



Sapere utile

# **IFOA**

## **Istituto Formazione Operatori Aziendali**

### **BIG DATA e Analisi dei Dati**

#### **Lezione 2.1 – File systems**

Mauro Bellone,  
Robotics and AI researcher

[bellonemauro@gmail.com](mailto:bellonemauro@gmail.com)  
[www.maurobellone.com](http://www.maurobellone.com)

## **GitHub repo attivo**

Link per scaricare i tutorial che faremo durante questo corso:

[https://github.com/bellonemauro/Tutorial\\_corsoIFOA2021\\_big](https://github.com/bellonemauro/Tutorial_corsoIFOA2021_big)

## Obiettivo

- ✓ Comprensione del file system per sistemi computazionali
- ✓ Comprensione delle problematiche relative al processing e la memorizzazione di big data con i file system tradizionali
- ✓ Introduzione al file system di Hadoop e mapReduce

# Livelli di astrazione del software

Utenti e altre applicazioni di sistema

Interfaccia utente

GUI

batch

cmd

Chiamate di sistema

Servizi

Alloc. risorse

Prot. e sicurezza

Operazioni I/O

Comunicazione

File system

Detect errori

Sistema operativo

Hardware

# Livelli di astrazione del software

Utenti e altre applicazioni di sistema

Interfaccia utente

GUI

batch

cmd

Chiamate di sistema

Servizi

Alloc. risorse

Prot. e sicurezza

Operazioni I/O

Comunicazione

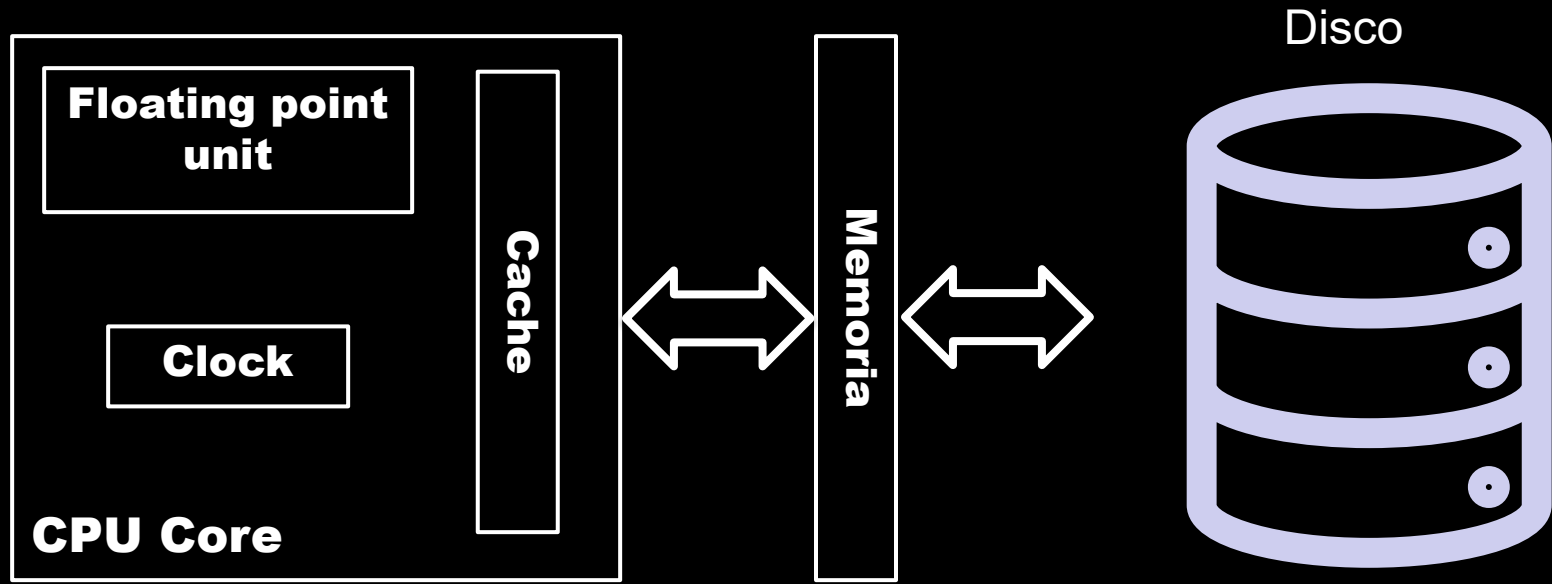
File system

Detect errori

Sistema operativo

Hardware





# File

Il file è una collezione nominate di informazioni memorizzate in una memoria secondaria.





# File

Il file è una collezione nominate di informazioni memorizzate in una memoria secondaria.

Dal punto di vista dell'utente un file è la più piccola quantità di memoria allottabile in una memoria (secondaria), dalla quale i dati possono essere letti o scritti.



# File

Il file è una collezione nominate di informazioni memorizzate in una memoria secondaria.

Dal punto di vista dell'utente un file è la più piccola quantità di memoria allottabile in una memoria (secondaria), dalla quale i dati possono essere letti o scritti.

Tipi di informazione memorizzabile: dati numerici, caratteri o programmi (any!)



# File

Il file è una collezione nominate di informazioni memorizzate in una memoria secondaria.

Dal punto di vista dell'utente un file è la più piccola quantità di memoria allottabile in una memoria (secondaria), dalla quale i dati possono essere letti o scritti.

Tipi di informazione memorizzabile: dati numerici, caratteri o programmi (any!)

Un file ha una struttura definita che dipende dal suo tipo (da cui l'estensione)



## Attributi di un file

- **Nome:** tag leggibile
- **Identificatore:** tag unico (tenuto nel file system e univoco per ogni file)
- **Tipo:** documento, eseguibile etc.
- **Locazione:** puntatore alla locazione di memoria iniziale del file
- **Dimensione:** dimensione corrente in memoria
- **Protezione:** controllo di sicurezza per accesso in lettura, scrittura, esecuzione
- **Data, ora e identificazione utenti:** attribute temporali e monitoraggio di uso

## Operazioni su un file

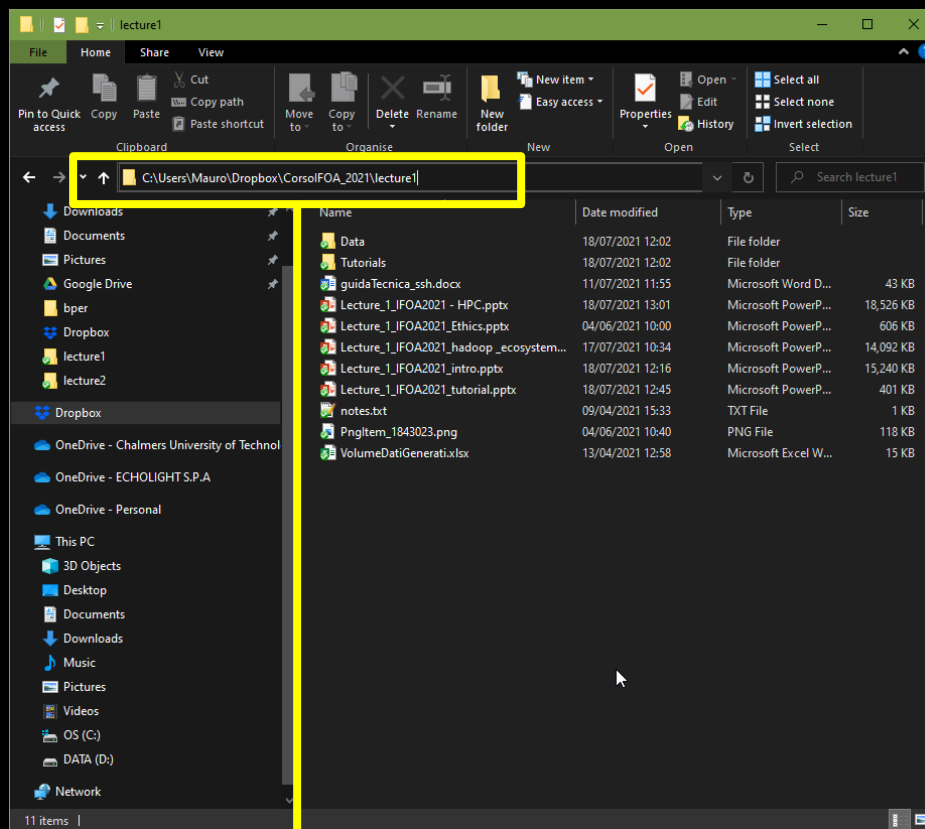
- **Creazione:** tag leggibile
- **Scrittura:** tag unico (tenuto nel file system e univoco per ogni file)
- **Lettura:** Si ritorna il puntatore al cluster di memoria iniziale del file da leggere
- **Spostamento:** Il puntatore viene aggiornato su un altro valore
- **Cancellazione:** Rilascia lo spazio allocato per quell file
- **Troncaggio:** La lunghezza del file viene ridotta
- **Rinominazione, apertura, chiusura:** Copia il blocco di memoria nella RAM o lo dealloca

## **Directories**

Troppi file in una lista sono complessi da gestire, abbiamo bisogno di raggruppamenti di files detti “cartelle”

# Directories

Troppi file in una lista sono complessi da gestire, abbiamo bisogno di raggruppamenti di files detti “cartelle”



`C:\Users\Mauro\Dropbox\CorsoIFOA_2021\lecture1`

# Struttura dei dischi

A - Traccia

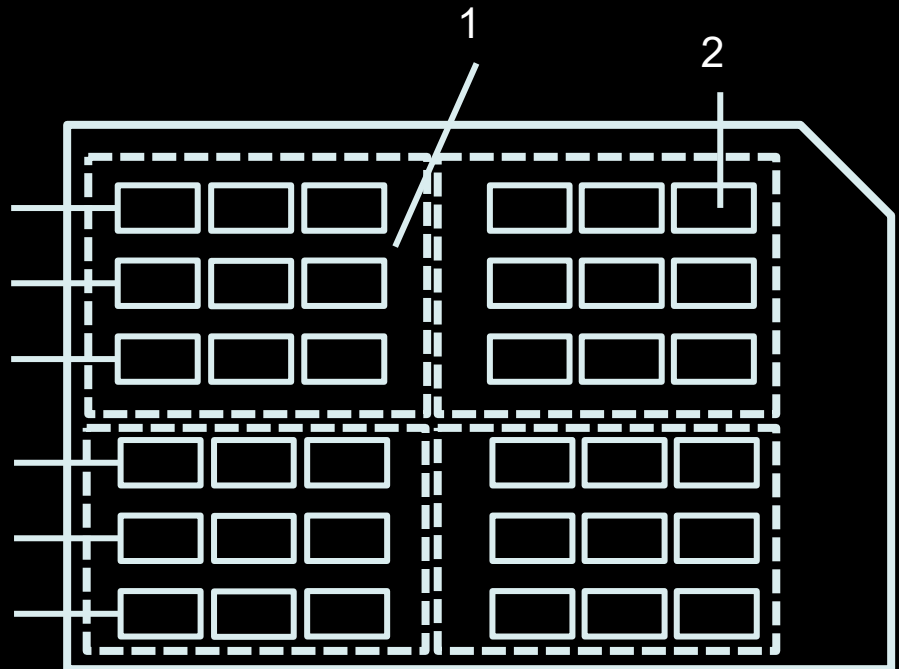
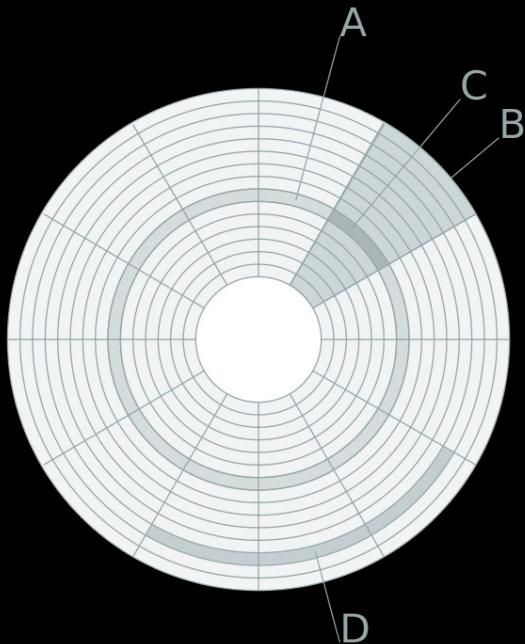
B - Settore geometrico

C - Settore di traccia

D - Cluster

1 - Blocco

2 - Cluster

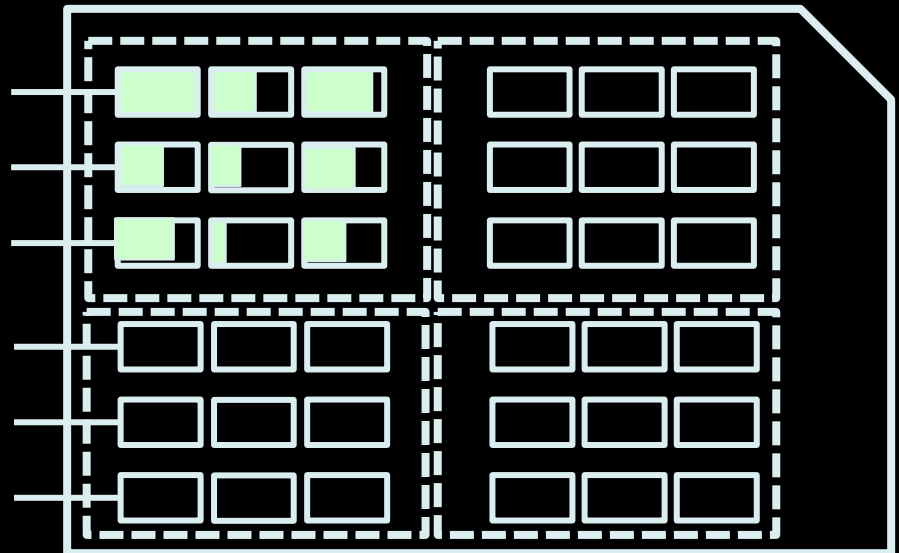




# Memorizzazione

Un file viene memorizzato in uno o più clusters, se il cluster è pieno si alloca un altro cluster, se il cluster non è pieno, lo spazio rimanente nel cluster resta non allocato.

- ✓ Uso del disco non ottimo
- ✓ Frammentazione

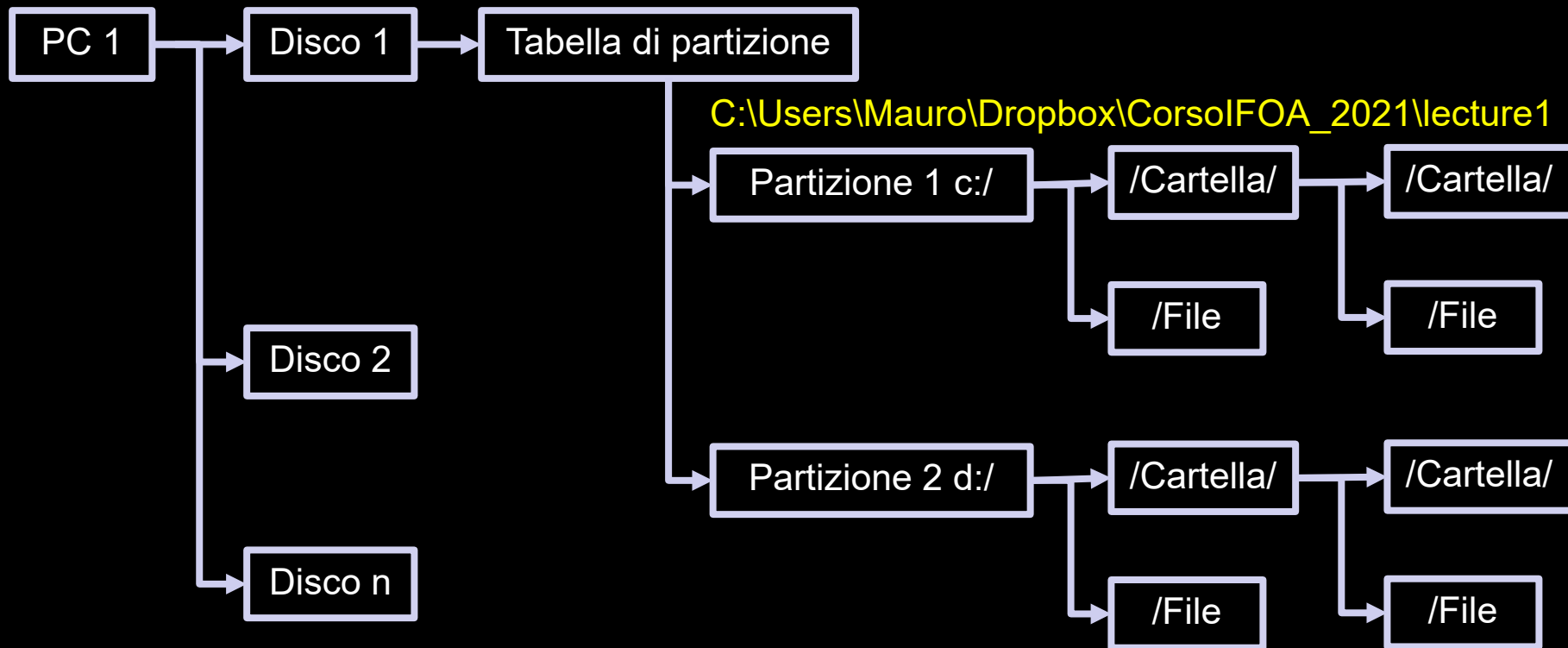


## **Cos'è il file system**

Il file system è un meccanismo standard di allocazione e organizzazione dei dati su memoria

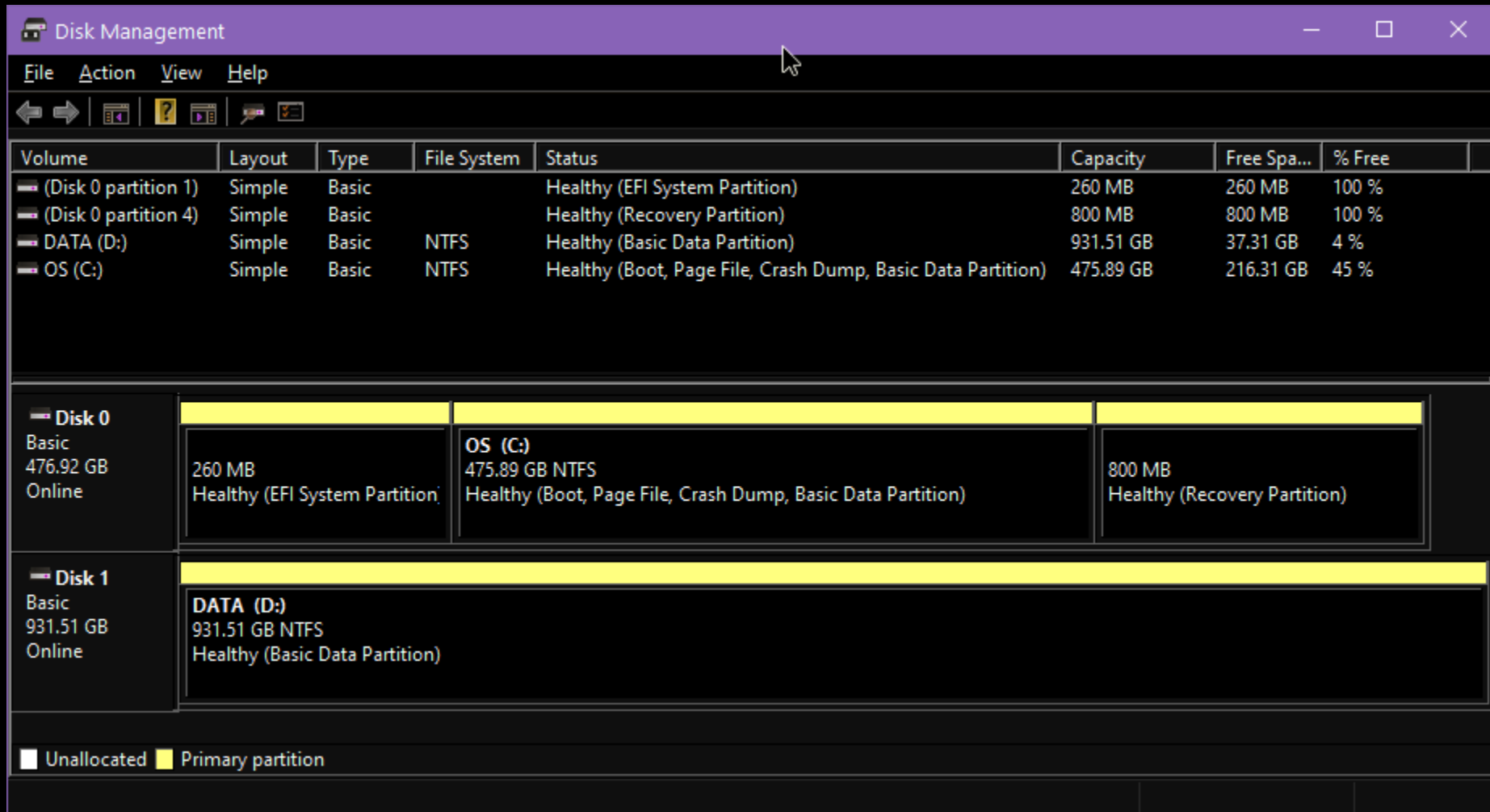
# Cos'è il file system

Il file system è un meccanismo standard di allocazione e organizzazione dei dati su memoria



# Cos'è il file system

Il file system è un meccanismo standard di allocazione e organizzazione dei dati su memoria



The screenshot displays the Windows Disk Management console. At the top, a table lists the volumes on the system. Below this, the details for Disk 0 and Disk 1 are shown, including their layout, type, file system, status, capacity, and free space. A legend at the bottom indicates that white represents unallocated space and yellow represents primary partitions.

Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free
(Disk 0 partition 1)	Simple	Basic		Healthy (EFI System Partition)	260 MB	260 MB	100 %
(Disk 0 partition 4)	Simple	Basic		Healthy (Recovery Partition)	800 MB	800 MB	100 %
DATA (D:)	Simple	Basic	NTFS	Healthy (Basic Data Partition)	931.51 GB	37.31 GB	4 %
OS (C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Basic Data Partition)	475.89 GB	216.31 GB	45 %

Disk	Layout	Type	File System	Status	Capacity	Free Space	% Free
Disk 0	Simple	Basic		Healthy (EFI System Partition)	260 MB	260 MB	100 %
Disk 0	Simple	Basic		Healthy (Recovery Partition)	800 MB	800 MB	100 %
Disk 0	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Basic Data Partition)	475.89 GB	216.31 GB	45 %
Disk 1	Simple	Basic	NTFS	Healthy (Basic Data Partition)	931.51 GB	37.31 GB	4 %

Legend:  Unallocated  Primary partition

## **Tipi di file system**

- ✓ File Partition Table FAT 8/12/16/32
- ✓ New Technology File System NTFS
- ✓ EXT2/3/4
- ✓ exFAT

# File Partition Table FAT 8/12/16/32

## ✓ File Partition Table FAT 8/12/16/32

I file system sono costituiti da una tabella di voci con degli attributi (es. isDirectory, isReadOnly, hidden etc)

Attributo	Dimensione
Filename	8B
Extension	3B
Attributi	1B
Riservato	1B
Data di creazione	3B
Ora di creazione	3B
Ultimo accesso	2B
Data ultima modifica	2B
Ora ultima modifica	2B
Primo cluster	2B
Dimensione file	4B
First cluster	MSB, 2B
First cluster	LSB, 2B

# Clusters

Tipicamente la dimensione del cluster varia tra 2 to 32 KiB per FAT8 e FAT12 e aumenta a 256kB per FAT16 e FAT32

FAT12	FAT16	FAT32	Descrizione
0x000	0x0000	0x00000000	Cluster libero
0x001	0x0001	0x00000001	Valore riservato
0x002-0xFE0	0x0002-0xFFEF	0x00000002-0x0FFFFFFEF	Cluster dati
0xFF0-0xFF6	0xFFFF0-0xFFFF6	0xFFFFFFFF0-0xFFFFFFFF6	Valori riservati
0xFF7	0xFFFF7	0xFFFFFFFF7	Cluster danneggiato
0xFF8-0xFFFF	0xFFFF8-0xFFFFF	0xFFFFFFFF8-0xFFFFFFFFF	Ultimo cluster

# Clusters

Tipicamente la dimensione del cluster varia tra 2 to 32 KiB per FAT8 e FAT12 e aumenta a 256kB per FAT16 e FAT32

FS	Dimensione file max	Dimensione volume max	
FAT12	32 MB (8kB clusters)	10 bit = 4096 clusters	32 MB (8kB clusters)
FAT16	2 GB / GB	16 bit = 65536 clusters	16 GB (256kB clusters)
FAT32	4 GB	32 bit = 4 Giga-clusters	32 GB - windows format 2TB - altri OS 16 TB - teorici



# Clusters

Tipicamente la dimensione del cluster varia tra 2 to 32 KiB per FAT8 e FAT12 e aumenta a 256kB per FAT16 e FAT32

FS	Dimensione file max	Dimensione volume max	
FAT12	32 MB (8kB clusters)	10 bit = 4096 clusters	32 MB (8kB clusters)
FAT16	2 GB / GB	16 bit = 65536 clusters	16 GB (256kB clusters)
FAT32	4 GB	32 bit = 4 Giga-clusters	32 GB - windows format 2TB - altri OS 16 TB - teorici

## Limiti del file system

- Dimensione massima del file supportato
- Dimensione massima della memoria
- Massima profondità dell'albero delle cartelle

