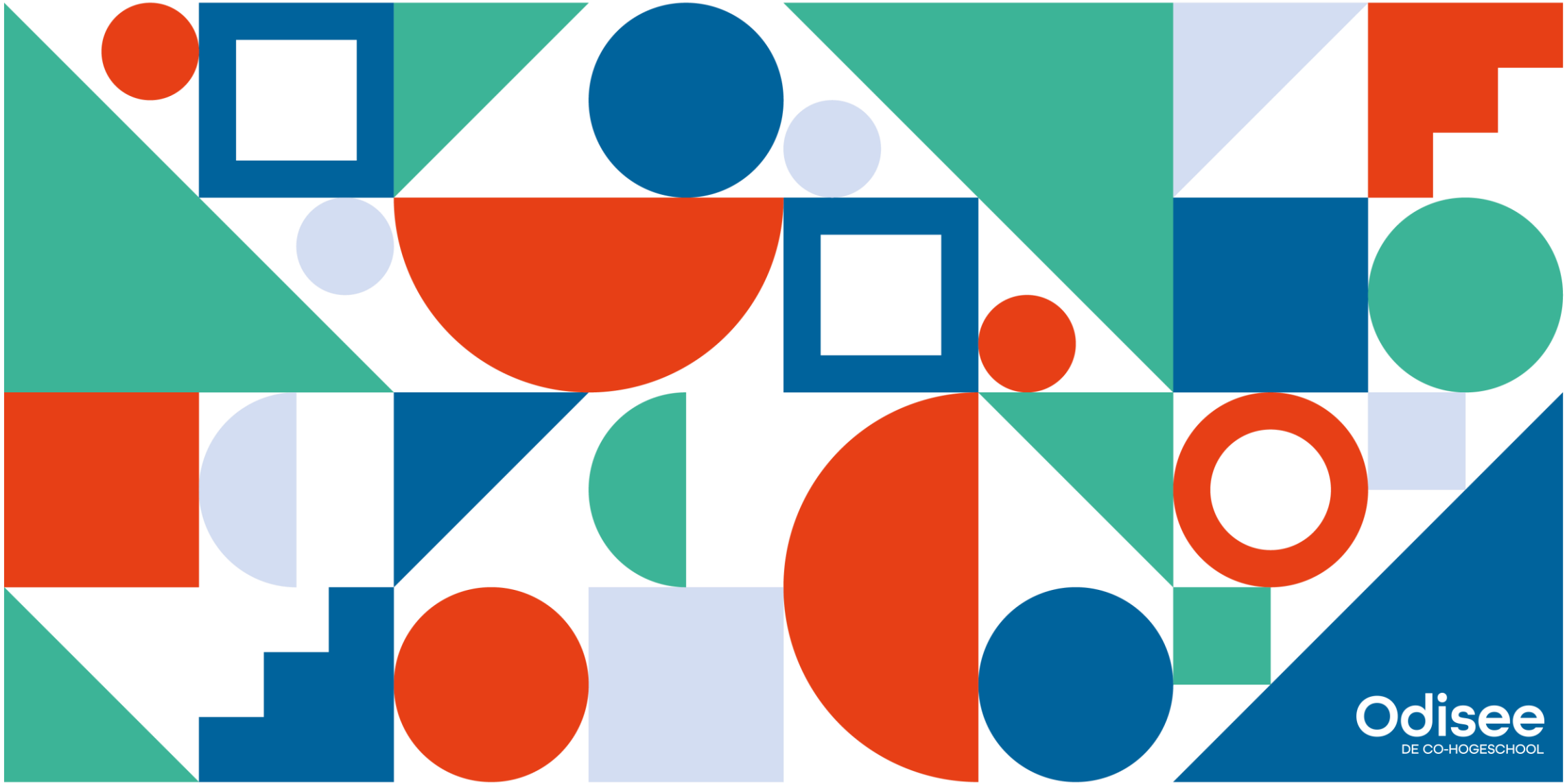
