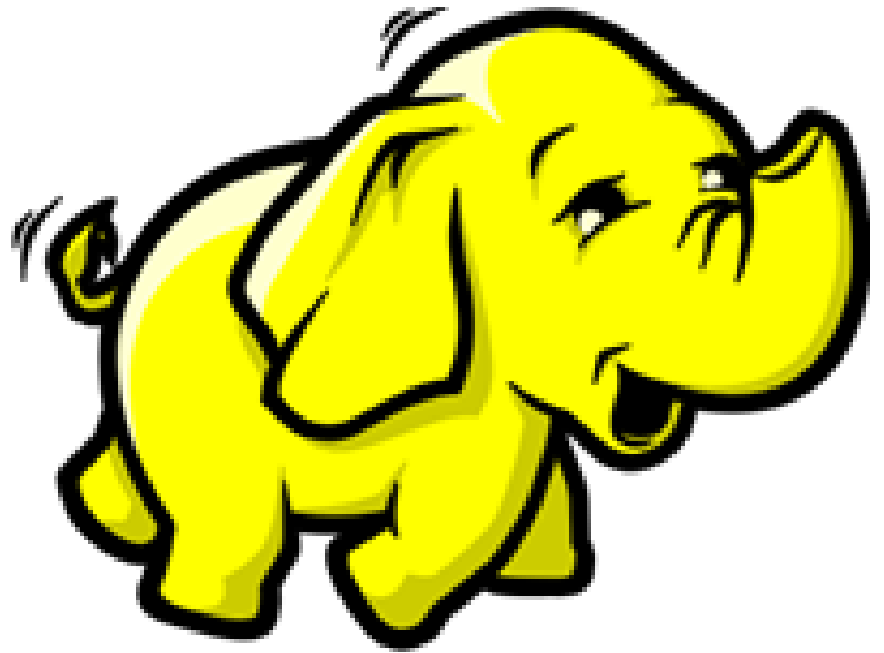


Hadoop/Hbase

Joseph ELIA

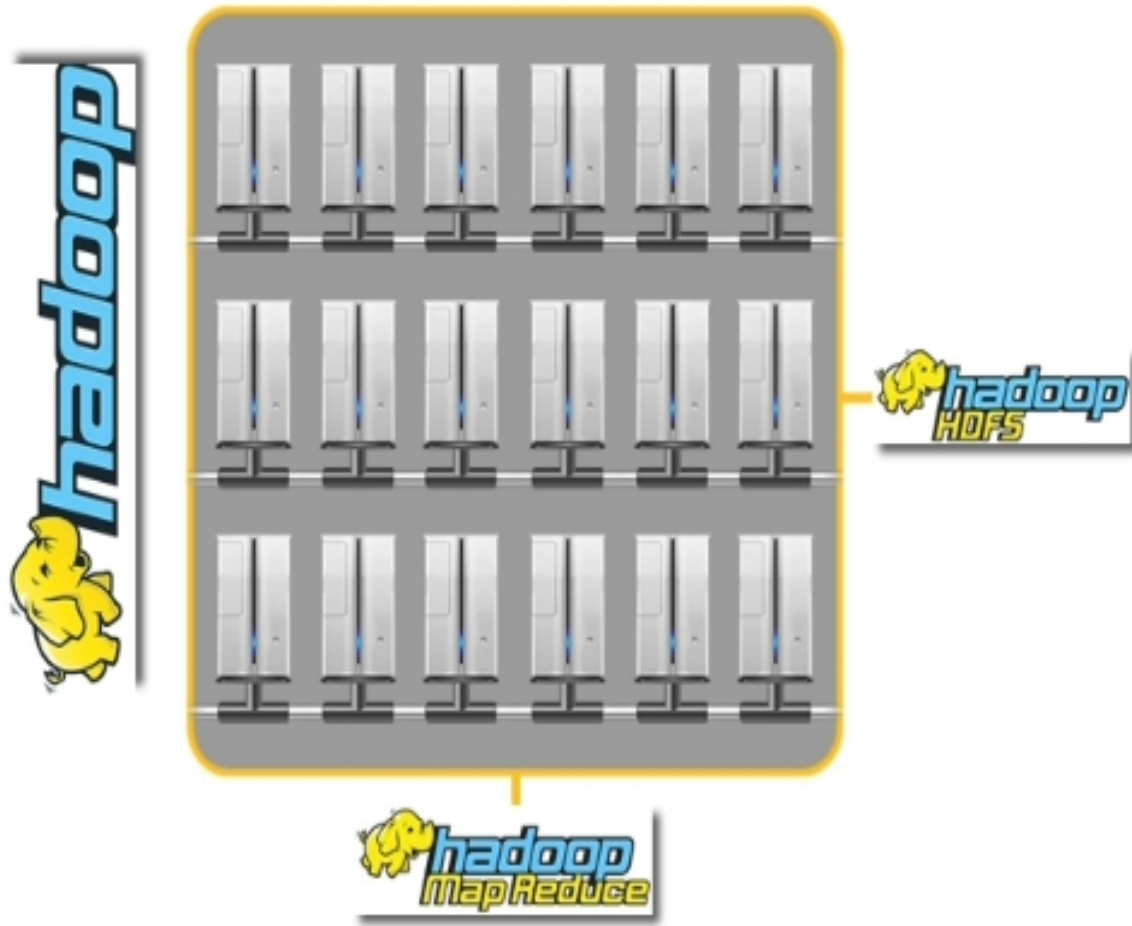
C'est quoi Hadoop?