# New Technology File System (NTFS)

- ✓ Dimensione massima del file supportato **16 EiB**
- ✓ Dimensione massima della memoria  **$2^{64}$  clusters**
- ✓ Introduzione del journaling (evita la corruzione)
- ✓ Supporta permessi di lettura scrittura su files e criptaggio

## Extended file allocation table - exFAT

- ✓ Dimensione massima del file supportato **16 EiB**
- ✓ Ottimizzato per sistemi USB ad alta capacità
- ✓ Largamente supportato su non-Windows OS
- ✓ Installato di default sulle memorie SDXC (schede di memoria SD per camere e smartphone)

## **Fourth extended filesystem - Ext4**

- ✓ Dimensione massima del file supportato **16 TB**
- ✓ Dimensione massima della memoria **1 EB (4 kB cluster size)**
- ✓ Supportato da tutti i sistemi operativi

## To wrap up

- ✓ I file system hanno il compito di gestire l'allocazione delle informazioni
- ✓ I file system hanno una capacità limitata data dalla quantità di memoria fisica indirizzabile
- ✓ Tutti i file system hanno delle intrinseche limitazioni

## To wrap up

- ✓ I file system hanno il compito di gestire l'allocazione delle informazioni
- ✓ I file system hanno una capacità limitata data dalla quantità di memoria fisica indirizzabile
- ✓ Tutti i file system hanno delle intrinseche limitazioni

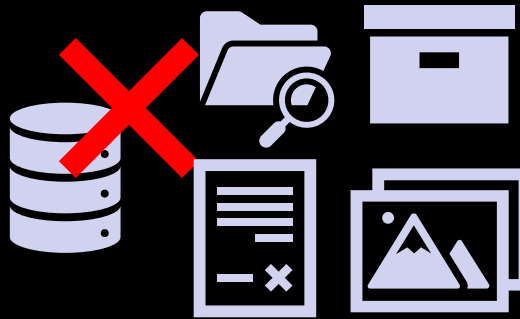
**In termini di big data tutte le limitazioni sono amplificate dalla quantità e dal flusso di informazioni**

# Problemi

- ✓ Memorizzazione
- ✓ Processamento

# Problemi

- ✓ Memorizzazione
- ✓ Processamento

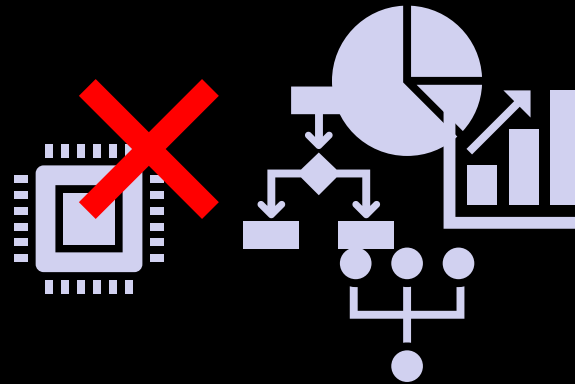


È un problema a causa del grande volume di informazioni, nessuno dei file system più evoluti può gestire quantità di memoria e informazioni dell'ordine dei ZB



# Problemi

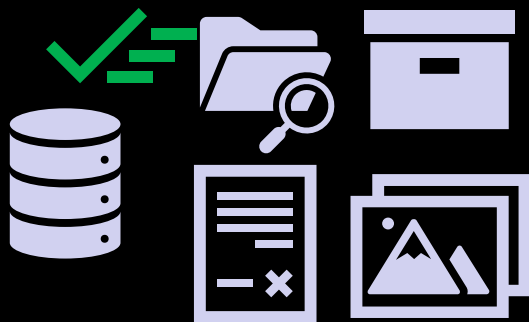
- ✓ Memorizzazione
- ✓ **Processamento**



Il processamento potrebbe richiedere un tempo molto lungo

# Hadoop file system

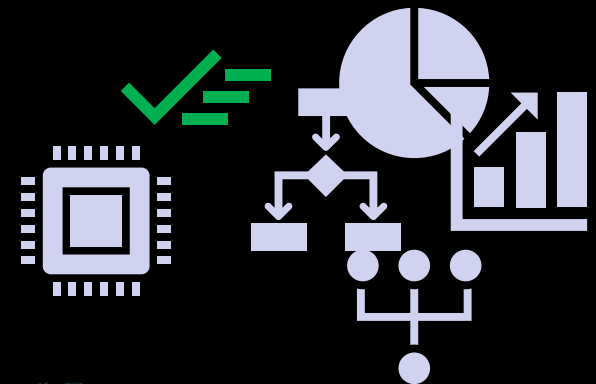
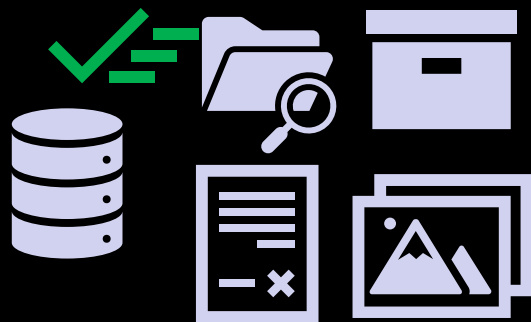
Il file system di Hadoop ci permette di memorizzare dati in maniera distribuita risolvendo il problema della memorizzazione



## Hadoop file system

Il file system di Hadoop ci permette di memorizzare dati in maniera distribuita  
risolvendo il problema della memorizzazione

Mentre mapReduce ci aiuta a risolvere il problema di processare l'informazione





Sapere utile

# **IFOA**

## **Istituto Formazione Operatori Aziendali**

### **BIG DATA e Analisi dei Dati**

#### **Lezione 2.2 – Hadoop distributed file system**

Mauro Bellone,  
Robotics and AI researcher

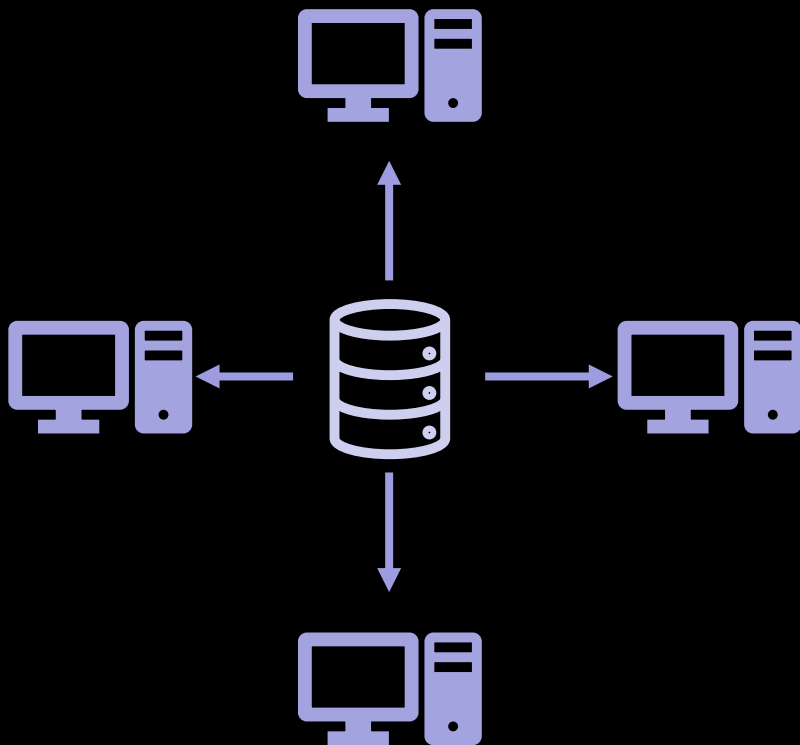
[bellonemauro@gmail.com](mailto:bellonemauro@gmail.com)  
[www.maurobellone.com](http://www.maurobellone.com)

## **Obiettivo**

- ✓ Comprensione del file system di hadoop
- ✓ Comprensione delle basi di lavoro sul file system di Hadoop

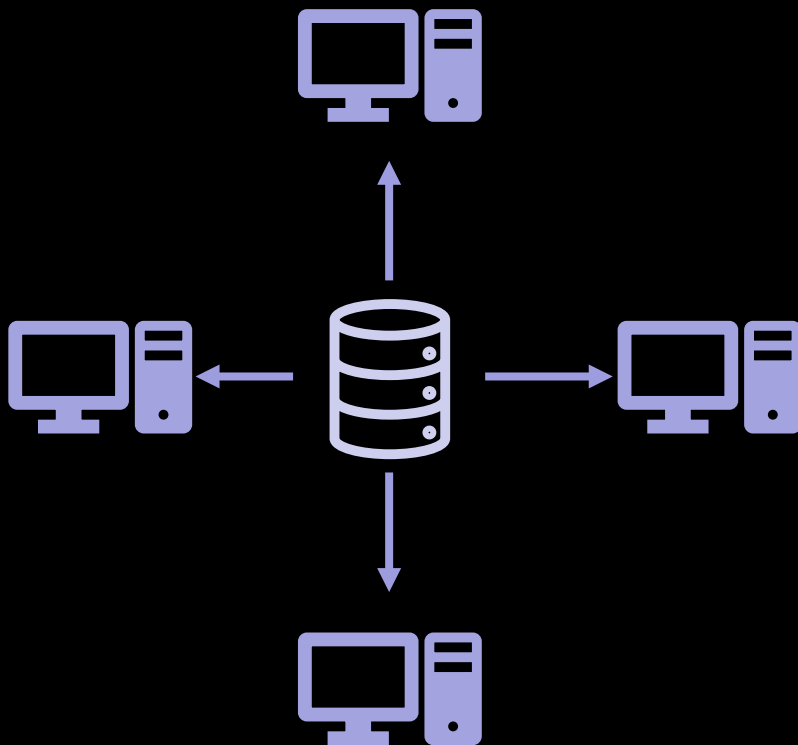
## HDFS – Hadoop distributed file system

Il file system di Hadoop è specificatamente progettato per memorizzare grandi dataset in commodity hardware (no servers, mainframes etc.)



## HDFS – Hadoop distributed file system

Il file system di Hadoop è specificatamente progettato per memorizzare grandi dataset in commodity hardware (no servers, mainframes etc.)

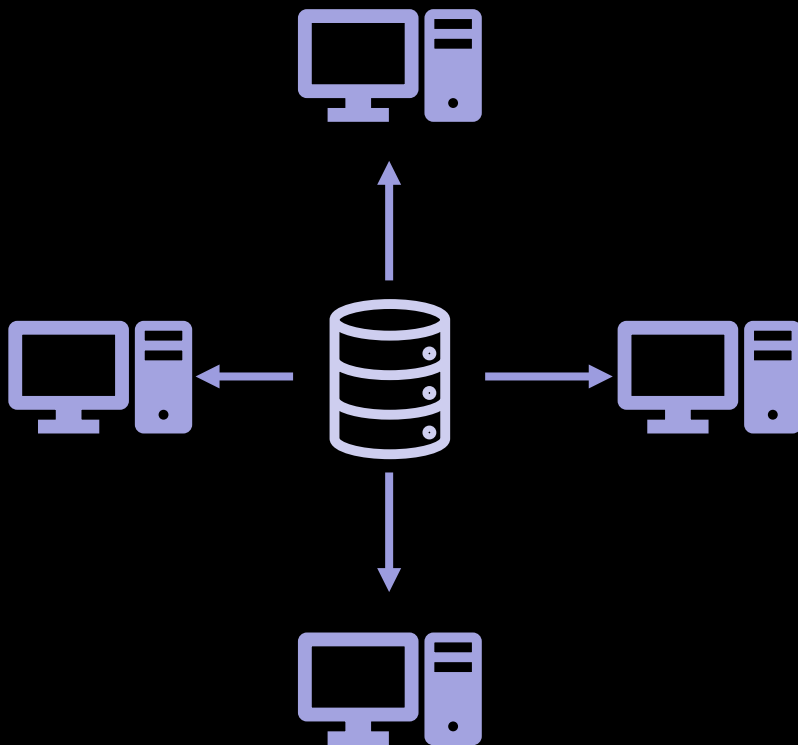


Requisiti di Progetto:

✓ **Grandi file (TB)**

## HDFS – Requisiti di progetto

Il file system di Hadoop è specificatamente progettato per memorizzare grandi dataset in commodity hardware (no servers, mainframes etc.)



Requisiti di Progetto:

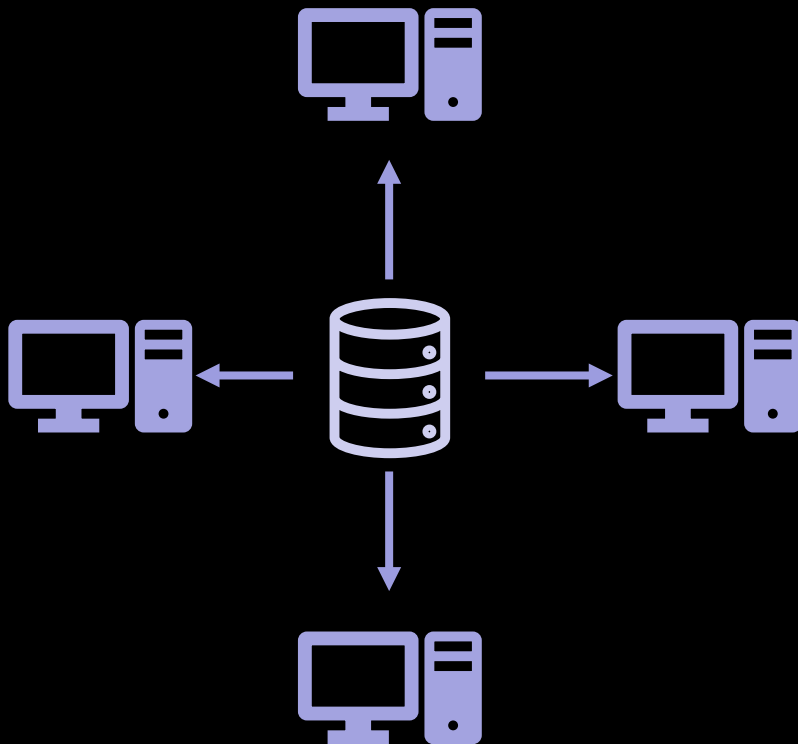
✓ **Grandi file (TB)**





## HDFS – Requisiti di progetto

Il file system di Hadoop è specificatamente progettato per memorizzare grandi dataset in commodity hardware (no servers, mainframes etc.)



Requisiti di Progetto:

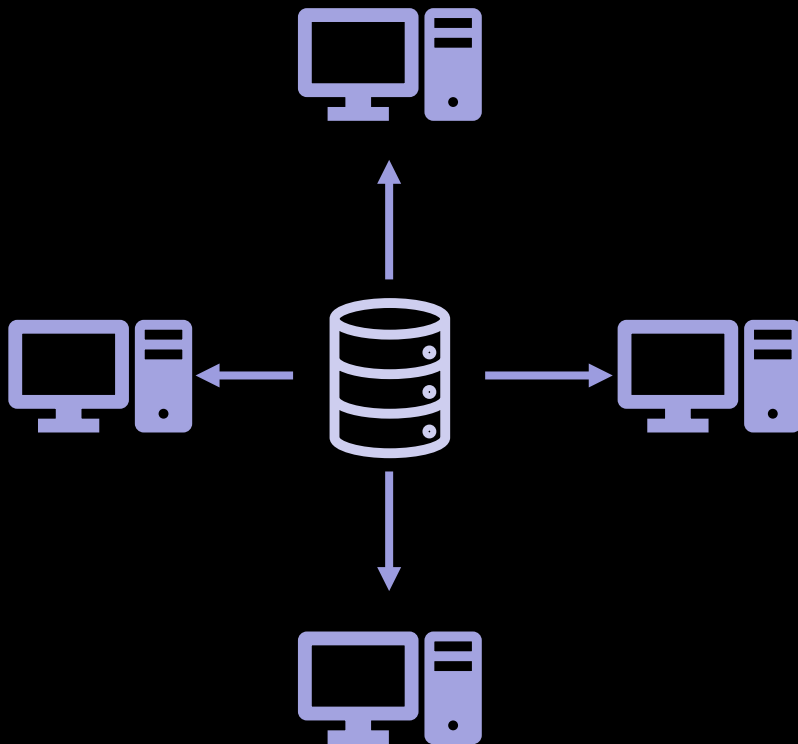
✓ Grandi file (TB)

✓ **Accesso dati in streaming**

**Write-once read-many times pattern**

## HDFS – Requisiti di progetto

Il file system di Hadoop è specificatamente progettato per memorizzare grandi dataset in commodity hardware (no servers, mainframes etc.)

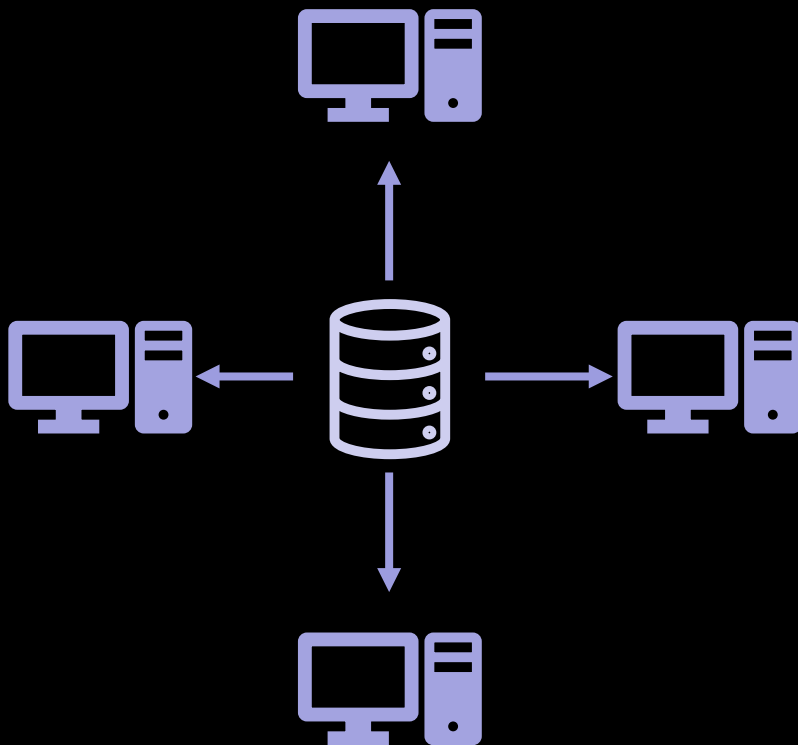


Requisiti di Progetto:

- ✓ Grandi file (TB)
- ✓ Accesso dati in streaming
- ✓ **Commodity hardware**

## HDFS – Requisiti di progetto

Il file system di Hadoop è specificatamente progettato per memorizzare grandi dataset in commodity hardware (no servers, mainframes etc.)



Requisiti di Progetto:

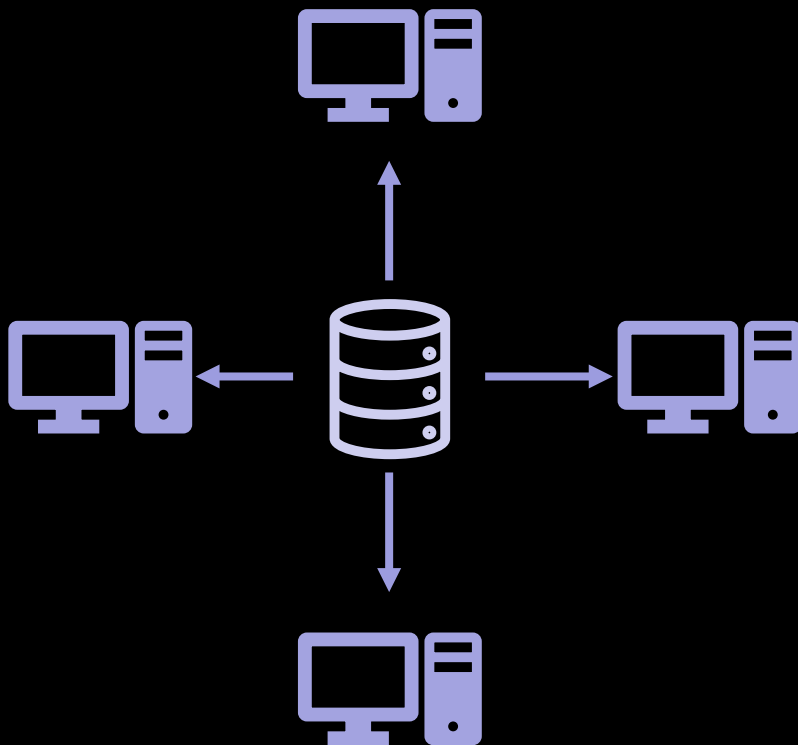
- ✓ Grandi file (TB)
- ✓ Accesso dati in streaming
- ✓ Commodity hardware

Limiti: non funziona bene con

✓ **Molti file piccoli**

## HDFS – Requisiti di progetto

Il file system di Hadoop è specificatamente progettato per memorizzare grandi dataset in commodity hardware (no servers, mainframes etc.)



### Requisiti di Progetto:

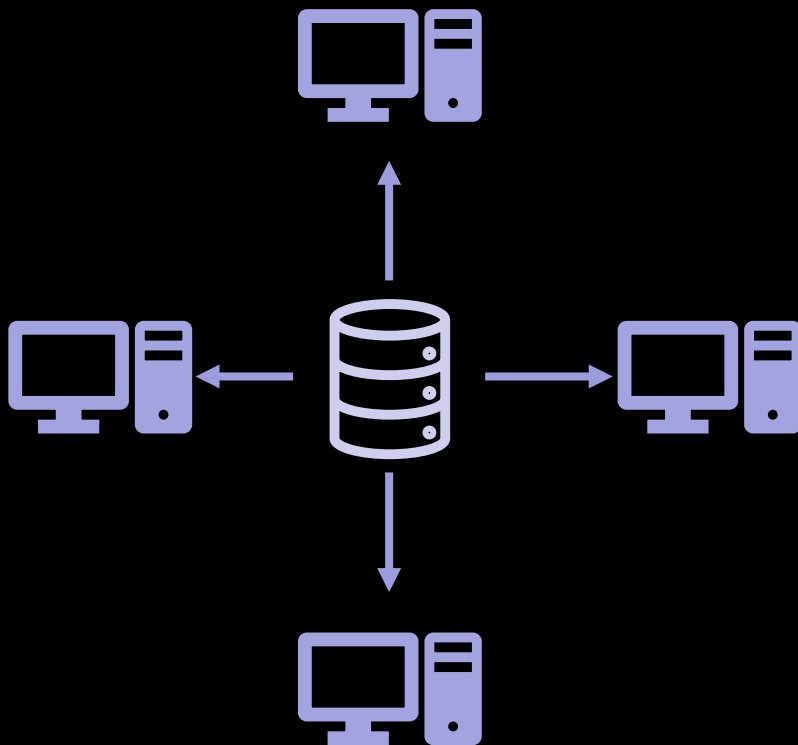
- ✓ Grandi file (TB)
- ✓ Accesso dati in streaming
- ✓ Commodity hardware

Limiti: non funziona bene con

- ✓ Molti file piccoli
- ✓ **Queries a bassa latenza**

## HDFS – Requisiti di progetto

Il file system di Hadoop è specificatamente progettato per memorizzare grandi dataset in commodity hardware (no servers, mainframes etc.)



