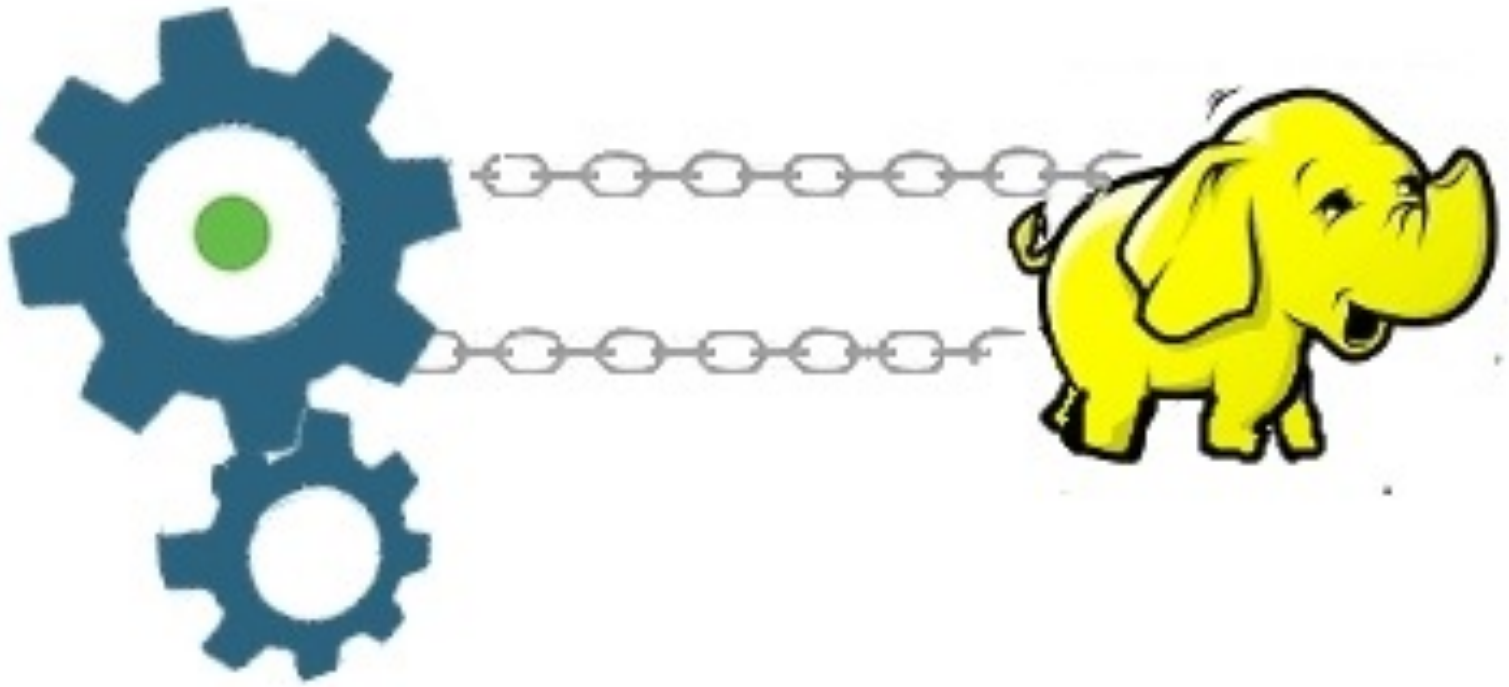
HADOOP



HADOOP Architecture

- HADOOP est une plateforme open-source basée sur une architecture lambda offrant des fonctionnalités de calcul distribué, d'extensibilité et de tolérance aux pannes.
- Il est écrit en java mais il supporte différents langages C, C++, Python …
- Il a été conçu pour gérer de gros volumes de données (Peta octets et même Hexa octets).
- HADOOP est très économique, de gros volumes de données peuvent être traités à partir de matériel populaire.
- Selon la loi de Moore, n'importe quel matériel peut tomber en panne et si il le peut alors il tombera en panne. HADOOP intègre des mécanismes de haute disponibilité pour garantir le service.
- HADOOP est un « cluster » de machines où le nombre de machines n'a pas de limite.
- HADOOP a séduit de nombreuses entreprises et communautés.
- De nombreux logiciels vont naître autour de cette plateforme et diversifier les traitements possibles.

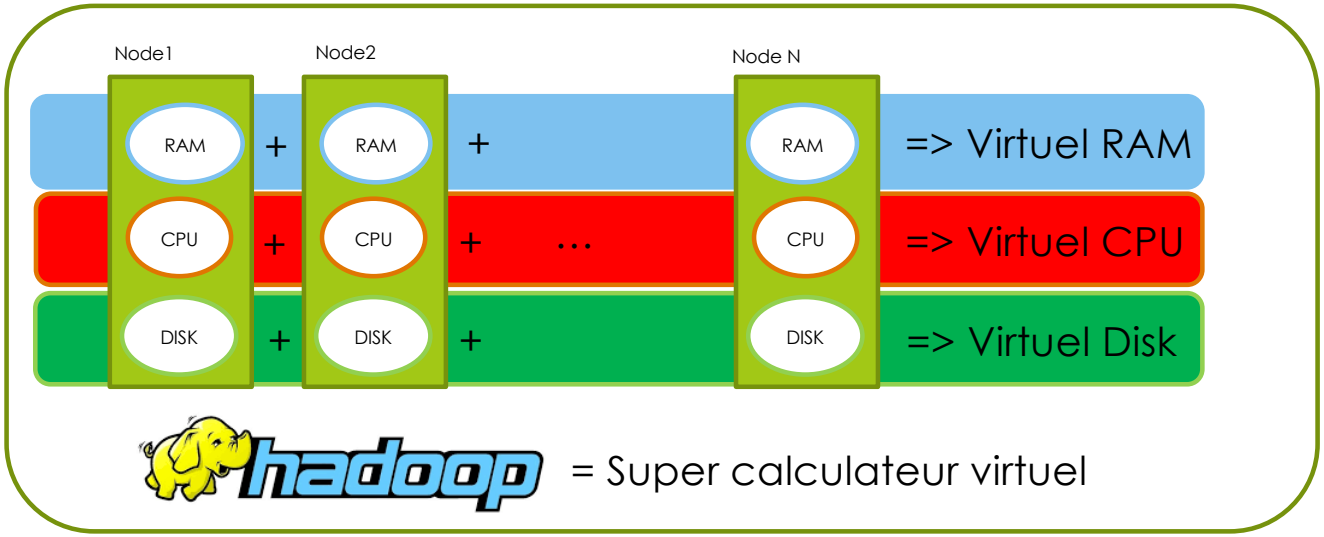
Les distributions HADOOP

- Officiellement, HADOOP est la distribution offerte par la fondation Apache.