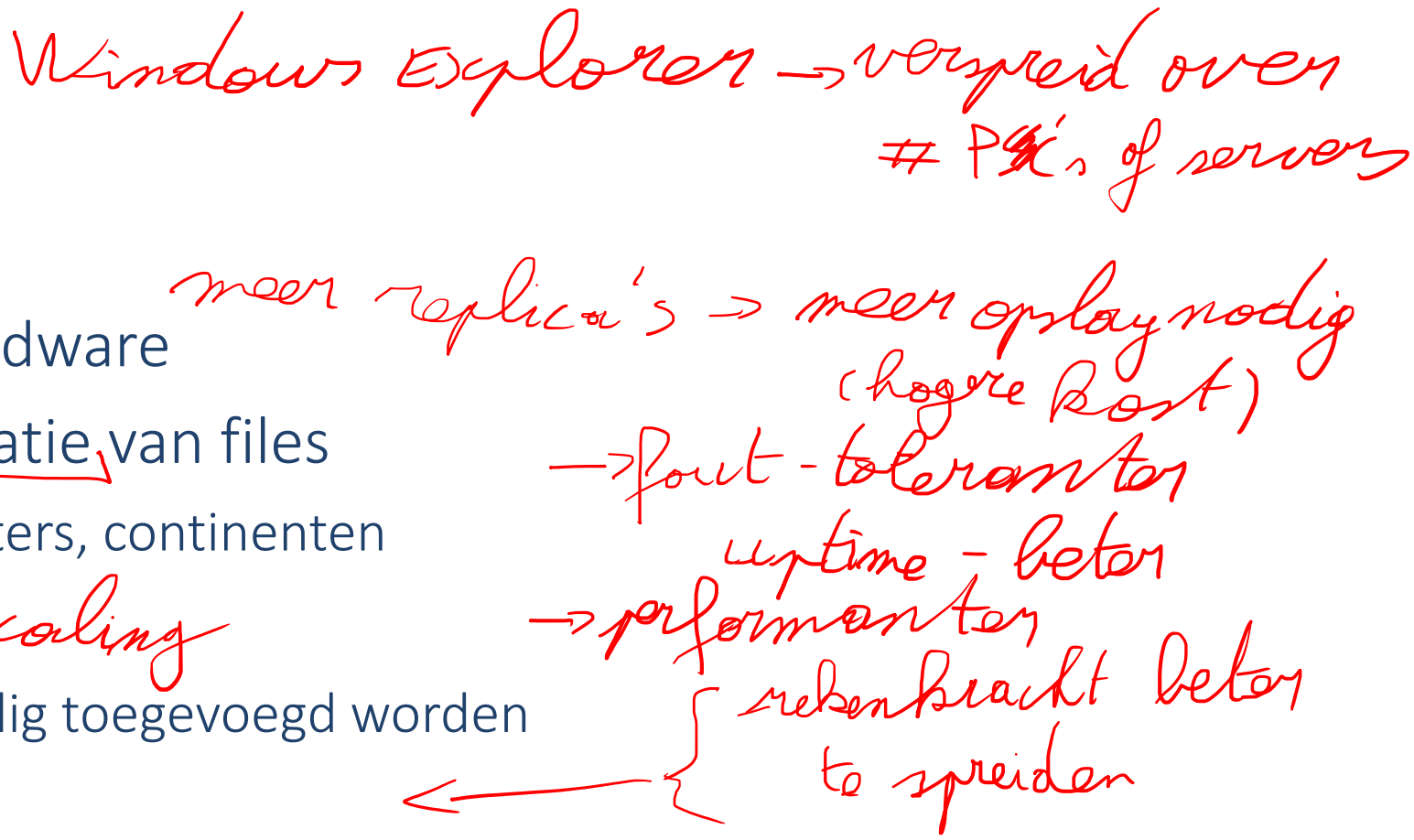