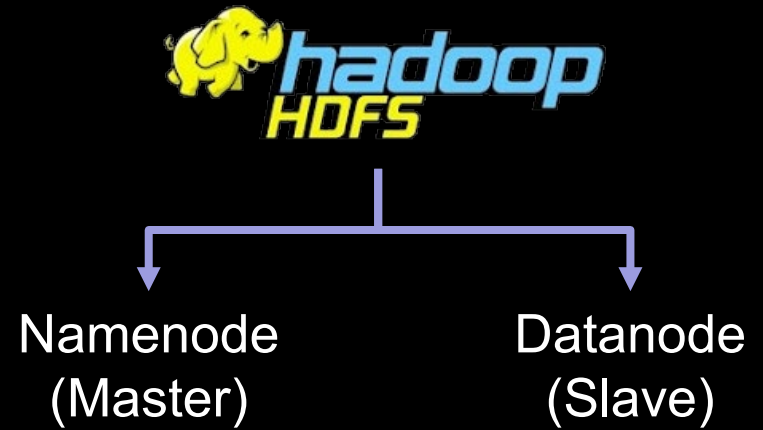
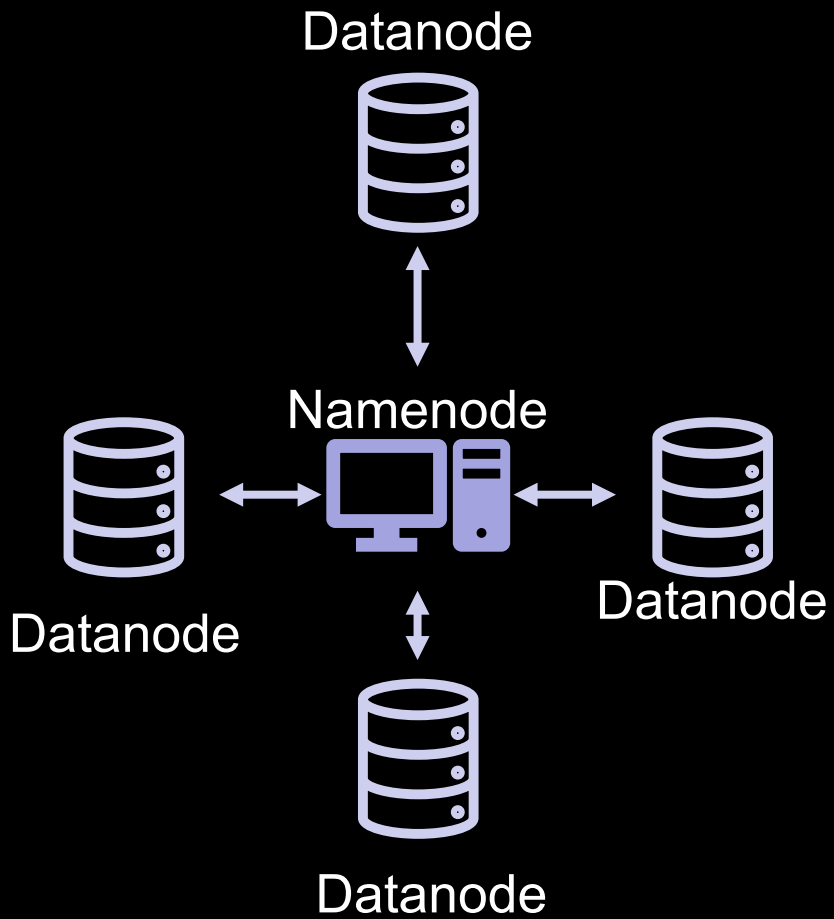
### Requisiti di Progetto:

- ✓ Grandi file (TB)
- ✓ Accesso dati in streaming
- ✓ Commodity hardware

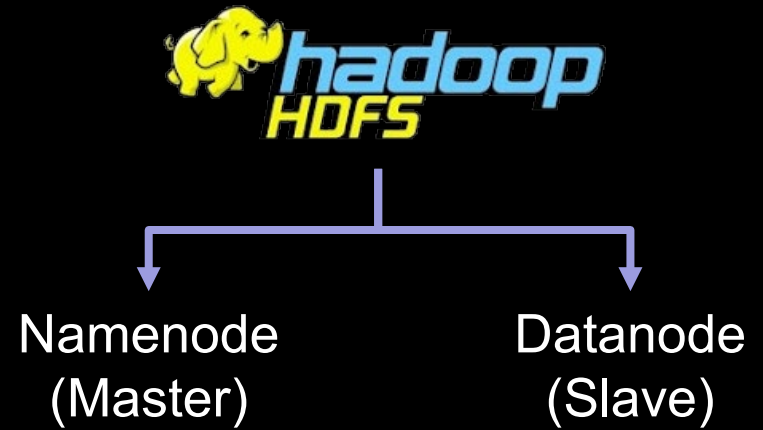
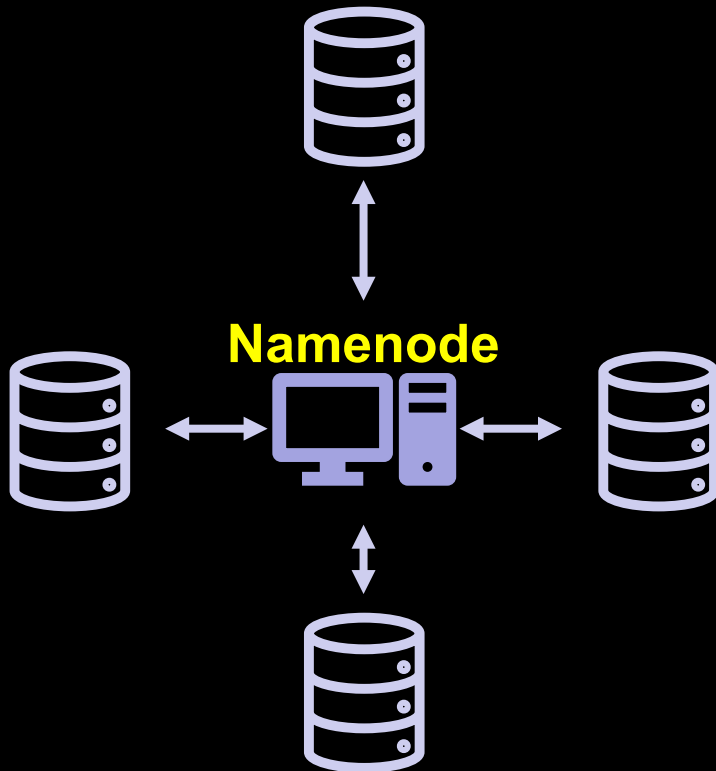
### Limiti: non funziona bene con

- ✓ Molti file piccoli
- ✓ Queries a bassa latenza

## HDFS – Namenode e datanode

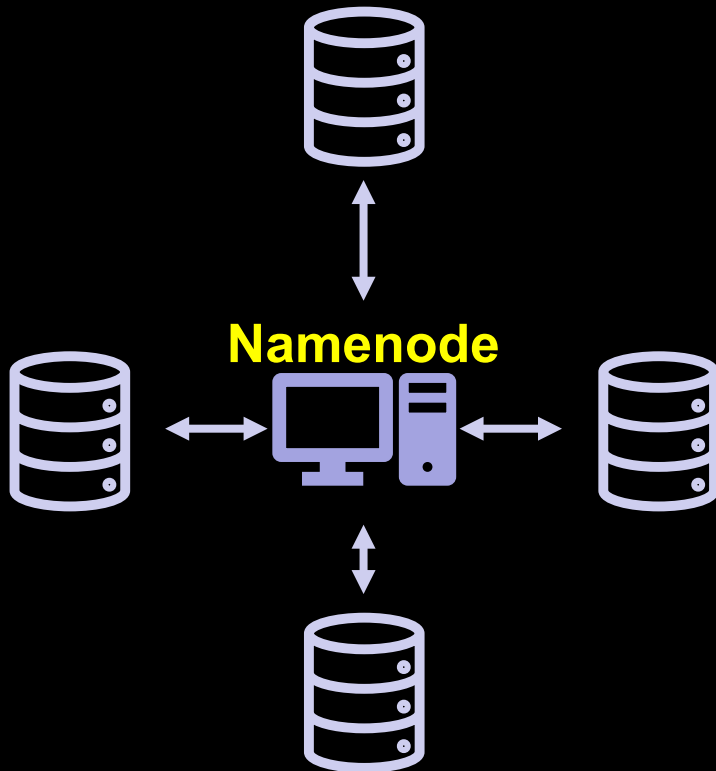


## HDFS – Namenode



✓ È un master daemon

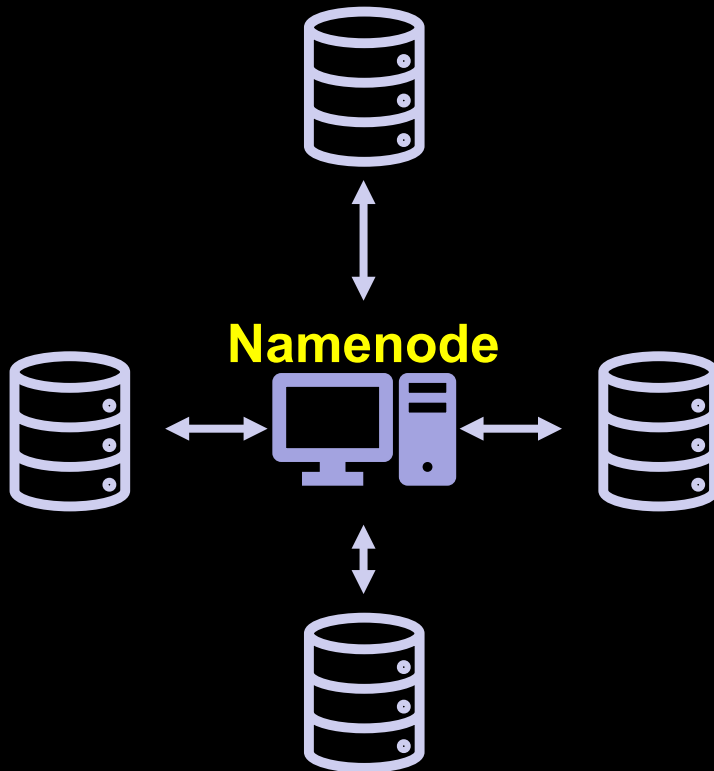
## HDFS – Namenode



- ✓ È un master daemon
- ✓ **Solo 1 namenode può essere attivo**



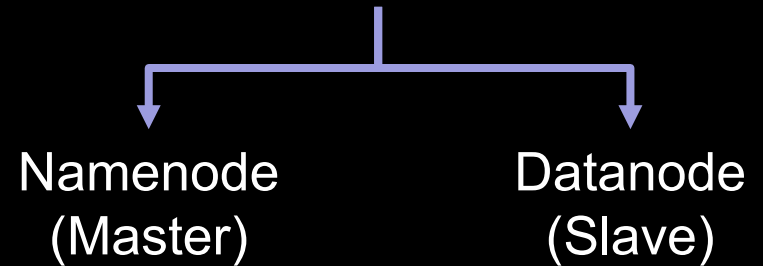
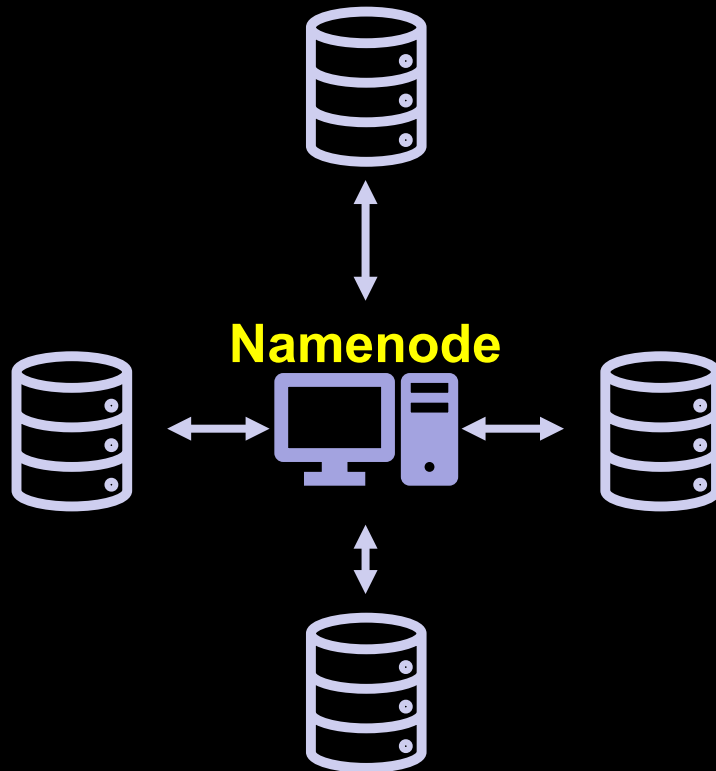
## HDFS – Namenode



- ✓ È un master daemon
- ✓ **Solo 1 namenode può essere attivo**

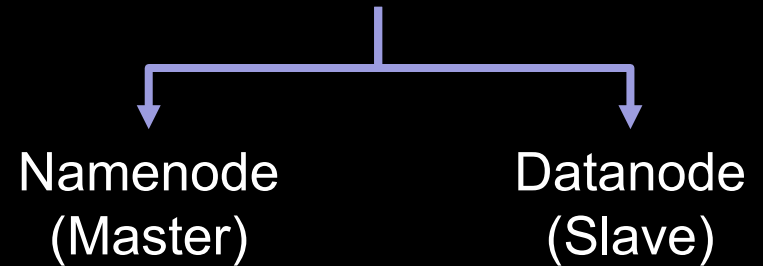
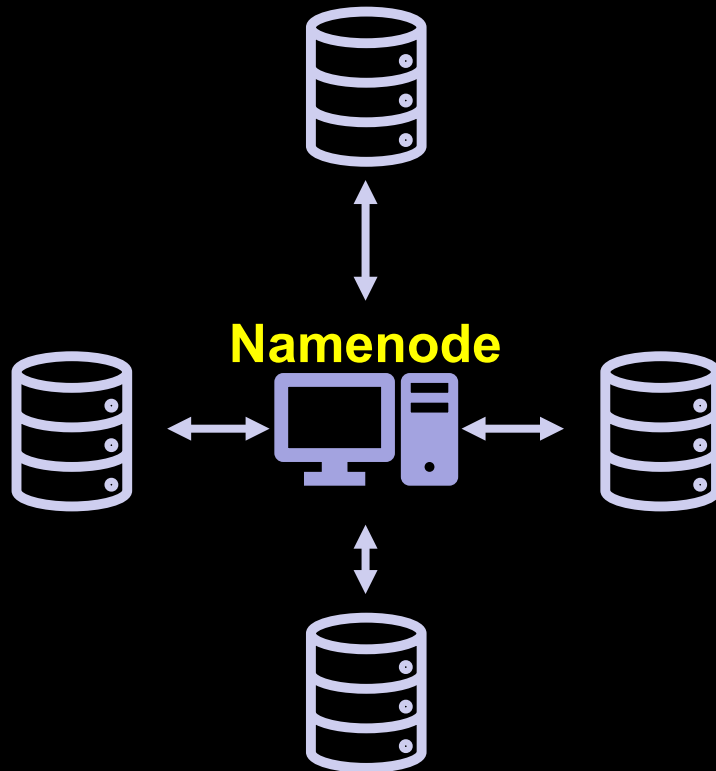
**Namenode failure !!!**

## HDFS – Namenode



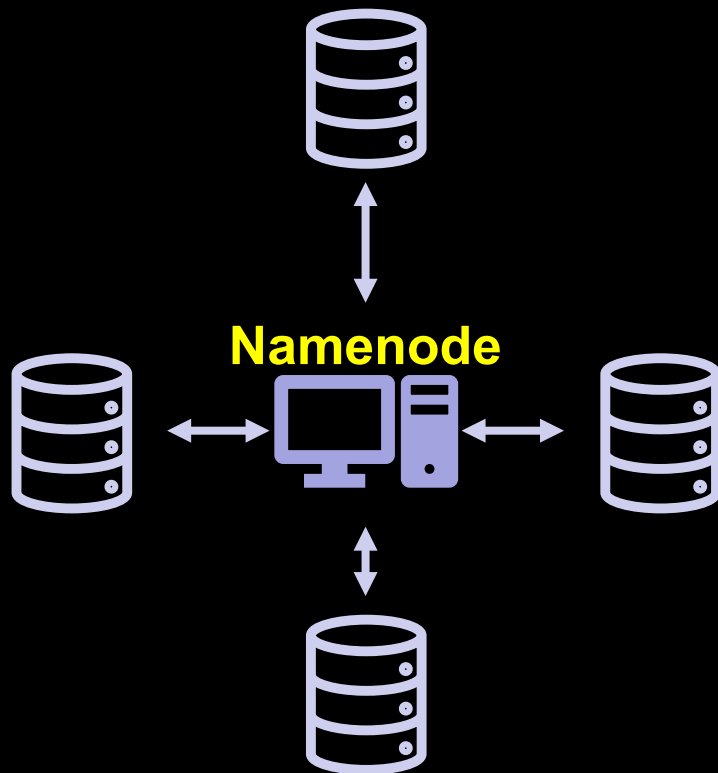
- ✓ È un master daemon
- ✓ Solo 1 namenode può essere attivo
- ✓ **Gestisce tutti i data nodes**

## HDFS – Namenode



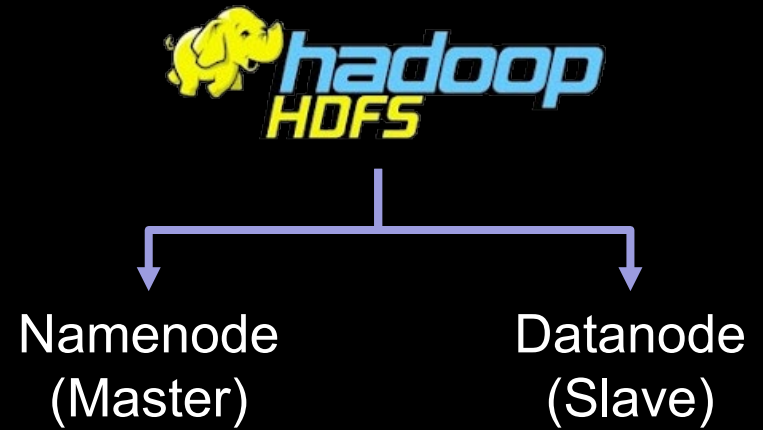
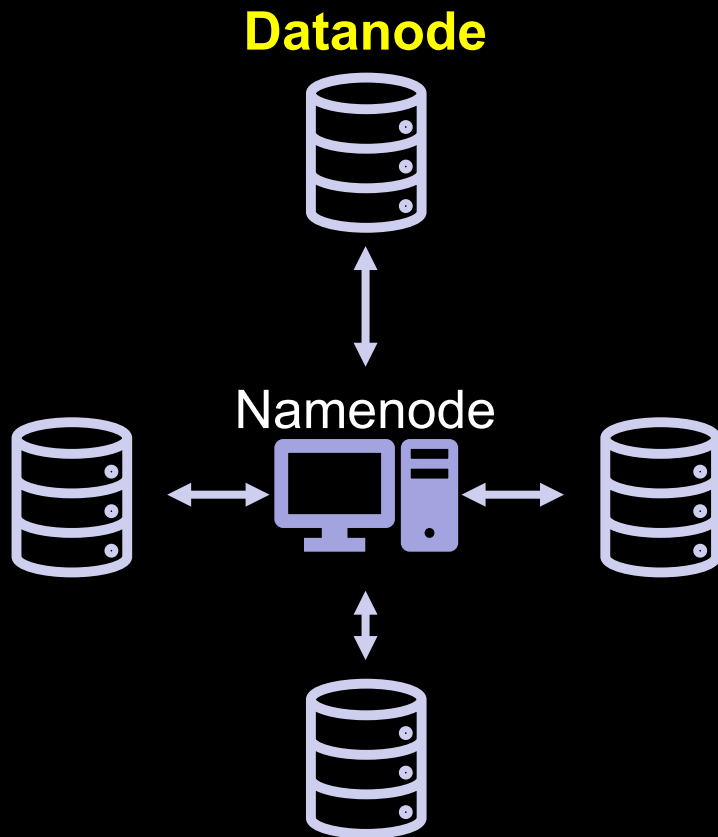
- ✓ È un master daemon
- ✓ Solo 1 namenode può essere attivo
- ✓ Gestisce tutti i data nodes
- ✓ **Contiene tutti i metadata (NO DATA)**

## HDFS – Namenode



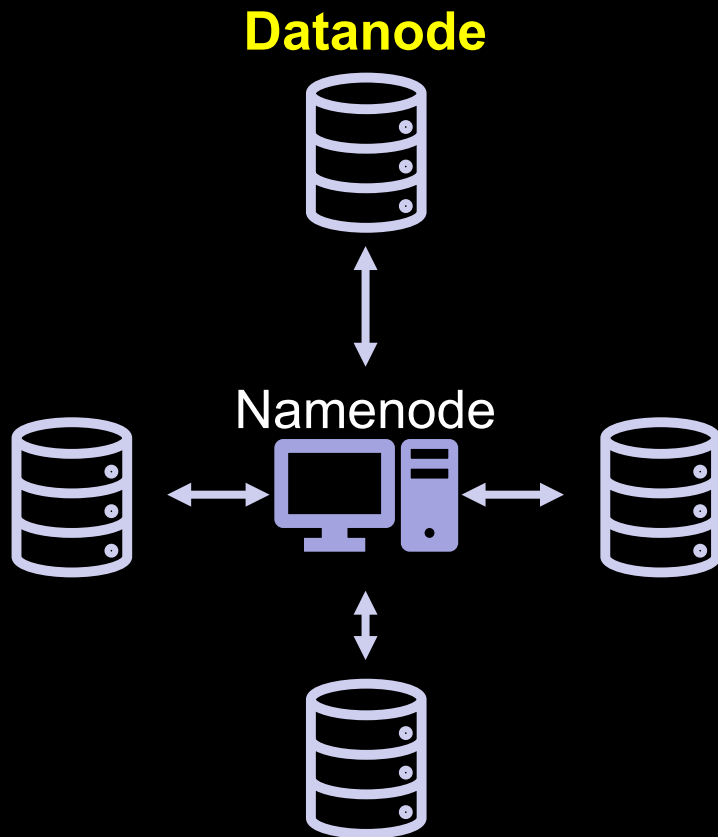
- ✓ È un master daemon
- ✓ Solo 1 namenode può essere attivo
- ✓ Gestisce tutti i data nodes
- ✓ Contiene tutti i metadata (NO DATA)

## HDFS – Datanode



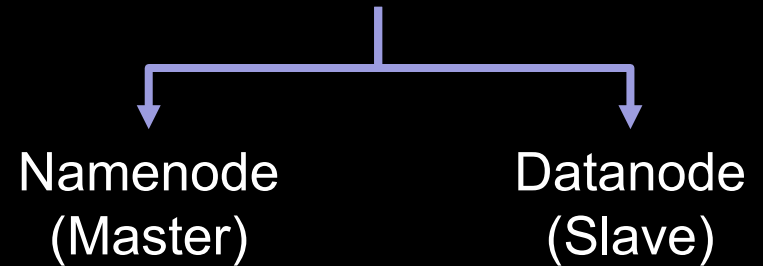
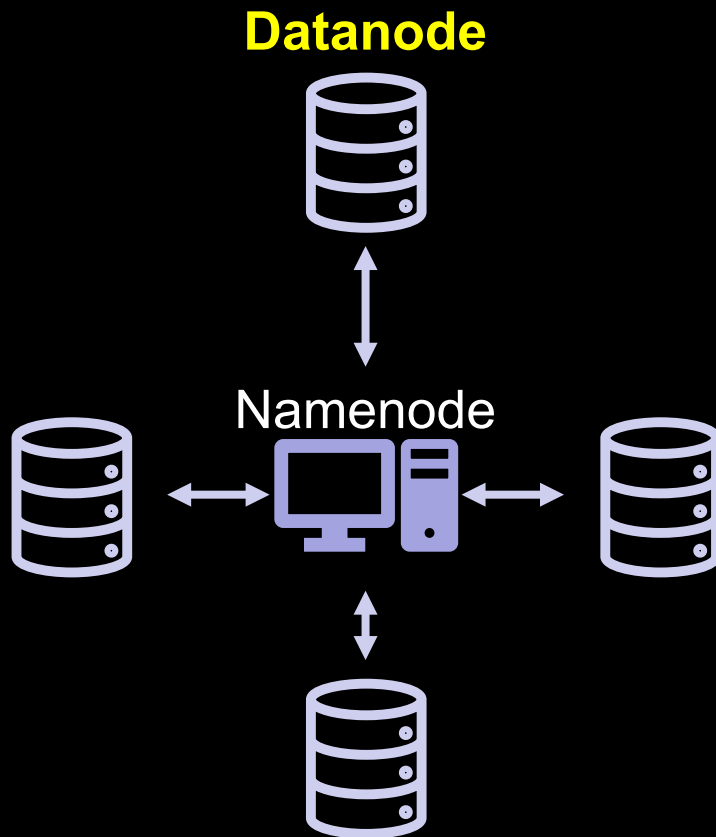
✓ È un slave daemon

## HDFS – Datanode



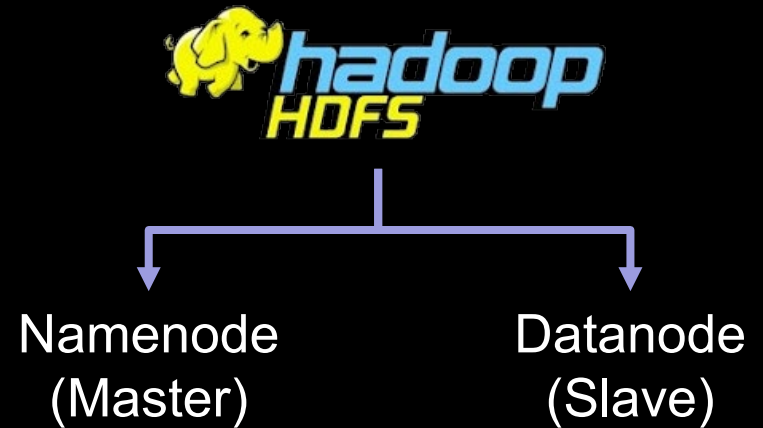
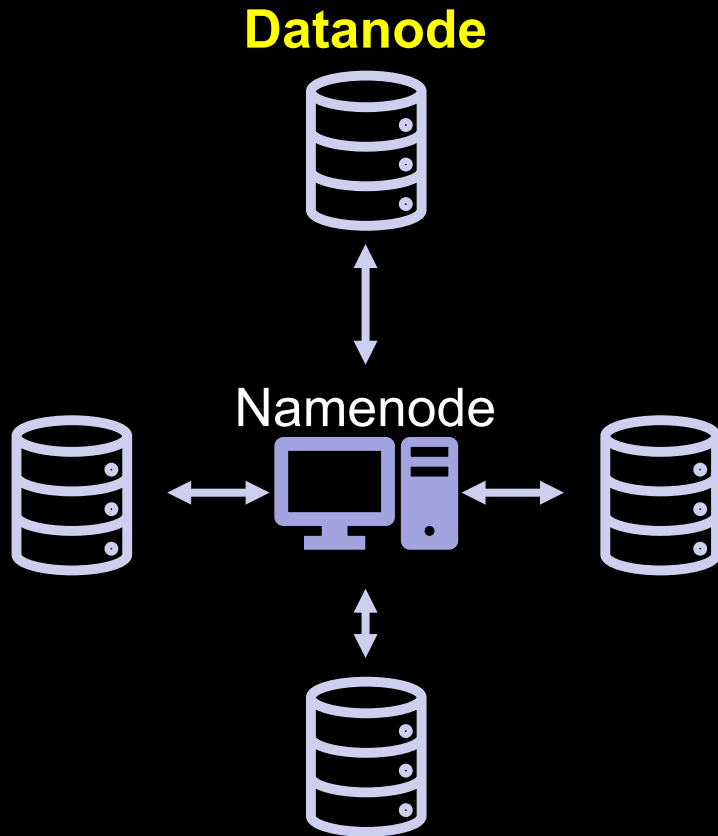
- ✓ È un slave daemon
- ✓ **Ci sono molti datanode**

