

# Conférence Thématique



## *Big Data, Hadoop & Spark*

Master 2 SID – Big Data  
Université Toulouse III – Paul Sabatier

12 Octobre 2017

# ekito



Alexia Audevert

Data & Enthusiasm

@aaudevert



Présidente meet-up  
Toulouse Data Science



Co-organisatrice du  
devfest Toulouse



- Lieu *d'échange* et de *partage* autour de la *valorisation* des *données* massives et de *l'analyse prédictive*
- + 1200 membres
- Les évènements du TDS
  - ✓ *Conférences thématiques* (1 par mois)
  - ✓ *Data Kaggle*
  - ✓ *Data NoBlabla* / ateliers
  - ✓ *Data Mojito* / soirée networking



# SOMMAIRE

1

## Big Data & son écosystème

- Introduction
- Ecosystème
- Data Lake
- Cas d'utilisation

2

## Hadoop

3

## Spark

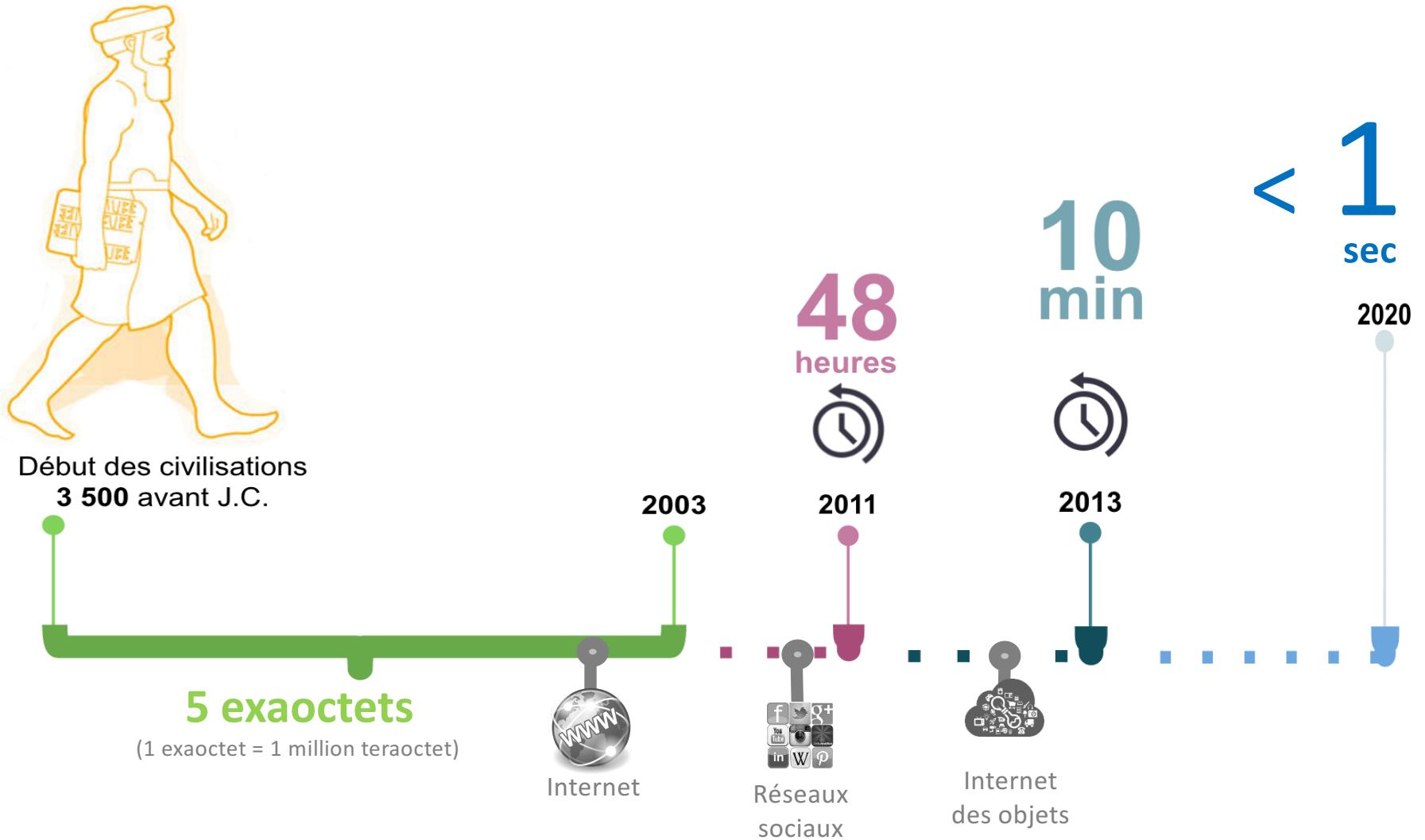
4

## Conclusion & Questions



Données vs Data vs Datum

# L'ÉVOLUTION DES DONNÉES



**44,000,000**  
MESSAGES PROCESSED  
**486,000**  
PHOTOS

MORE THAN  
**21,000,000**  
MESSAGES SENT

MORE THAN  
**195,000**  
MINUTES OF AUDIO CHATTING  
ON WECHAT

MORE THAN  
**69,500**  
HOURS OF  
VIDEO WATCHED  
ON NETFLIX



AROUND  
**56,000**  
PHOTOS  
UPLOADED

**NETFLIX**

MORE THAN  
**48,000**  
APPS DOWNLOADED  
ON IPHONE



**9,800**  
ARTICLES PINNED  
ON PINTEREST

**26**  
NEW REVIEWS  
POSTED ON YELP

**120**  
NEW ACCOUNTS  
OPENED ON  
LINKEDIN

MORE THAN  
**140**  
SUBMISSIONS  
ON REDDIT

MORE THAN  
**2,315,000**  
SEARCHES

**3,125,000**  
 **243,055**



**GO-Globe™**  
CUSTOM WEB DEVELOPMENT



MORE THAN  
**100**  
NEW DOMAINS  
REGISTERED

MORE THAN  
**280,000**  
SNAPS SENT  
ON SNAPCHAT

**You**  
**Tube**

MORE THAN  
**2,700,000**  
VIDEO VIEWS AND  
**139,000** HOURS  
OF VIDEO WATCHED

**14** NEW  
SONGS ADDED  
ON SPOTIFY

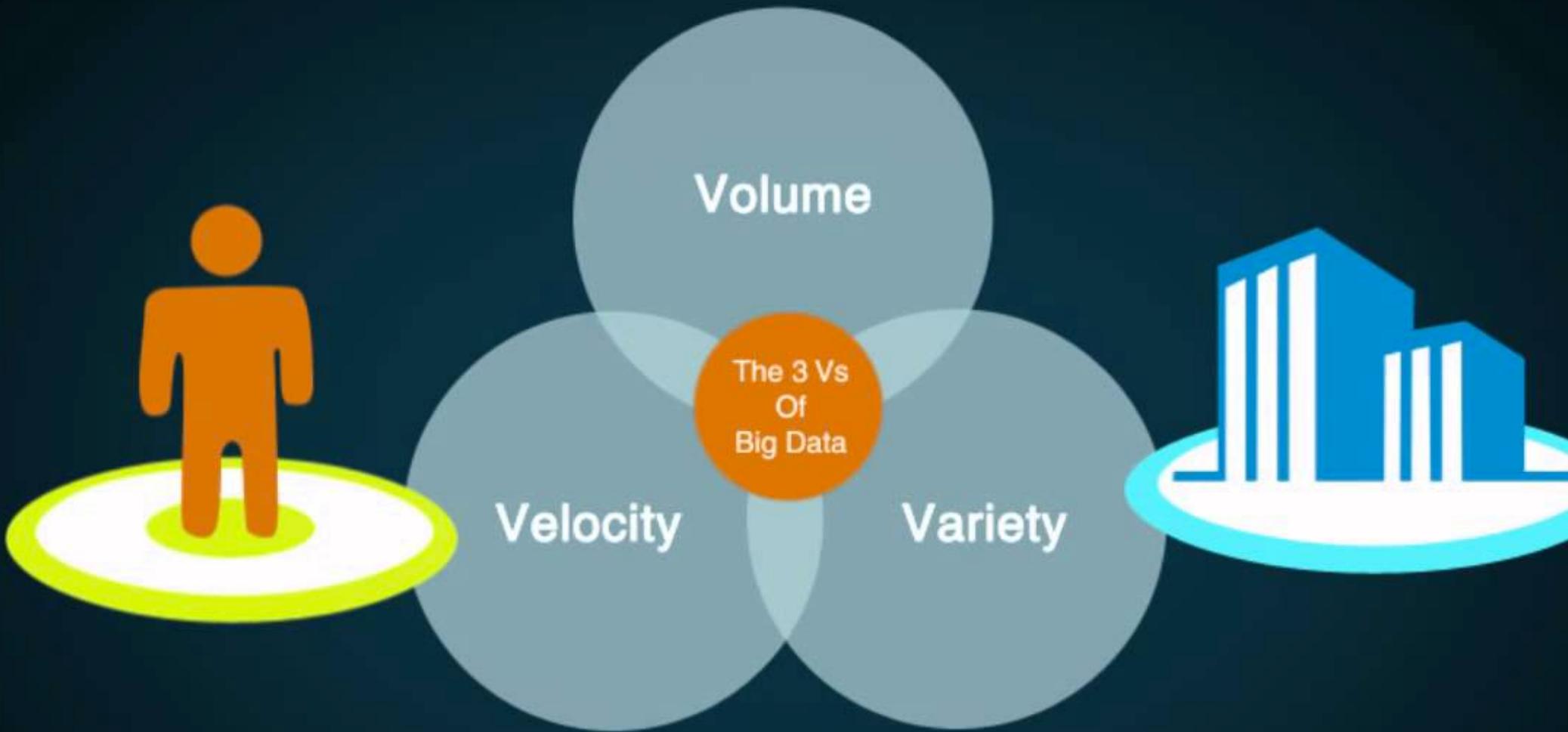
MORE THAN  
**300** HOURS  
OF VIDEO ARE UPLOADED



MORE THAN  
**150,000,000**  
E-MAILS ARE SENT



MORE THAN  
**430,000**  
TWEETS SENT



# QUE SE CACHE DERRIÈRE LE BUZZWORD BIG DATA ?

Une variété de sources de données...



... des nouvelles technologies et des outils pour exploiter et analyser ces données

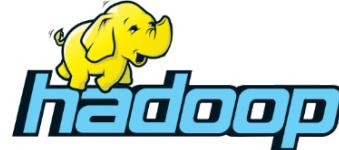


... et des outils & technologies pour les visualiser et les utiliser

Internal & External



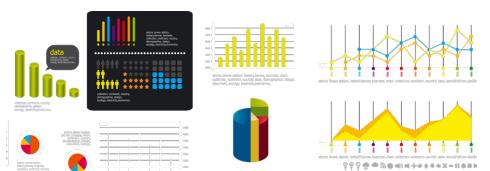
Calculators, Storage... Big Analytics



Platforms & Apps



Visualisation Interfaces



# Le BIG DATA n'est pas une technologie



Mais la capacité de collecter, stocker, traiter, valoriser, rapidement à moindre coût de gros volumes de donnée où la taille unitaire d'une donnée est insignifiante.