- Plusieurs compagnies et organisations vendent leur propre distribution basée sur la distribution Apache
  - Amazon Web Services : Amazon utilise HADOOP avec leur solution de stockage EC2.
  - Cloudera et Hortonworks : Des compagnies qui participent activement au développement de l'éco-système HADOOP.
  - BigInsights Enterprise Edition, solution fournie par IBM
  - Pivotal HD, distribution implémentée par Pivotal pour les architectures Kappa.
  - Serengeti, solution développé par VMWare
  - …

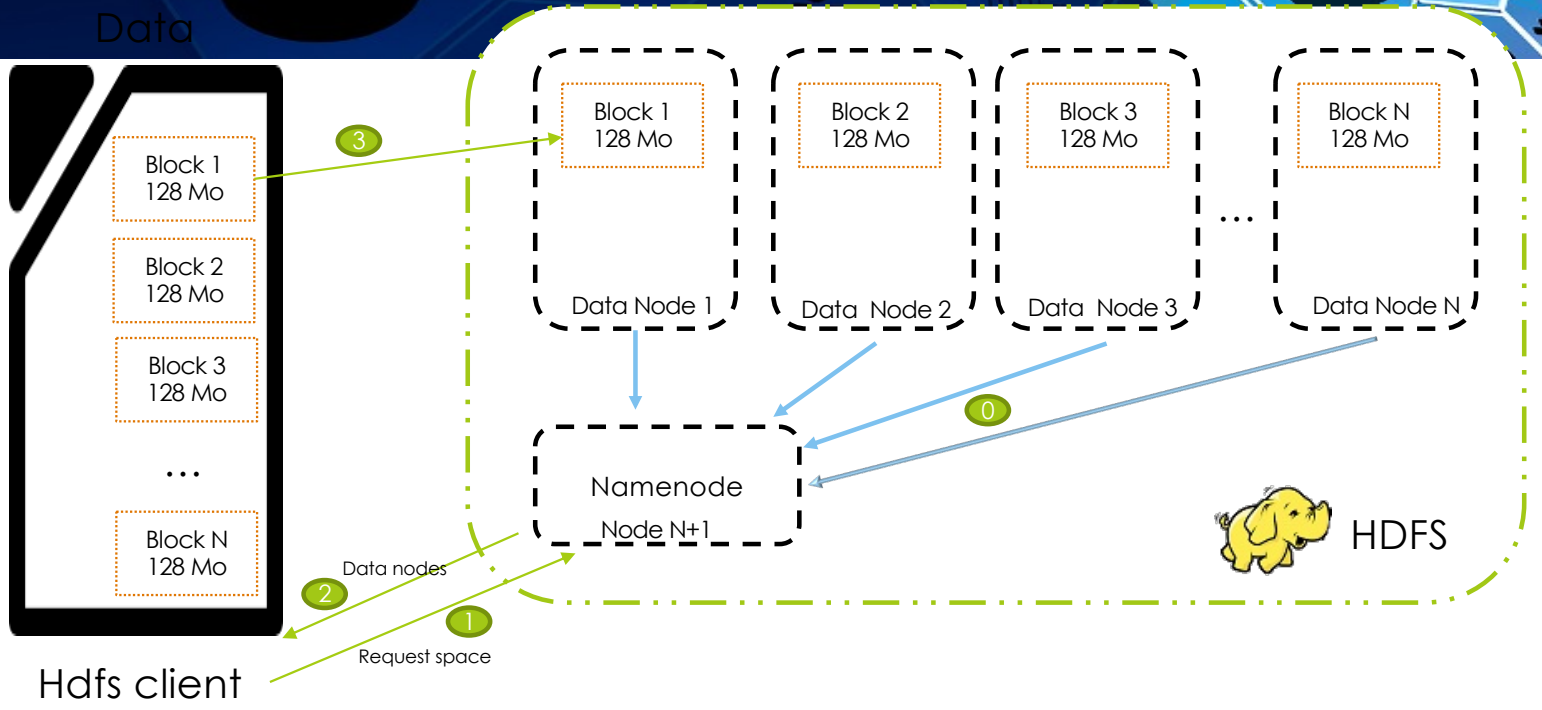
HADOOP architecture

- Pour traiter de gros volumes de données, HADOOP virtualise l'ensemble des ressources de calcul, de mémoire et de stockage de plusieurs machines en un super calculateur virtuel.



- A partir de 2.0, HADOOP s'appuie sur 3 composants.
  - Un stockage distribué HDFS
  - Un moteur de calcul distribué
  - Un gestionnaire de ressource

HADOOP Distributed File System HDFS : exemple écriture d'un fichier



0 Au démarrage, tous les Datanodes notifient le Namenode de la liste des blocks qu'ils contiennent.

1 Pour charger des données dans HDFS, le client contacte le Namenode et lui demande de créer un fichier dans l'arborescence du système de fichiers.