- Système distribué pour le stockage et le traitement de grands ensembles de données.
- Traitement Batch
- Basé sur le système de fichiers de Google (GFS)
- Sous licence de Apache

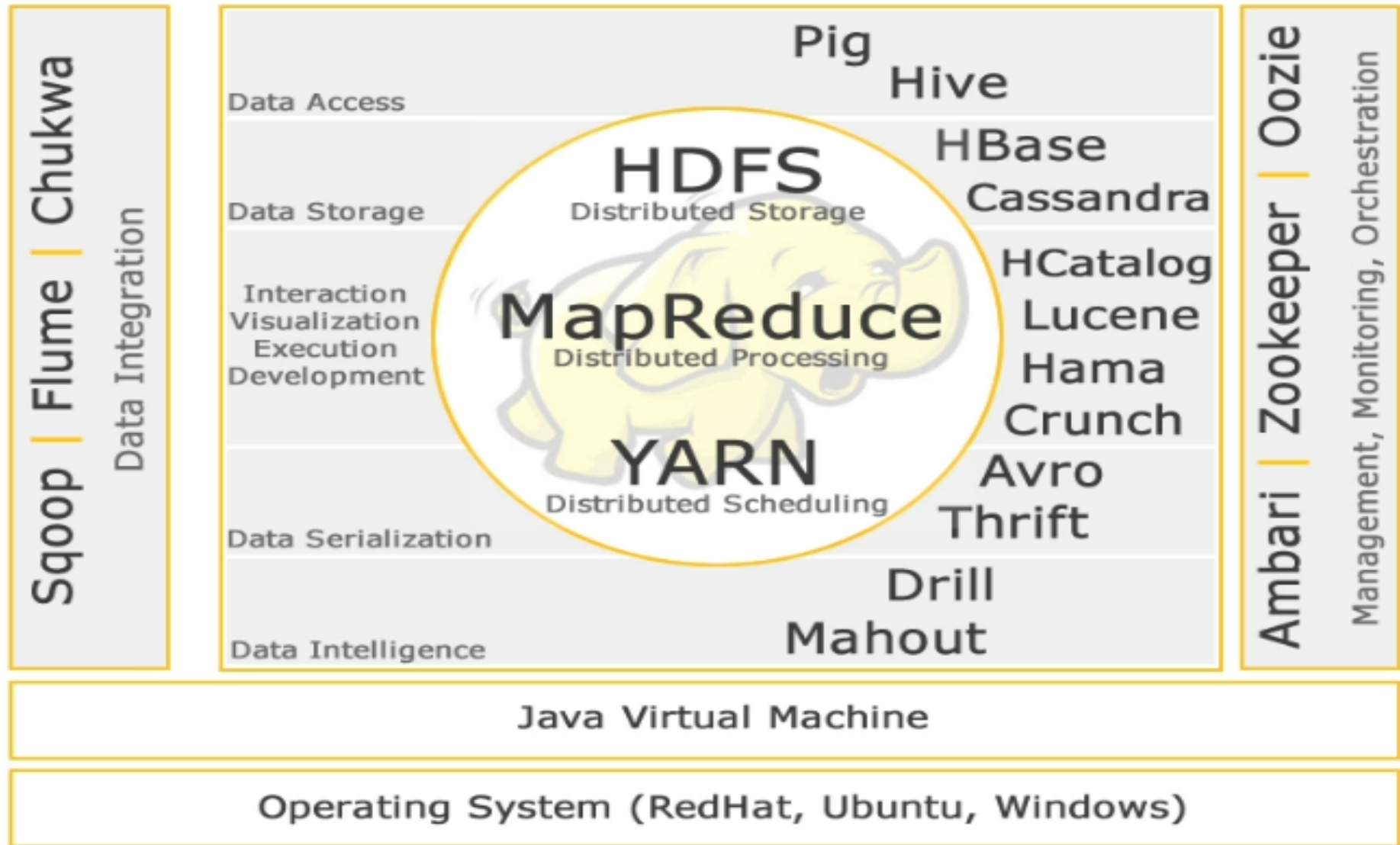


L'architecture de Hadoop

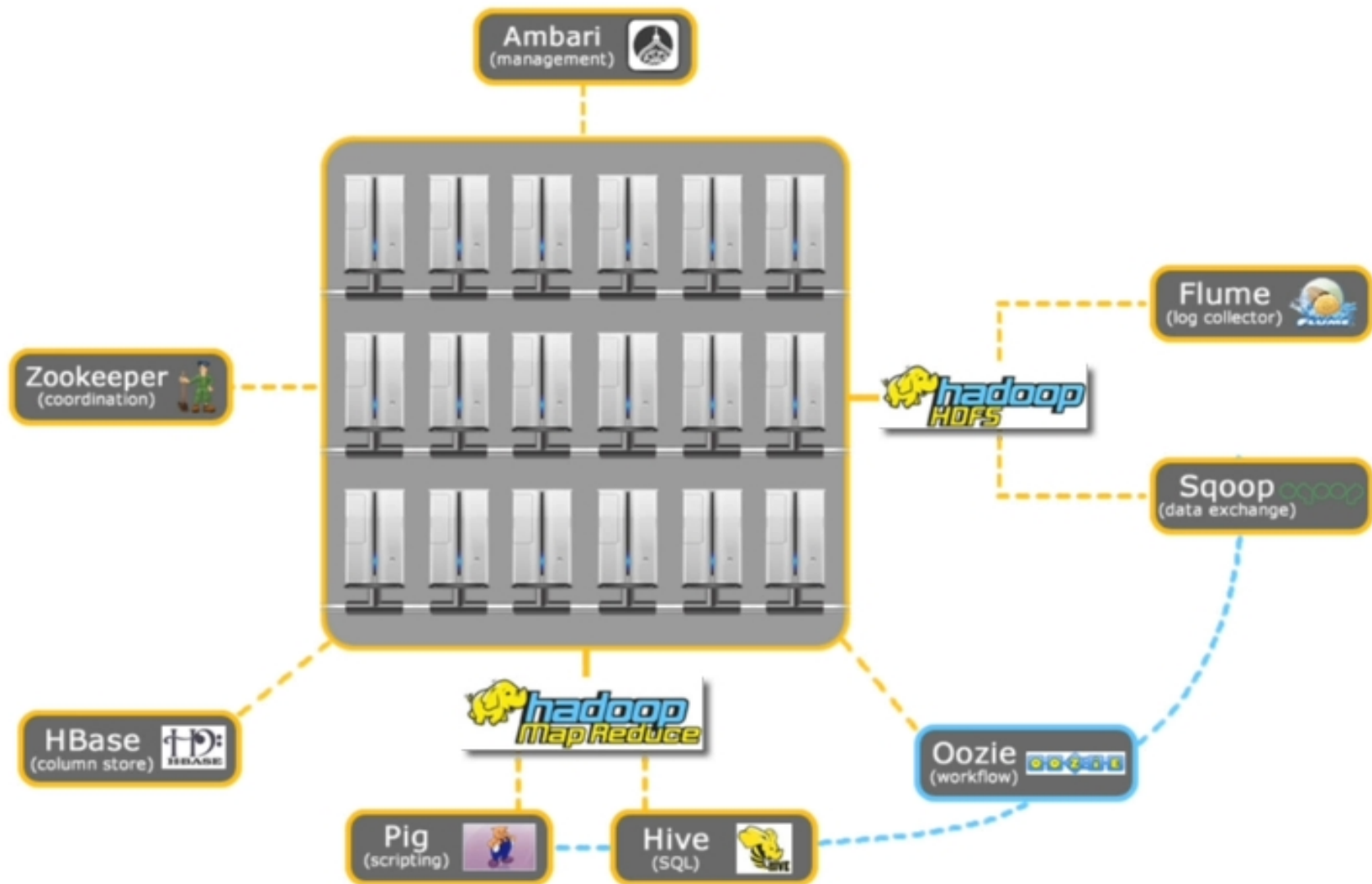
- Évolutif
- Flexible
- Tolérance aux pannes
- Intelligent