Odisee
DE CO-HOGESCHOOL

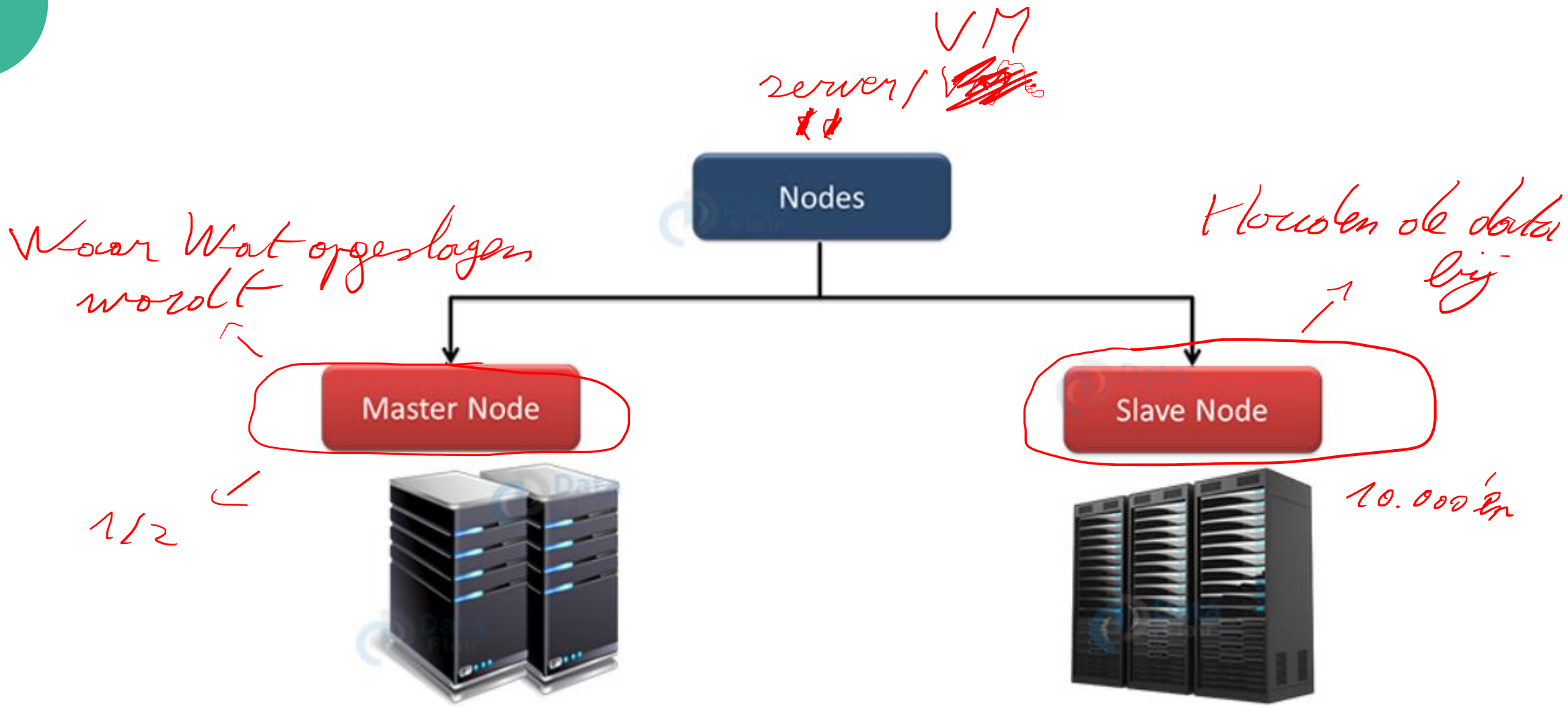
HDFS