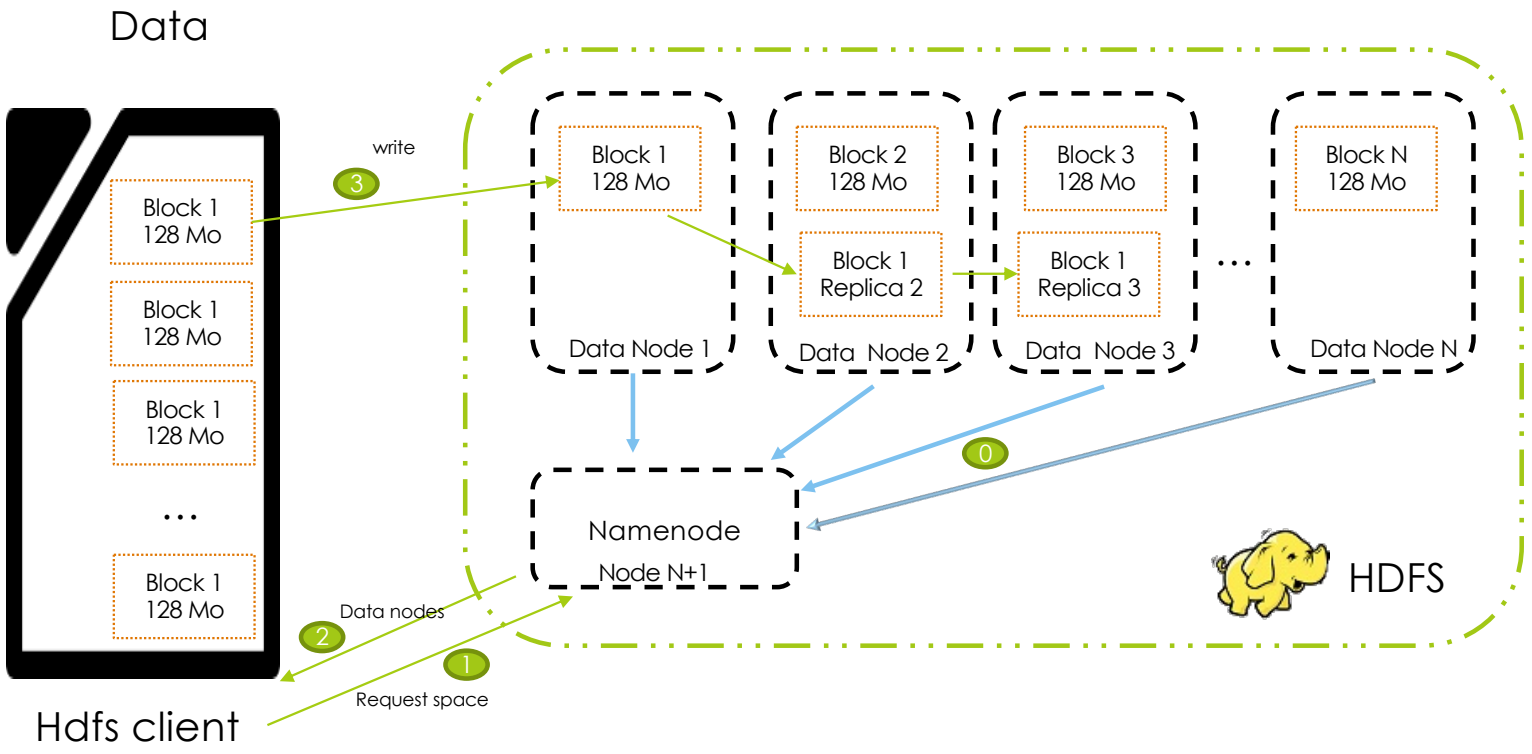
2 Le Namenode alloue un bloc de données (ex 128 Mo) et retourne au client une liste de blocs disponibles sur lesquels les données peuvent être stockées.

3 Le client contact le Datanode recommandé par le Namenode et lui envoie ses données.

- Si le fichier de données est plus grand que 128Mo alors le client HDFS renouvellera une nouvelle demande de bloc au Namenode et procédera à l'écriture comme décrit précédemment.

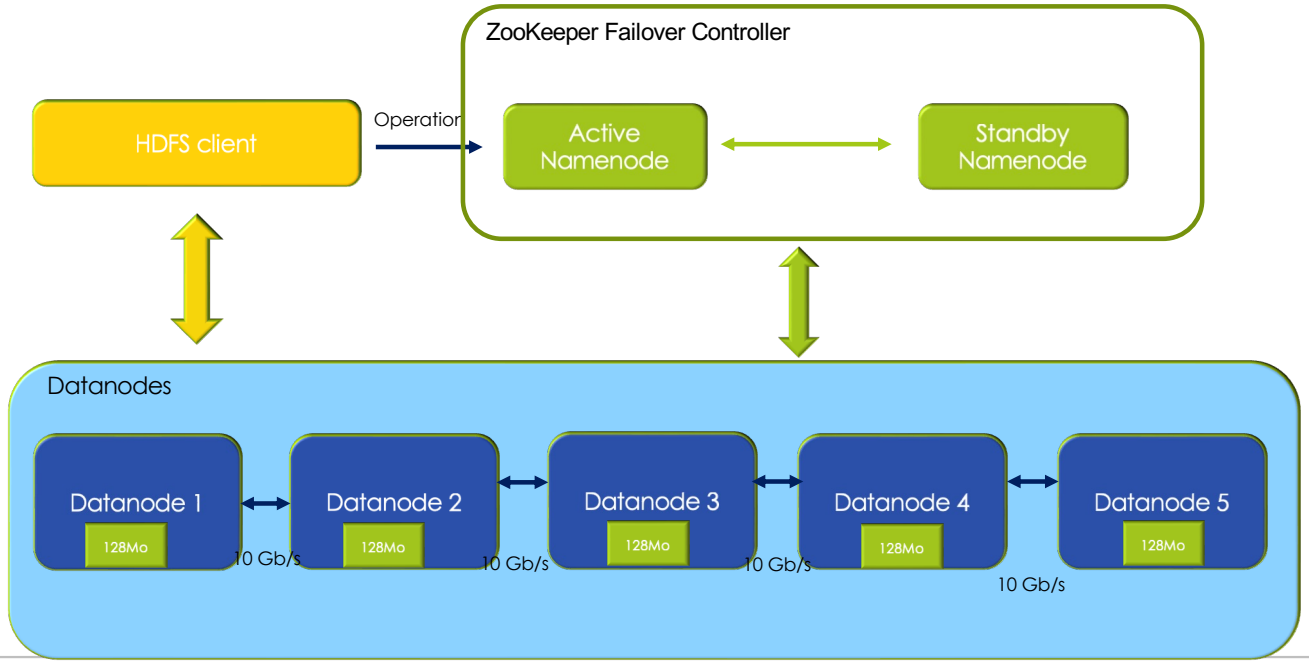
HADOOP Distributed File System HDFS : exemple écriture d'un fichier