## HDFS – Datanode



- ✓ È un slave daemon
- ✓ Ci sono molti datanode
- ✓ **Contiene i dati reali**

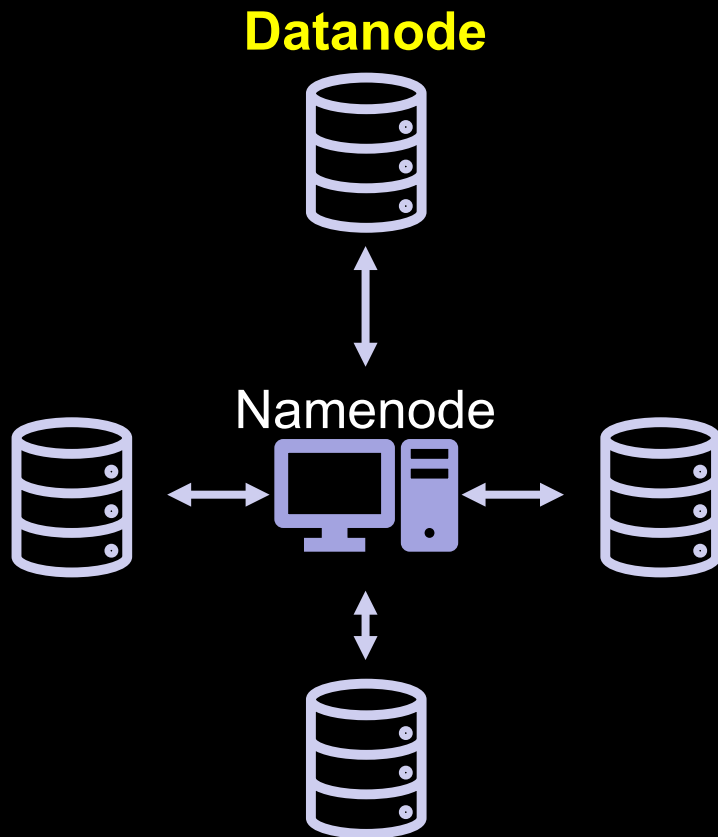
## HDFS – Datanode



- ✓ È un slave daemon
- ✓ Ci sono molti datanode
- ✓ Contiene i dati reali
- ✓ **Fault tolerance**



## HDFS – Datanode



- ✓ È un slave daemon
- ✓ Ci sono molti datanode
- ✓ Contiene i dati reali
- ✓ Fault tolerance

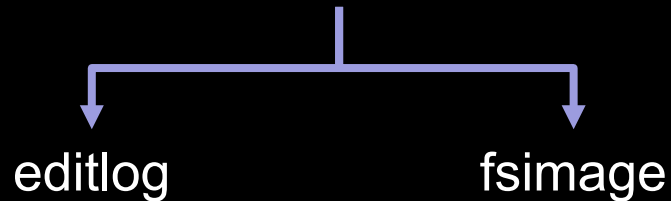
## **HDFS – Metadata**

I metadata sono allocati nel namenode e ci danno informazioni su locazione, dimensione e altre informazioni sui files contenuti nei datanodes

## HDFS – Metadata

I metadata sono allocati nel namenode e ci danno informazioni su locazione, dimensione e altre informazioni sui files contenuti nei datanodes

Tutti i metadata sono gestiti  
tramite 2 files



## HDFS – Metadata

I metadata sono allocati nel namenode e ci danno informazioni su locazione, dimensione e altre informazioni sui files contenuti nei datanodes

Tutti i metadata sono gestiti  
tramite 2 files

editlog

fsimage

Tiene traccia delle modifiche  
**in corso** fatte al HDFS

Tiene traccia delle di **tutte** le  
modifiche fatte al HDFS

## **HDFS – Fault tolerance**

*Progettate un sistema come se niente possa  
funzionare a dovere!!!*

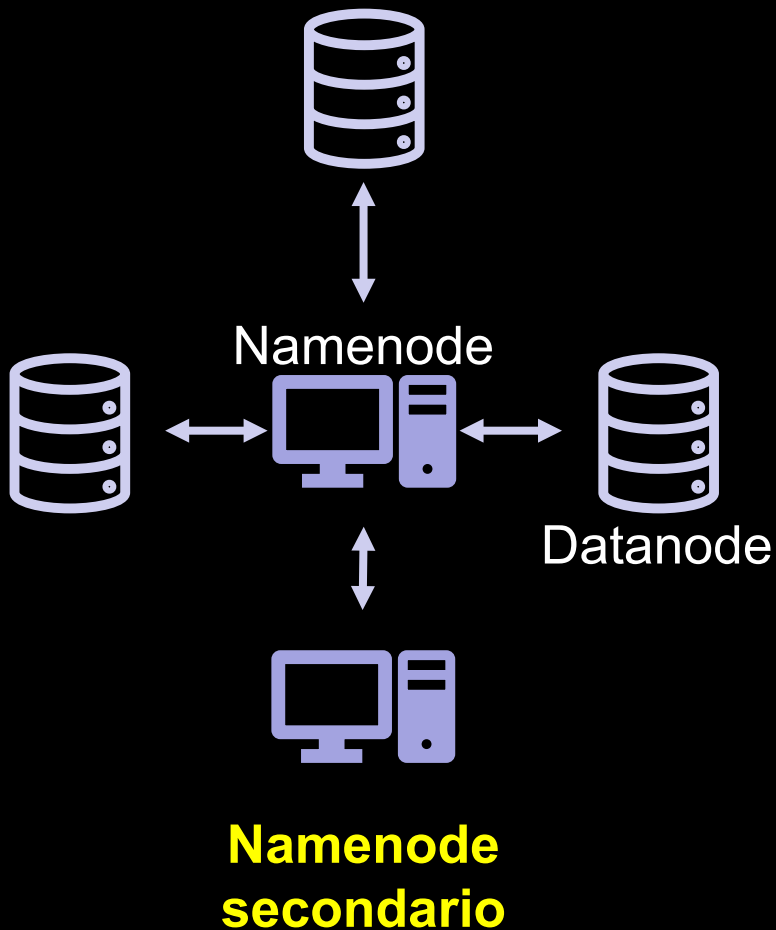
## **HDFS – Fault tolerance**

Casi di guasto:

1. Crash del namenode (hardware failure, mancanza di corrente etc. )
2. Esaurita la capacità di memoria del namenode

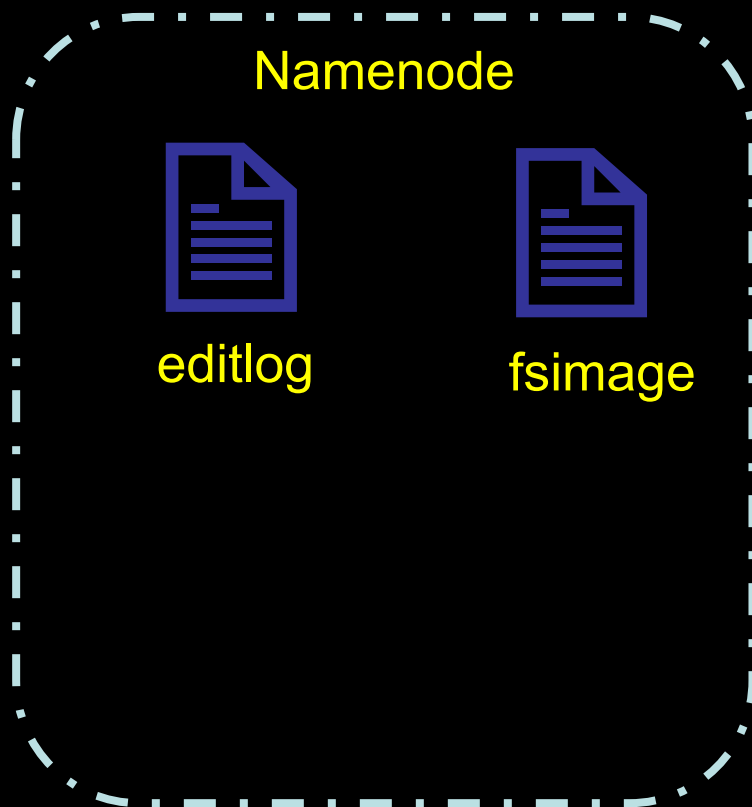
## HDFS – Fault tolerance

Un namenode secondario tiene traccia dei files di editlog e fsimage



## HDFS – Fault tolerance

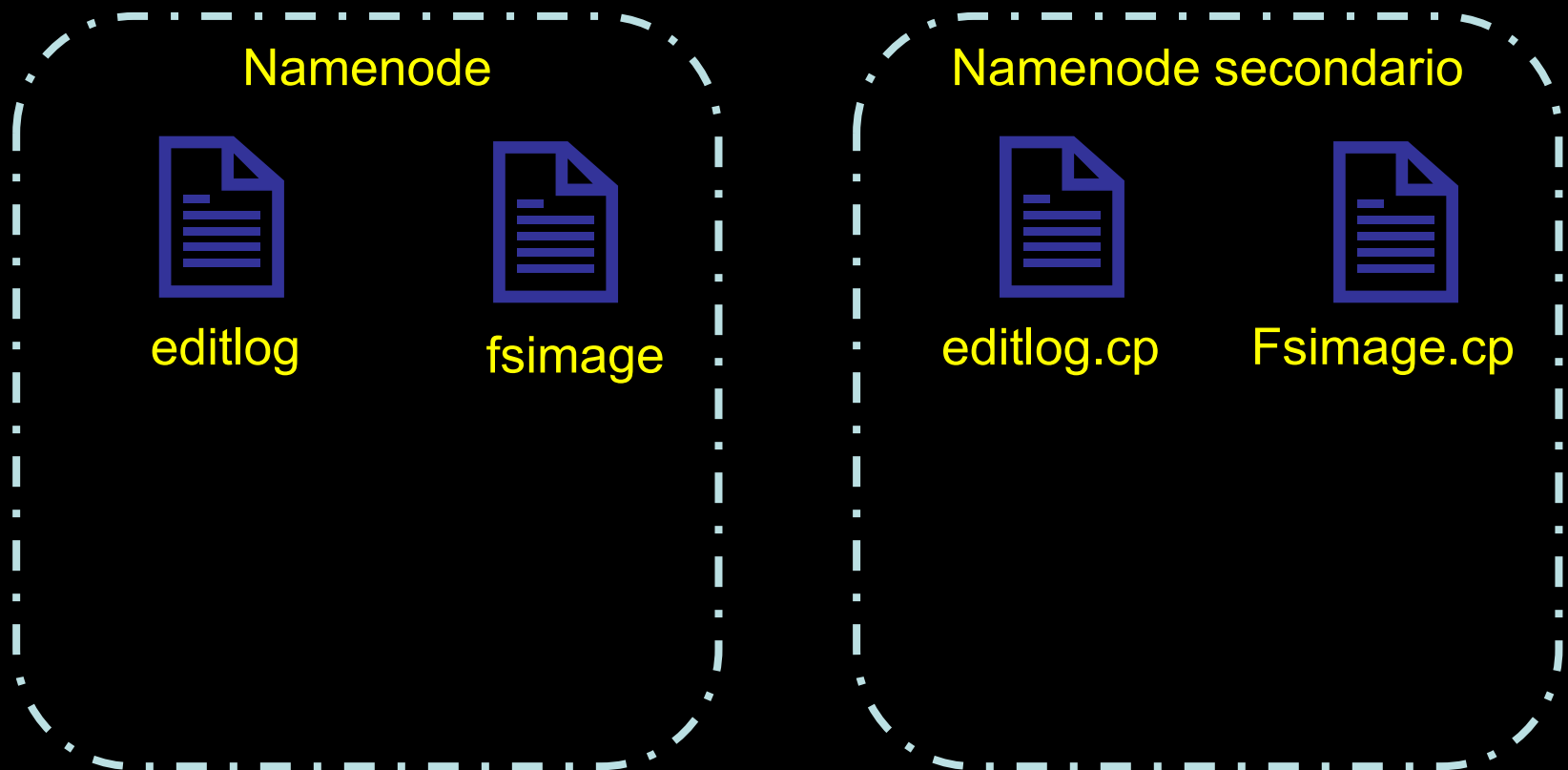
Un namenode secondario tiene traccia dei files di editlog e fsimage





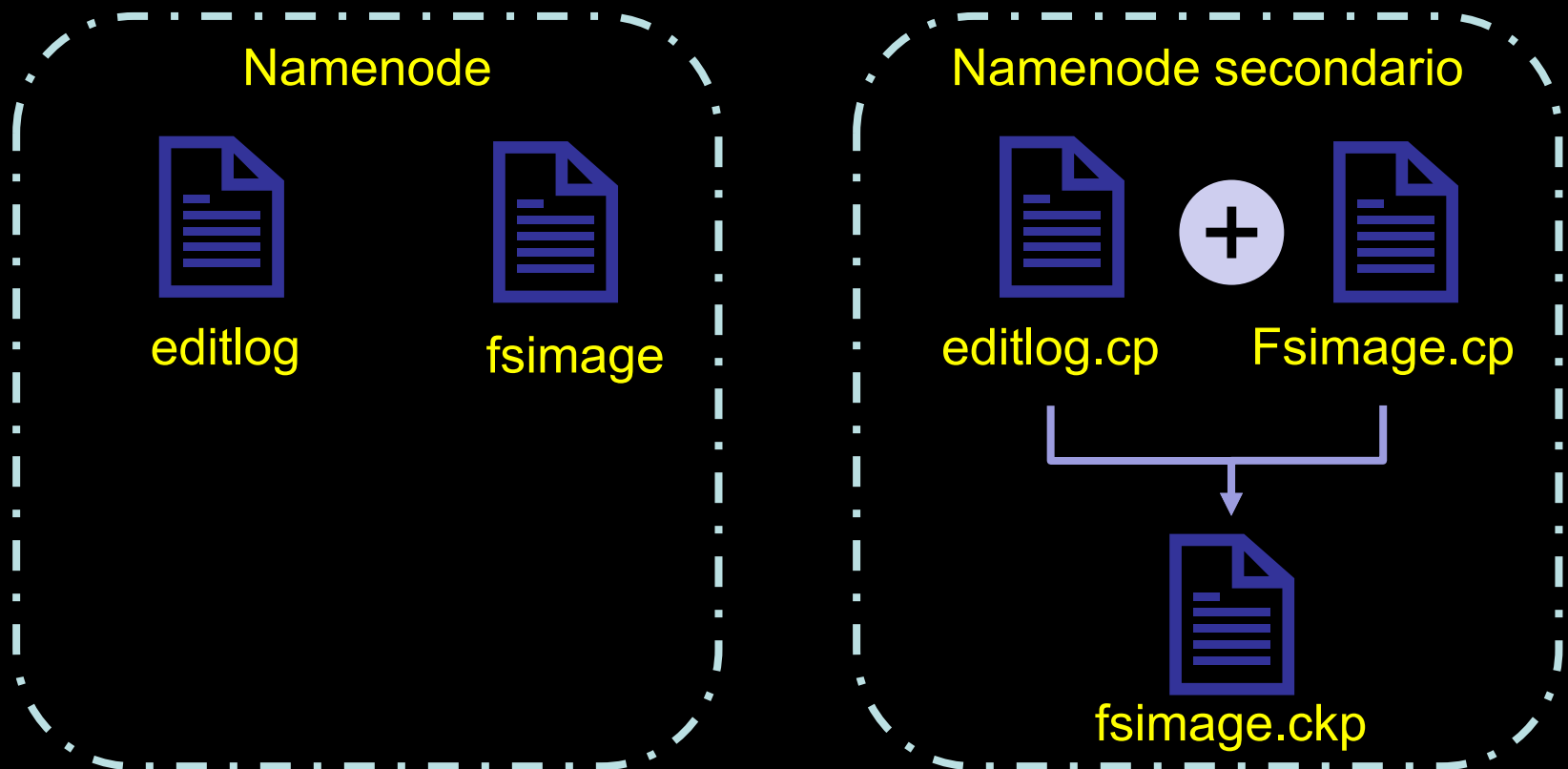
## HDFS – Fault tolerance

Un namenode secondario tiene traccia dei files di editlog e fsimage



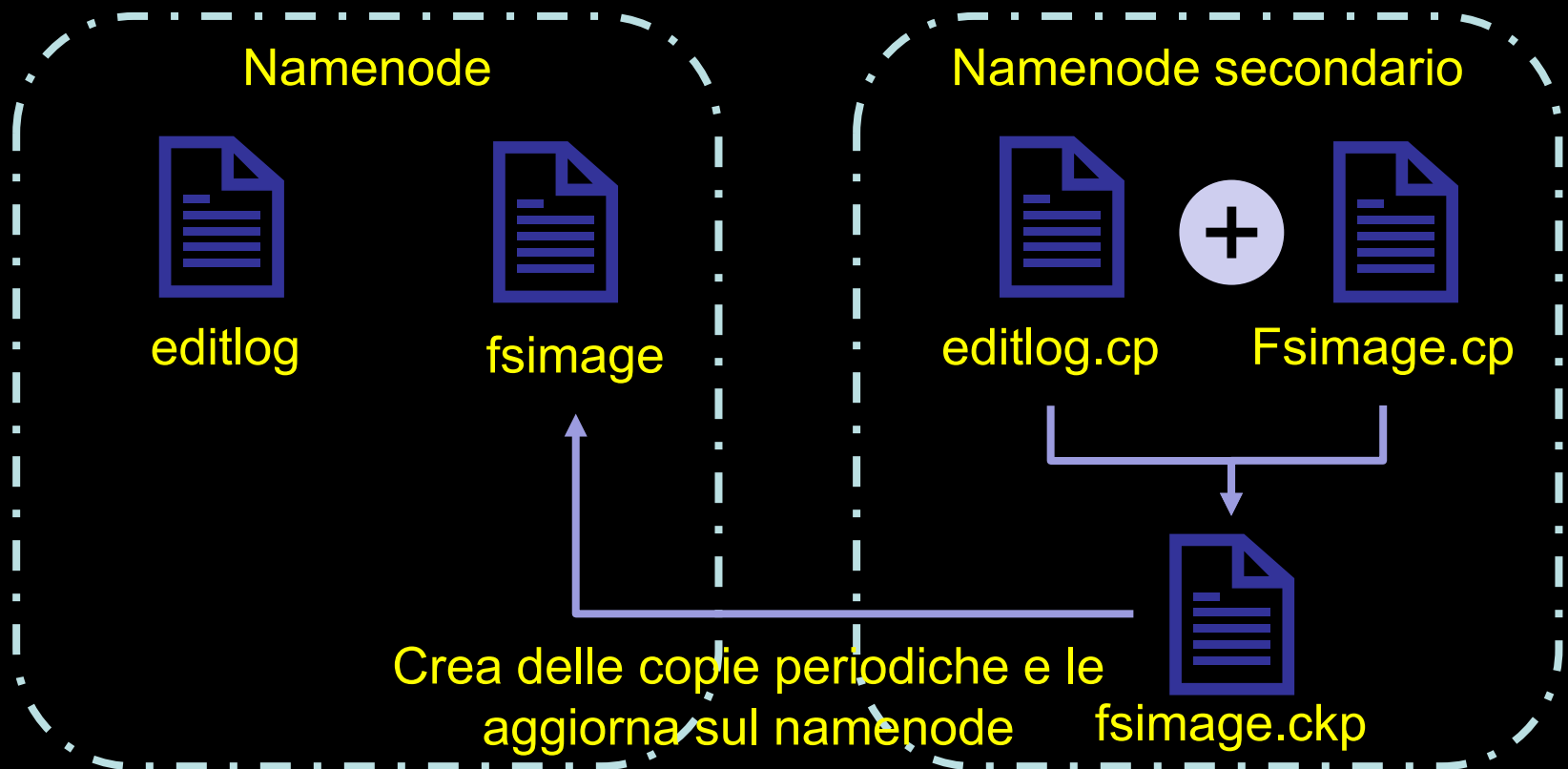
## HDFS – Fault tolerance

Un namenode secondario tiene traccia dei files di editlog e fsimage



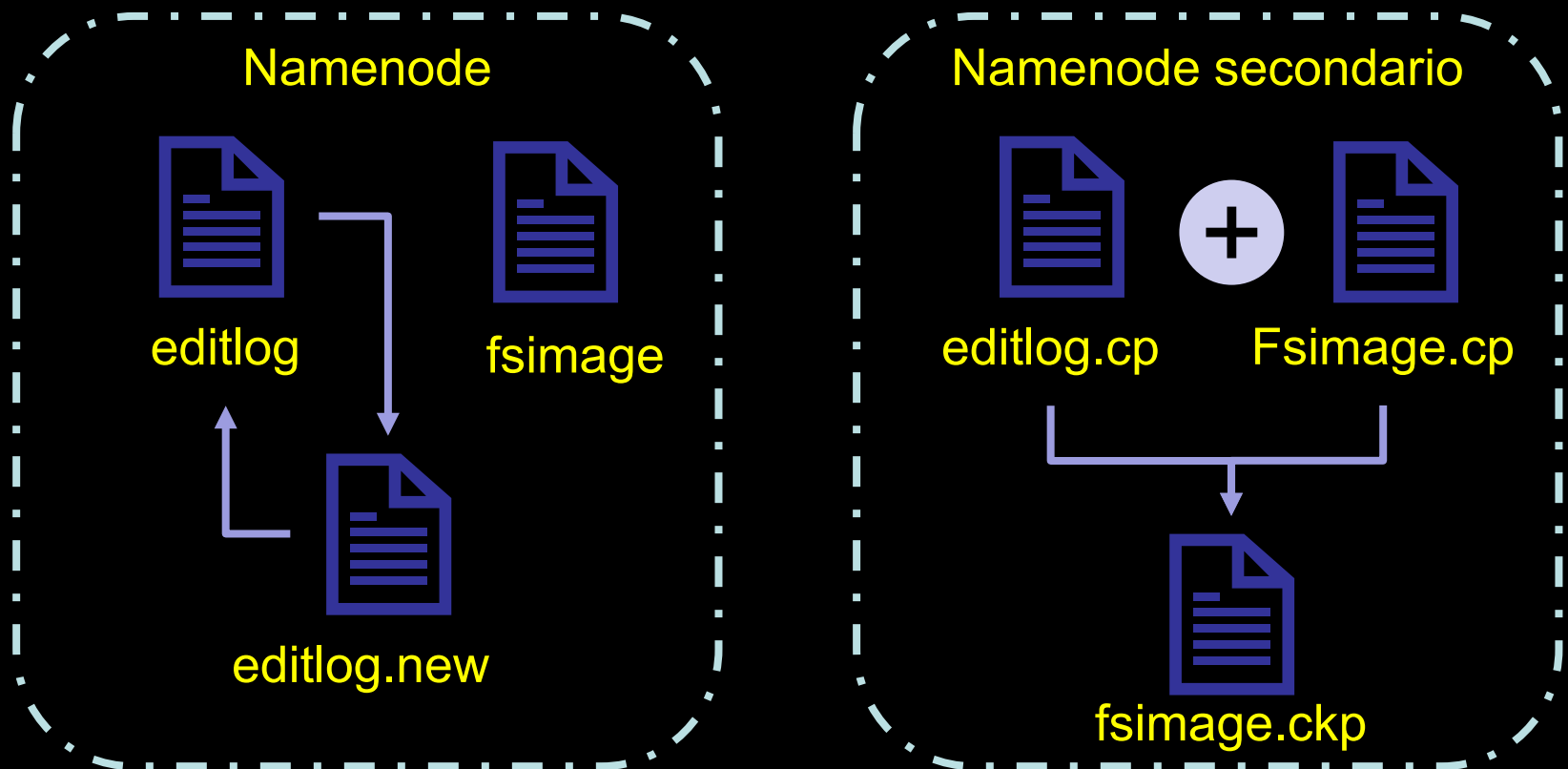
## HDFS – Fault tolerance

Un namenode secondario tiene traccia dei files di editlog e fsimage



## HDFS – Fault tolerance

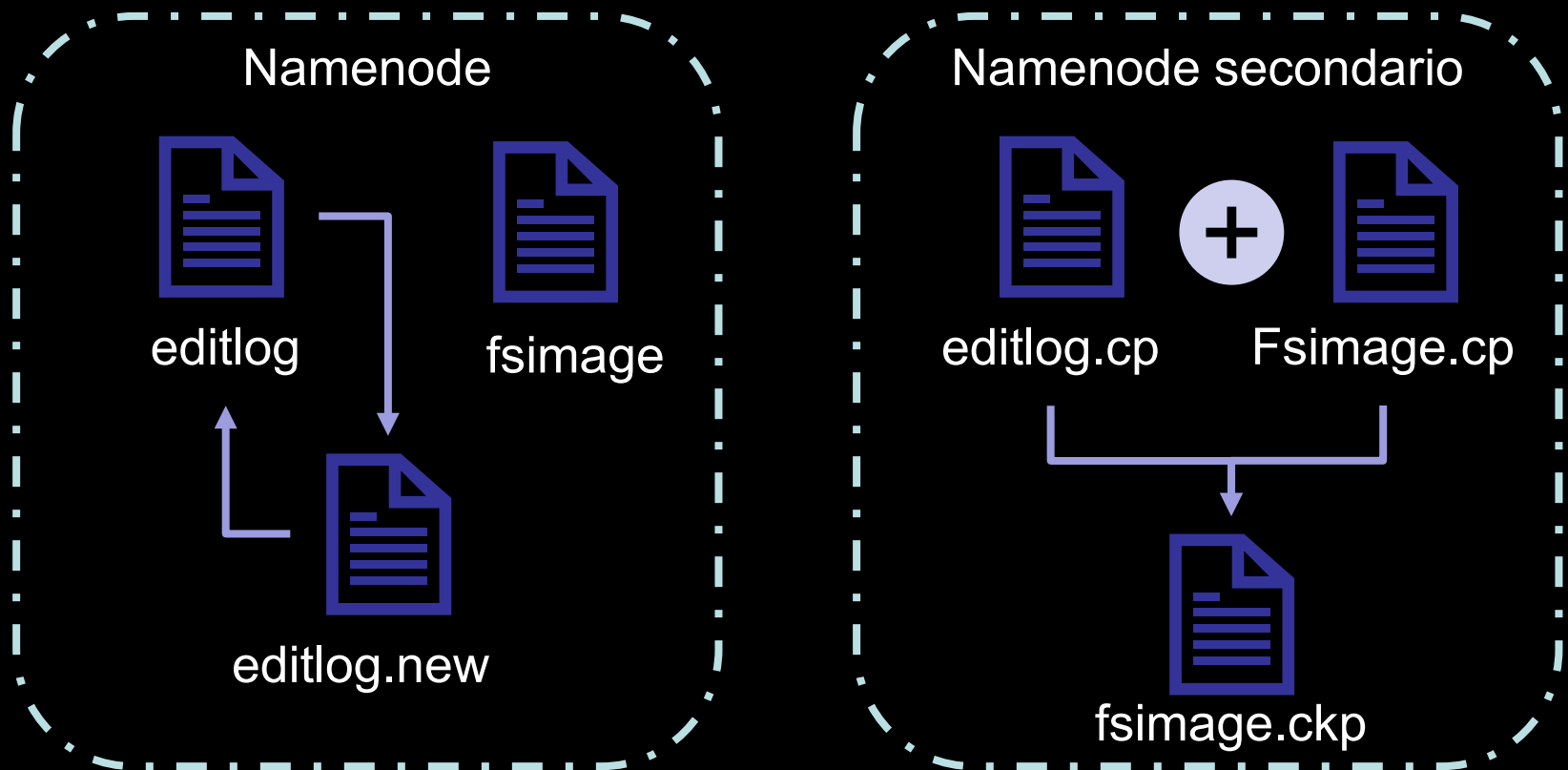
Un namenode secondario tiene traccia dei files di editlog e fsimage