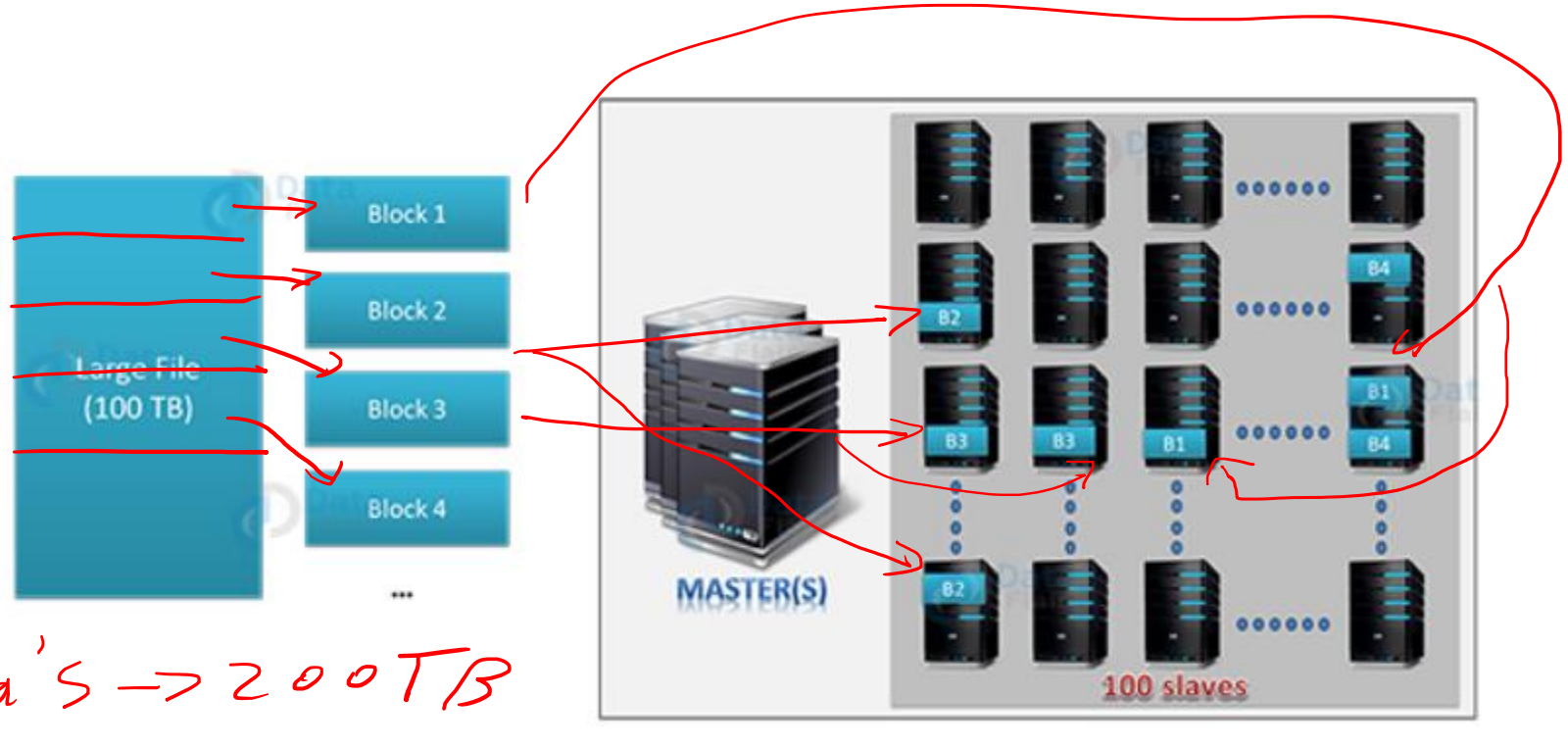
Jens Baetens

Wat is HDFS?