# SOMMAIRE

1

## Big Data & son écosystème

- Introduction
- Ecosystème
- Data Lake
- Cas d'utilisation

2

## Hadoop

3

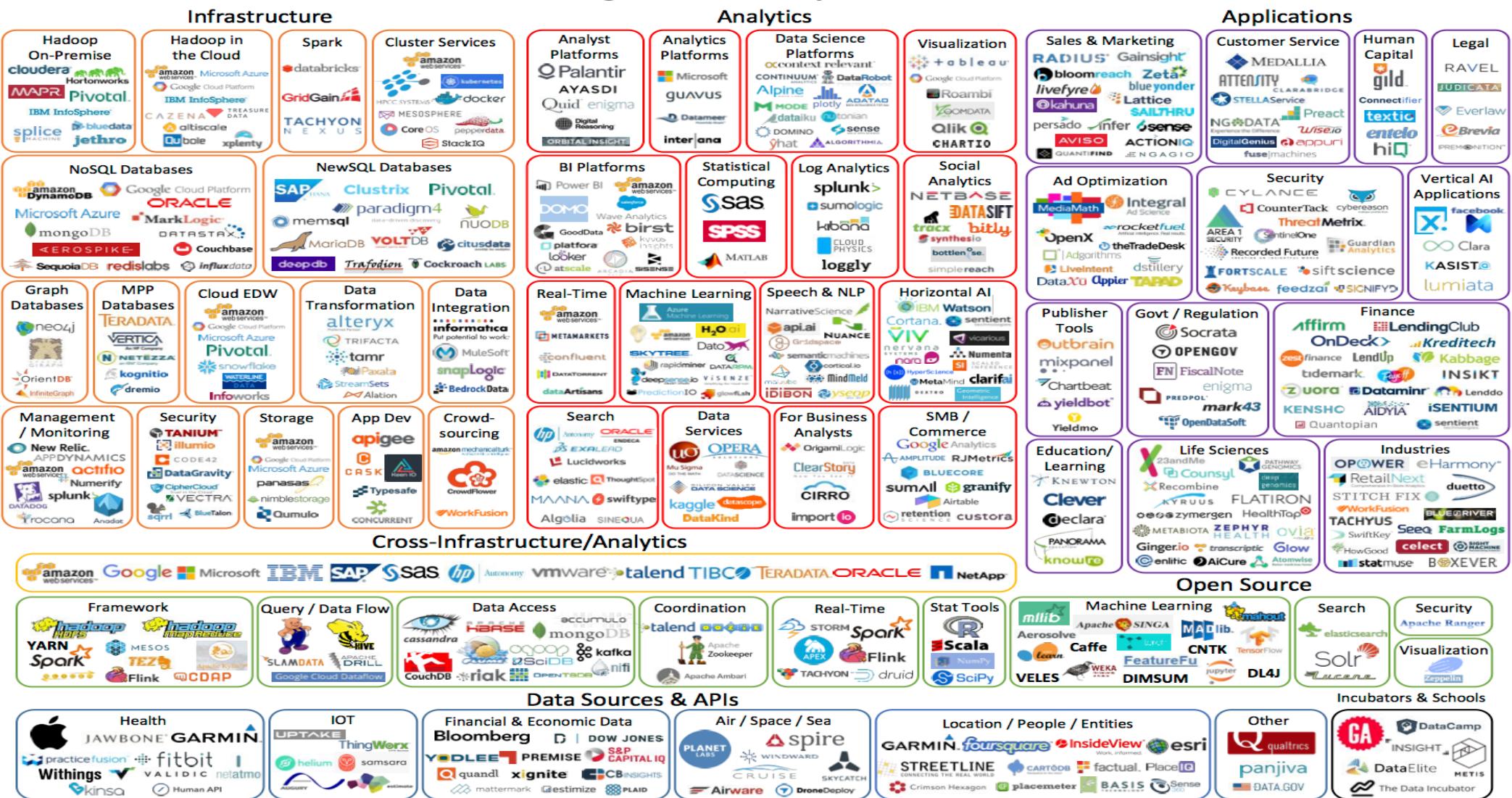
## Spark

4

## Conclusion & Questions

# L'ÉCOSYSTÈME BIG DATA

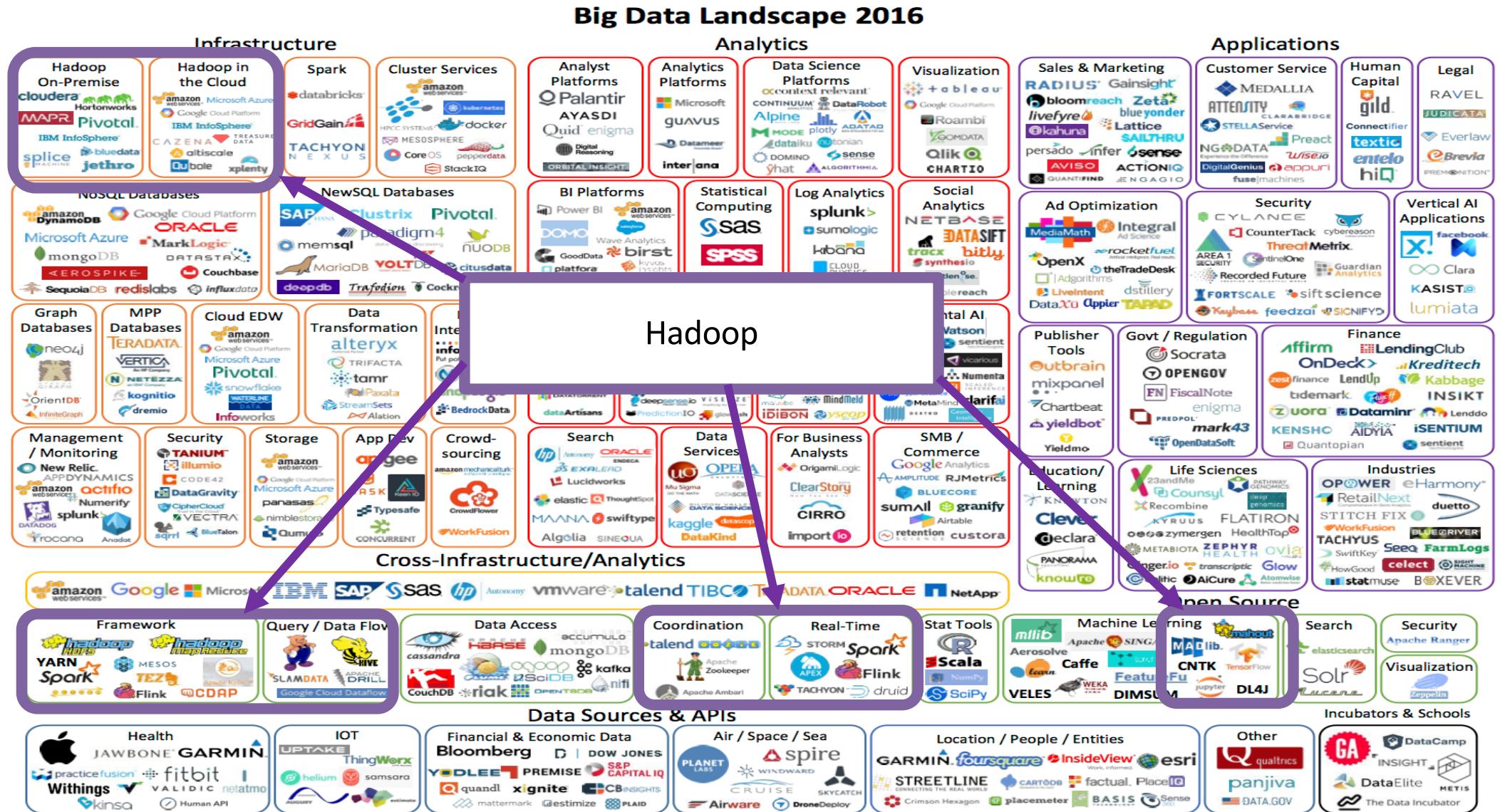
## Big Data Landscape 2016



© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)

FIRSTMARK

# L'ÉCOSYSTÈME BIG DATA

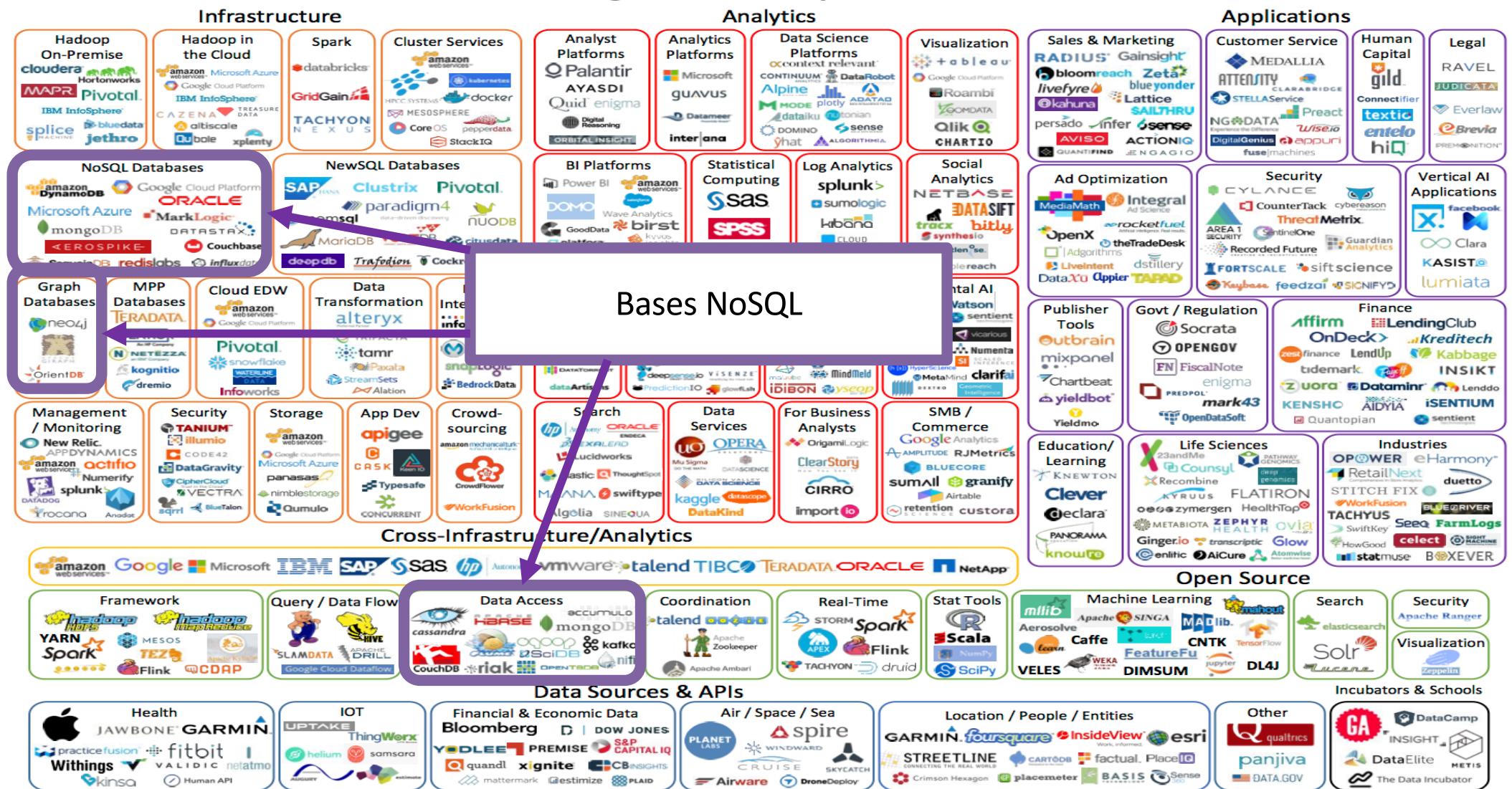


© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)