Questo significa che bisogna anche aggiornare il file editlog

## HDFS – Fault tolerance

Un namenode secondario tiene traccia dei files di editlog e fsimage



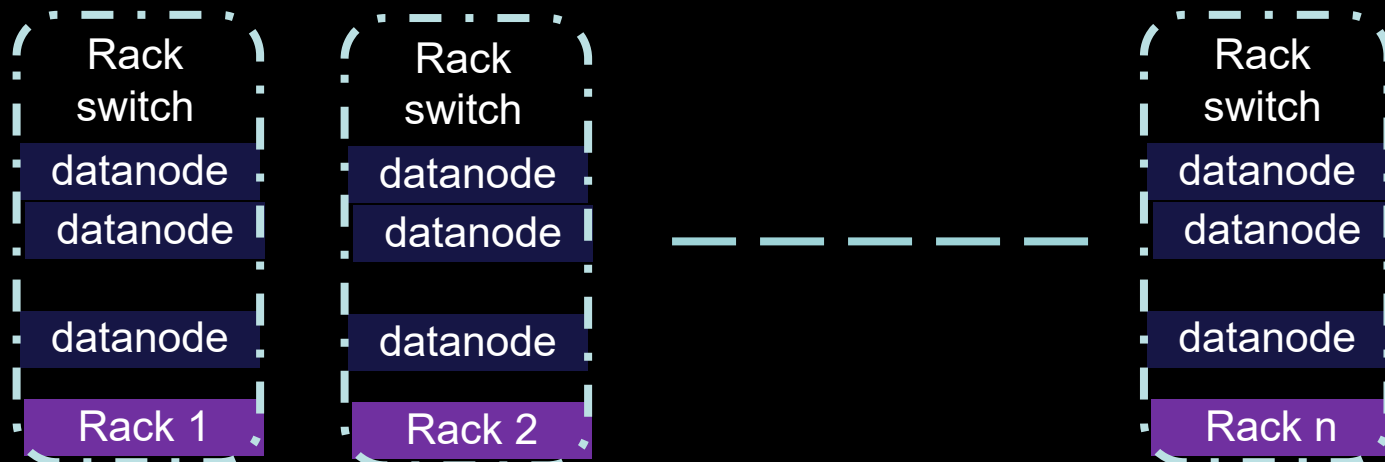
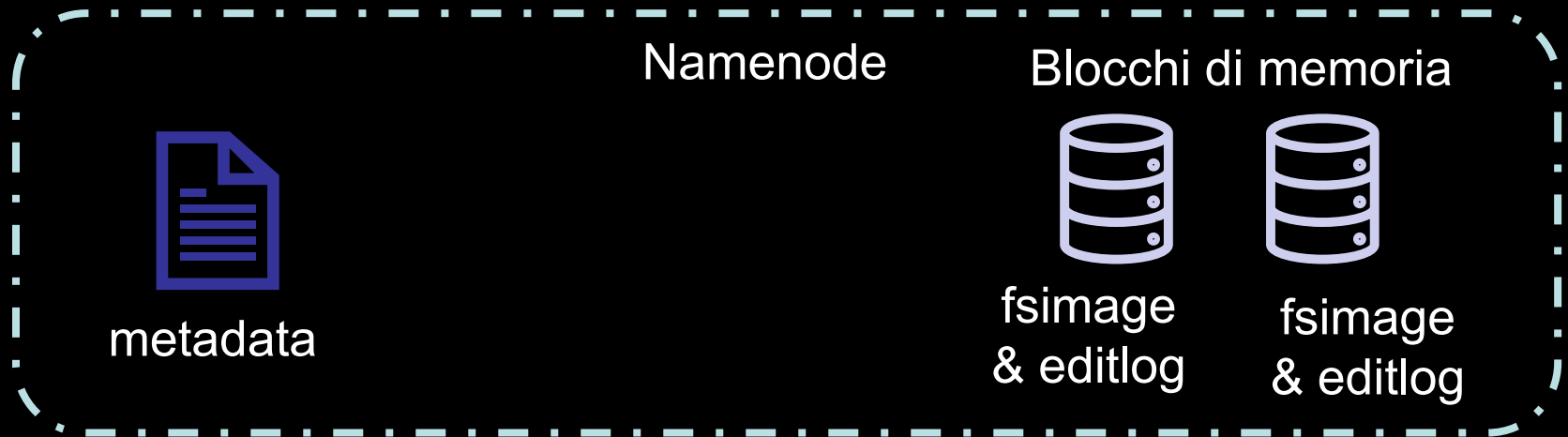
Tutto il processo tipicamente è eseguito ogni ora nel cluster

## **HDFS – Architettura del cluster**

## HDFS – Architettura del cluster

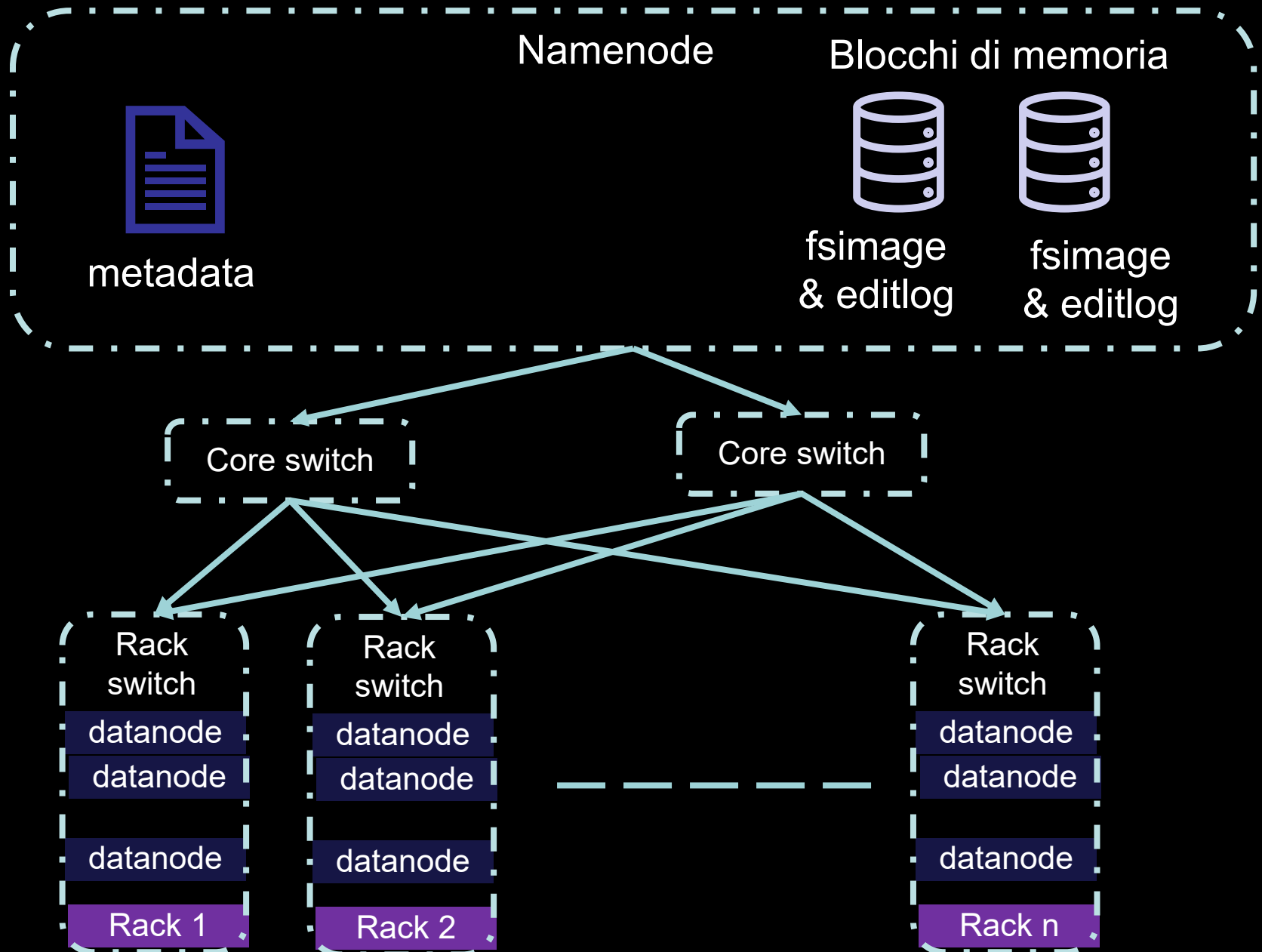


# HDFS – Architettura del cluster





# HDFS – Architettura del cluster

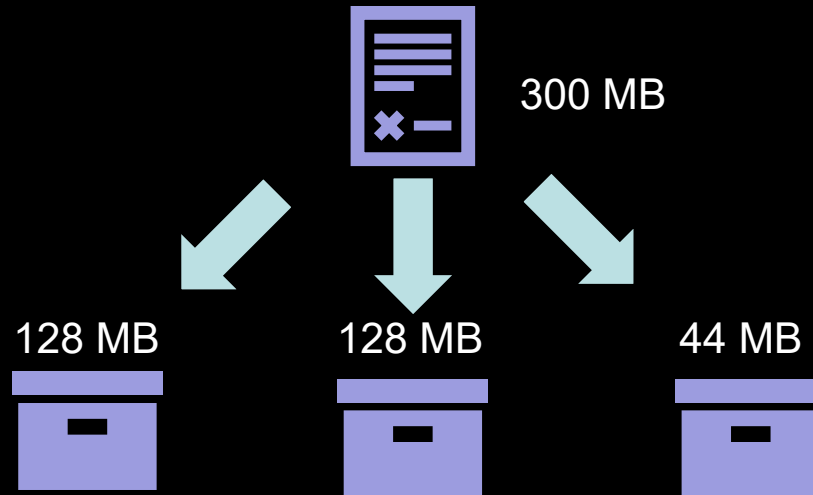


## **HDFS – Blocchi di dati**

Il file system di hadoop divide grandi files in piccoli blocchi (data block)

## HDFS – Blocchi di dati

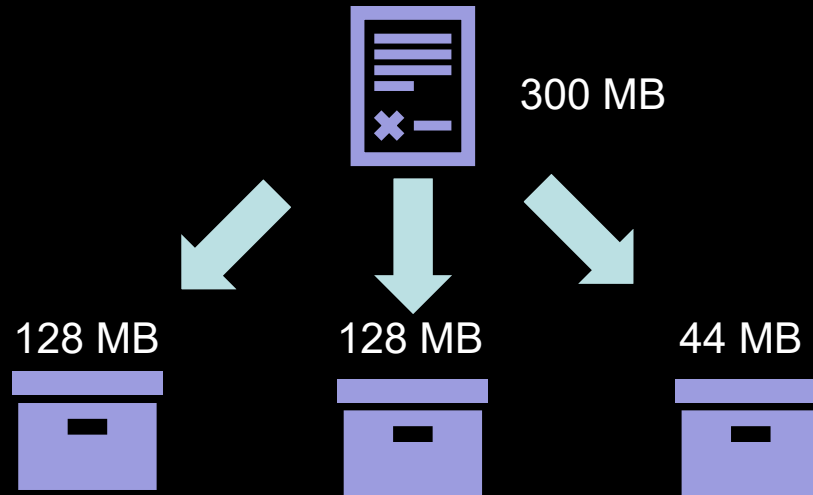
Il file system di hadoop divide grandi files in piccoli blocchi (data block)



By default i dati sono divisi in blocchi di 128 MB

## HDFS – Blocchi di dati

Il file system di hadoop divide grandi files in piccoli blocchi (data block)



By default i dati sono divisi in blocchi di 128 MB

Blocchi più piccoli causano overhead del sistema

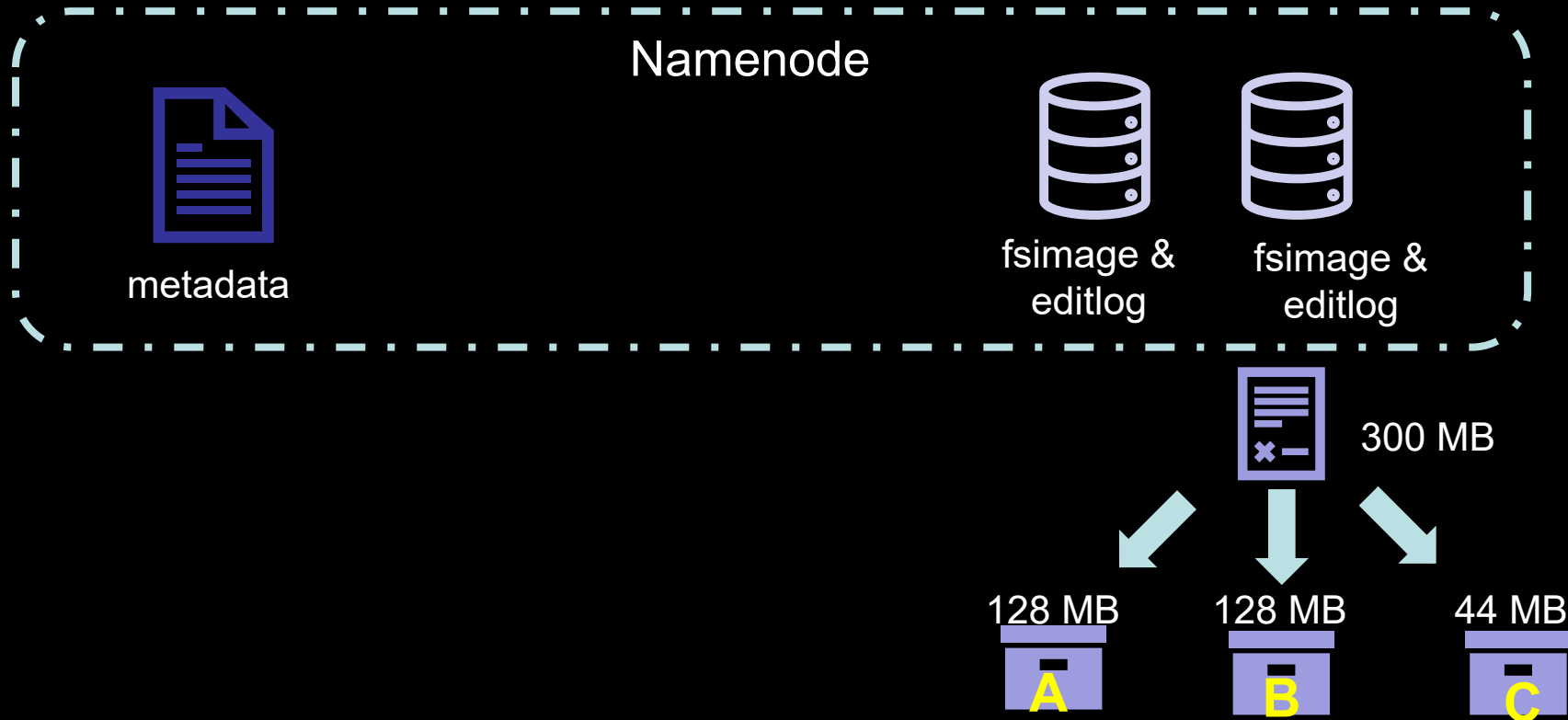
Blocchi più grandi allungherebbero tempi di processamento

## **HDFS – Ridondanza**

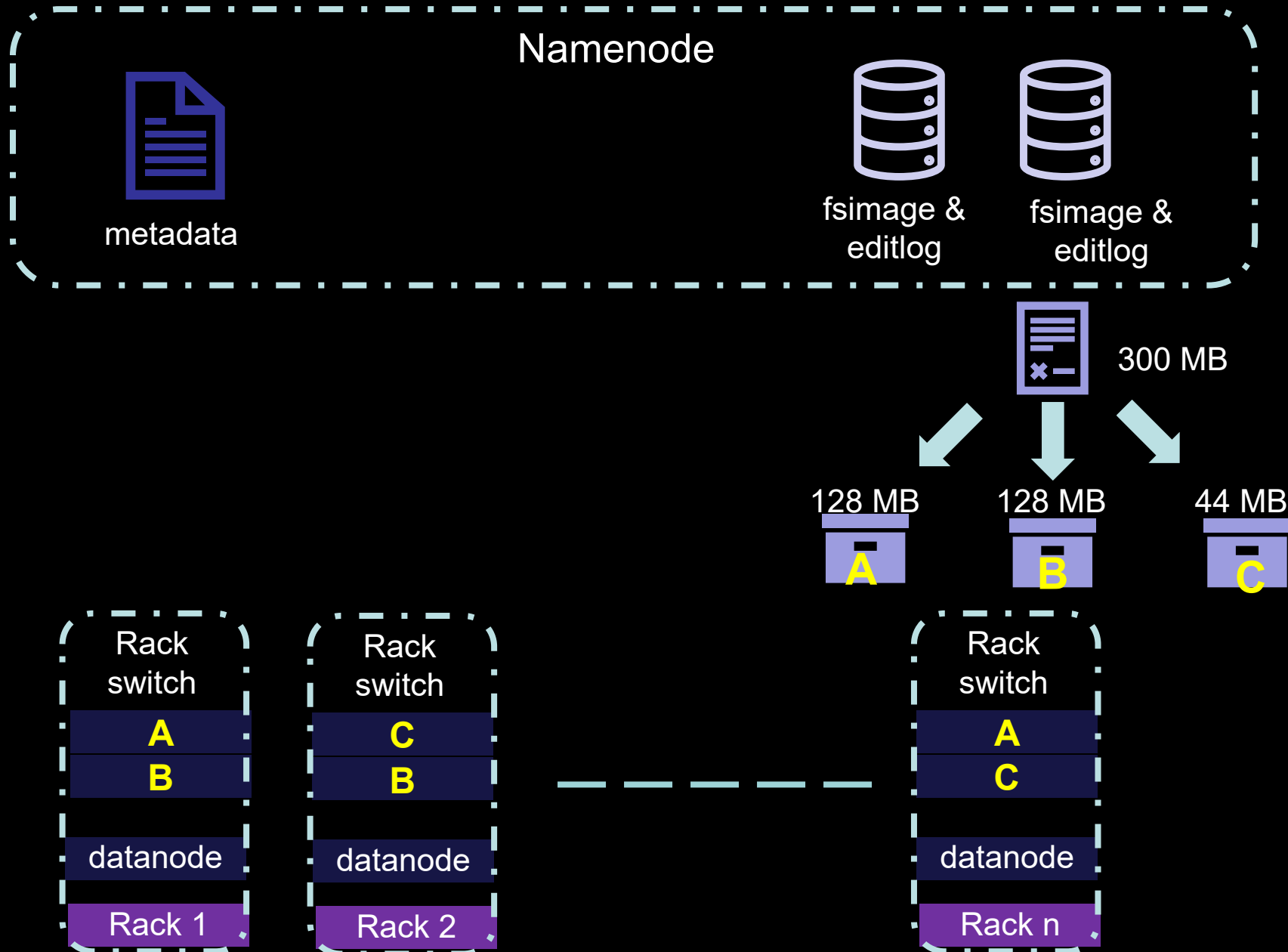
## HDFS – Ridondanza



## HDFS – Ridondanza



# HDFS – Ridondanza

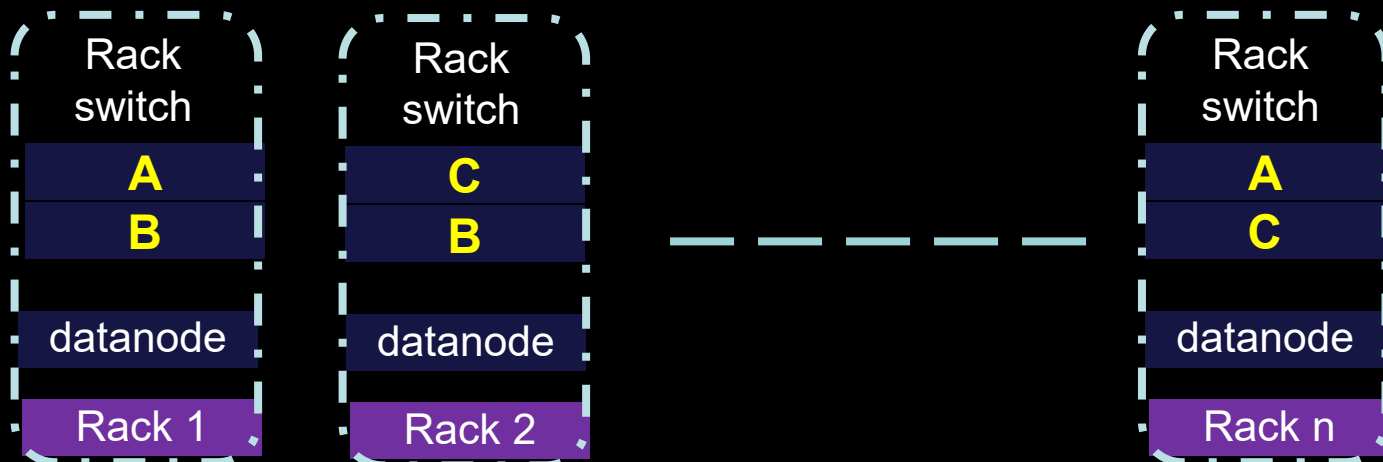
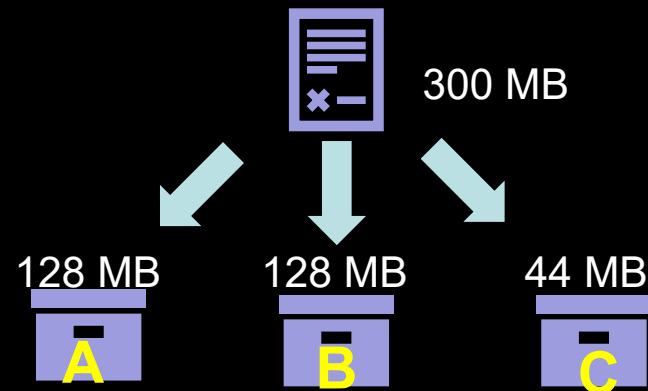




## HDFS – Ridondanza



By default il fattore di replicazione è 3, quindi abbiamo sempre almeno 3 copie dello stesso dato su rack diversi



# HDFS – Architettura

## HDFS – Architettura

Namenode



Metadata (nome, repliche, ...)  
/home/pippo/dati/

Il namenode effettua tutte le operazioni di creazione e aggiornamento dei metadata con informazioni di lettura, scrittura e accesso su ogni blocco

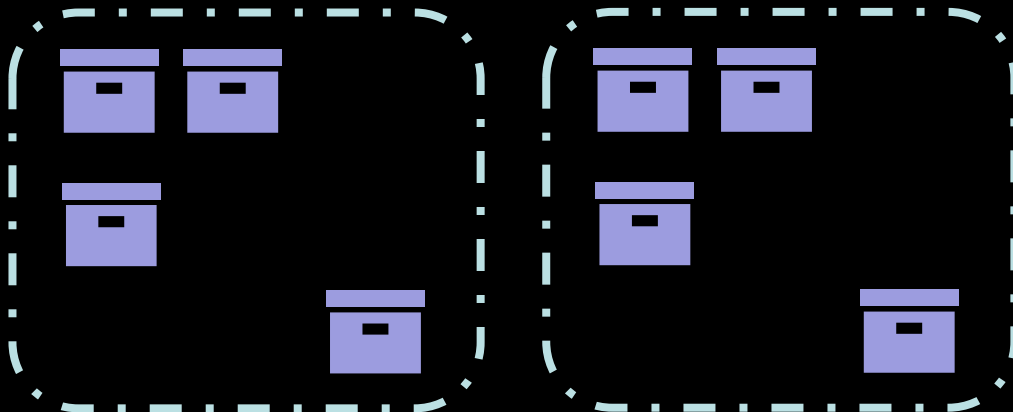
# HDFS – Architettura

Namenode



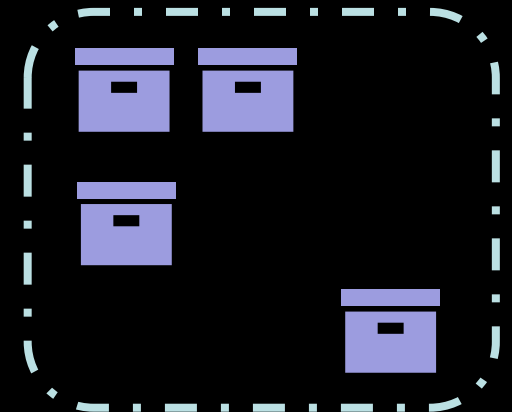
Metadata (nome, repliche, ...)  
/home/pippo/dati/