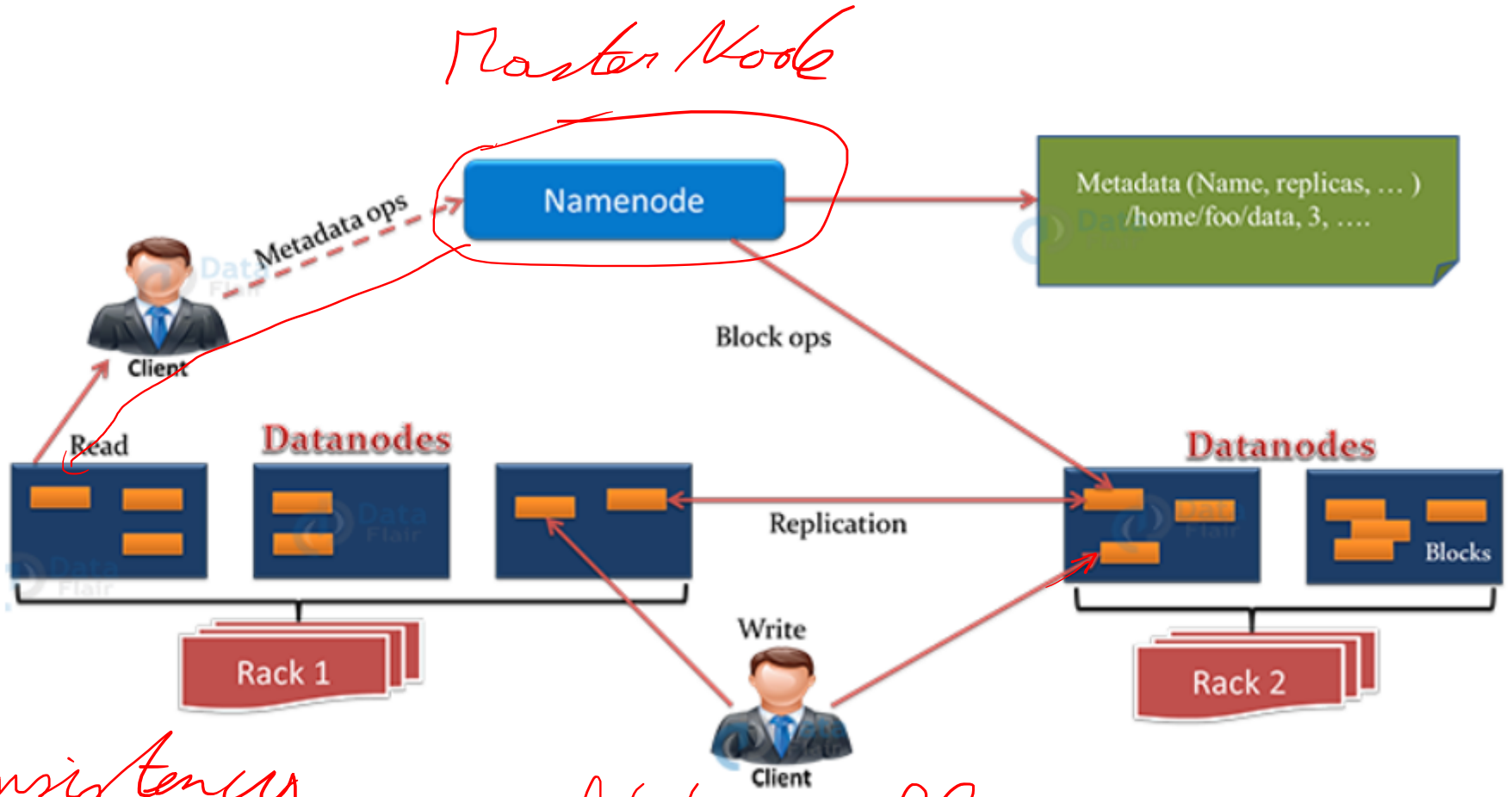
- Distributed File Storage
- Cluster van commodity hardware
- Fault Tolerance door replicatie van files
  - Verschillende racks, datacenters, continenten
- Scalable
  - Extra nodes kunnen eenvoudig toegevoegd worden
- Parallelle data access




Nodes
Master Node
Slave Node



2 replica's -> 200TB



Lazy Consistency Evaluation
→ overgangsfase waar niet alle blokken consistent zijn

altijd naar alle blokken → data synchroon te houden
↳ # ms duren

Features – Distributed Storage

- Onderverdeel files in kleinere delen (Blocks)
- Verdeel de blokken over de nodes
- Repliceer de blokken het gewenste aantal keren (minstens 1 op een andere rack)

Features - Blocks

- Default block size is 128 MB
  - File van 150 MB wordt dus gesplitst in 128 MB en 22 MB en niet 75/75
- Beheer van de blokken volledig door de namenode
- Voordeel van grotere block-sizes is dat
  - de file sneller ingelezen wordt
  - Map reduce voert functie uit per block dus niet te veel blocks gewenst.