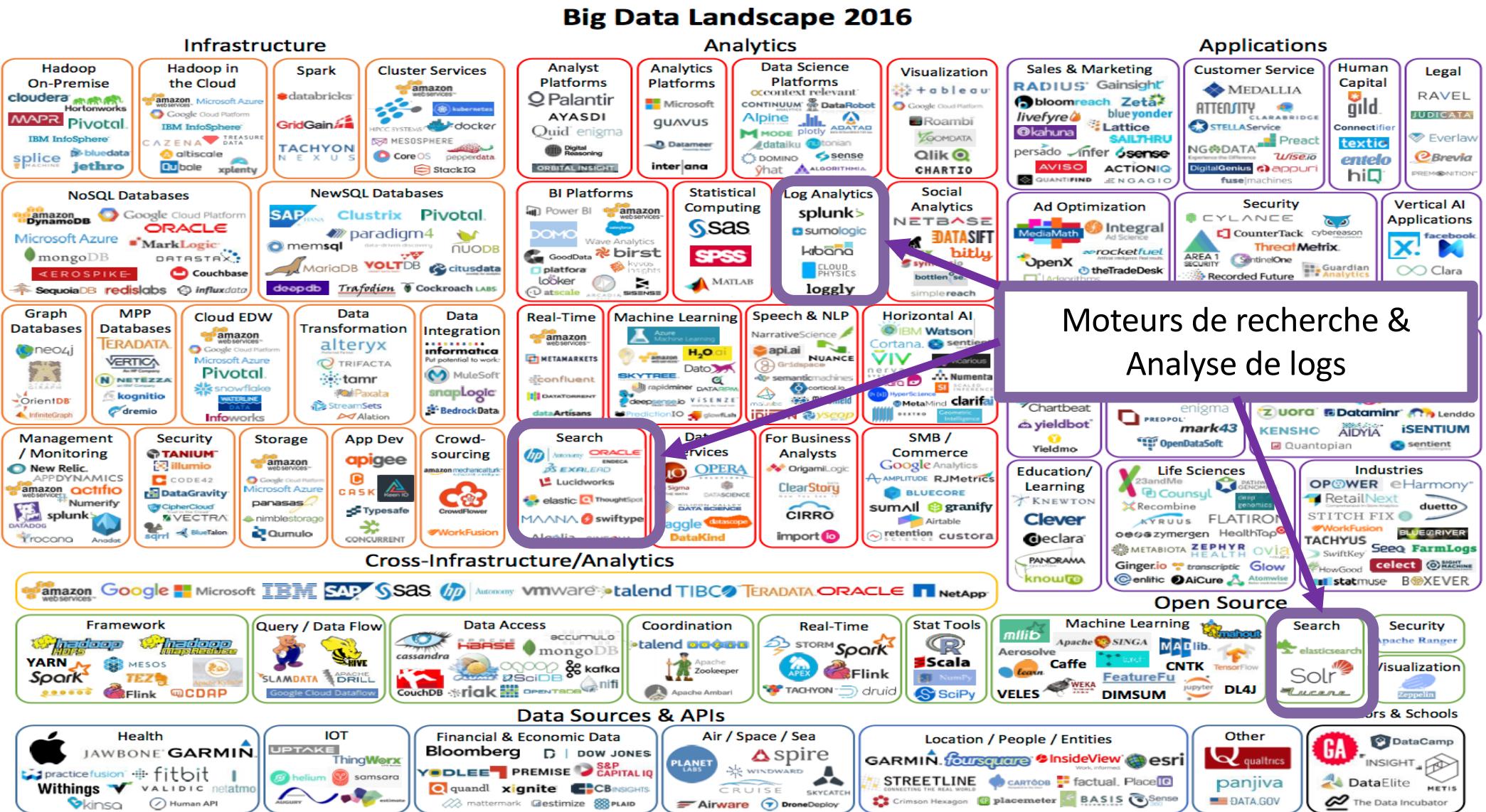
FIRSTMARK

# L'ÉCOSYSTÈME BIG DATA

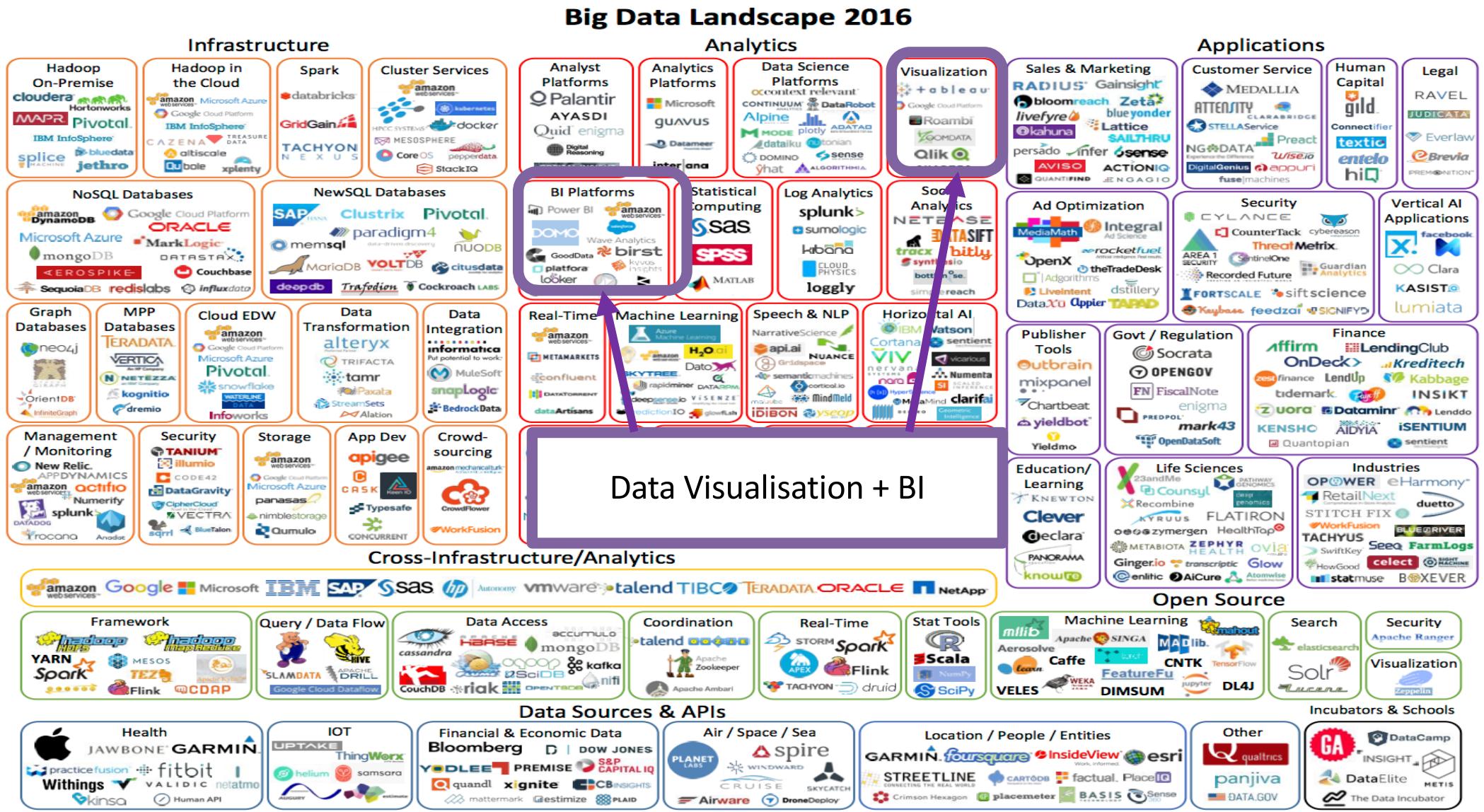
## Big Data Landscape 2016



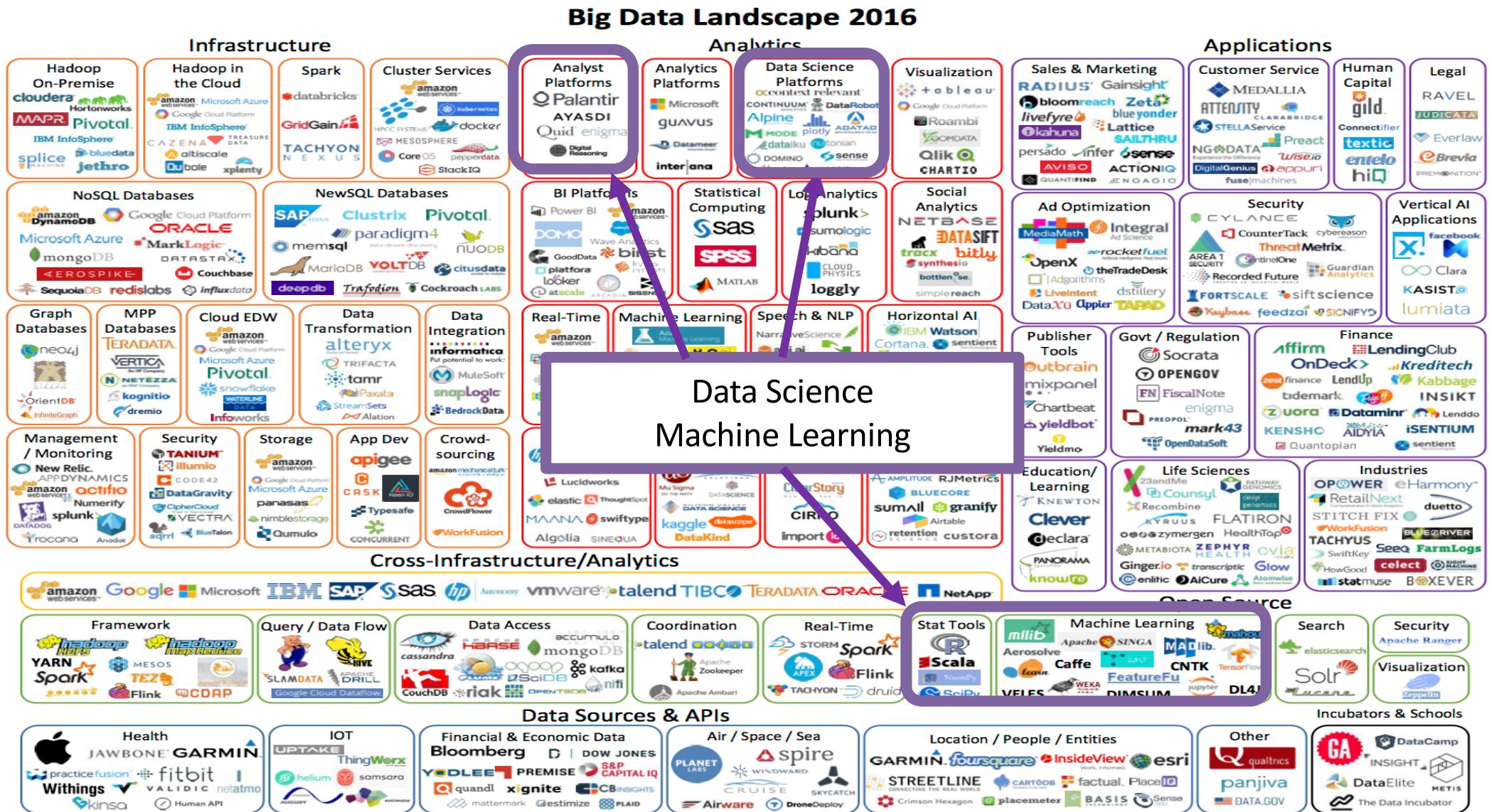
# L'ÉCOSYSTÈME BIG DATA



# L'ÉCOSYSTÈME BIG DATA



# L'ÉCOSYSTÈME BIG DATA



# SOMMAIRE

1

## Big Data & son écosystème

- Introduction
- Ecosystème
- Data Lake
- Cas d'utilisation

2

## Hadoop

3

## Spark

4

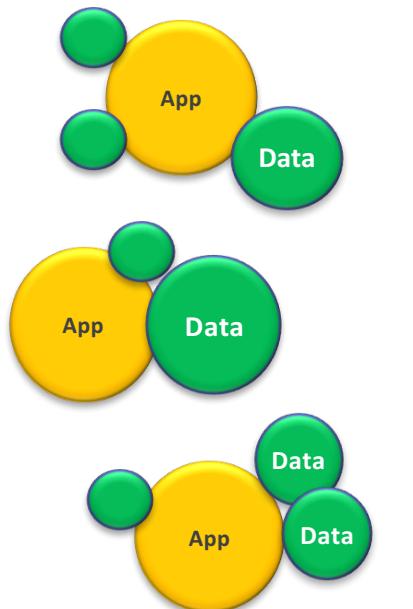
## Conclusion & Questions

# VERS UNE NOUVELLE GESTION DES DONNÉES

---

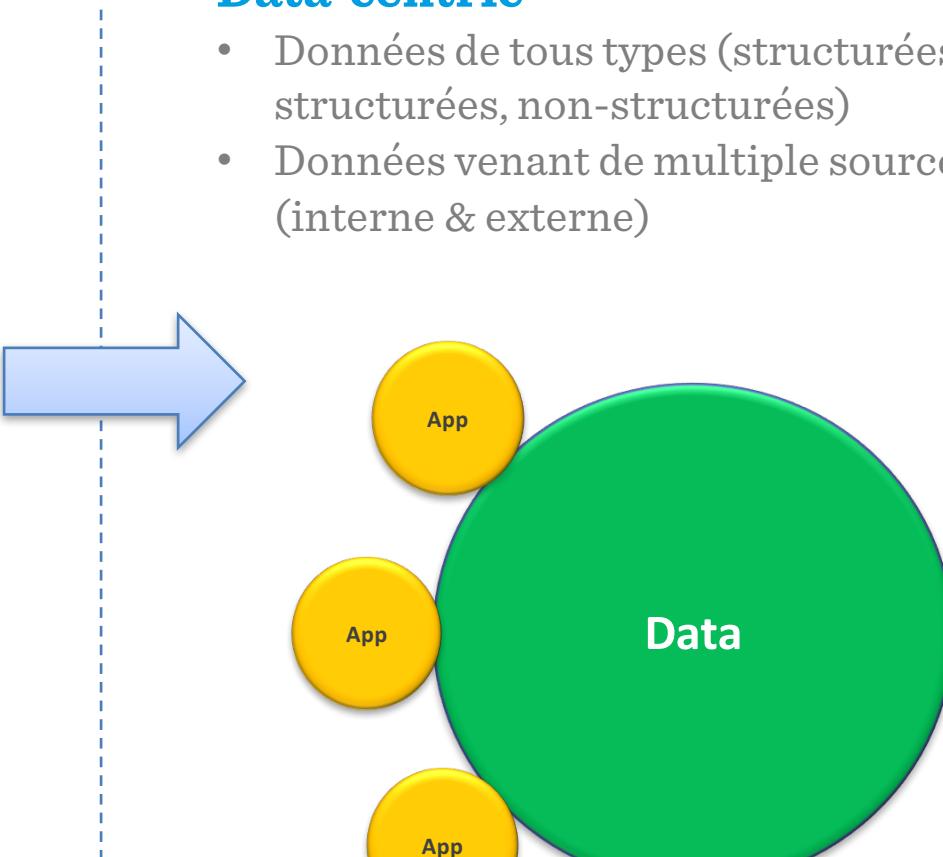
## Process-centric

- Données structurées
- Données venant de sources Internes
- Données “importantes” uniquement
- Multiple copies des Données