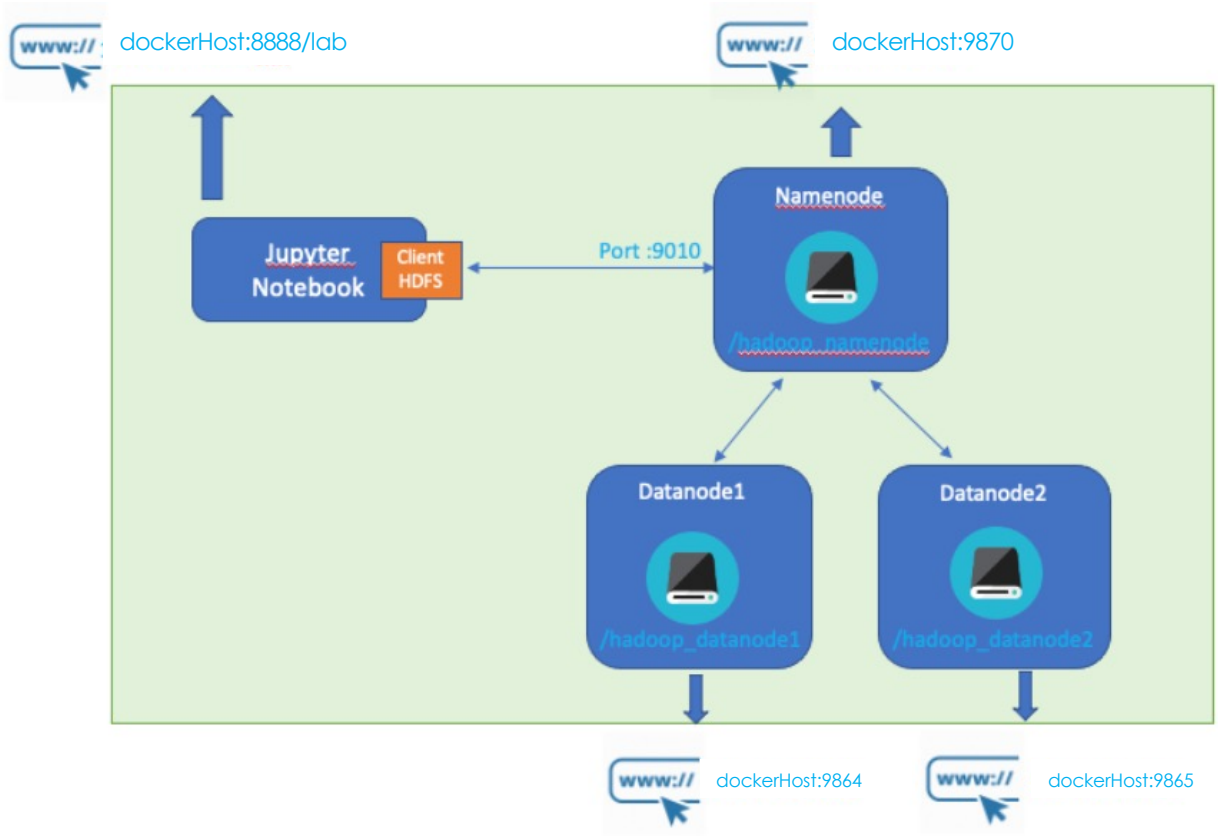
- Pour assurer la disponibilité des données HDFS permet de répliquer les blocs de données entre les Datanodes (par défaut 3).
- La réplication des données s'opèrent lorsque que la première copie du bloc de données est envoyé sur le premier Datanode. Chaque Datanode qui stockeront un réplica, liste définie par le Namenode, recevront le bloc à partir du Datanode précédent.



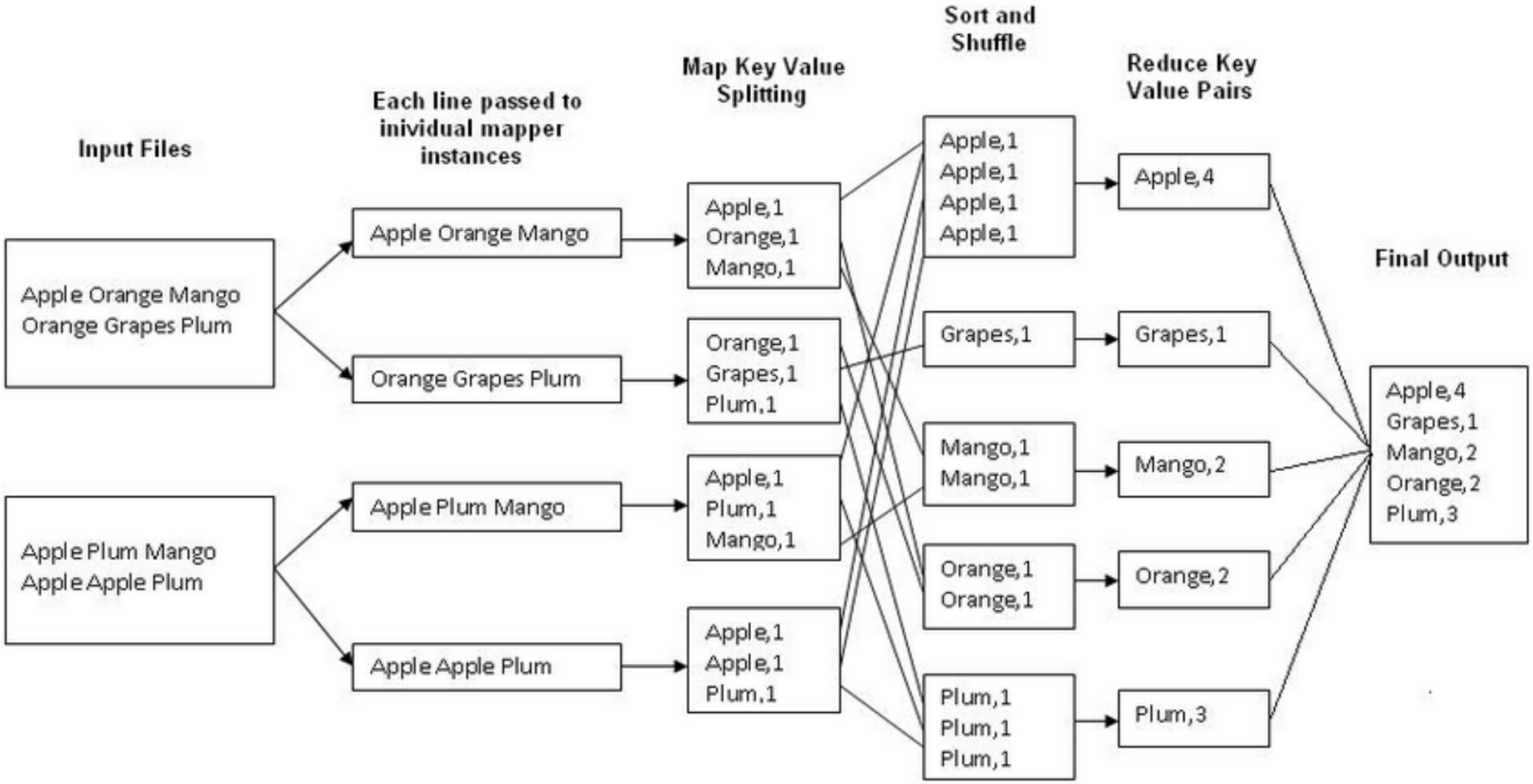
HADOOP Distributed File System HDFS : Haute disponibilité

- Toutes les écritures dans HDFS passent par le Namenode.
- Si le Namenode tombe en panne, nous perdons l'accés aux données aussi bien en écriture qu'en lecture.
- Dans HADOOP 2.0, il est possible d'exécuter 2 Namenodes simultanément pour fournir la haute disponibilité, avec un actif et le second en passif.
- La solution de haute disponibilité repose sur Apache ZooKeeper. Zookeeper a le rôle d'un coordinateur qui supervise l'état du Namenode actif et gère en cas de panne le mécanisme de basculement vers le Namenode passif.