Datanodes



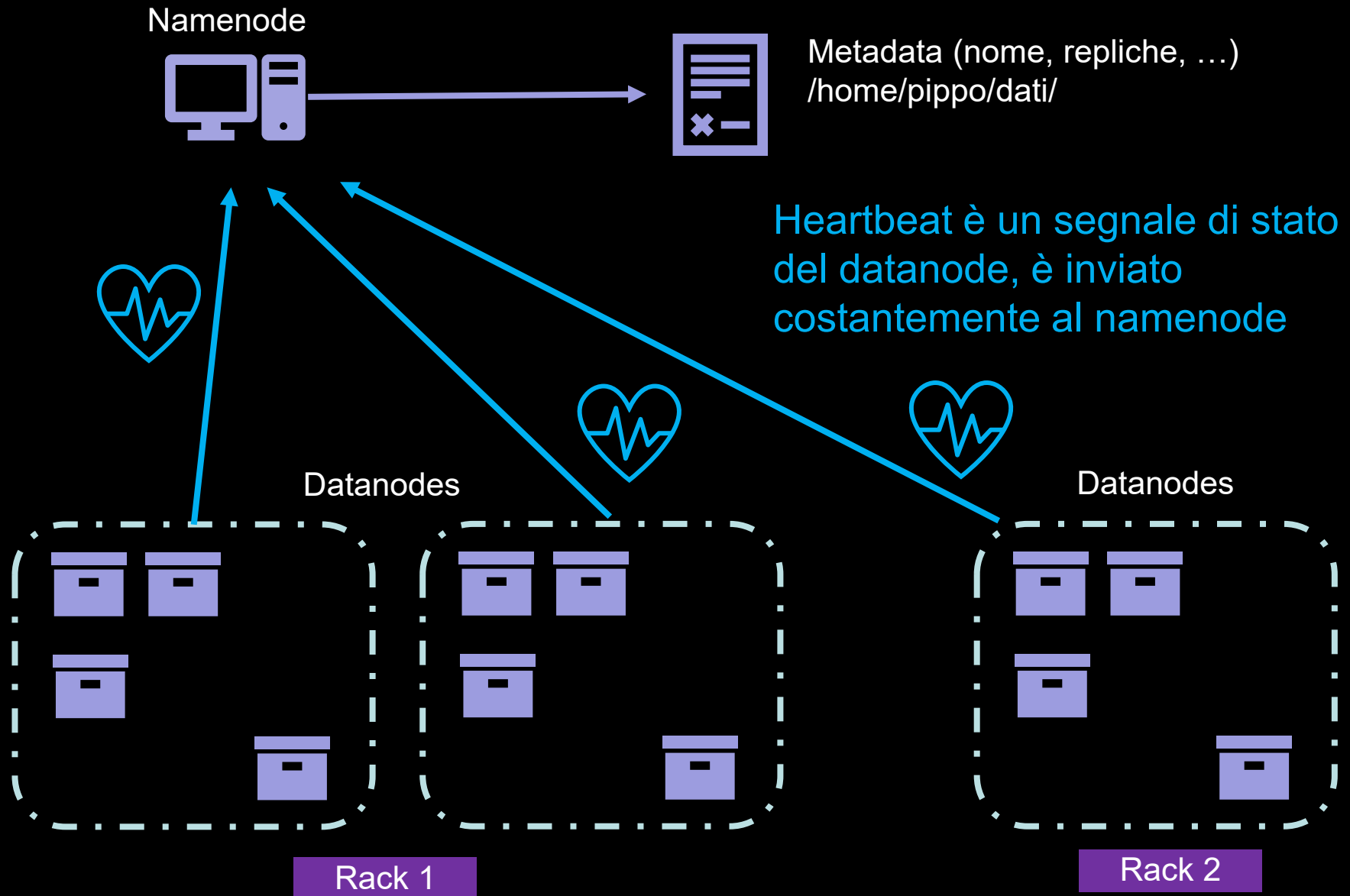
Rack 1

Datanodes

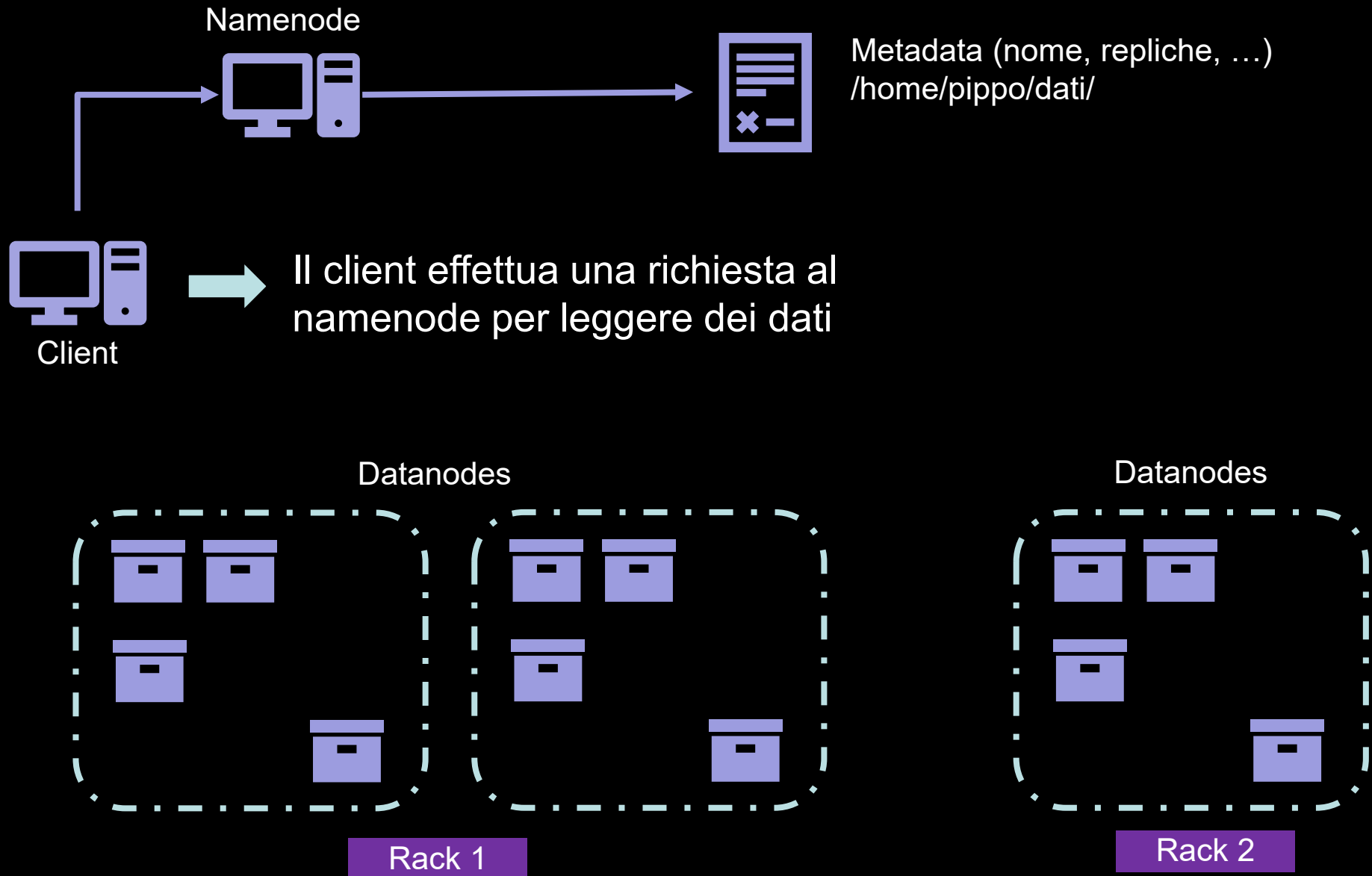


Rack 2

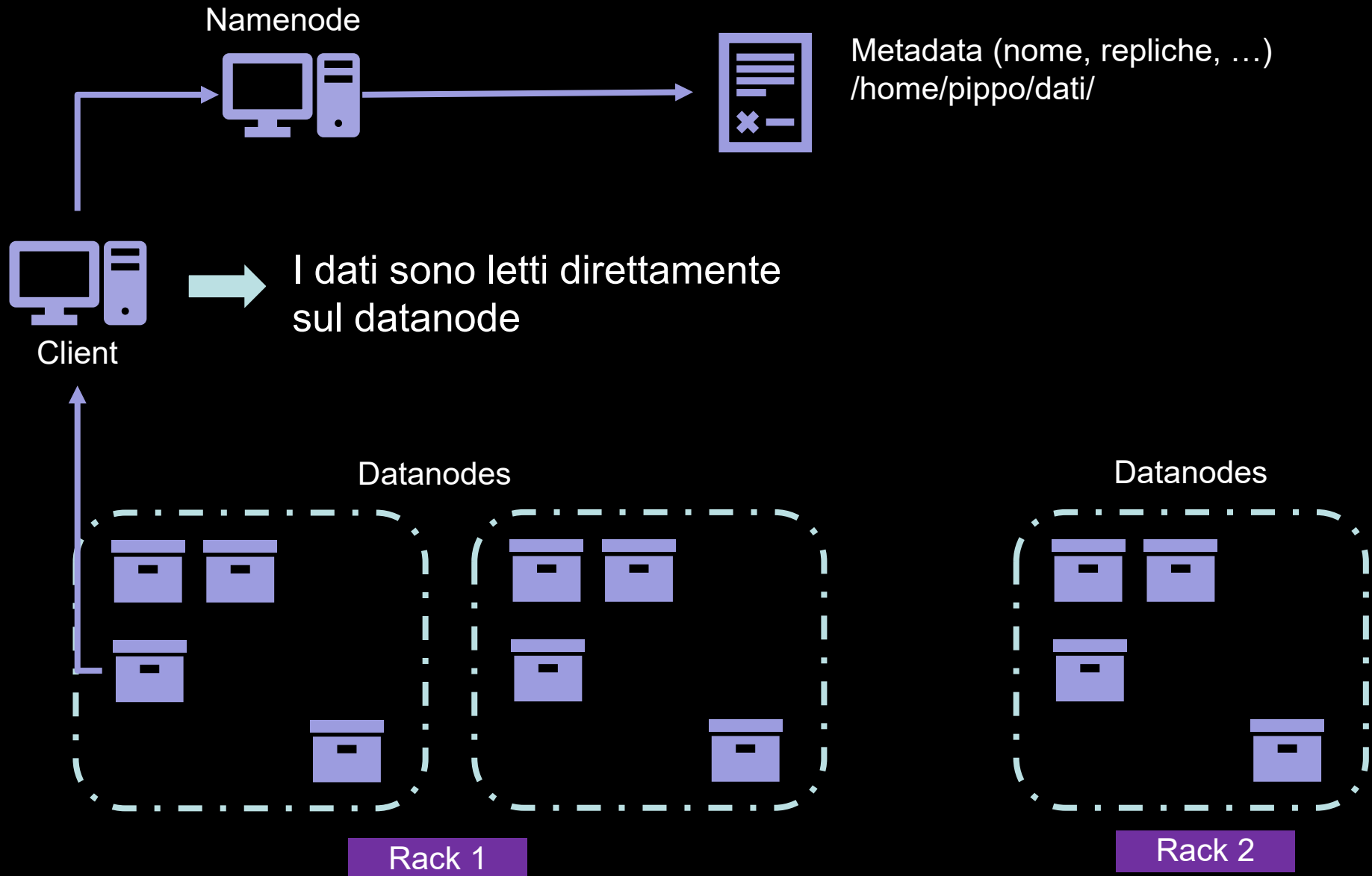
# HDFS – Architettura



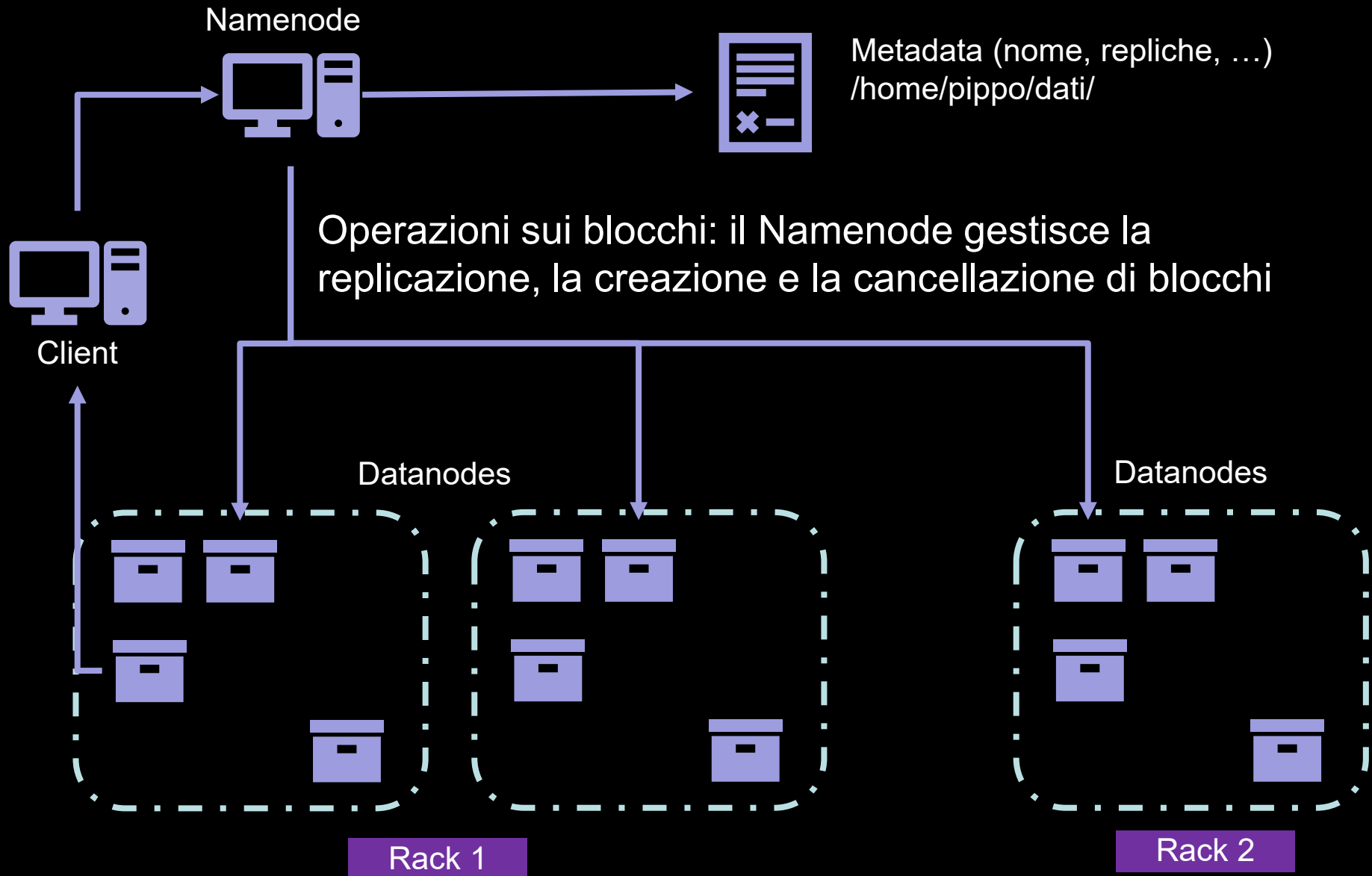
# HDFS – Architettura



# HDFS – Architettura

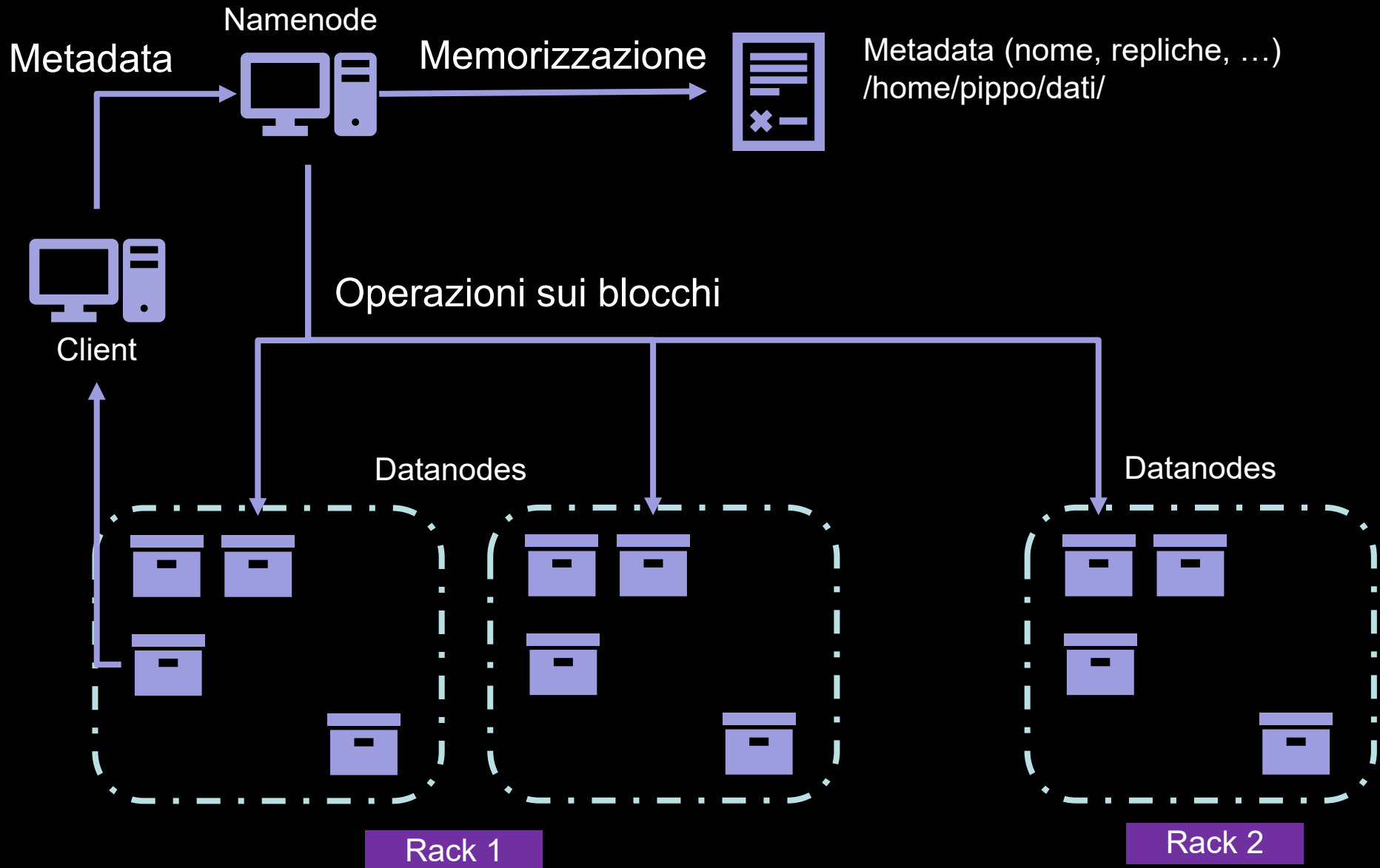


# HDFS – Architettura





# HDFS – Architettura



## HDFS – Lettura

## HDFS – Lettura

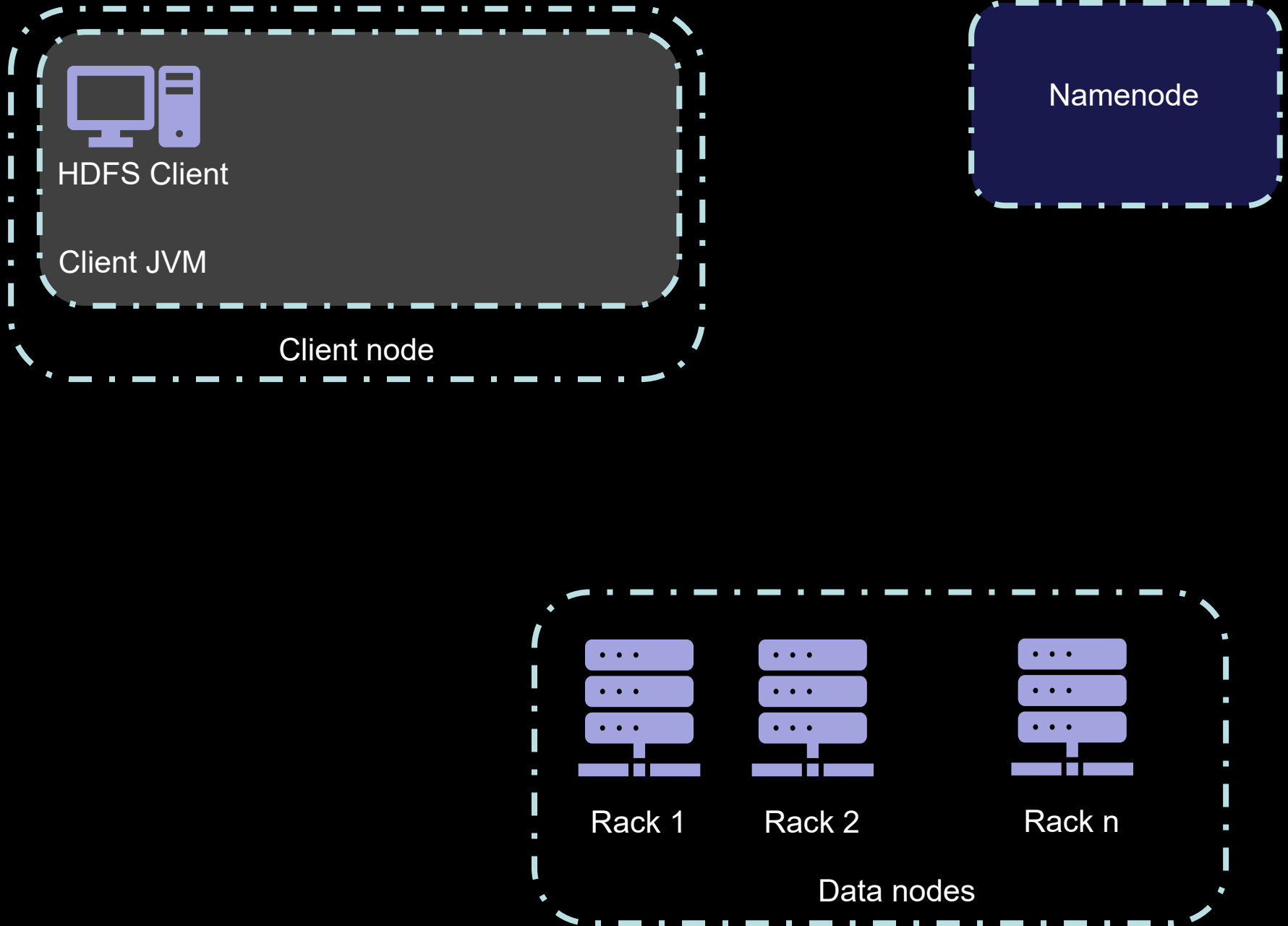


HDFS Client