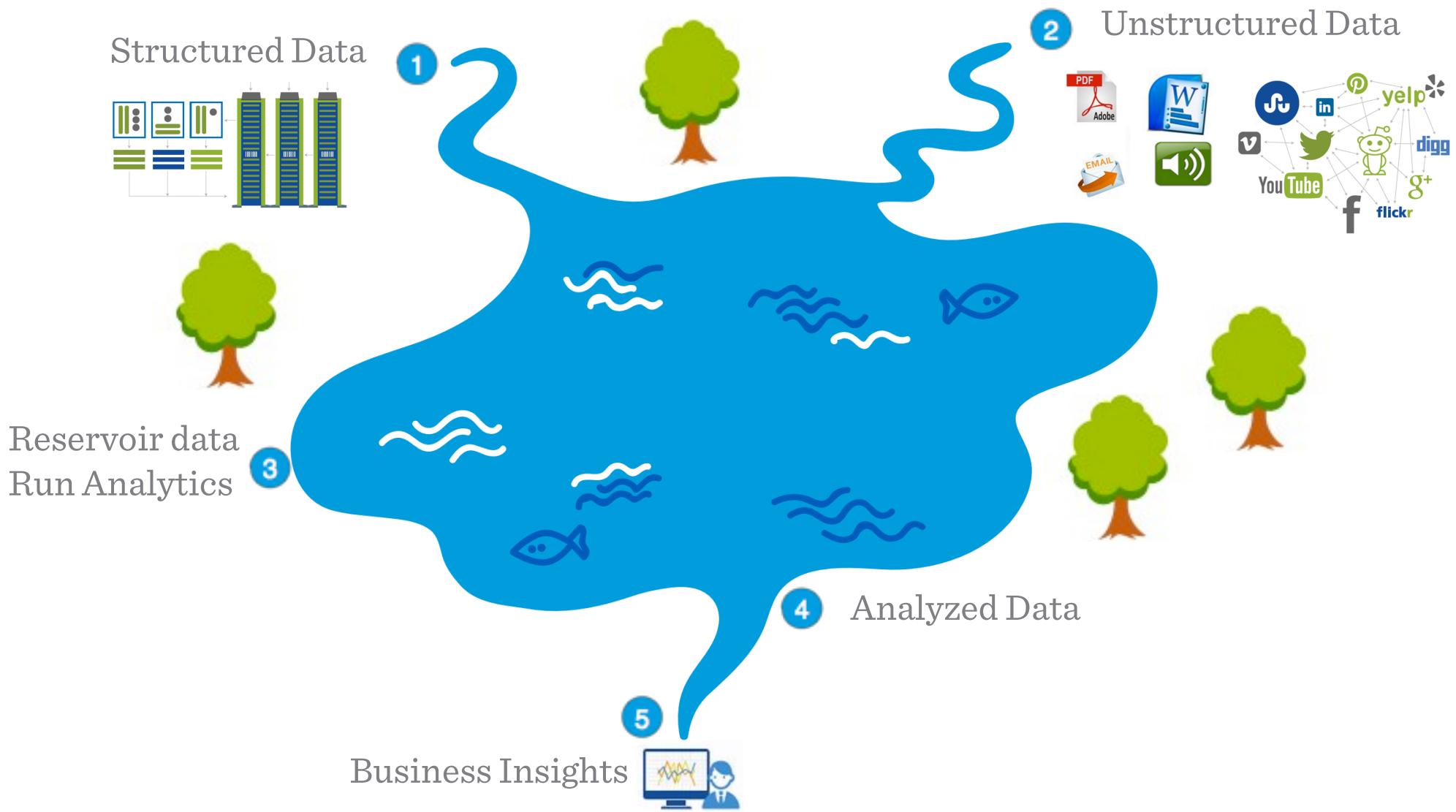


## Data-centric

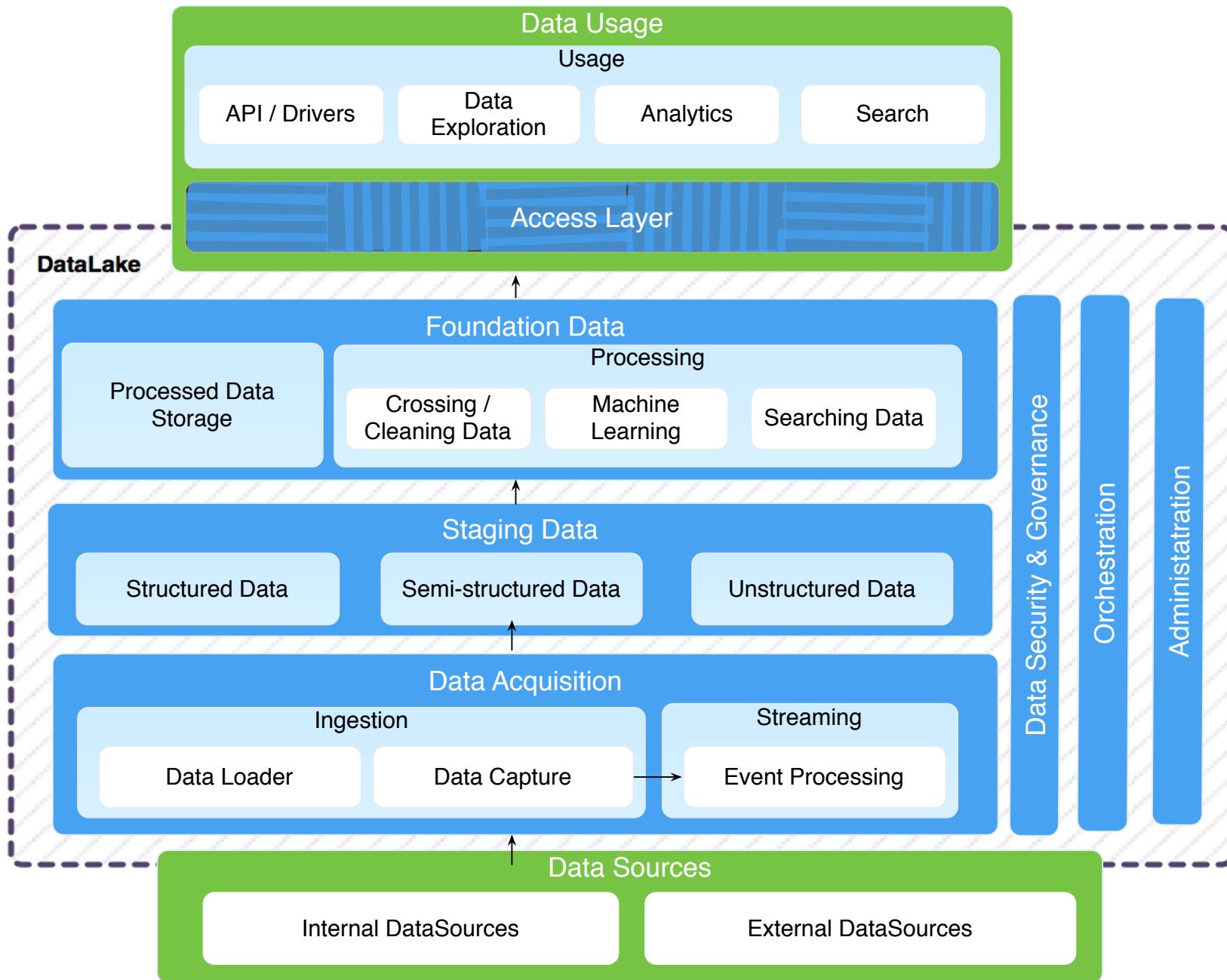
- Données de tous types (structurées, semi-structurées, non-structurées)
- Données venant de multiple sources de données (interne & externe)



# DATA LAKE



# DATA LAKE



# SOMMAIRE

1

## Big Data & son écosystème

- Introduction
- Ecosystème
- Data Lake
- Cas d'utilisation

2

## Hadoop

3

## Spark

4

## Conclusion & Questions

# QUELQUES CAS D'UTILISATION

---

1

## Réduction des couts :

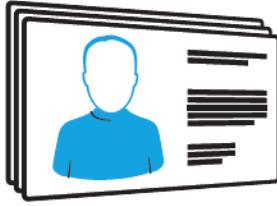
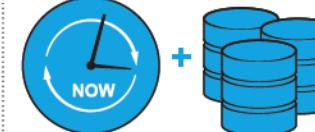
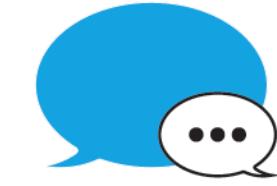
- Archivage
- Déchargement d'entrepôt de données
- ETL (Extract-Transform-Load)
- Fail-Over

2

## Elargir le champs des possibles :

- Analyser et tirer de la valeur des données de l'entreprise
- Analyser des données exogènes de l'entreprise et les corréler avec des données internes

# QUELQUES CAS D'UTILISATION

<p><b>Profile Management</b></p> 	<p><b>Personalization</b></p> 	<p><b>360 Degree Customer View</b></p> 	<p><b>Internet of Things</b></p> 	<p><b>Mobile Applications</b></p> 
<p><b>Content Management</b></p> 	<p><b>Catalog</b></p> 	<p><b>Real Time Big Data</b></p> 	<p><b>Digital Communication</b></p> 	<p><b>Fraud Detection</b></p> 

# SOMMAIRE

1

Big Data & son écosystème

2

Hadoop

- Introduction
- Composants Primaires
- Écosystème
- Distributions

3

Spark

4

Conclusion & Questions

# INTRODUCTION A HADOOP

---

## *Framework OpenSource Apache Hadoop*

- stocker et traiter de grands ensembles de données
- de façon distribuée (Cluster)
- sur du matériel standard



## *Composé de nombreux projets Apache Software Foundation*

- Répondant à une fonctionnalité bien précise
- Associés à leur propre communauté de développeurs
- Possèdent leur propre cycle de développement



# INTRODUCTION A HADOOP

---

***Le projet Hadoop consiste en deux grandes parties :***

- Stockage des données: **HDFS** (**H**adoop **D**istributed **F**ile **S**ystem)
- Traitement des données: **Map Reduce**



## ***Principe***

- **Diviser** et **sauvegarder** les données sur un **cluster**
- **Traiter** les données directement ***là où elles sont stockées***
- **Scalabilité** : possibilité d'**ajouter/retirer** des **machines** au cluster

# CLUSTER HADOOP

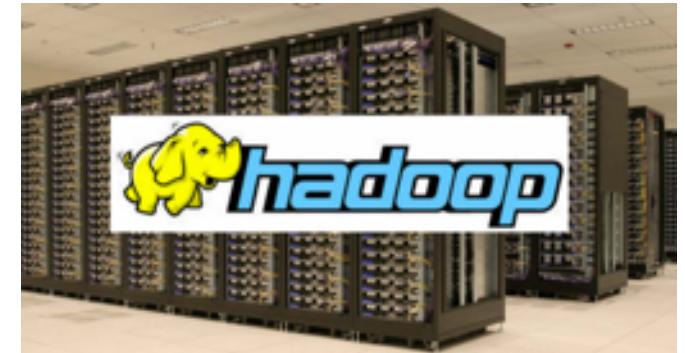
---

## Cluster Hadoop

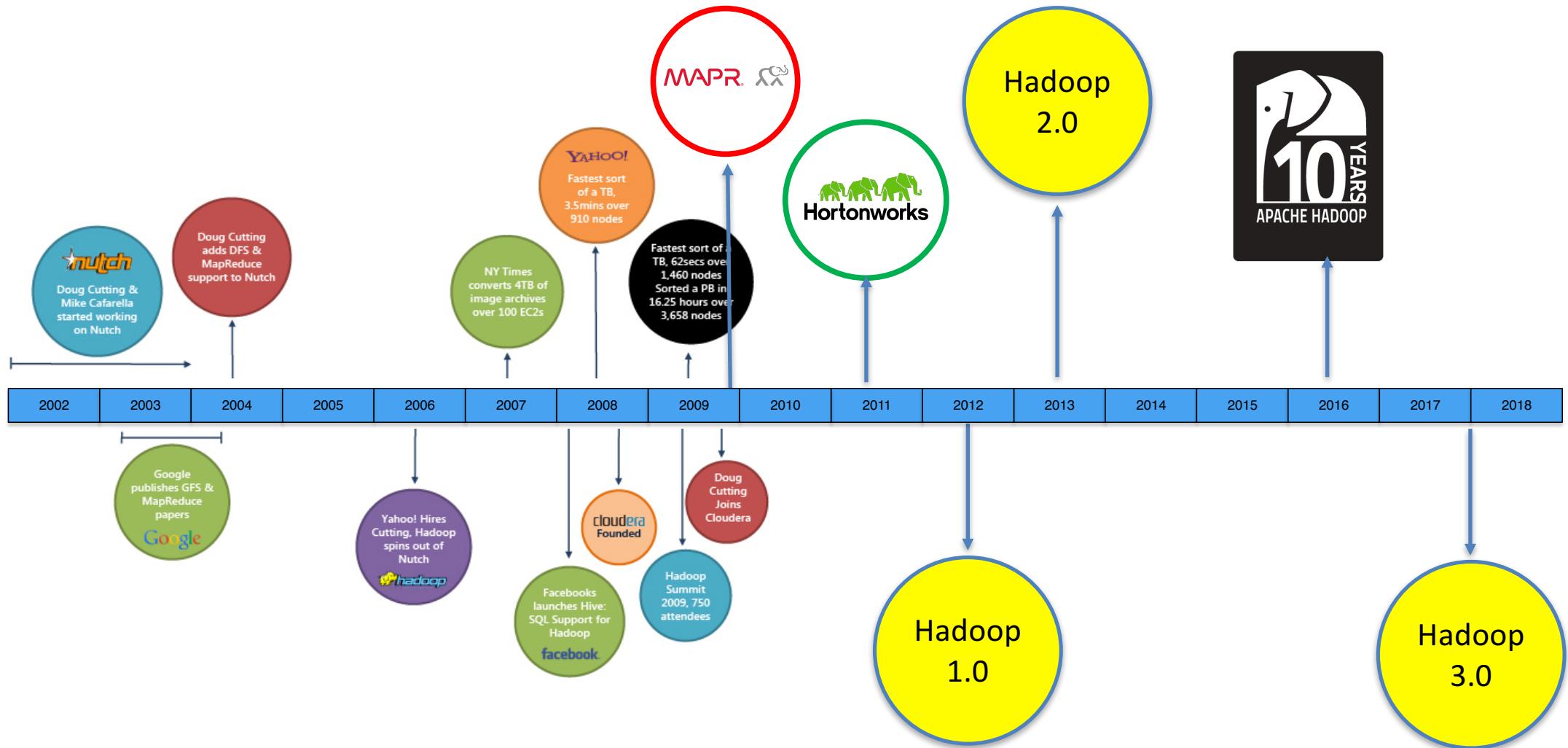
- *Ensemble de machines* : serveurs d'entrée de gamme (commodités)
- Système « *Shared Nothing* » : Le seul élément partagé est le réseau qui connecte les machines
- Une machine est appelé un « *Node* »

Un cluster est composé de :

- *Master Nodes*
  - Gèrent l'infrastructure
- *Worker/Slave Nodes*
  - Contiennent les données distribuées
  - Exécutent les traitements sur les données.