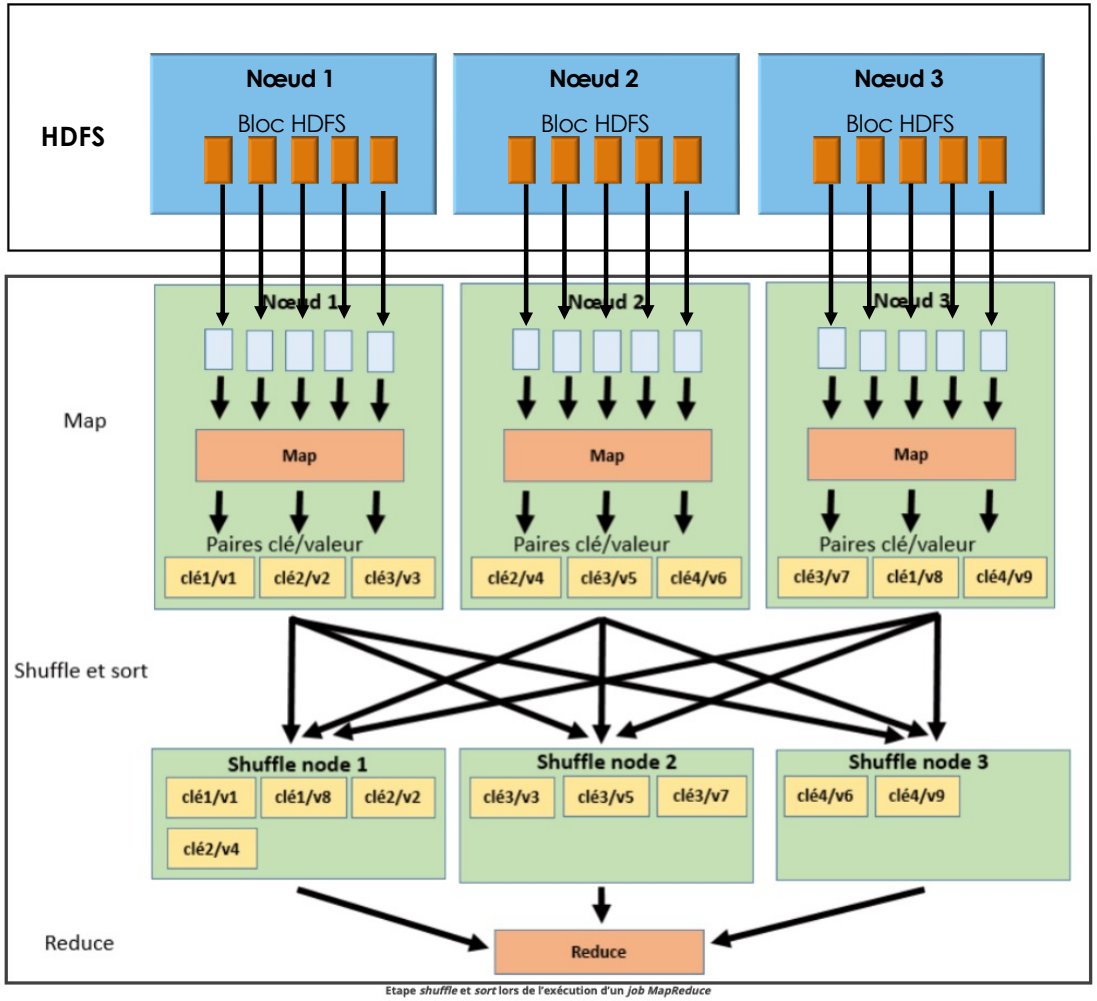

www://
dockerHost:8888/lab
www://
dockerHost:9870
Jupyter
Notebook
Client
HDFS
Port :9010
Namenode
/hadoop_namenode
Datanode1
/hadoop_datanode1
Datanode2
/hadoop_datanode2
www://
dockerHost:9864
www://
dockerHost:9865