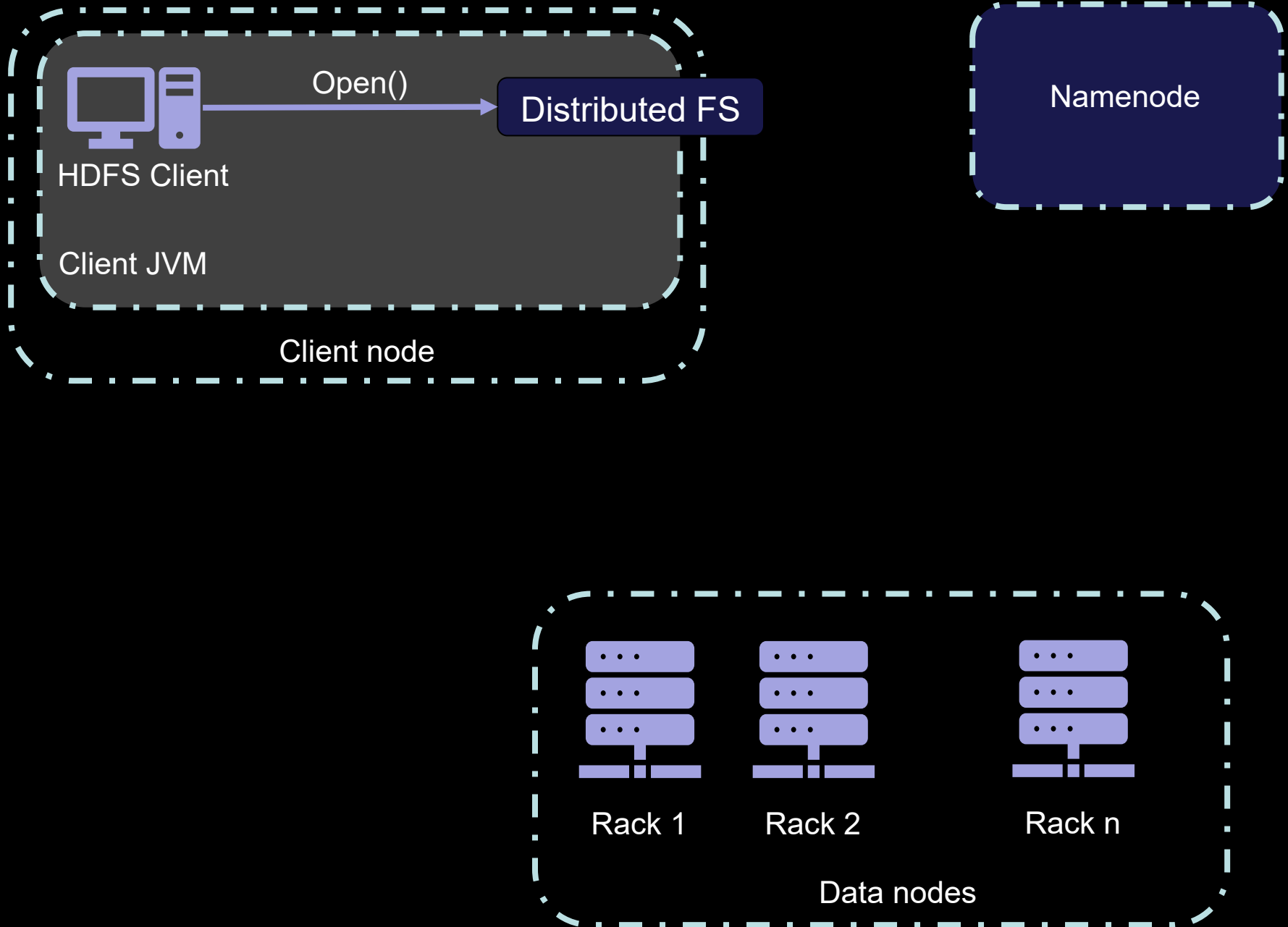
Client JVM

Client node

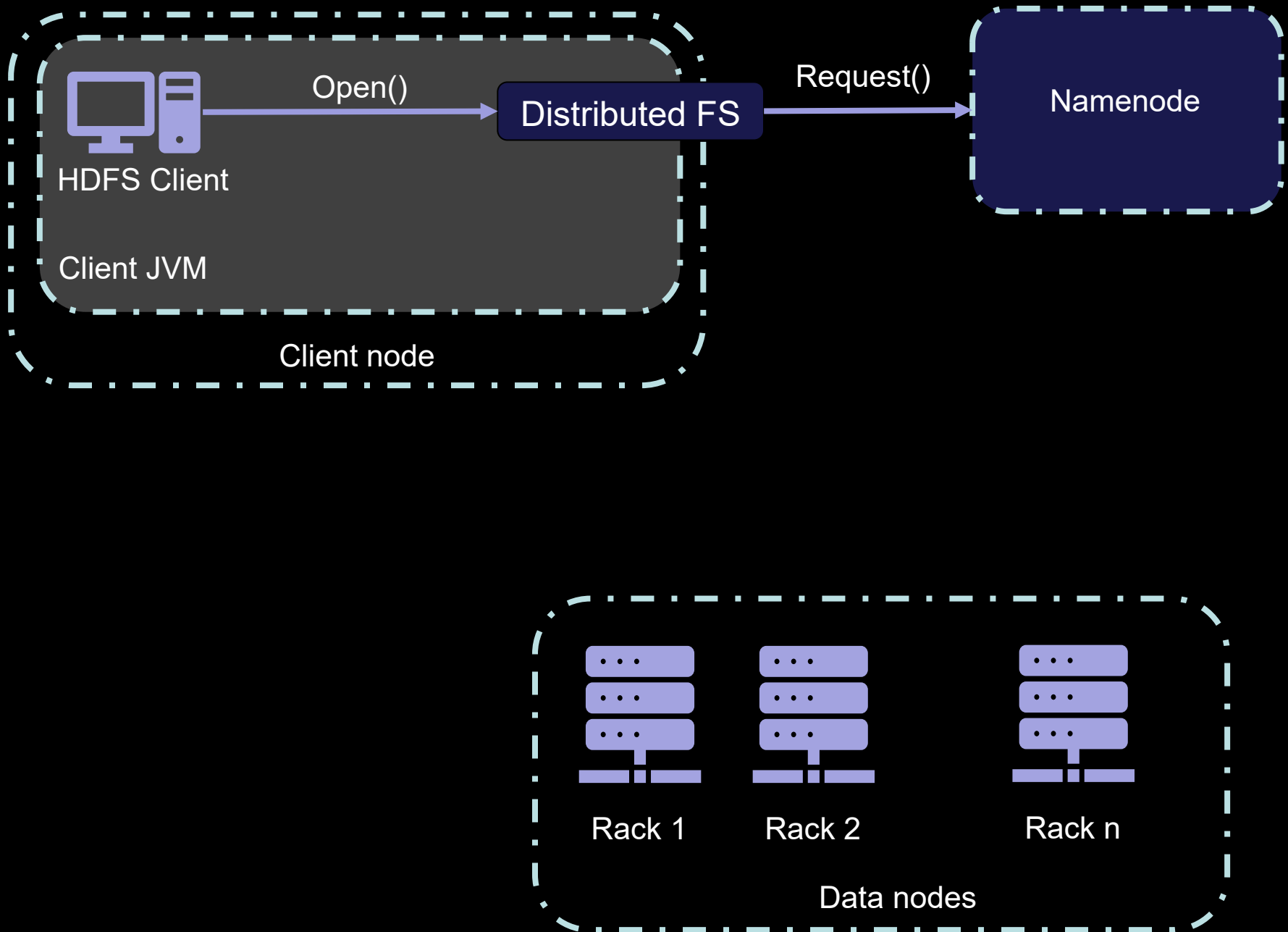
## HDFS – Lettura



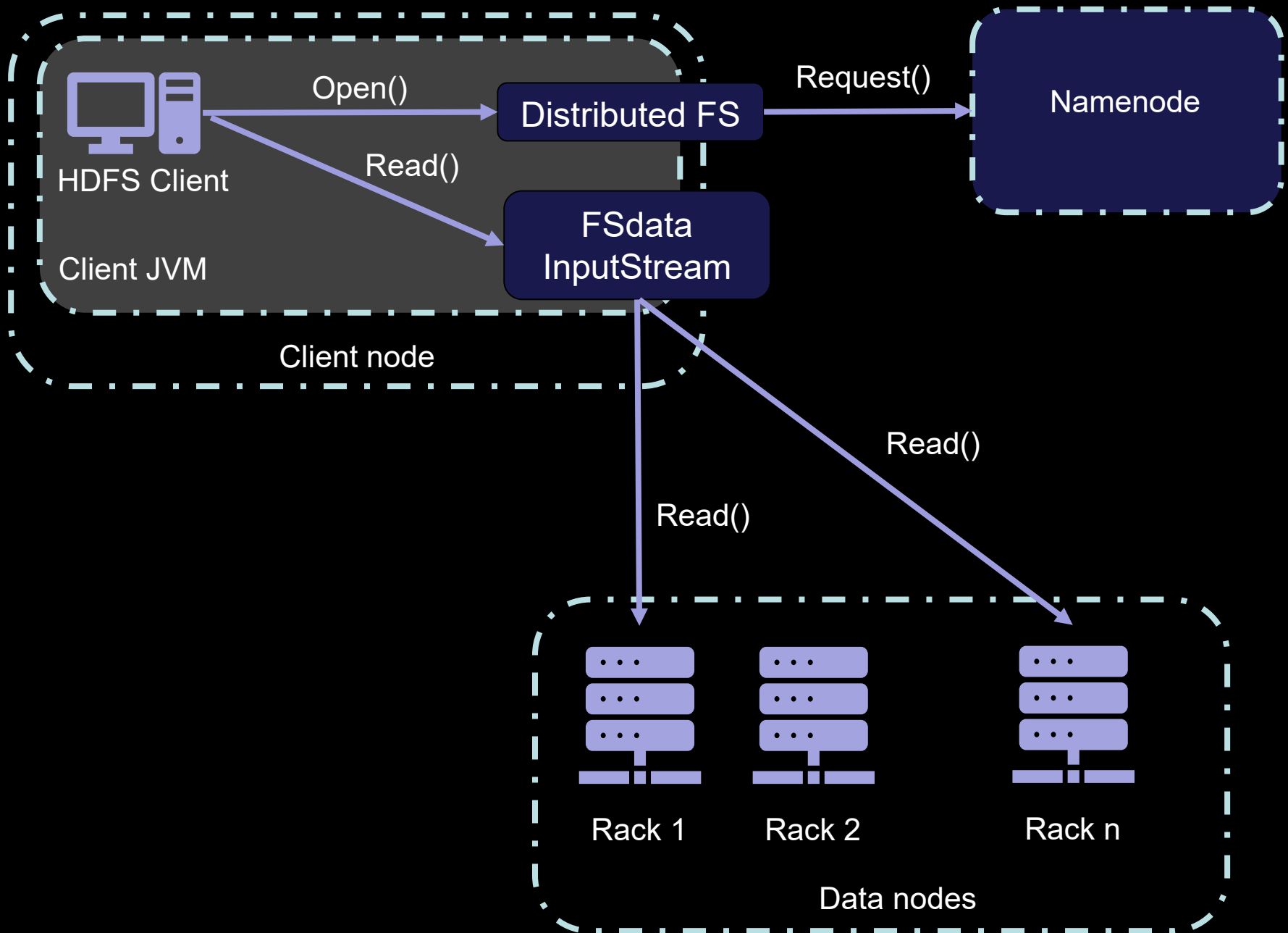
## HDFS – Lettura



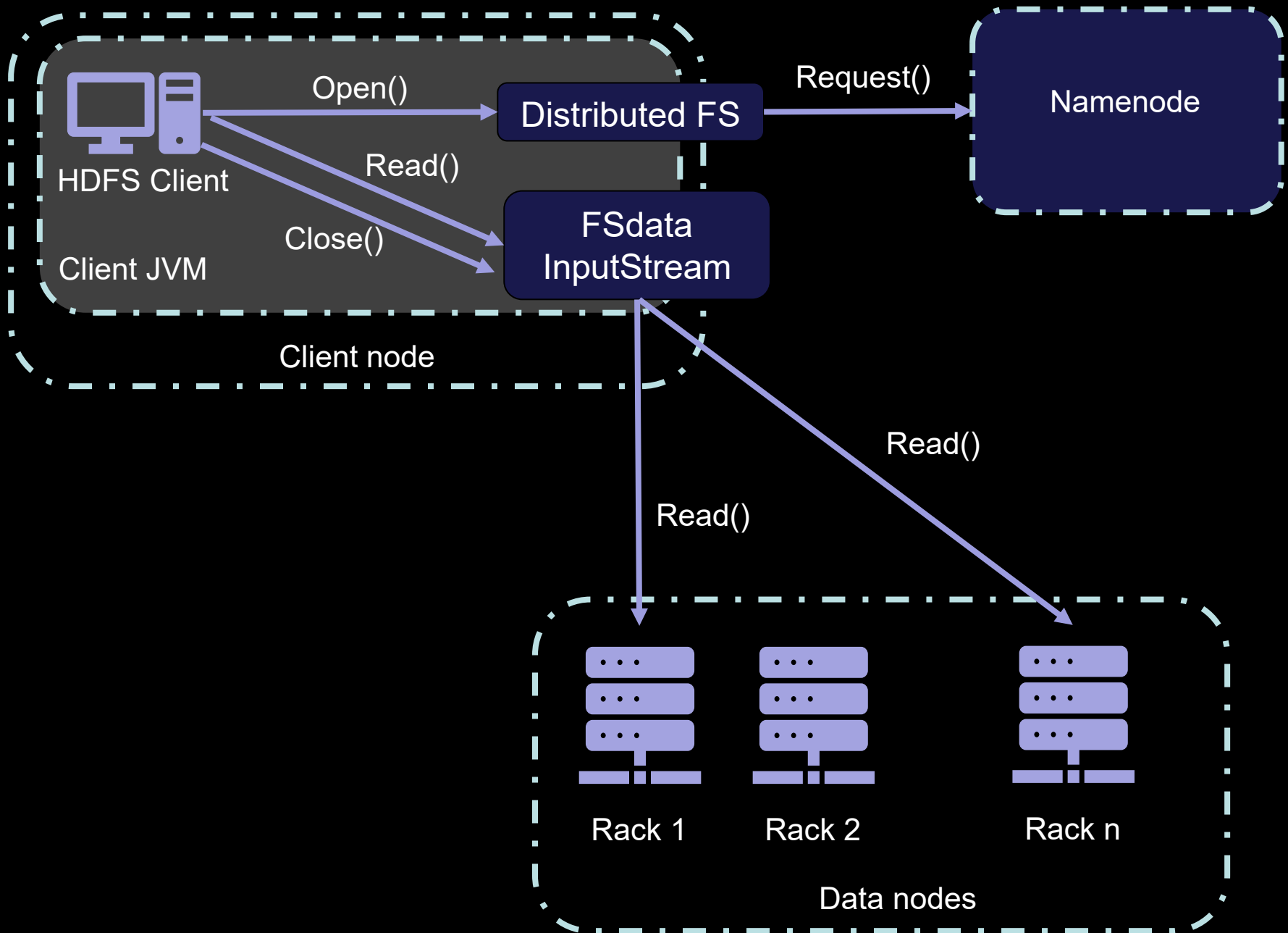
## HDFS – Lettura



## HDFS – Lettura

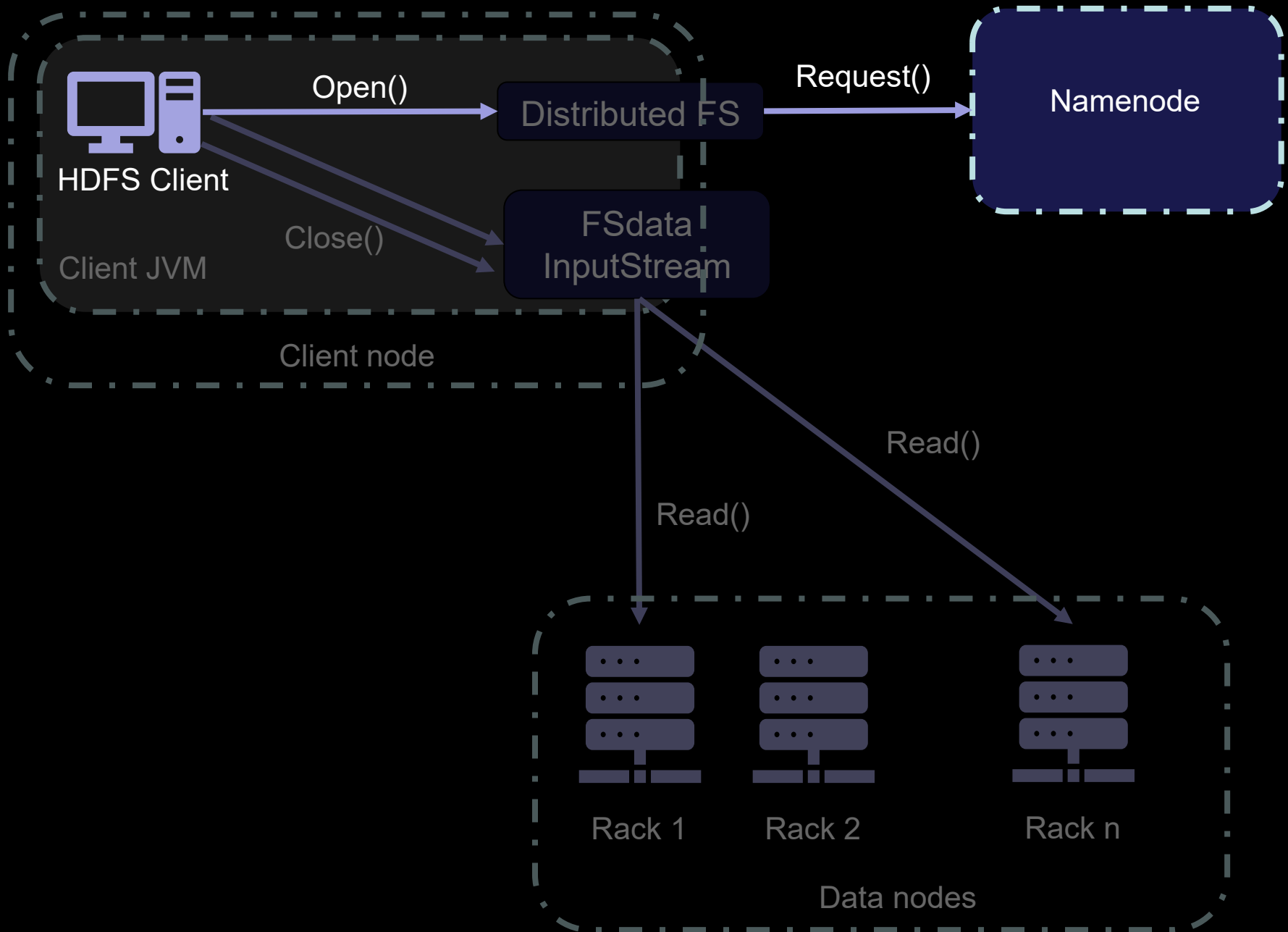


## HDFS – Lettura

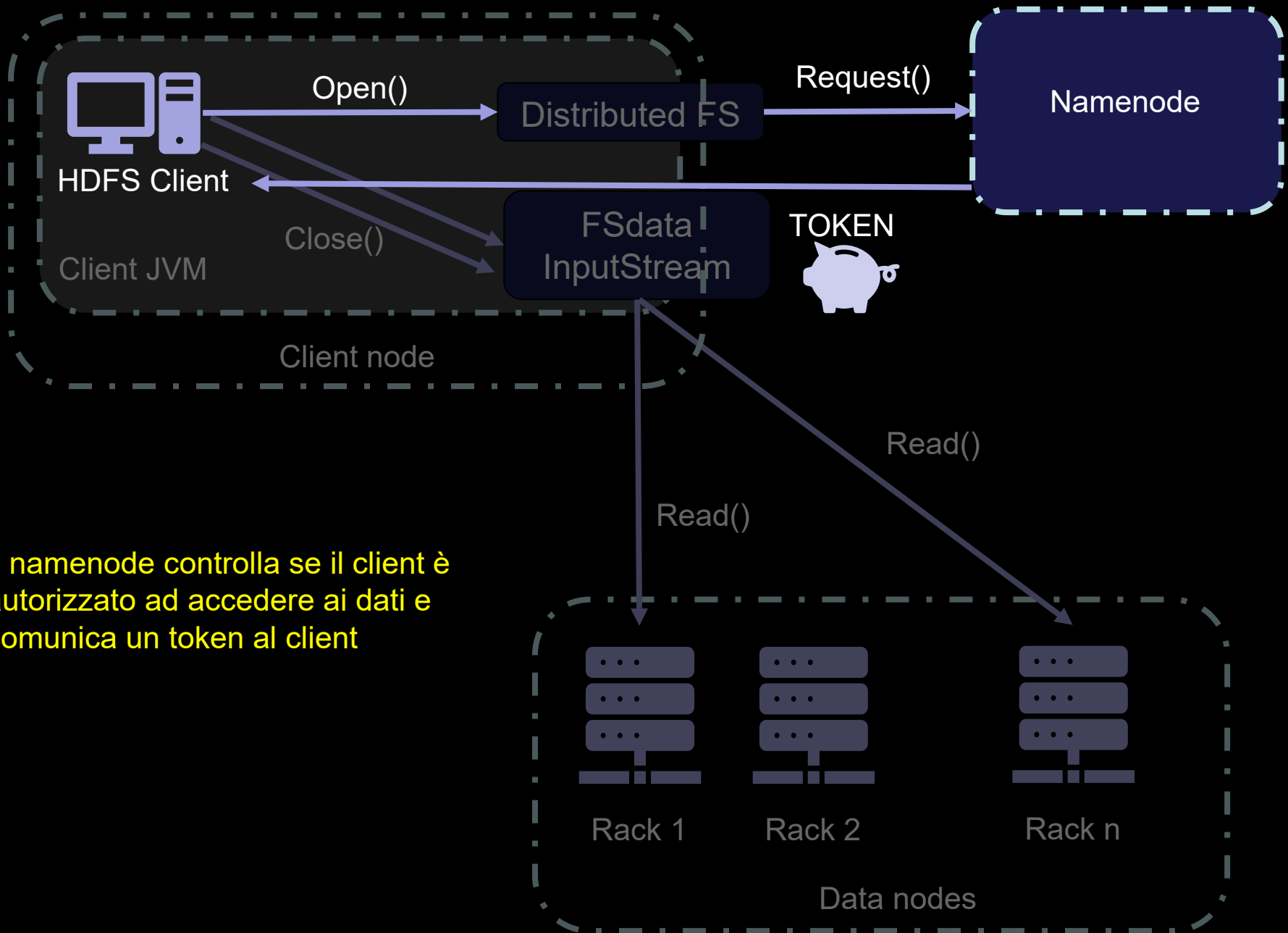




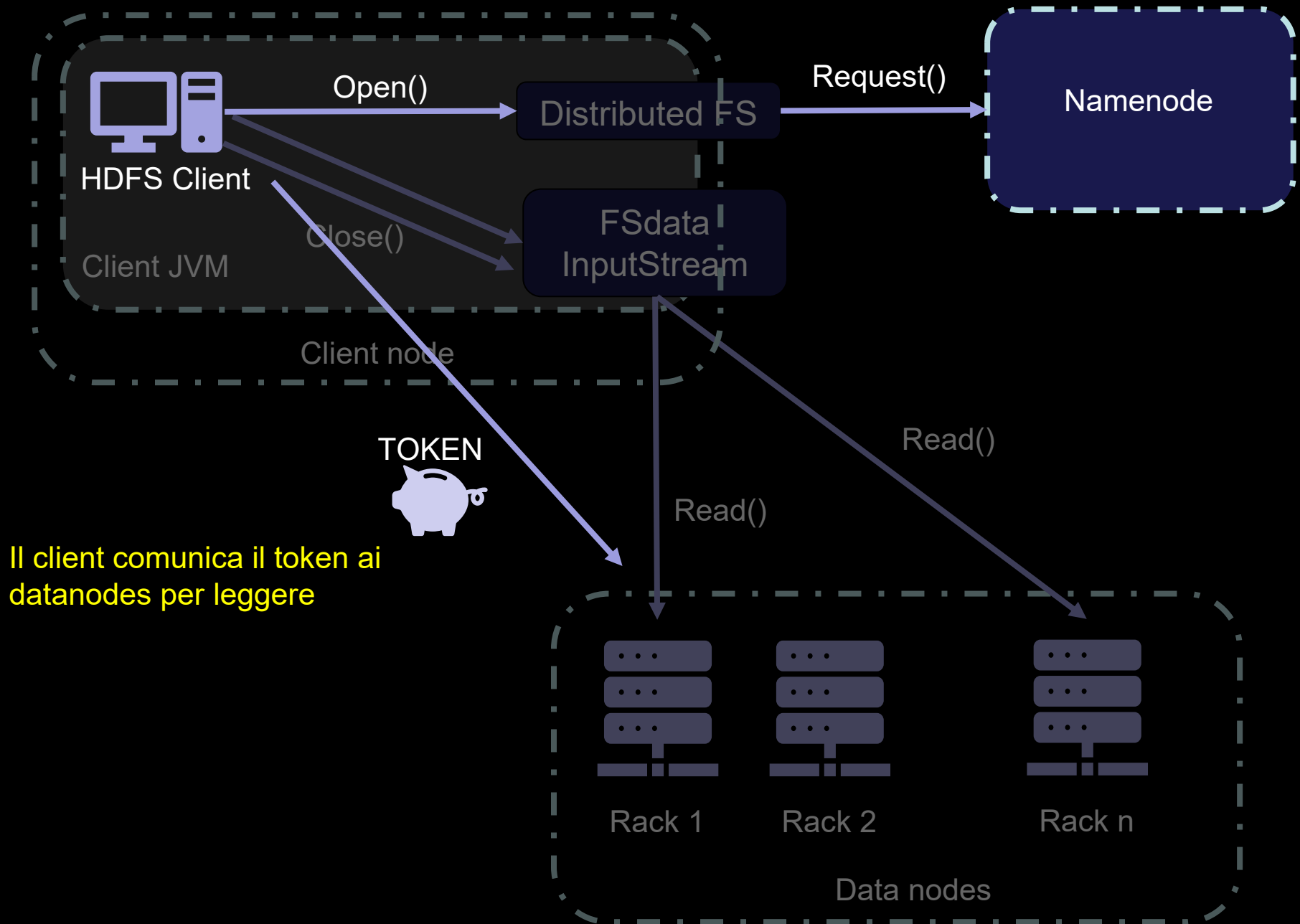
## HDFS – Lettura – RPC (remote procedure call)