# L'HISTOIRE D'HADOOP



# QUIZZ!!!

---

**Pourquoi le nom Hadoop ?**

Nom de la peluche du fils de Doug Cutting



**Pourquoi le nom Lucène ?**

Deuxième nom de sa femme & Prénom de sa grand-mère

# SOMMAIRE

1

Big Data & son écosystème

2

Hadoop

- Introduction
- Composants Primaires
- Écosystème
- Distributions

3

Spark

4

Conclusion & Questions

---

**HDFS** est un système de fichiers distribué, extensible et portable.

- Ecrit en **Java**
- Permet de **stocker** de très gros volumes de données (données structurés ou non) au sein d'un Cluster

Les données sont **découpées et distribuées** dans un cluster Hadoop :

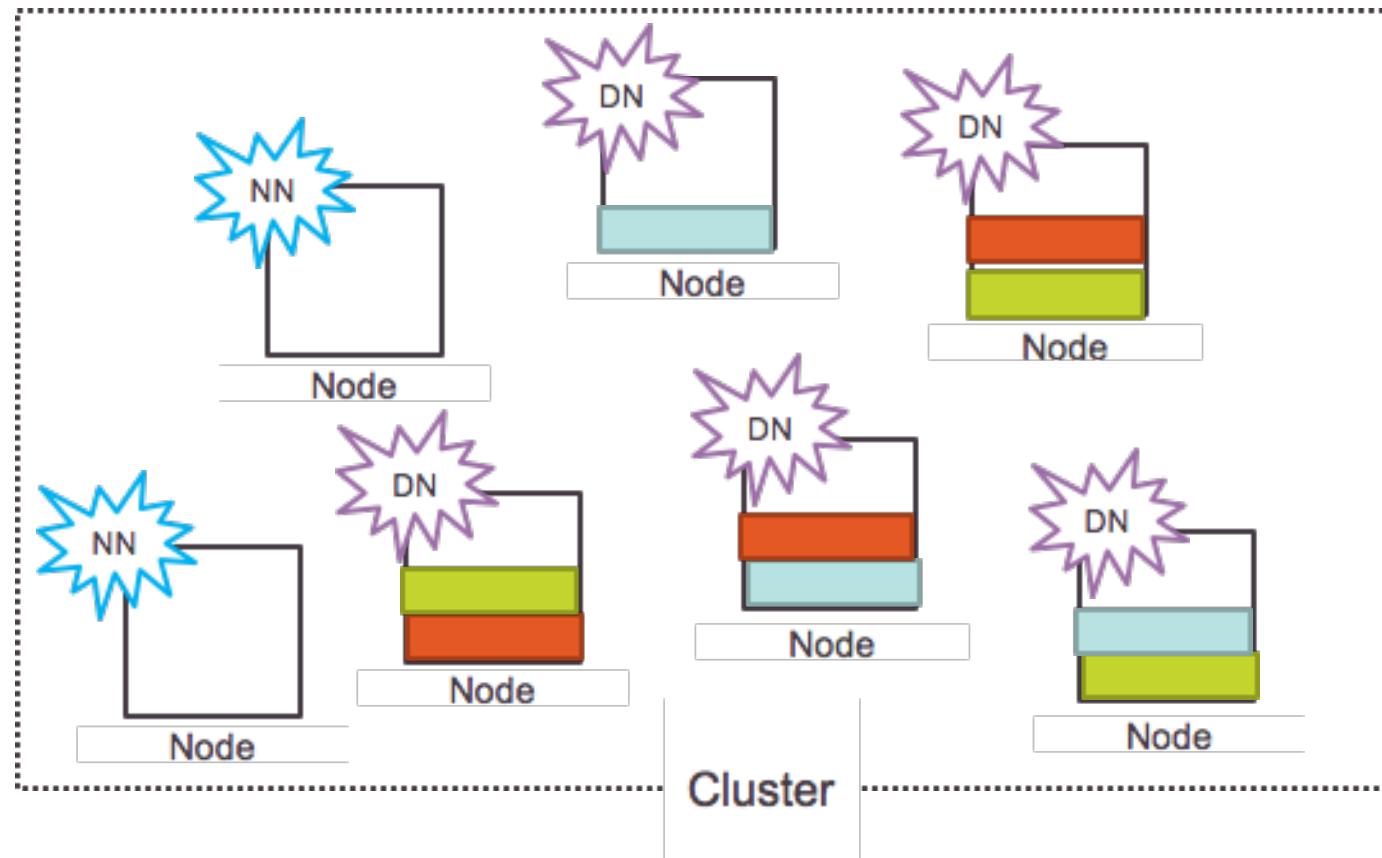
- **Block Size** : par défaut 128 Mo
- **Replication Factor** : nombre de copies d'une donnée (par défaut 3 : 1 primaire et 2 secondaires)

Dans HDFS, les données sont de type « **write-once** »

# HDFS



bloc\_1 -> 128Mo  
bloc\_2 -> 128 Mo  
bloc\_3 -> 64 Mo  
Fichier (320 Mo)



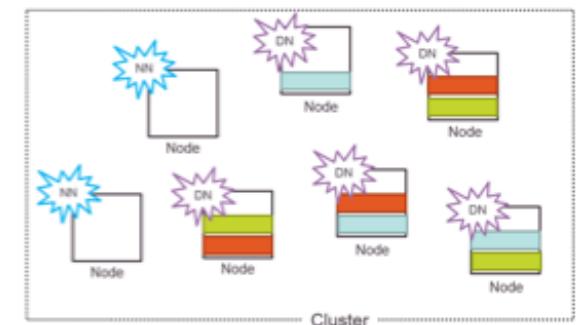


## **NameNode** : Responsable de la localisation des données

- **Démon** s'exécutant sur une machine séparée
- Contient des **méta-données**
- Permet de retrouver les nœuds qui exécutent les blocs d'un fichier
- NameNode est **duplicqué**, non seulement sur son propre disque, mais également quelque part sur le système de fichiers du réseau (**Secondary NameNode**).

## **DataNode** : Stocke et restitue les blocs de données

- **Démon** sur chaque nœud du cluster



# MAP REDUCE

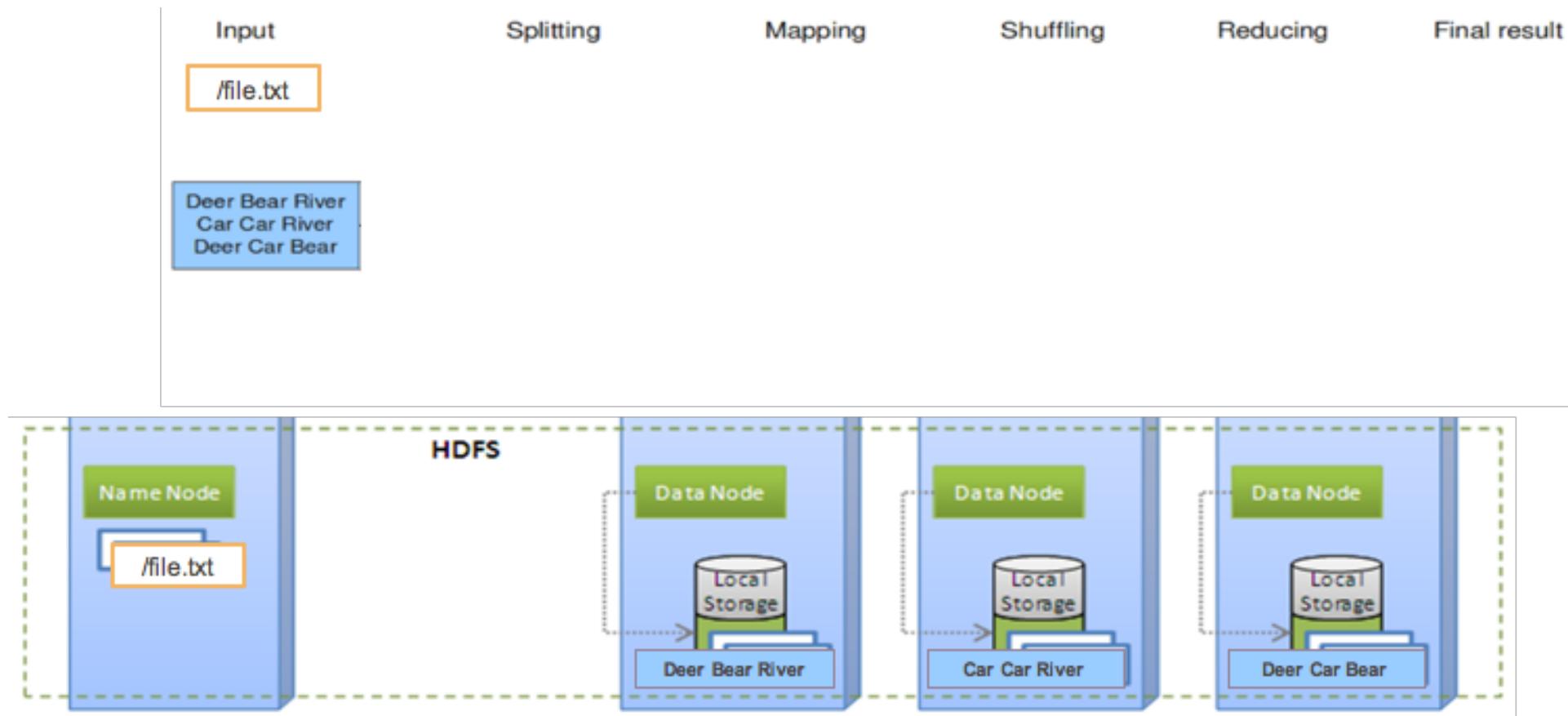
---



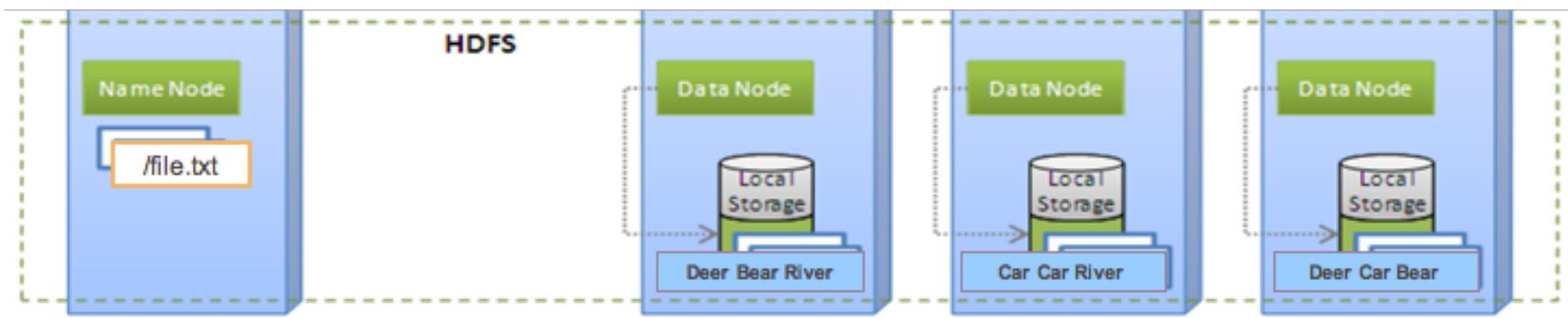
## **Map Reduce :**

- Concept issu des langages fonctionnels
- Utilisé par Google pour son outil de recherche Web
- **Co-localiser les données & les traitements**
- **Parallélisation** automatique des programmes Hadoop  
-> Gestion transparente du mode **distribué**
- Traitement rapide des **données volumineuses**
- **Fault Tolerant** : Tolérance aux pannes basée sur la réPLICATION

# MAP REDUCE : WORD COUNT



# MAP REDUCE : WORD COUNT



# MAP REDUCE

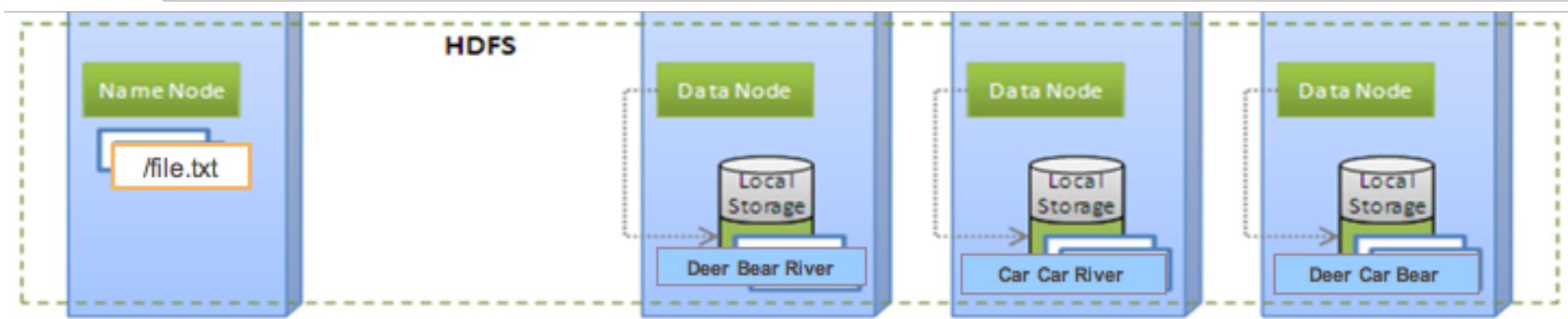
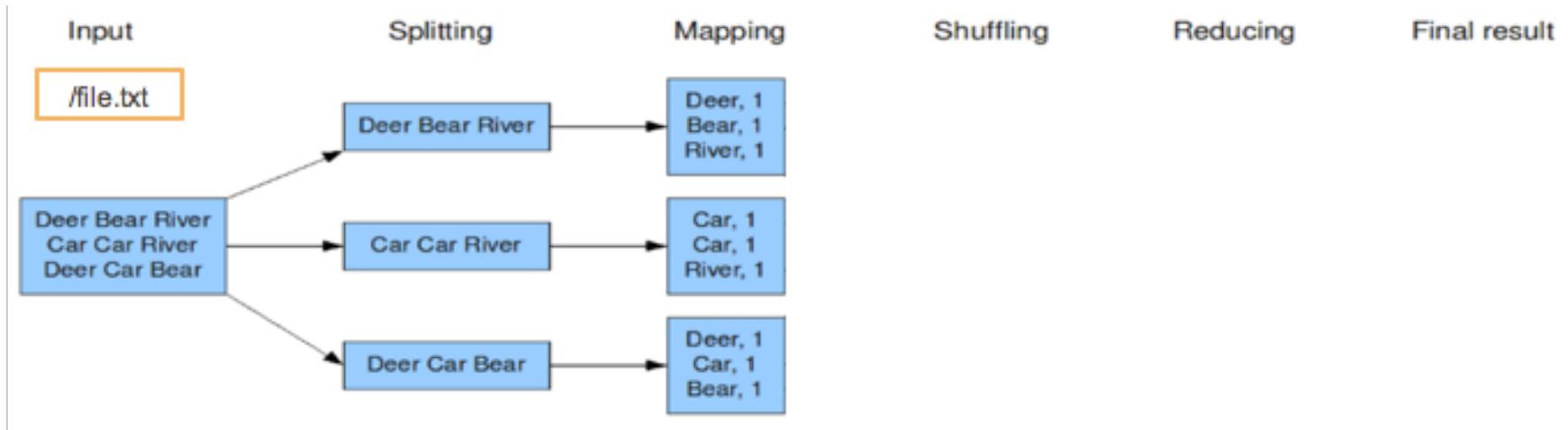