Moteur d'exécution : MapReduce

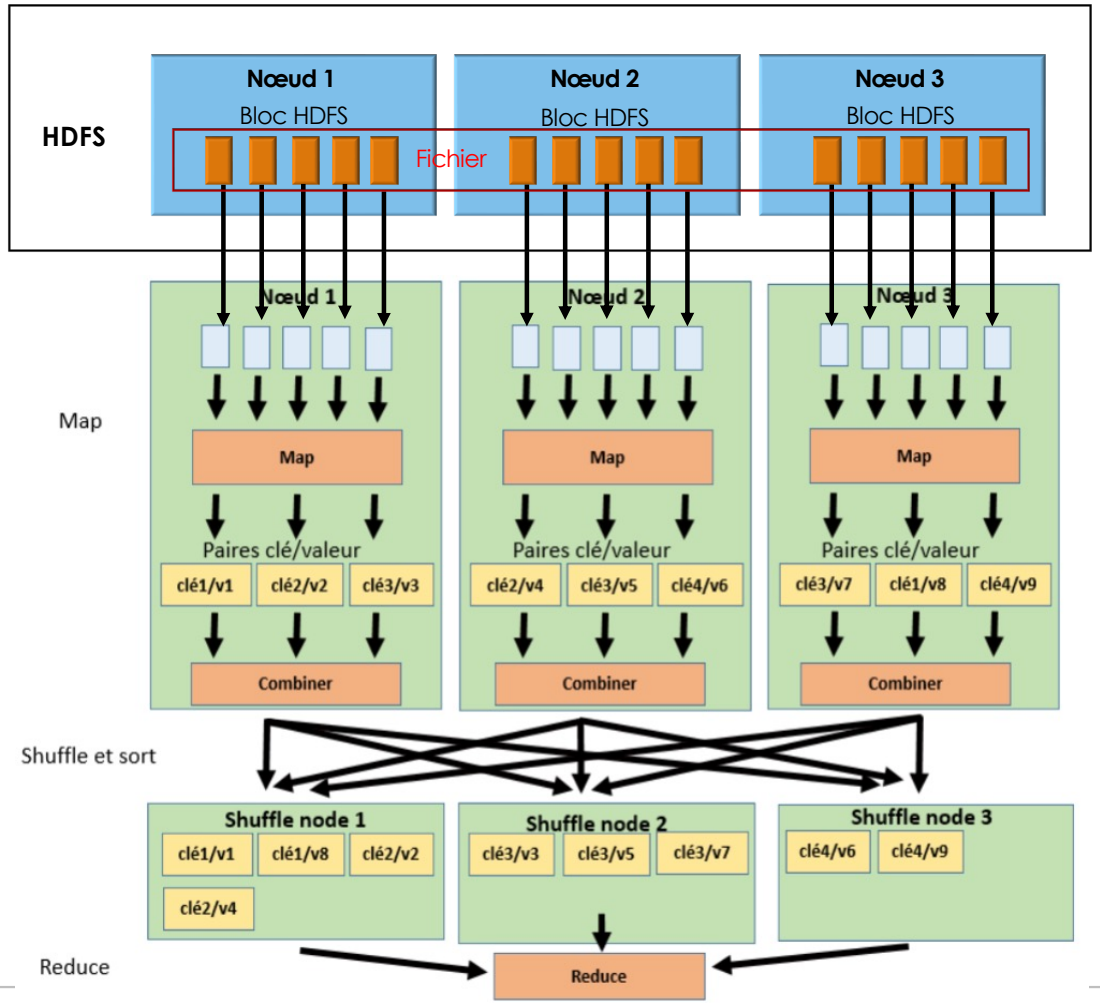


Source : http://javax4u.blogspot.fr/2012/11/hadoop.html

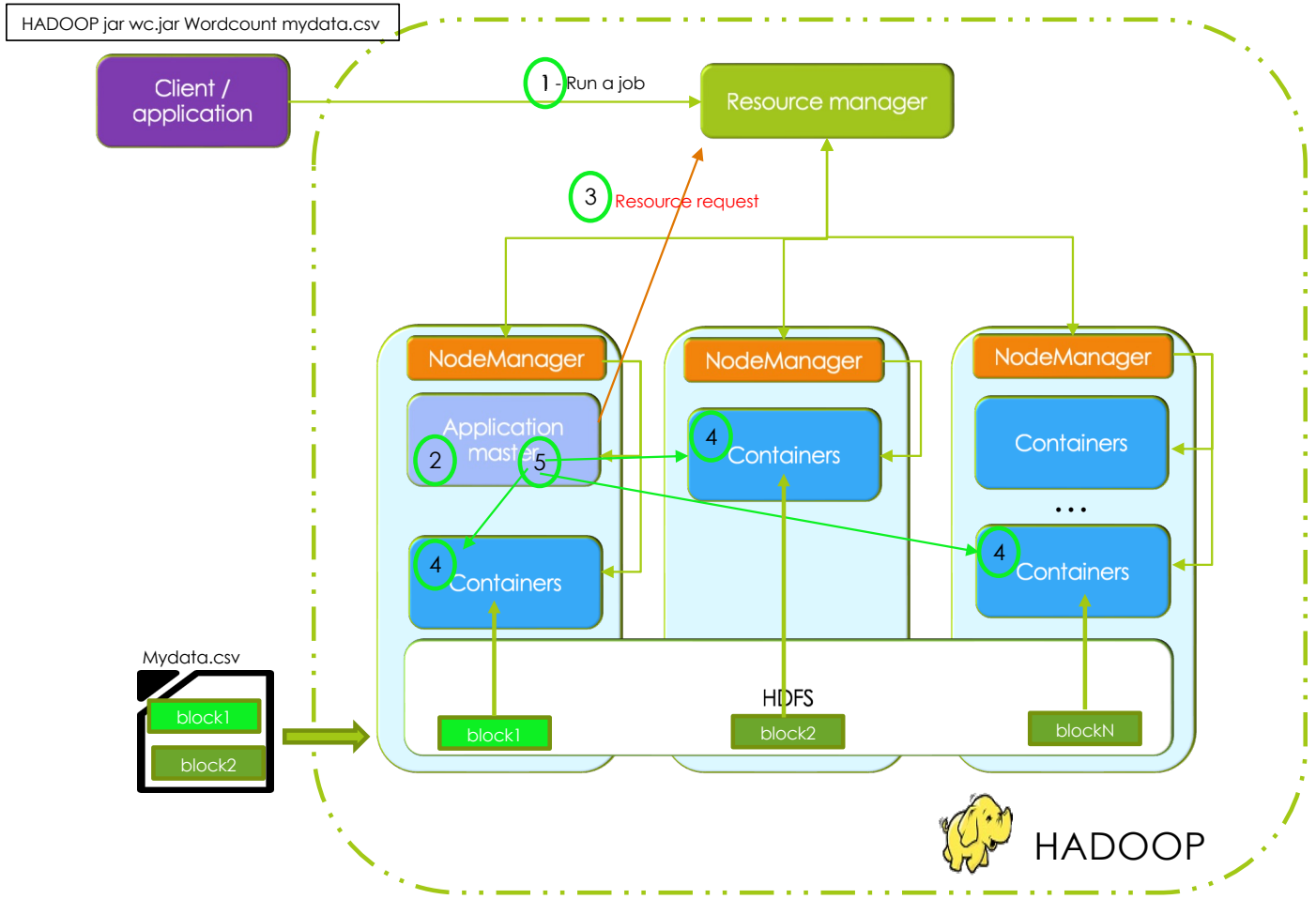
Moteur d'exécution : MapReduce v1



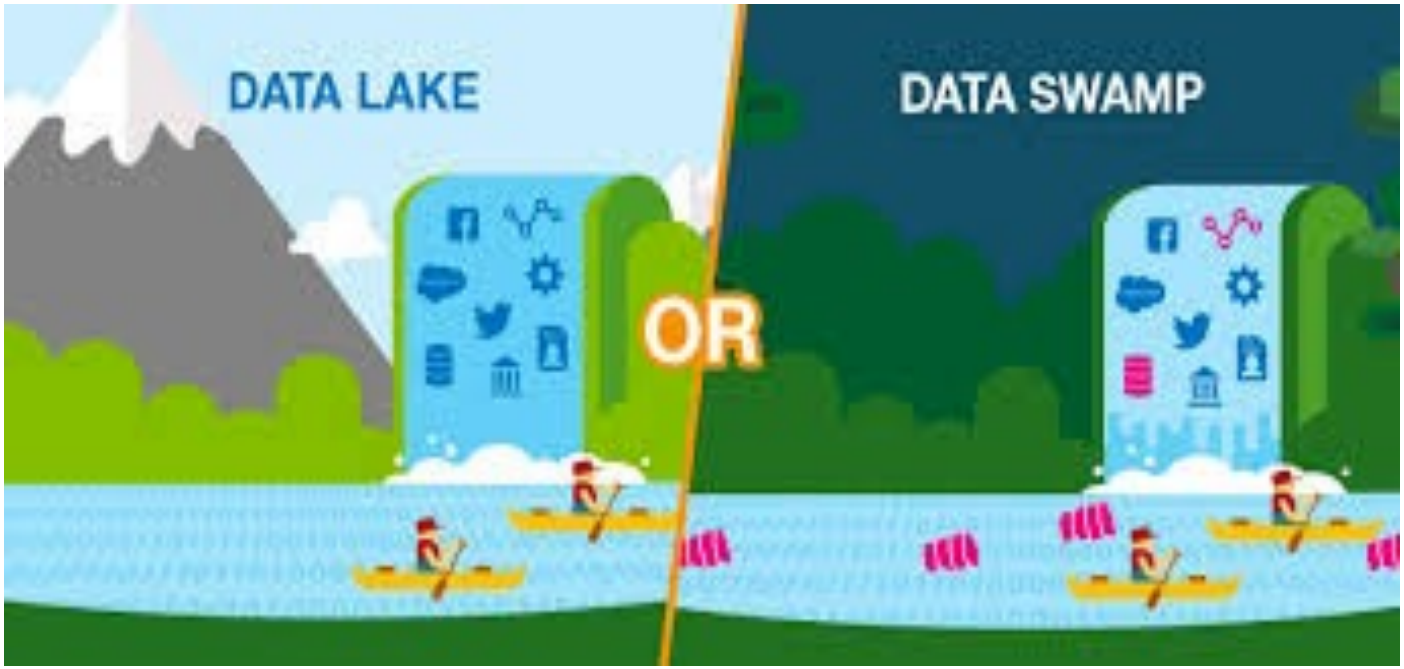
Etape *shuffle* et *sort* lors de l'exécution d'un *job MapReduce*