L'écosystème de Hadoop

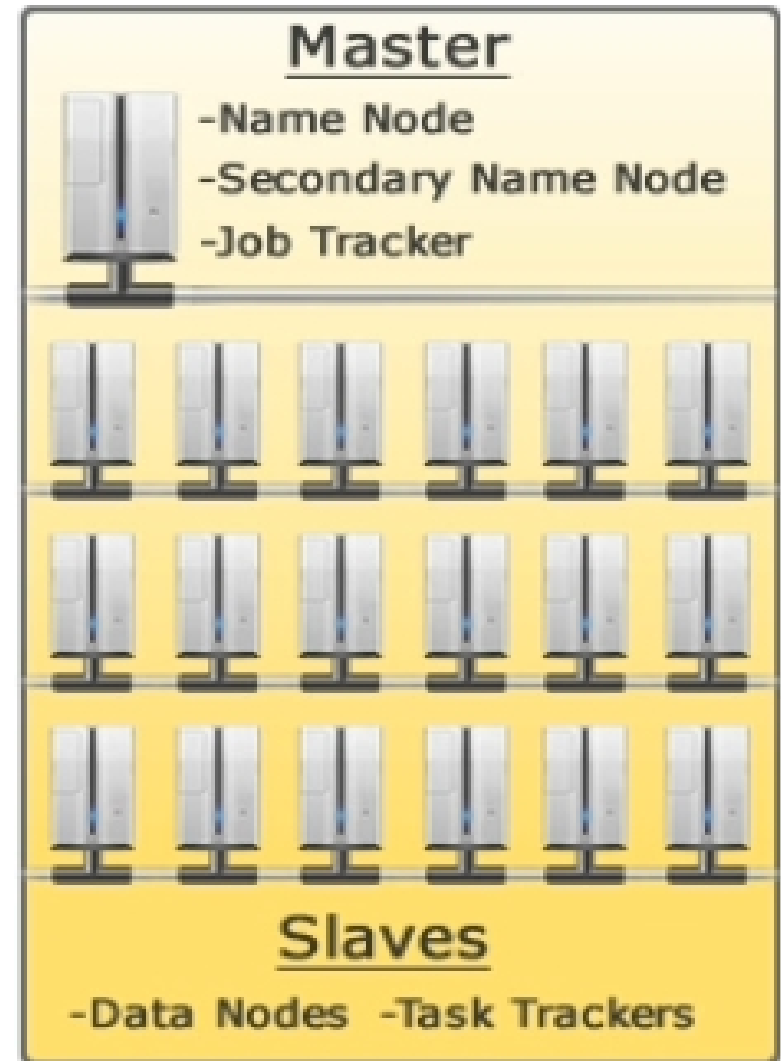


L'implémentation de Hadoop Stack



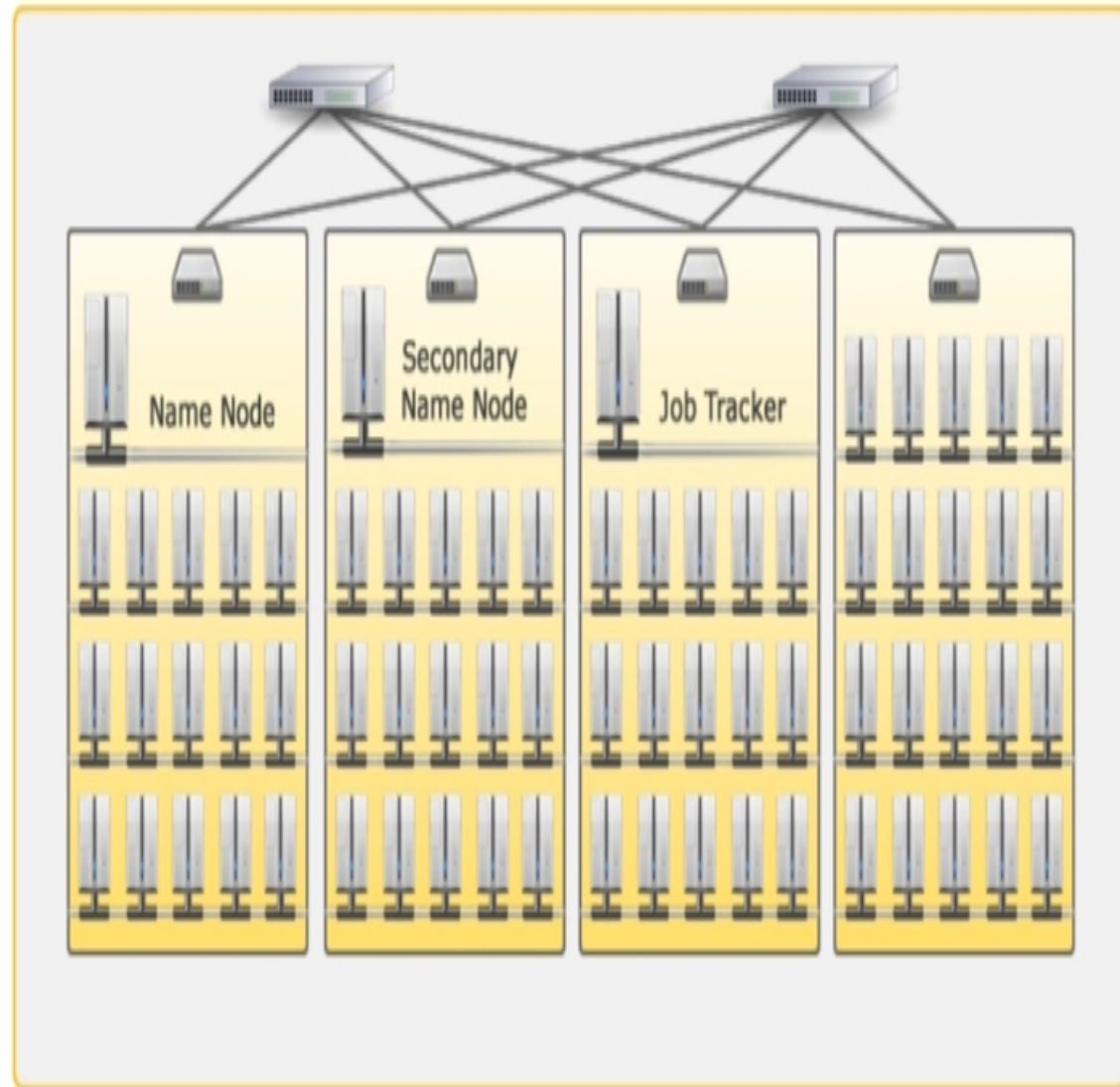
L'Architecture HDFS

- Architecture Master/Slave
- les opérations du système de fichiers sont gérés par le NameNode
- Les DataNodes effectuent toute les opérations des blocks
- Secondary NameNode: system restore pour le NameNode