---

**Map**: Décomposition d'une tache en un ensemble de tache plus petite produisant un sous ensemble du résultat final

- Composé de **Mappers**
- Fonctionnant en **parallèle**
- **Stockage sur disque** des données en entrée et sortie
- **Sorties** des Mappers = **enregistrements intermédiaires** sous forme d'un couple (clef, valeur)

# MAP REDUCE : WORD COUNT



# MAP REDUCE

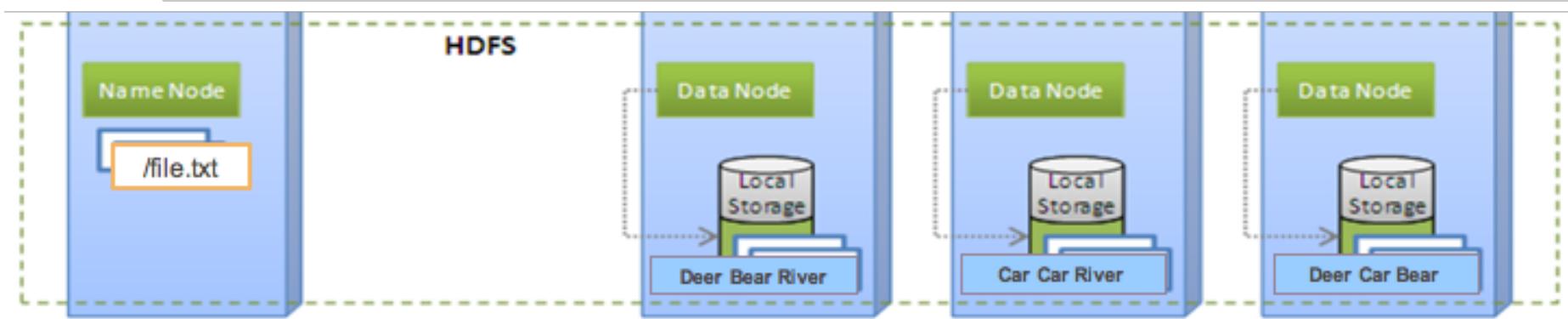
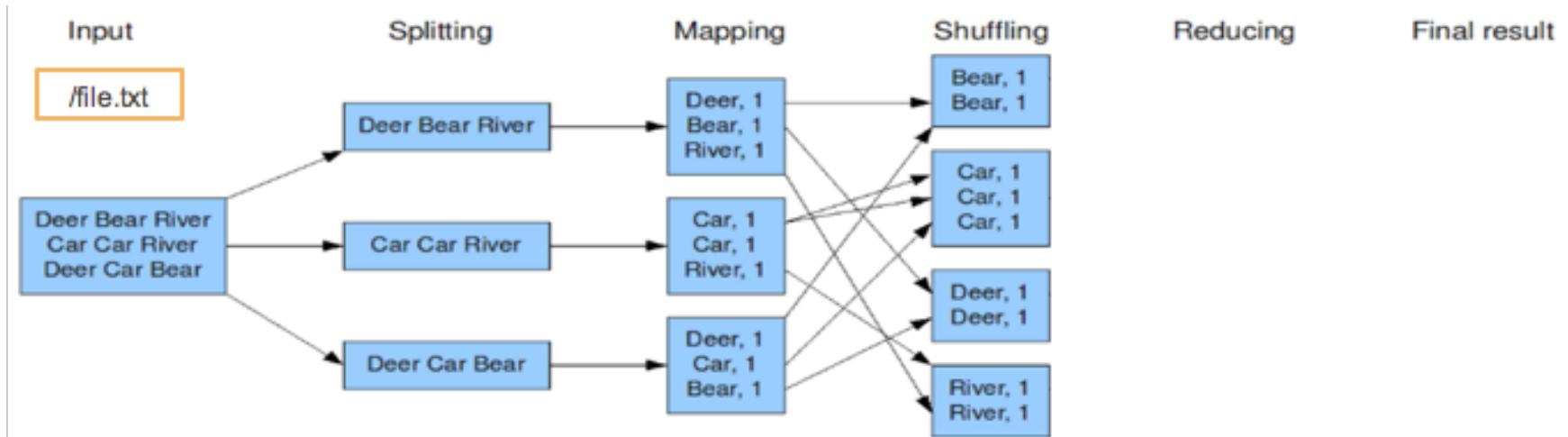


---

## **Shuffle & Sort : Mélange et Tri**

- *Tri par clef des données intermédiaires.*
- *Envoi des données ayant la même clef vers un seul et même reducer.*

# MAP REDUCE : WORD COUNT



# MAP REDUCE

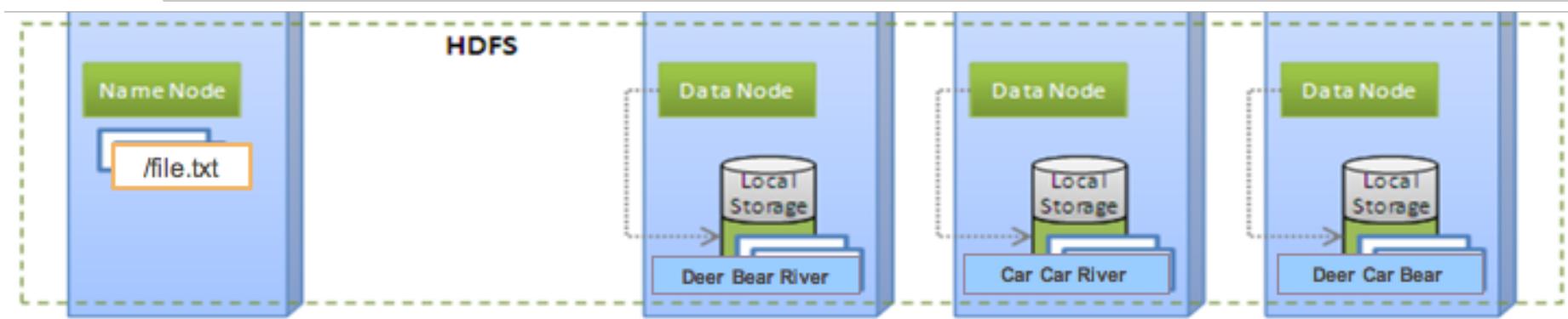
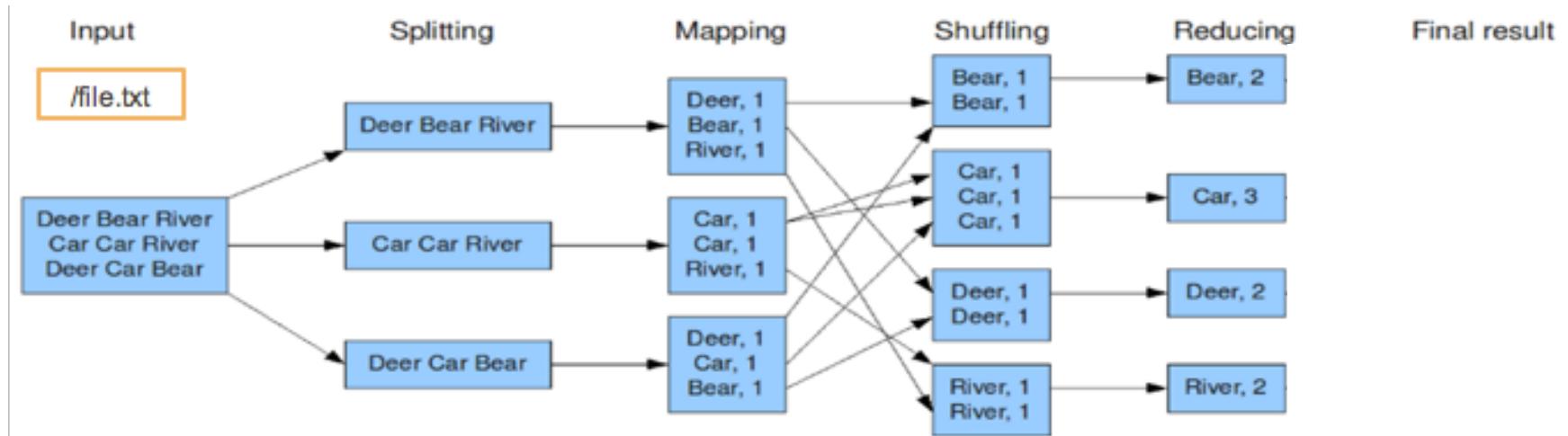
---



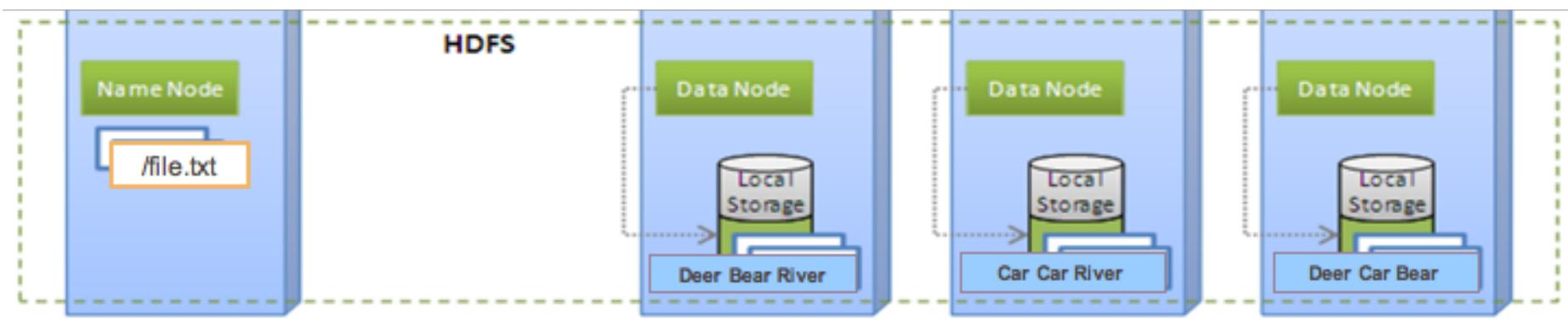
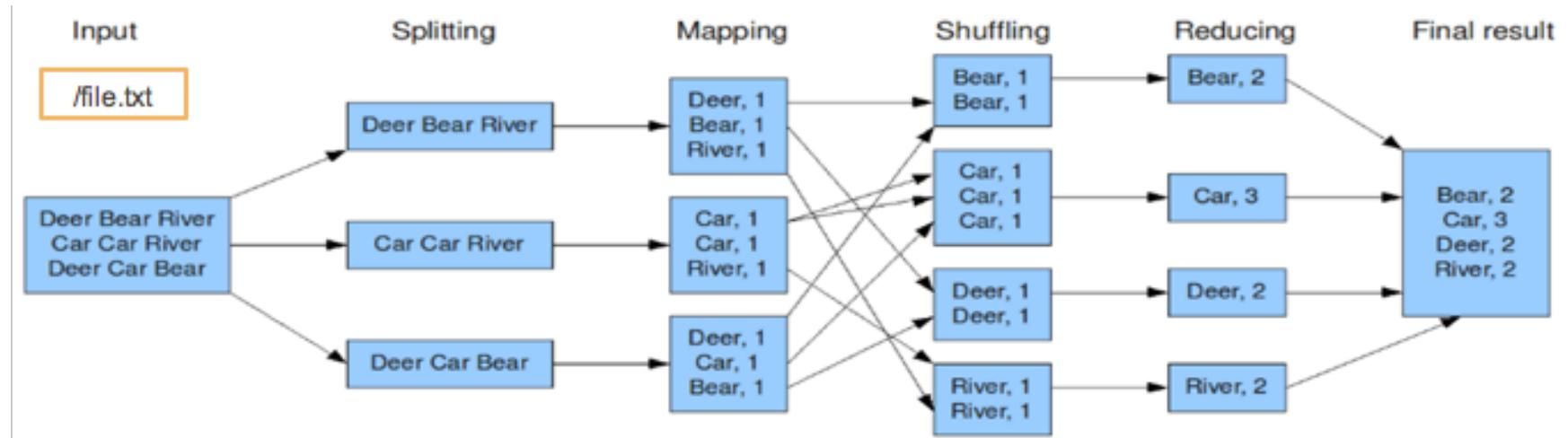
## **Reduce:**

- *Consolide (agrégation, filtre) les résultats issus du Mapper.*
- *Génère les **résultats finaux** et les écrit sur disque.*