→ anders verlies je te veel tijd met communicatie / netwerkactiviteit

Features - Replication

- Het aantal keer dat eenzelfde blok voorkomt over alle datanodes
- Dit verhoogt de beschikbaarheid van een blok omdat indien een node crashed, de data beschikbaar is op een andere node.

- Er wordt gepoogd minstens 1 replica op een andere node te plaatsen

- Default waarde is 3 → In onze VM is dit 1

Features – High Availability, Data Reliability en Fault Tolerance

- Datanode fails
  - Datanode stuurt heartbeat naar de namenode -> detecteren van crashed datanode
  - Datanode crashed tijdens opvragen gegevens -> vraag nieuwe locatie aan namenode

- Namenode fails
  - In de master-slave architectuur is de master een single point of failure
  - Vanaf Hadoop 2.0 is er een secondary namenode
- Consistency bij gebruik van meerdere namenodes vereist extra aandacht
  - Identieke gegevens in primary en secondary namenode
  - Wat bij terug online komen van primary namenode

Features - Scalability

- Vertical scaling
  - Meer HDD’s in een node
  - Heeft downtime nodig (om HDD te installeren)
- Horizontal scaling
  - Extra noden toevoegen aan cluster

Features – High Throughput

- Throughput = Hoeveel werk dat gedaan wordt per seconde/minuut/…
- Data wordt parallel gelezen, het werk wordt verdeeld door de verschillende systemen /nodes