HDFS Multi-Rack

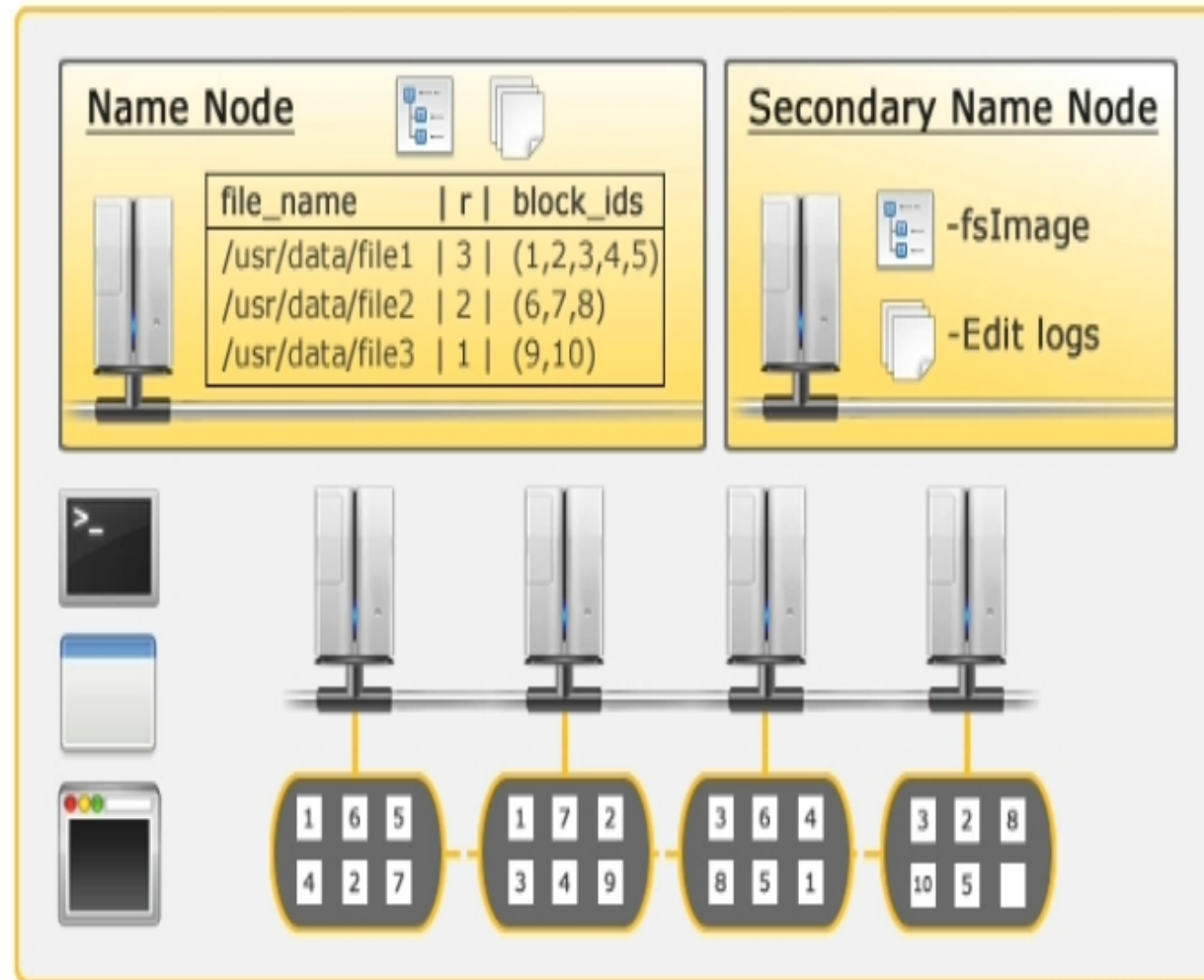
- Prévention des pertes de données grâce à la fonctionnalité de 'Rack-awareness'
- Gain de bande passante grâce à la fonctionnalité de 'Data locality'



L'Intérieur de HDFS

NameNode:

- Contient les métadonnées du système de fichiers
- Un point de défaillance unique
- Surveille la santé des datanodes