# MAP REDUCE : WORD COUNT



# MAP REDUCE : WORD COUNT



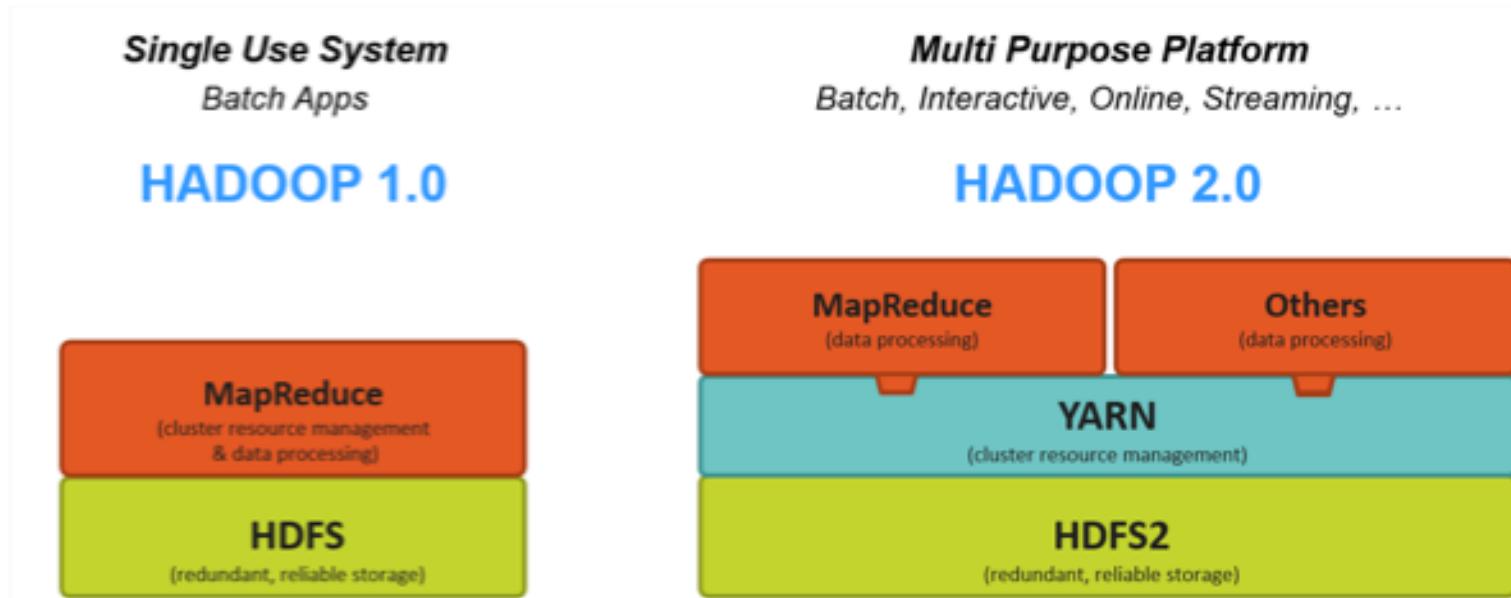
# YARN

**Yet-Another-Resource-Negotiator**

Intégré à **Hadoop** depuis la *v2*

**YARN** apporte une séparation entre :

- Gestion de l'état du cluster et des ressources.
- Gestion de l'exécution des jobs.



# SOMMAIRE

1

Big Data & son écosystème

2

Hadoop

- Introduction
- Composants Primaires
- Écosystème
- Distributions

3

Spark

4

Conclusion & Questions

# LANGAGE DE REQUÊTAGE

---

- **Au dessus du MapReduce**

- **Pig**

- Langage de script
    - Développé par Yahoo



- **Hive** : requêtes SQL

- HiveQL : langage SQL – Select only
    - Crée à l'origine par Facebook



- **SQL-on-Hadoop : Impala & Drill**

- Extraction des données directement à partir de HDFS avec SQL
  - Optimisé pour les requêtes à faible latence
  - Requêtes très performantes



# BASE NOSQL

## HBase

- Base de données NoSQL orientée colonnes
- Distribuée : basée sur Hadoop et HDFS



(Inspirée des publications de Google sur BigTable)

Trié selon la clé de la ligne et la clé de la colonne

Famille de colonne

Timestamp est entier long

2 Versions de la ligne

Nom de colonne

Row Key	Column Key	Timestamp	Value
1	info:name	1273516197868	Gaurav
1	info:age	1273871824184	28
1	info:age	1273871823022	34
1	info:sex	1273746281432	Male
2	Info:name	1273863723227	Harsh
3	Info:name	1273822456433	Raman

# ECOSYSTEME HADOOP : CONNEXION A HDFS

---

## *Sqoop*



- Import des données d'une base de données traditionnelle dans HDFS.
- Développé par Cloudera

## *Flume*

- Collecte d'un ensemble de données (des logs) à partir de plusieurs sources vers HDFS
- Développé par Cloudera



# ECOSYSTEME HADOOP

---

## Hue

- Front-end graphique pour le cluster
- Fournit
  - Un navigateur pour HDFS et HBase
  - Des éditeurs pour Hive, Pig, Impala et Sqoop



## Oozie

- Outil de gestion de workflow
- Gère et coordonne les jobs Hadoop

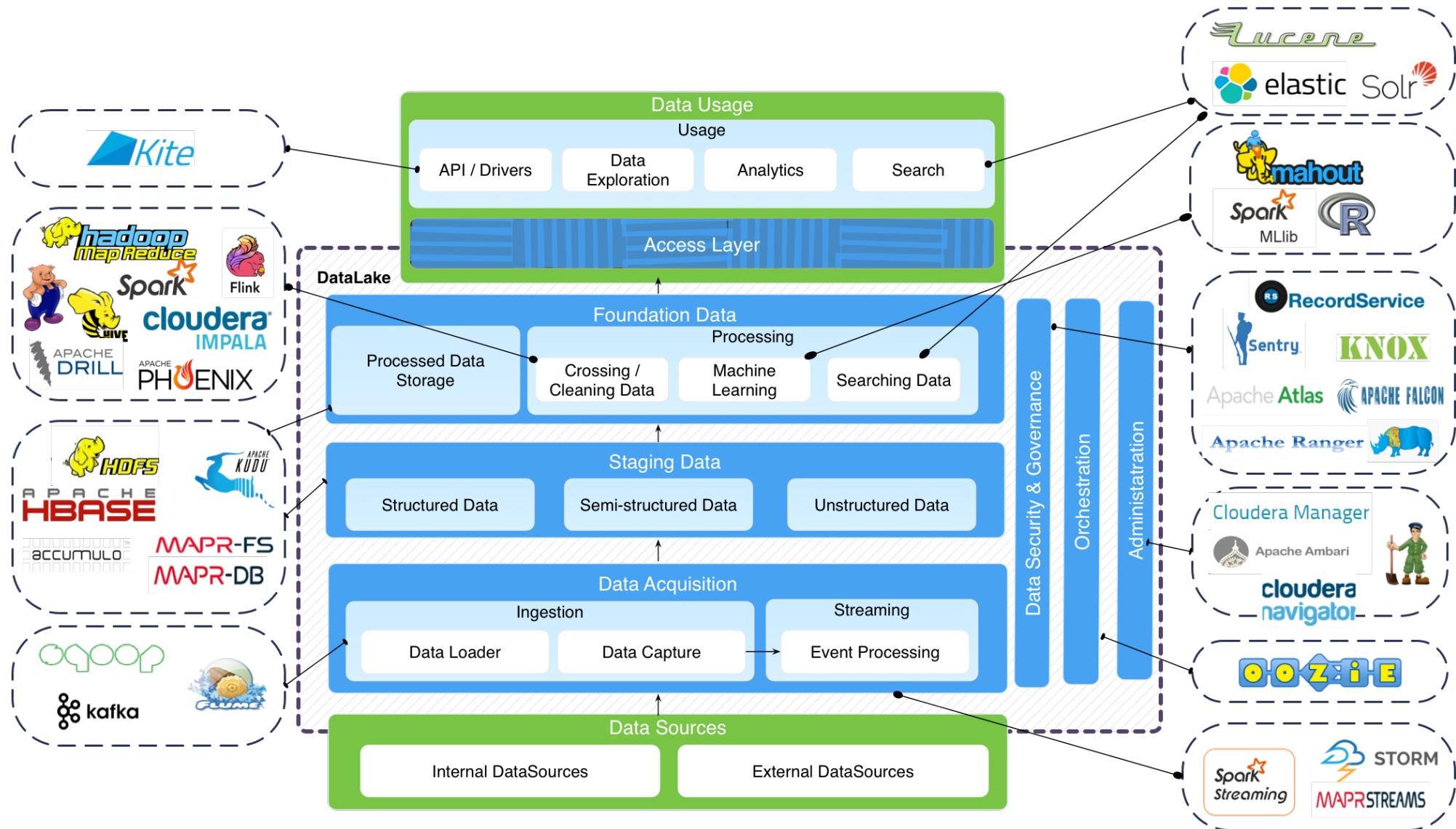


## Mahout

- Bibliothèque d'implémentation d'algorithmes d'apprentissage automatique et de datamining



# ECOSYSTEME HADOOP



# SOMMAIRE

1

Big Data & son écosystème

2

Hadoop

- Introduction
- Composants Primaires
- Écosystème
- Distributions

3

Spark

4

Conclusion & Questions

# LES DISTRIBUTIONS D'HADOOP

## Open Source

- Apache Hadoop



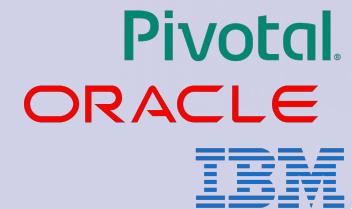
## Pure Players

- Cloudera
- Hortonworks
- MapR



## Software Publishers

- Pivotal Greenplum (HDP)
- IBM InfoSphere BigInsights (CDH)
- Oracle Big data appliance (CDH)