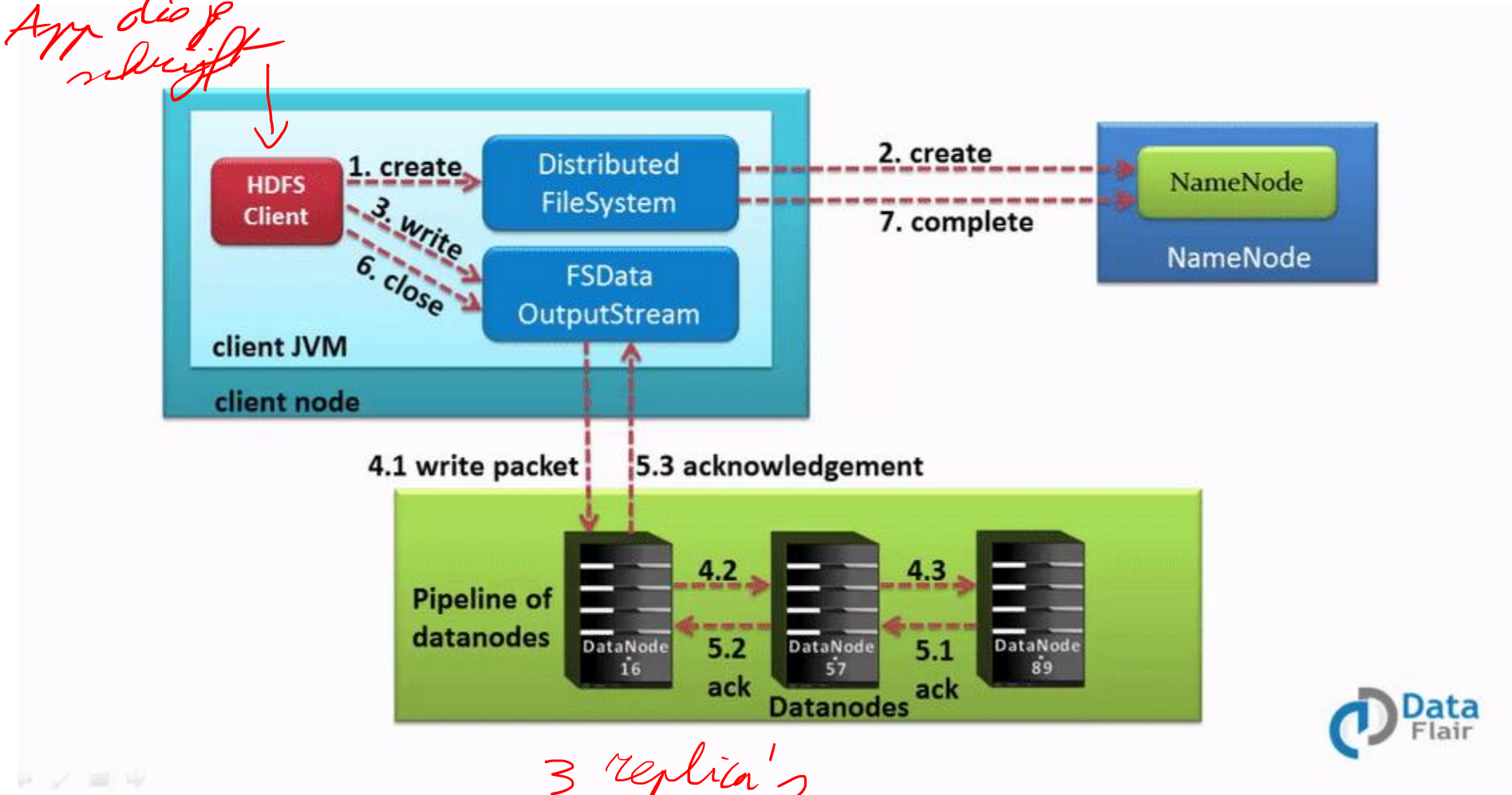
velocity

volume value veracity

variety

HDFS - Operations

App die je schrijft



3 replica's

↳ schrijven van replica's in de ondergrond

Hoe bewerken van een hdfs?

- Ofwel via commandline, communicatie met hdfs geconfigureerd in xml-files
  - https://www.informit.com/articles/article.aspx?p=2755708&seqNum=4
  - hadoop/etc/hadoop/core-site.xml
  - hadoop/etc/Hadoop/hdfs-site.xml
- Maak gebruik van libraries om met een HDFS te communiceren
  - Pydoop (gebruikt in mijn code)
  - Snakebite
  - Hdfs

```
from snakebite.client import Client

client = Client('localhost', 9000)
for x in client.ls(['/']):
    print x
```

Odisee
DE CO-HOGESCHOOL

Pydoop

- https://crs4.github.io/pydoop/api_docs/hdfs_api.html#hdfs-api
- Aanmaken van een connectie:
  - import pydoop.hdfs as hdfs
  - client = hdfs.hdfs(host='localhost', port=9000)
  - client.capacity()

Aanmaken van bestanden en folders

CLI

- Exists: hdfs dfs -test -d hdfs_path
- Mkdir: hdfs dfs -mkdir -p /bigdata/03_HDFS
- hadoop fs -put /path/in/linux /hdfs/path
- hadoop fs -get /hdfs/path /path/in/linux

Pydoop

- client.exists('03_HDFS’)
- client.create_directory('03_HDFS’)
- client.set_working_directory('03_HDFS’)
- localFS.copy(local_file, client, remote_file)
- localFS.move(local_file, client, remote_file)

Bekijken van het filesysteem

- hdfs dfs -ls command
- hdfs dfs –usage

- hdfs dfs -cat <hdfs_filename> | head -n 5

- client.list_directory("")
- client.used()
- client.working_directory()
- client.default_block_size()
- client.open_file(remote_file)

Aanpassen van het filesysteem

- hadoop fs -mv oldname newname
- hdfs dfs -rm path (-r voor folders)
- hdfs dfs –setrep –w 3 /tmp/logs/file.txt

- client.rename(“old", “new")
- client.delete("davinci3.txt")
- client.set_replication("davinci.txt", 5)