Secondary NameNode

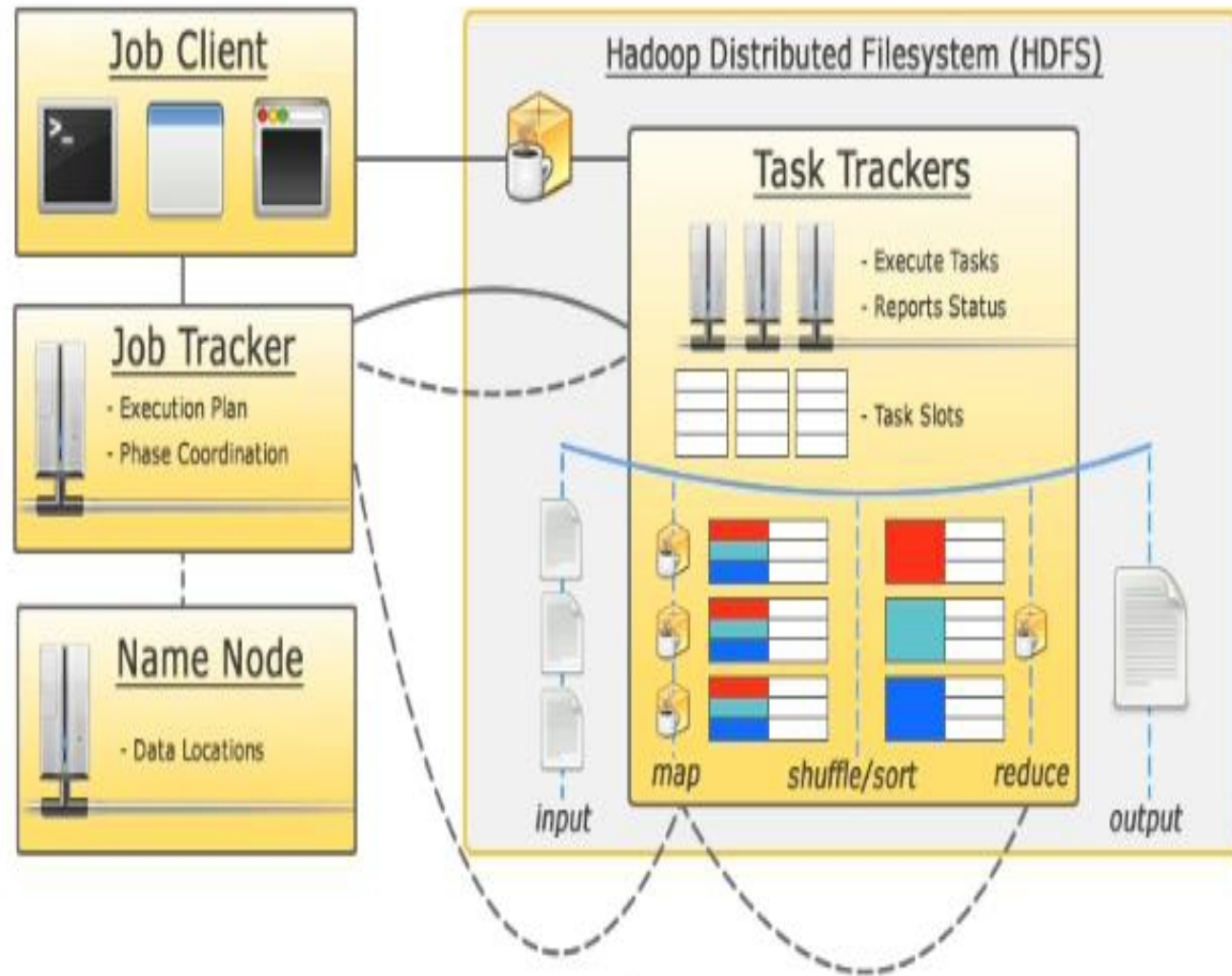
- Peut reconstruire le namenode
- Pas un serveur haute disponibilité
- Joue le rôle de HouseKeeper

Datanode

- Traite les demandes des clients
- Envoie des messages HeartBeat à namenode toutes les 3 secondes

L'Architecture MapReduce:

- Un framework responsable pour le stockage et le traitement des données
- Composé de trois composantes: «Job Client», «JOB Tracker» et «Task Trackers».




JOB Client: soumet les MapReduce Jobs au Job Tracker
(fonction Mapper et Reducer).