HDFS
Nœud 1
Bloc HDFS
Nœud 2
Bloc HDFS
Nœud 3
Bloc HDFS
Fichier
Map
Nœud 1
Nœud 2
Nœud 3
Map
Map
Map
Paires clé/valeur
clé1/v1
clé2/v2
clé3/v3
Paires clé/valeur
clé2/v4
clé3/v5
clé4/v6
Paires clé/valeur
clé3/v7
clé1/v8
clé4/v9
Combiner
Combiner
Combiner
Shuffle et sort
Shuffle node 1
clé1/v1
clé1/v8
clé2/v2
clé2/v4
Shuffle node 2
clé3/v3
clé3/v5
clé3/v7
Shuffle node 3
clé4/v6
clé4/v9
Reduce
Reduce

Gestionnaire de ressources: YARN

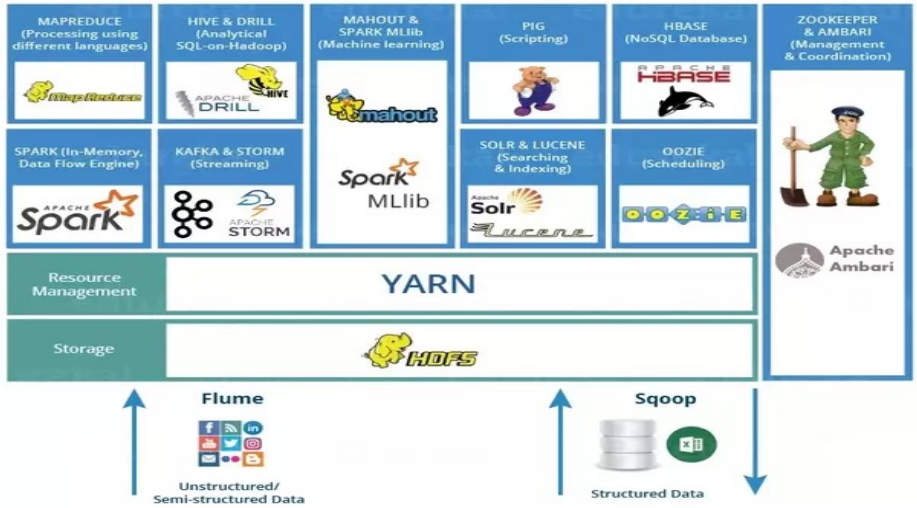


L'écosystème HADOOP

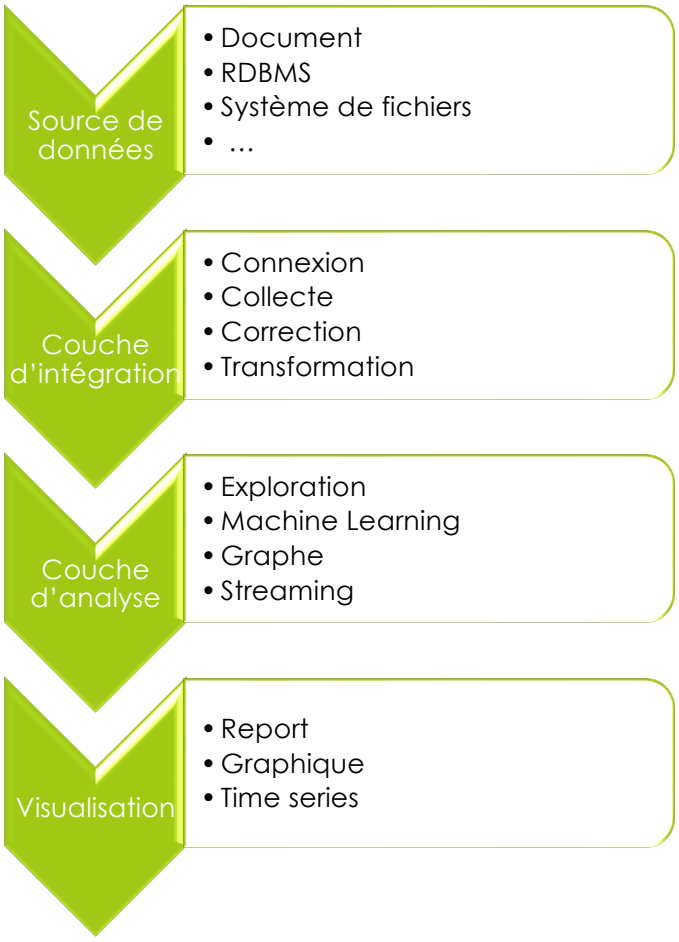


L'écosystème HADOOP

- Grâce à l'architecture flexible de HADOOP, il est possible de supporter une large variétée d'application via la containerisation.
- De nombreux projets naissent autour de HADOOP et font partis de l'eco-système HADOOP.
- Vous trouverez ci-dessous une architecture typique que l'on peut retrouver dans de nombreuses entreprises (Hortonworks / Cloudera, mapR).

