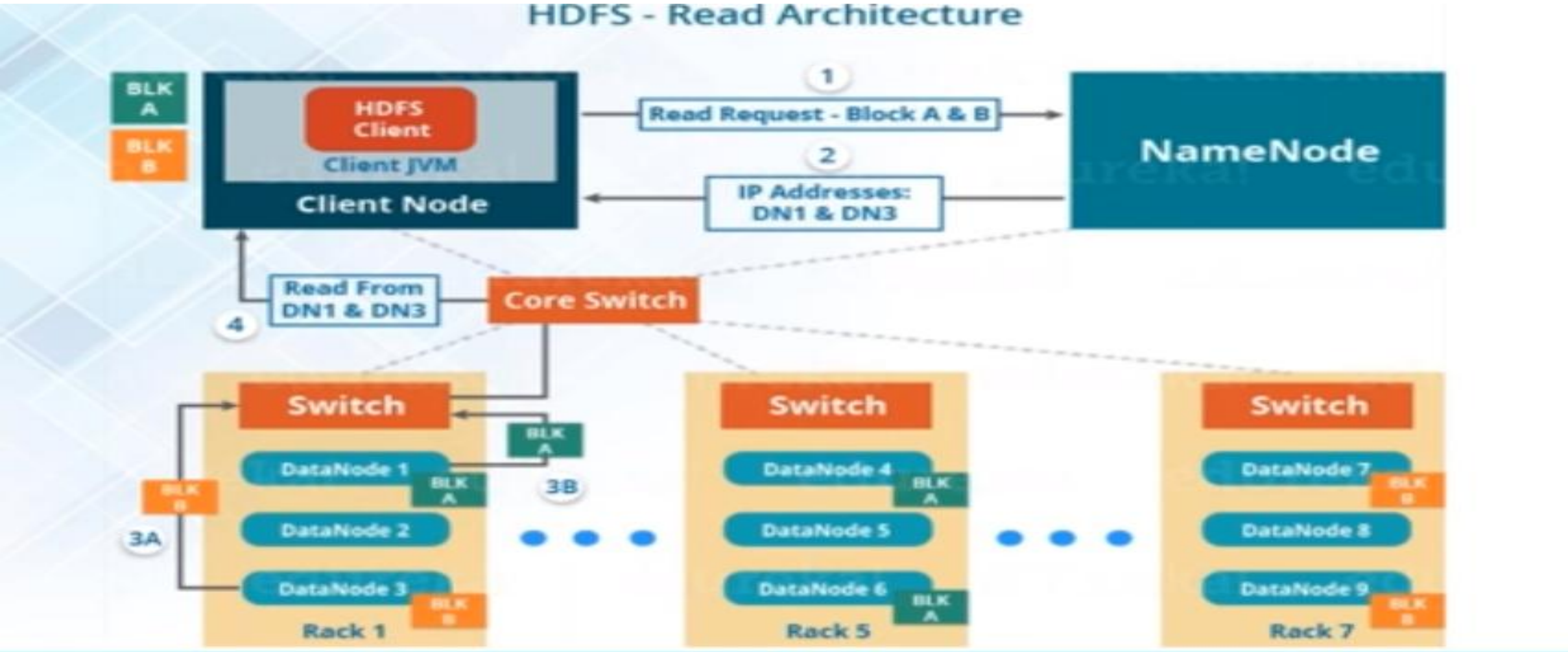


HDFS Multi - Block Write Pipeline

1 GB File = 3 GB Storage in HDFS
3 GB Network Traffic

For Block A: 1A -> 2A -> 3A -> 4A
For Block B: 1B -> 2B -> 3B -> 4B -> 5B -> 6B

HDFS (READ Mechanism)


HDFS - Read Architecture
BLK A
BLK B
HDFS Client
Client JVM
Client Node
1
Read Request - Block A & B
NameNode
2
IP Addresses: DN1 & DN3
Read From DN1 & DN3
4
Core Switch
Switch
DataNode 1
BLK A
DataNode 2
DataNode 3
BLK B
Rack 1
BLK B
3A
BLK A
3B
Switch
DataNode 4
BLK A
DataNode 5
DataNode 6
BLK A
Rack 5
Switch
DataNode 7
BLK B
DataNode 8
DataNode 9
BLK B
Rack 7