JOB Tracker:

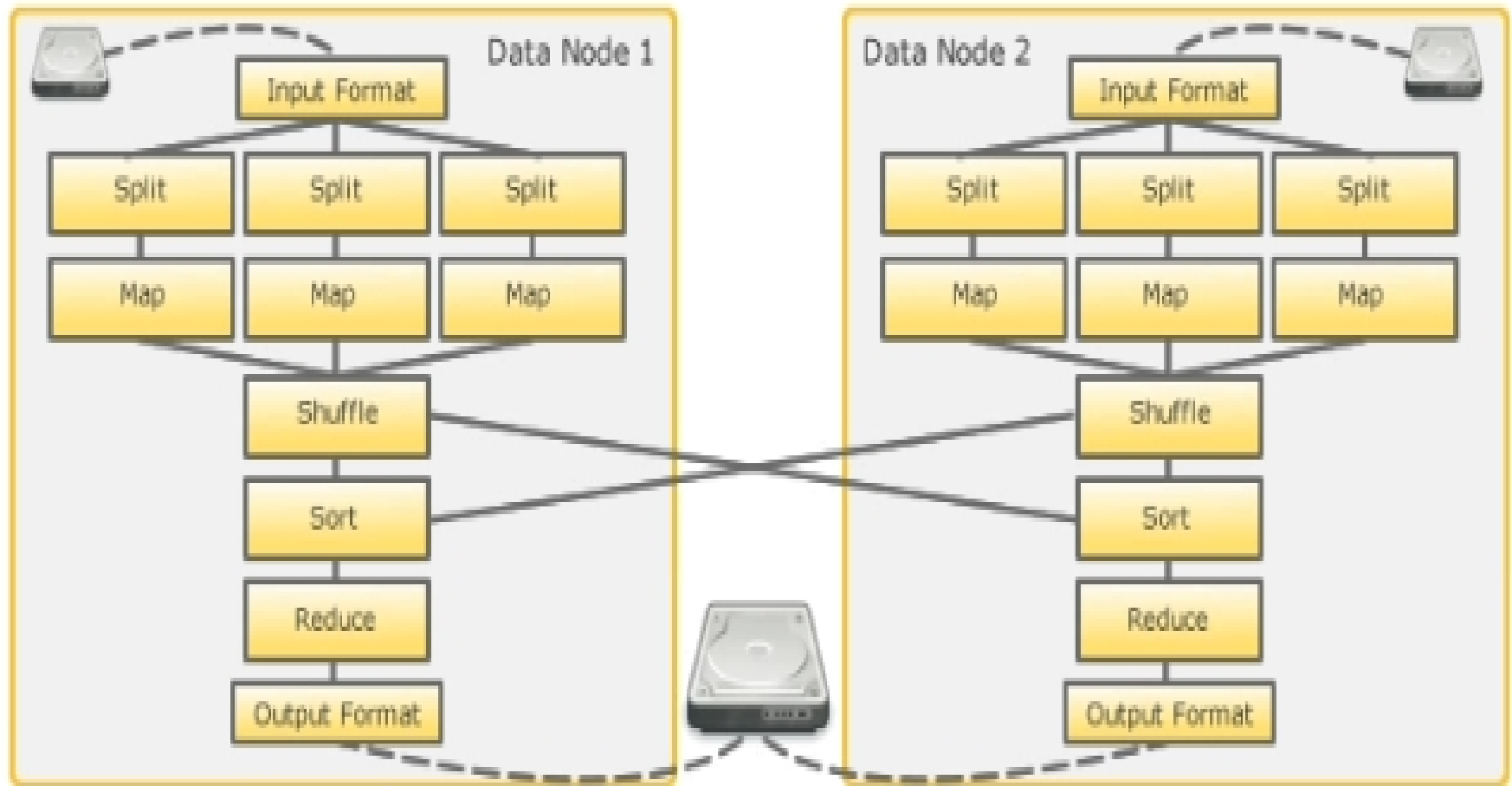
- Interroge le NameNode.
- Crée un plan d'exécution.
- Soumet des travaux au TaskTracker .
- Gère les phases de mapreduce.


TASK Tracker:

- Exécute les travaux fournis par le Job Tracker via les message HearBeat.
- Rapporte les progrès au Job Traccker.



Les Phases de MapReduce



1. **Input Format:** Le format de données défini pour être exécuté
 2. **Split:** diviser le input format puis l'envoie au Mapper
 3. **Map:** Transforme les divisions en paires clé/valeur
 4. **Shuffle/Sort:** Trie les output Map en fonction du clé
 5. **Reducer:** Regroupe les paires clé/valeur basées sur le code défini.
 6. **Output Format:** Détermine comment les résultats sont écrits dans le répertoire de sortie
- 

Hbase

- Base de données orientée-colonne distribuée
- Construit sur le dessus de HDFS
- Conçu pour les requêtes à faible latence et l'accès en temps réel
- Base de données NoSQL
- Écrit en Java



Orientées lignes vs Orientées Colonnes

Orientées lignes	Orientées Colonnes
OLTP (On-line Transaction Processing)	OLAP (On-line Analytical Processing)
Construit pour un petit nombre de colonnes / lignes	Taux de compression élevé en raison de peu de valeurs distinctes
Les données sont stockées en rangée de tous les champs	Les données sont stockées comme colonne de toutes les valeurs d'un champ

	Customer	Product	Amount
Row 1	Jane	TV	1000.00
Row 2	John	Clock	200.00
Row 3	Susan	PC	1000.00