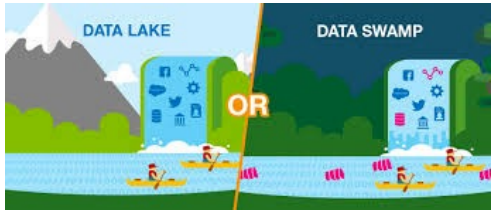


Traitement de données

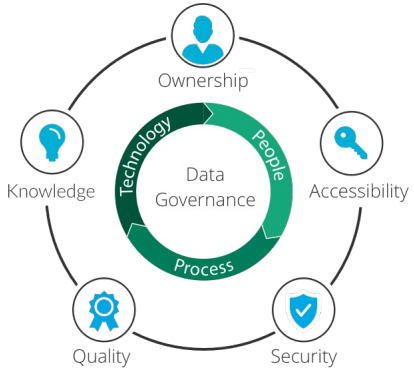


La gouvernance de données

- Les données sont difficilement exploitables sans organisation.

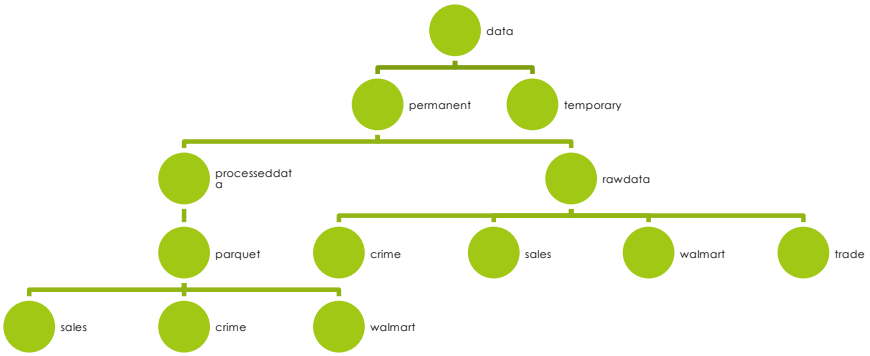


- Mise en place d'une gouvernance de données :
  - normes, principes et règles régissant divers types de données.
    - Médicale
    - bancaire
  - une stratégie d'entreprise pour gérer
    - les données
    - les flux
    - les accès
    - le stockage
    - les mises à jour



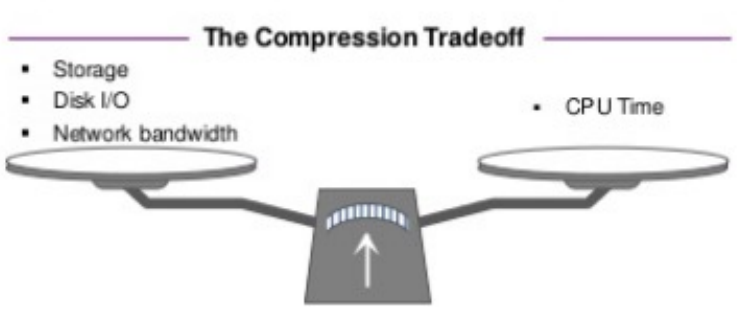
Comment organiser vos données ?

- Où se trouvent vos données ?
- Quels types de données possédez-vous ?
- Qu'arrive-t-il à vos données ?
- Vos données sont-elles exactes et sûres ?
- Exemple d'organisation pour les TP



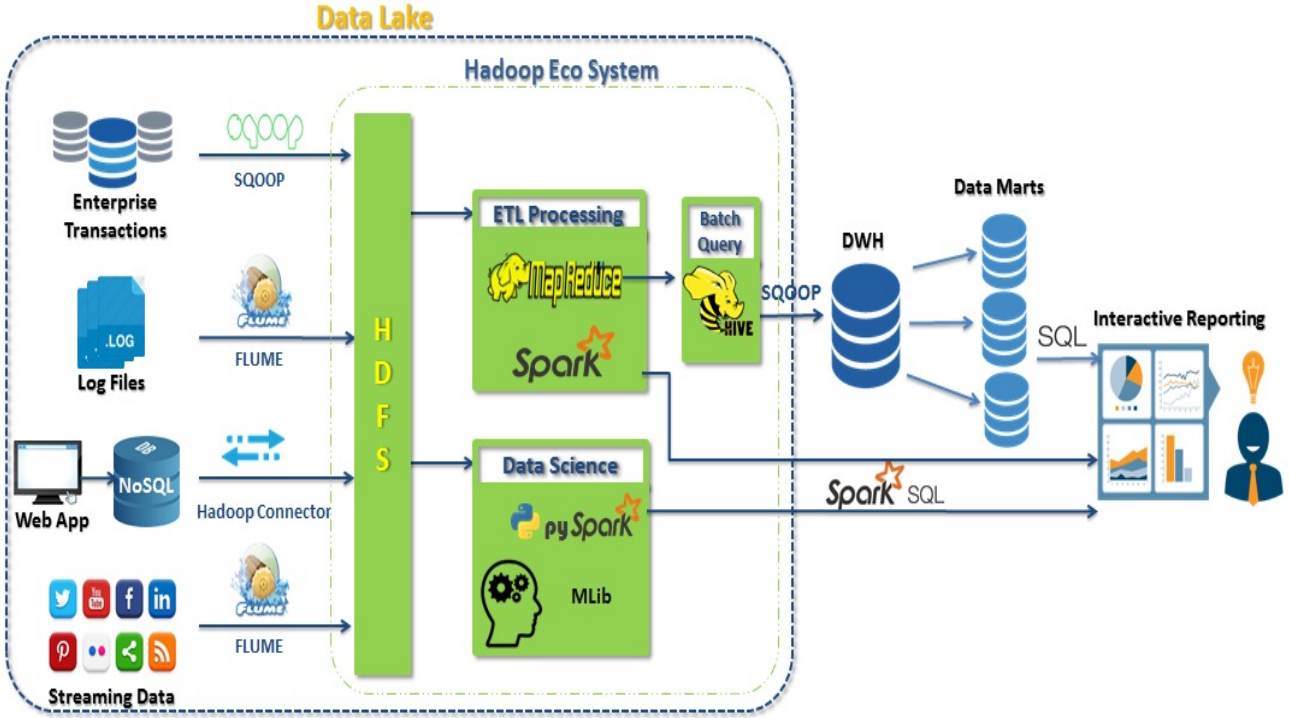
Les formats de fichiers

- Fichier non compressé : csv, texte, json
  - Volumineux en espace
  - Sécable en partition
- Fichier compressé
  - Gain en espace
  - Pas forcément sécable
- Fichier binaire
  - Produit d'une sérialisation et sécable : avro parquet



DATA LAKE by HADOOP

- Voici un exemple, d'une architecture Lambda avec un Data Lake basée sur l'eco-système HADOOP.



Conclusion


BIG DATA is GOING TO IMPACT US SOON !
CALL BRUCE WILLIS !!
DATA
www.nuxeo.com
Soulcié