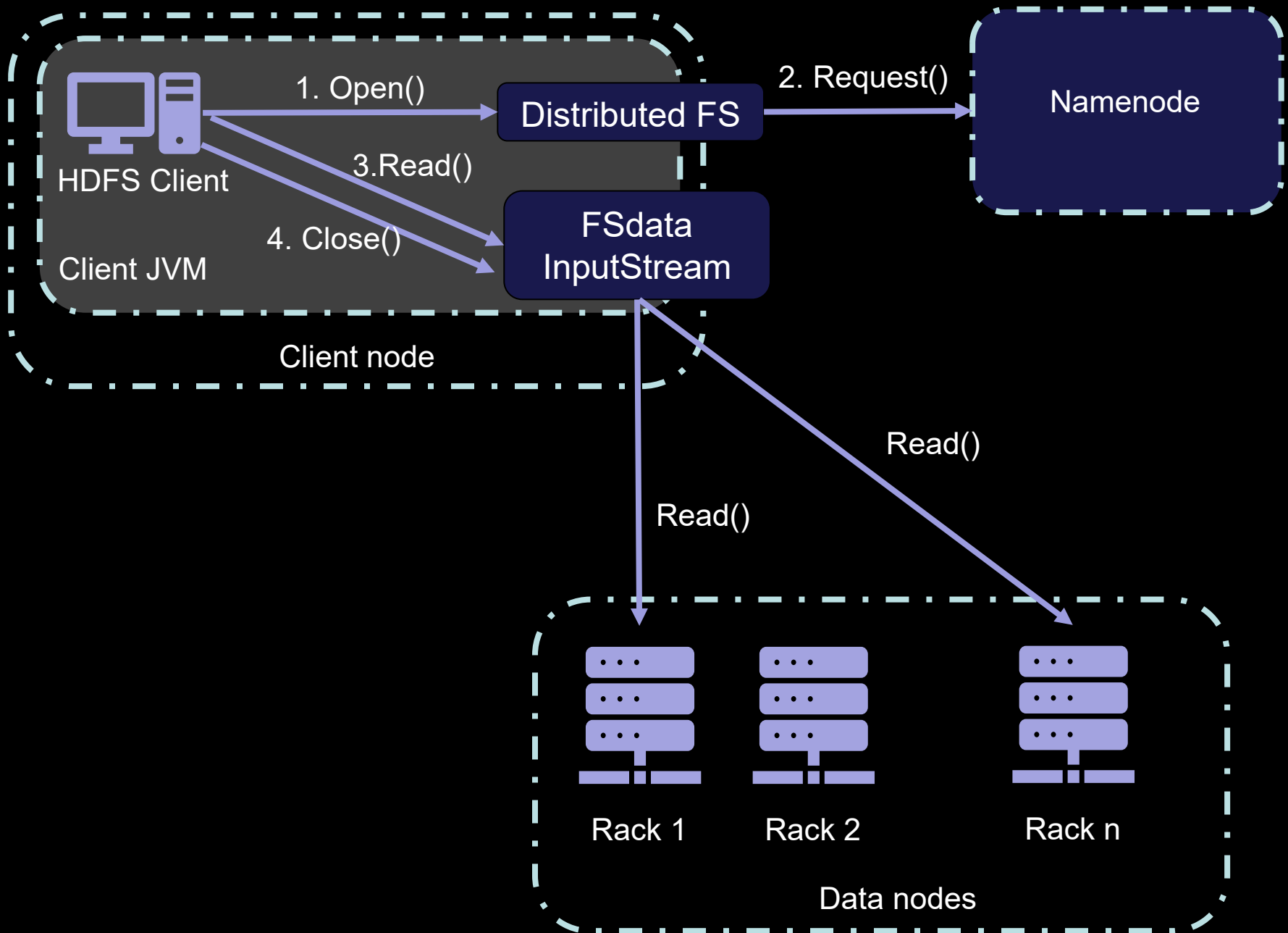
## HDFS – Lettura – RPC (remote procedure call)



## HDFS – Lettura – RPC (remote procedure call)



## HDFS – Lettura



## **HDFS – Scrittura**

## HDFS – Scrittura

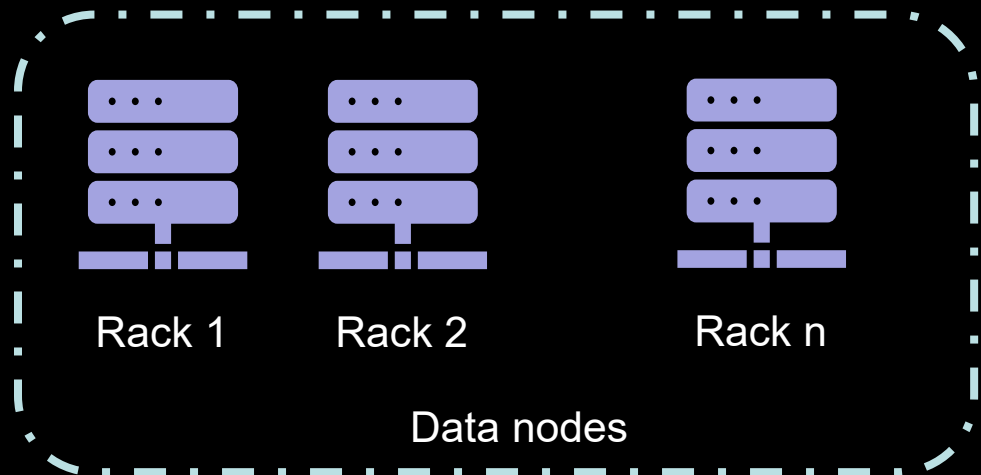
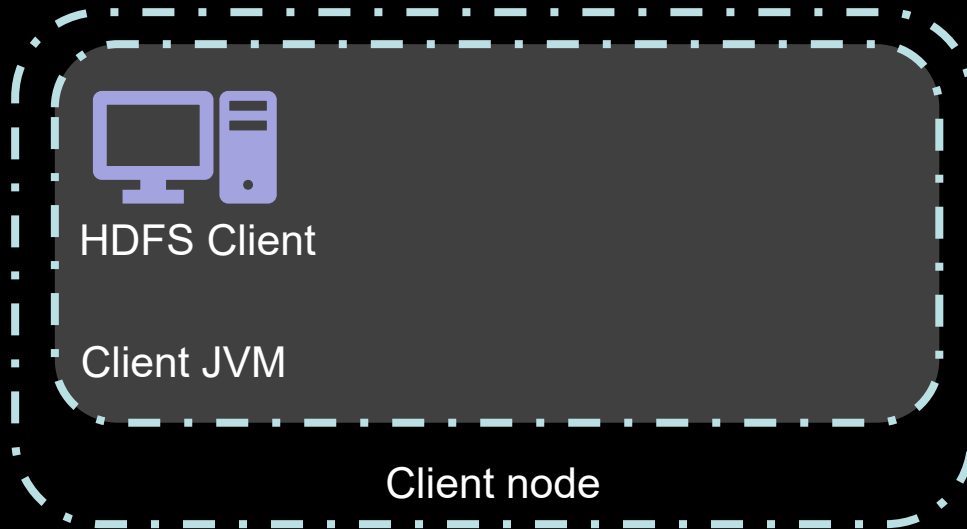


HDFS Client

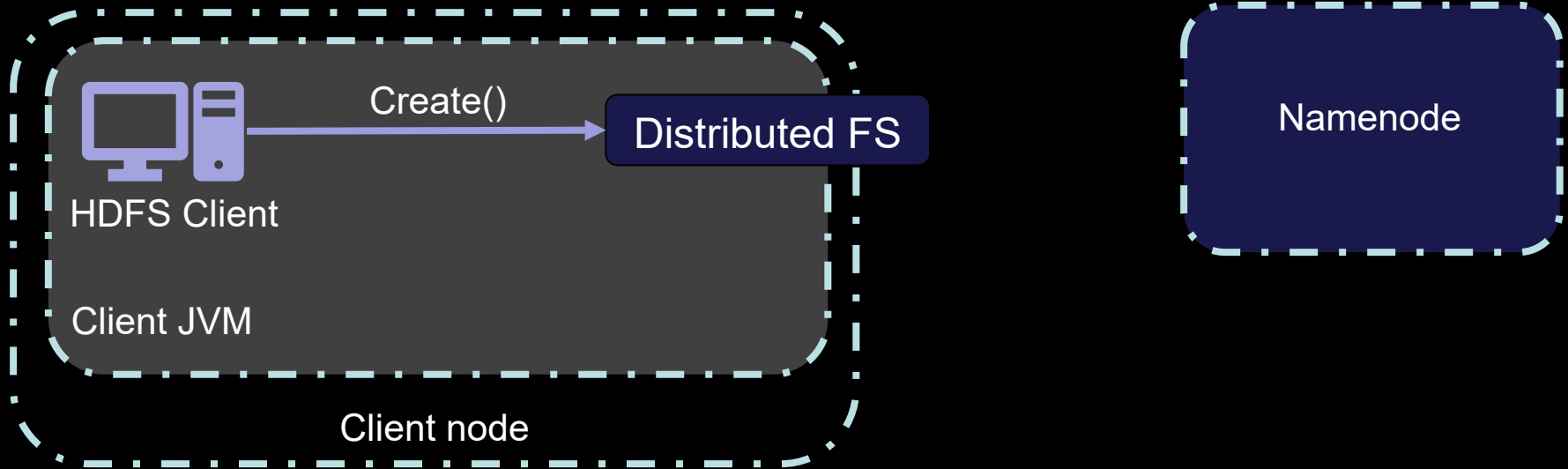
Client JVM

Client node

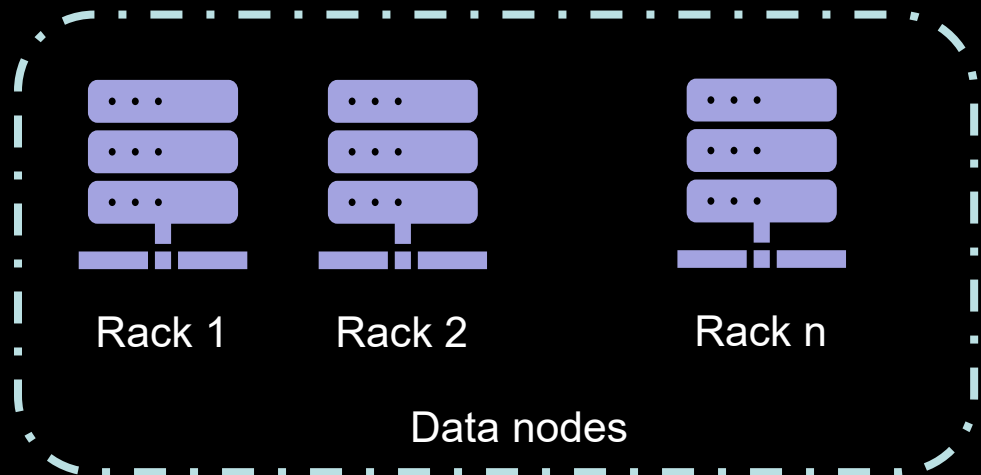
## HDFS – Scrittura



## HDFS – Scrittura

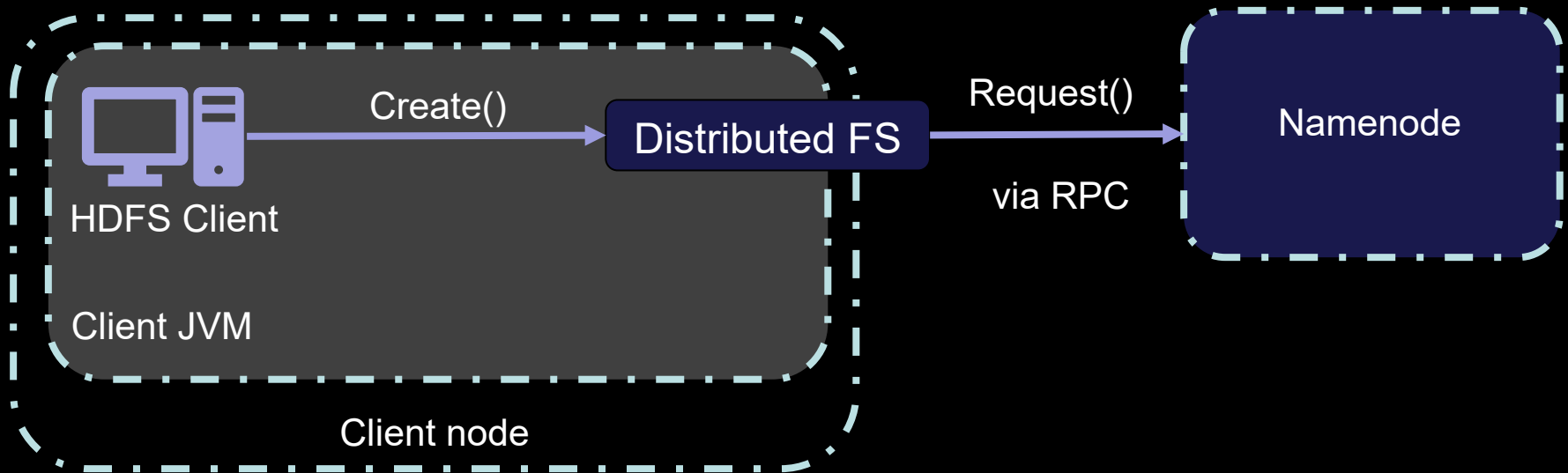


Il client crea un file sul file system distribuito



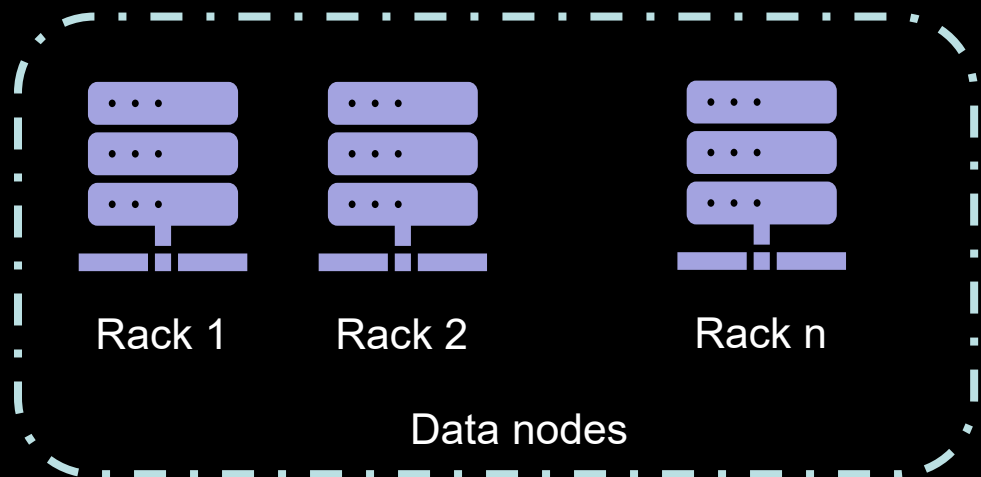


## HDFS – Scrittura

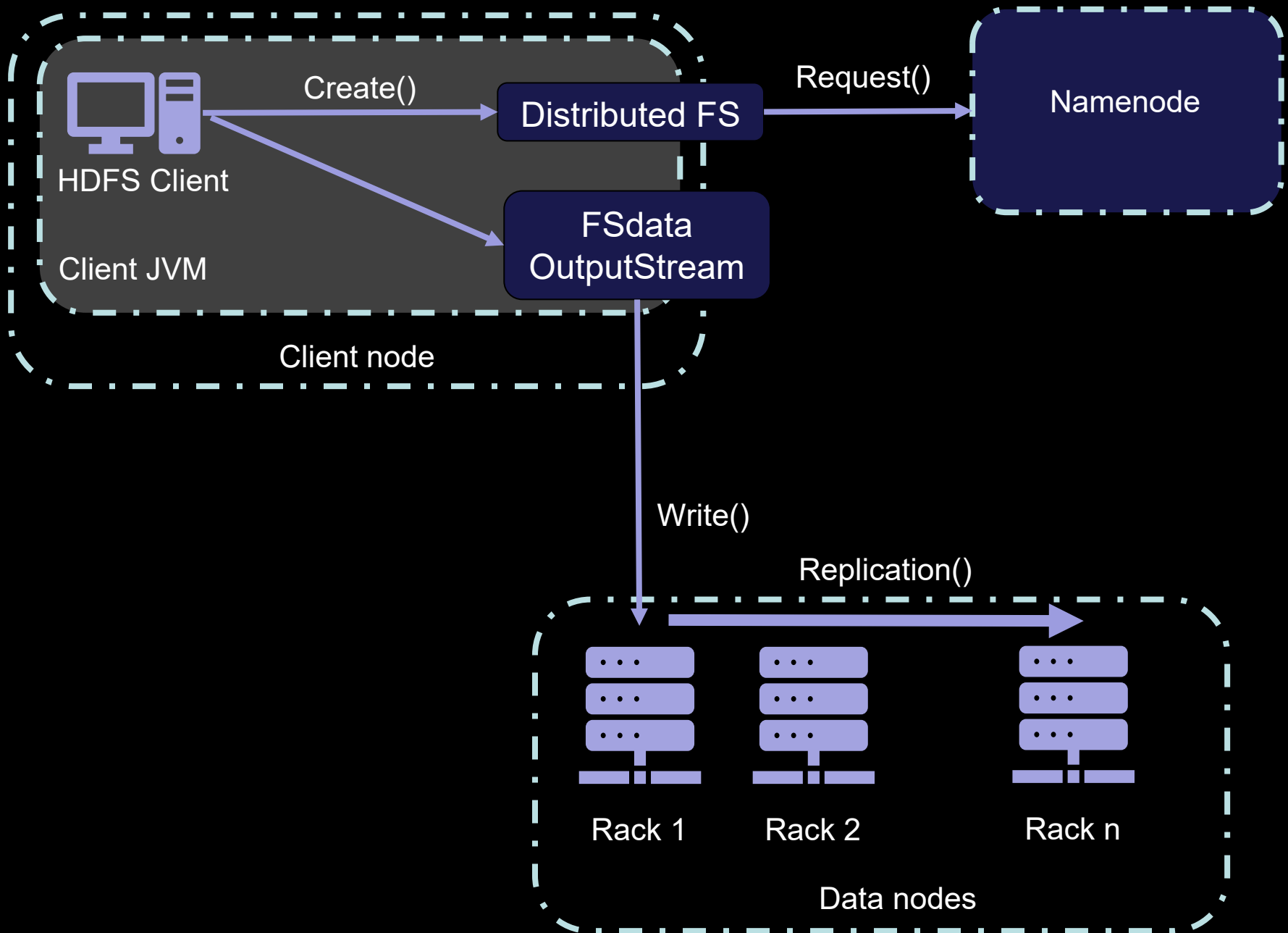


Il file system distribuito interagisce con il namenode per restituire una locazione dove creare il file

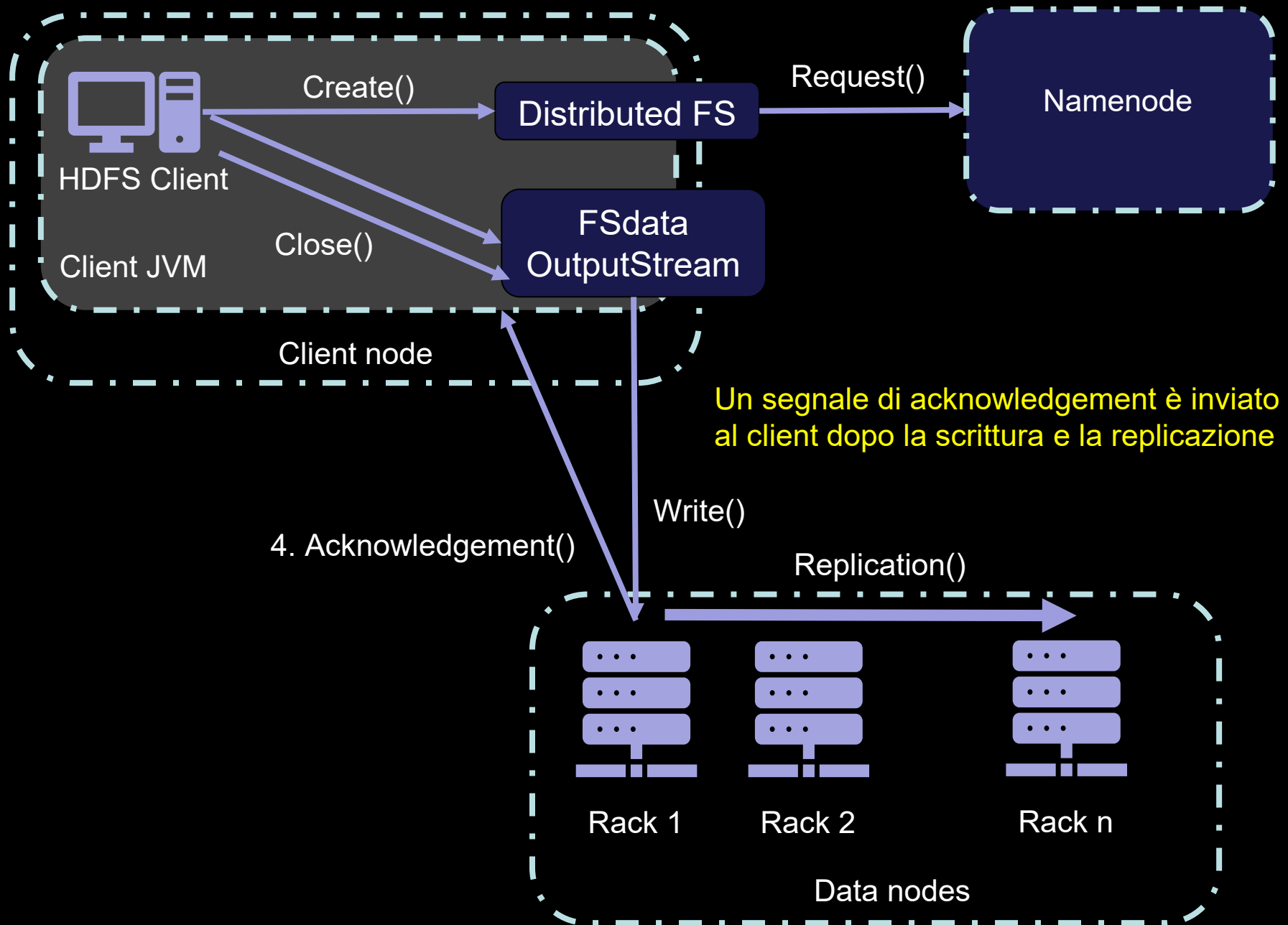
L'interazione è fatta sempre tramite chiamata RPC



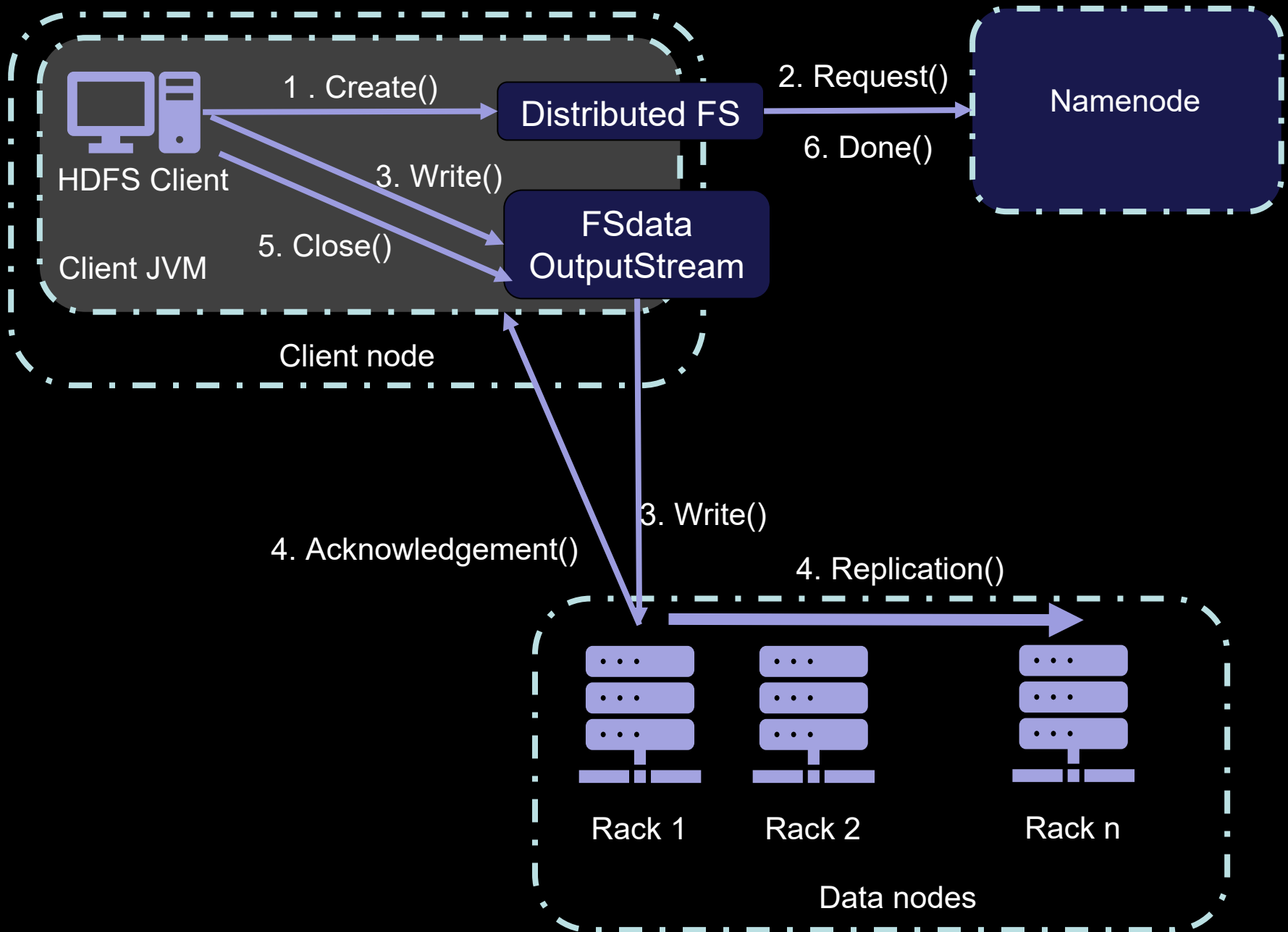
## HDFS – Scrittura



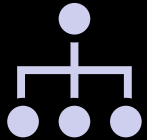
## HDFS – Scrittura



## HDFS – Scrittura



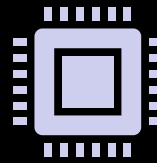
# HDFS - Vantaggi



Scalabile



Ridondante



Funziona su  
commodity hardware



Sicuro