Orientées lignes

Row 1	Jane	TV	1000.00
Row 2	John	Clock	200.00
Row 3	Susan	PC	1000.00

Orientées Colonnes

Customer	Jane	John	Susan
Product	TV	Clock	PC
Amount	1000.00	200.00	1000.00

L'Architecture de HBase

HBase

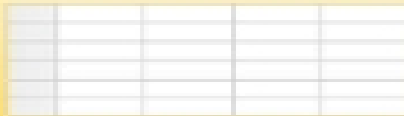
HMaster



- Manages & monitors cluster ops
- Assigns regions to region servers
- Load balancing & splitting

HRegionServer

- Hosts and manages regions



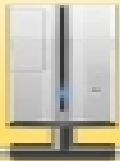
HRegionServer

- Automatically splits regions



HRegionServer

- Handles read/write requests



HRegionServer

- Clients communicate directly




Hadoop / HDFS



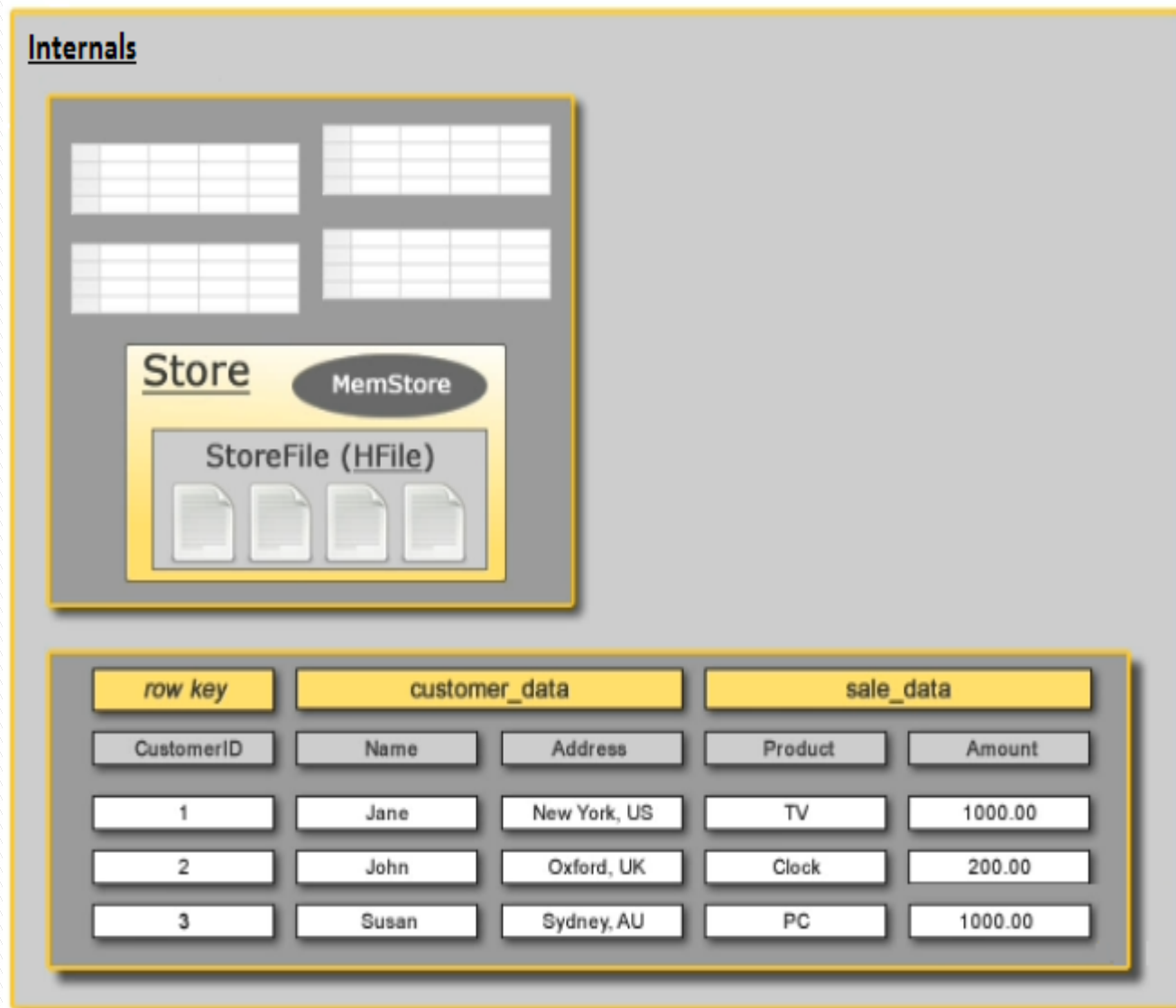
Master Server:

- Comparable au Namenode dans HDFS.
- Gère et surveille les opérations de cluster Hbase.
- Responsable de Load Balancing et de diviser les données.

Region Server:

- Comparable au DataNode dans HDFS.
 - Gère toutes les requêtes de lecture / écriture reçues des clients.
 - La communication entre le Master Server et les RegionServers passe par ZooKeeper.
- 

L'Intérieur de HBASE



Fin