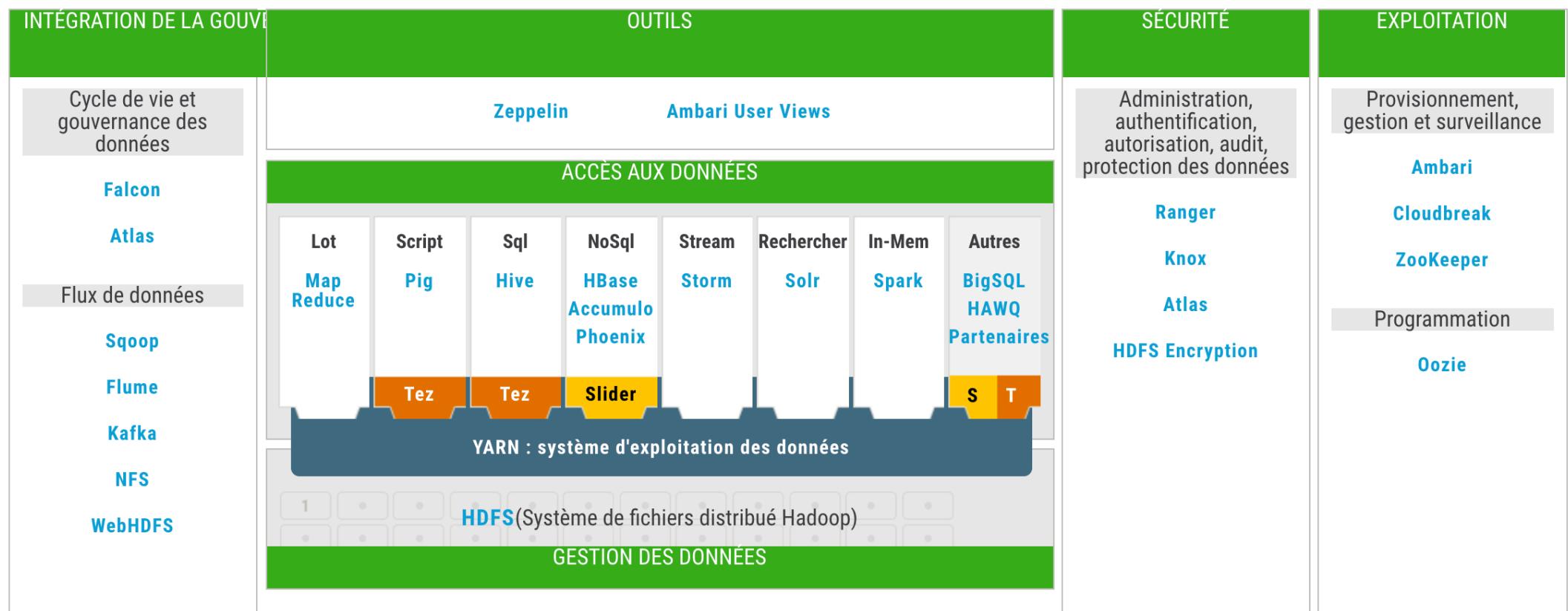


## Public Cloud

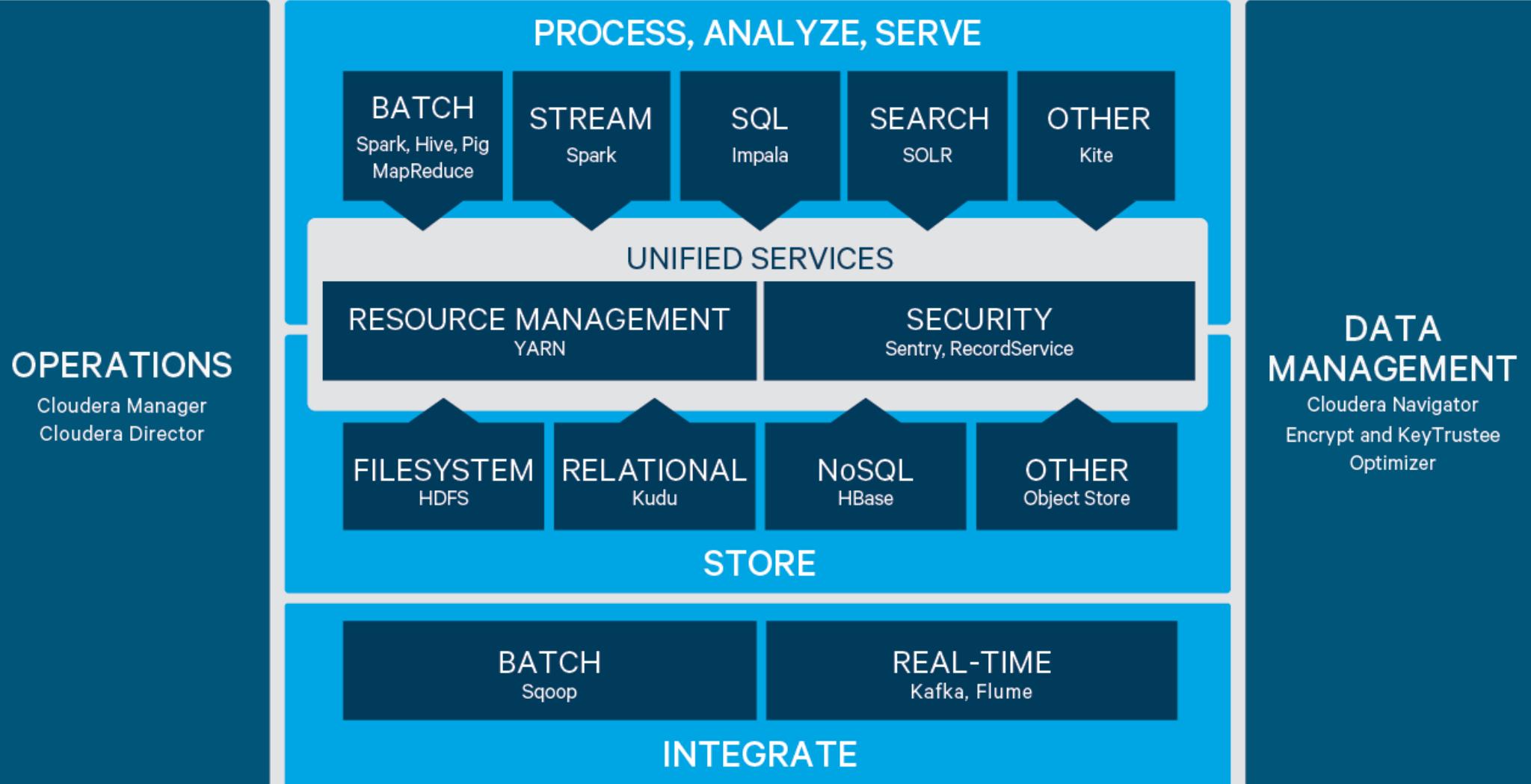
- Amazon Elastic MapReduce (Amazon & MapR)
- Microsoft Azure HDInsight (HW)
- Google Cloud Dataproc



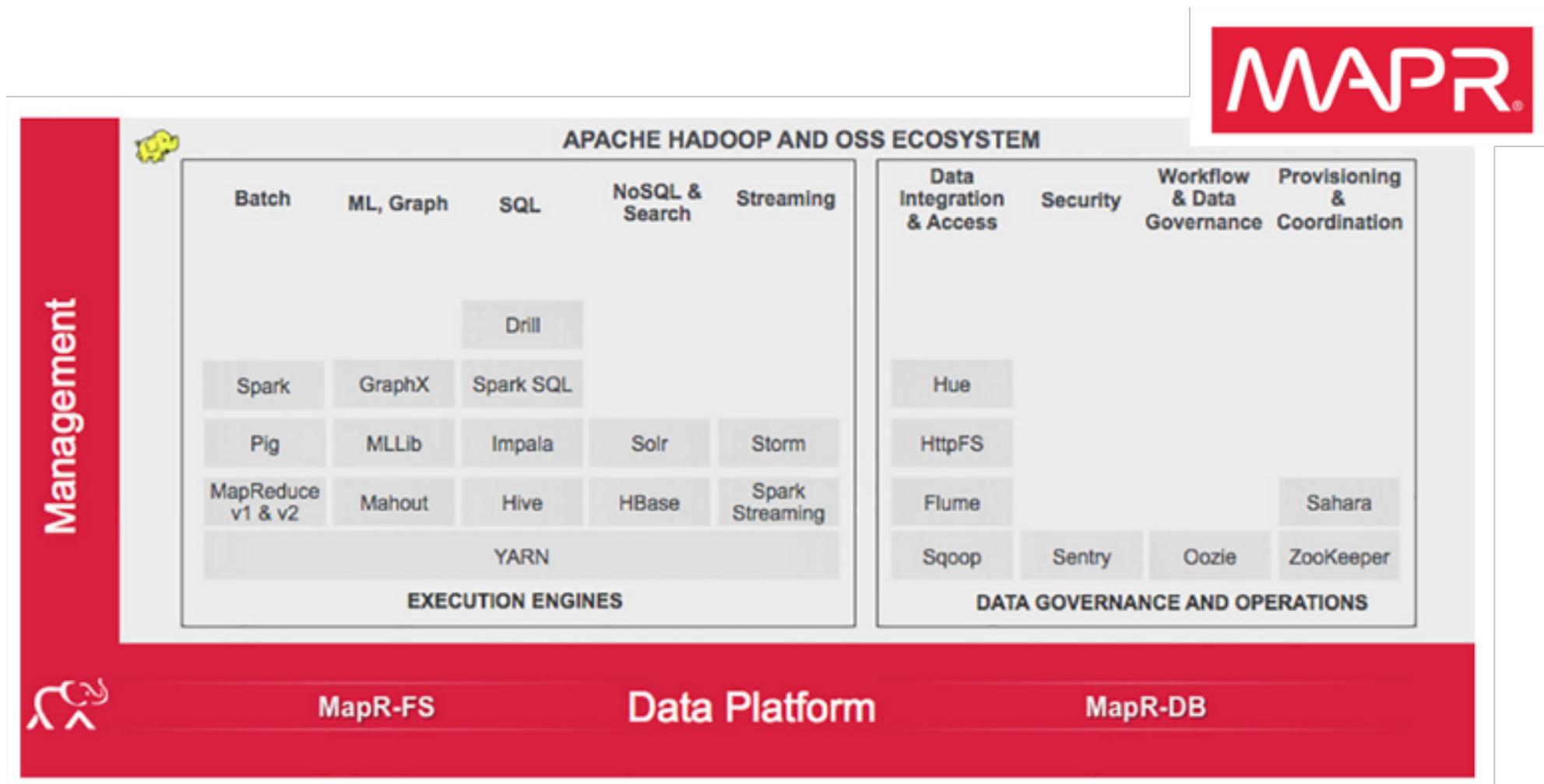
# LES DISTRIBUTIONS D'HADOOP: HORTONWORKS



# LES DISTRIBUTIONS D'HADOOP: CLOUDERA



# LES DISTRIBUTIONS D'HADOOP: MAPR



# SOMMAIRE

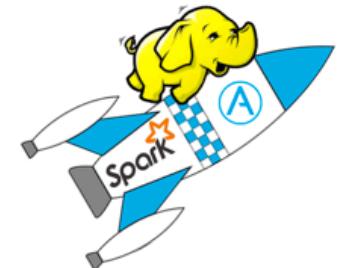
- 1 Big Data & son écosystème
- 2 Hadoop
- 3 Spark
- 4 Conclusion & Questions



# HISTORIQUE DE SPARK

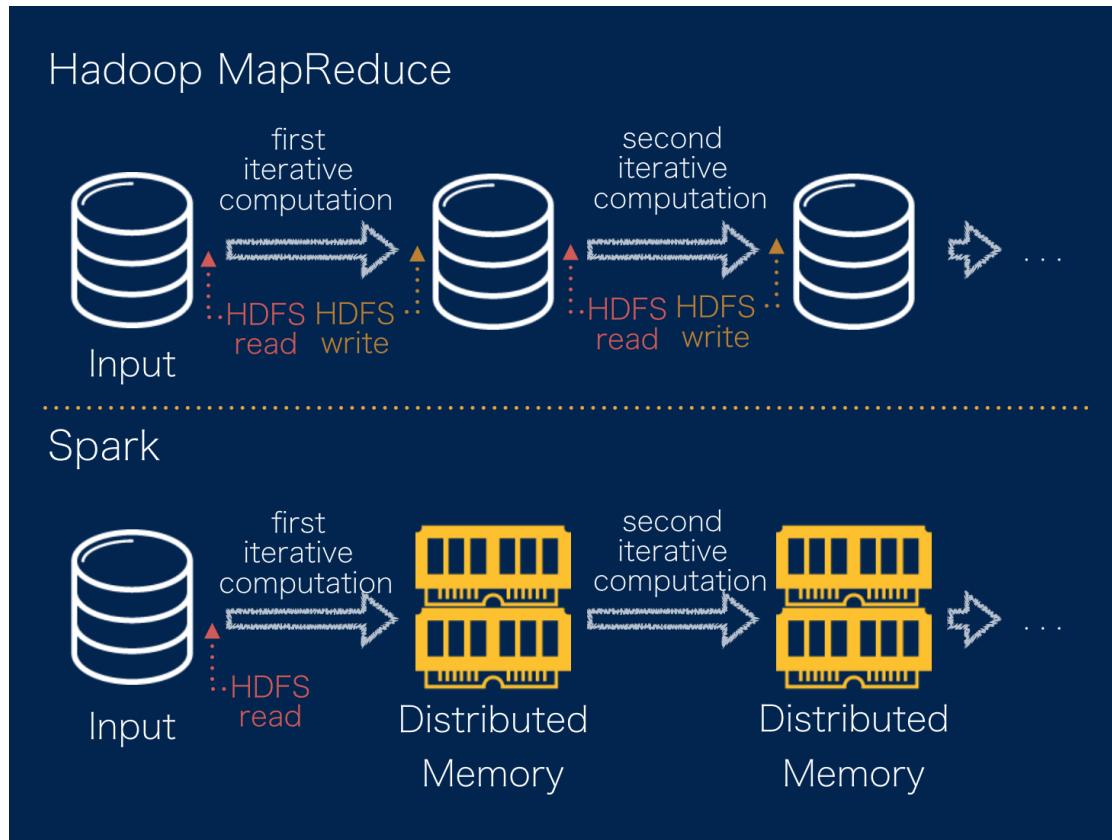
---

- Développé par **AMPLab**, de l'Université UC **Berkeley**, en 2009
- Passé **OpenSource** en 2010 sous forme de projet Apache
  - Release 1.0 – Mai 2014
  - Release 2.0 – mi 2016
- Juin 2013 : **Top Apache Project** (Apache Spark)
- **Extension du modèle MapReduce** (plus performant, in-memory)





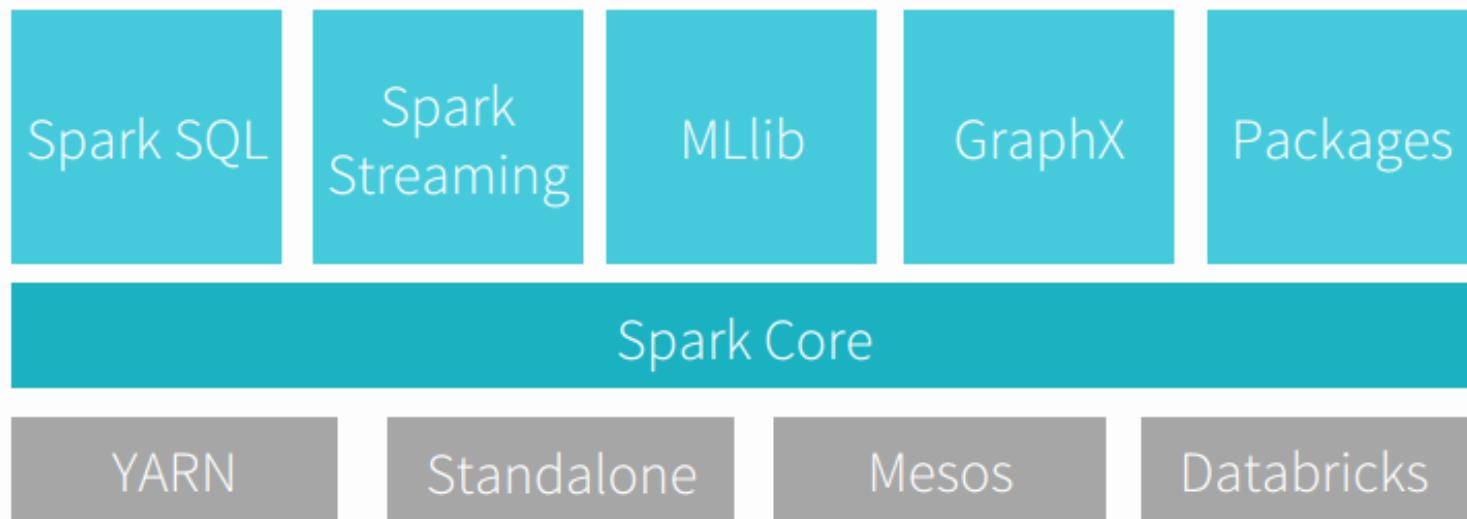
## SPARK VS MAP-REDUCE



**Alternative in-memory plus rapide que MapReduce de Hadoop**  
(100 x plus rapide en mémoire & 10 x plus vite sur disque)



# INTRODUCTION À SPARK



Framework généraliste / API en Scala, Java, Python et R  
Ecosystème riche (SparkSQL, Spark Streaming, MLlib, GraphX)



# RDD

## Resilient Distributed DataSet

- Collection d'objets distribués
  - Structure de donnée **Immutable**
  - **In memory** par défaut
  - Manipulés par des **opérateurs** : transformations / actions
  - **Tolérants aux pannes** : un RDD sait comment recréer et recalculer son ensemble de données
- Transformations**

  - Creation d'un jeu de donnée
  - Lazy par nature. N'est exécuté que lorsque d'une action est effectuée
  - Exemple :
    - Map(func)
    - Filter(func)
    - Distinct()

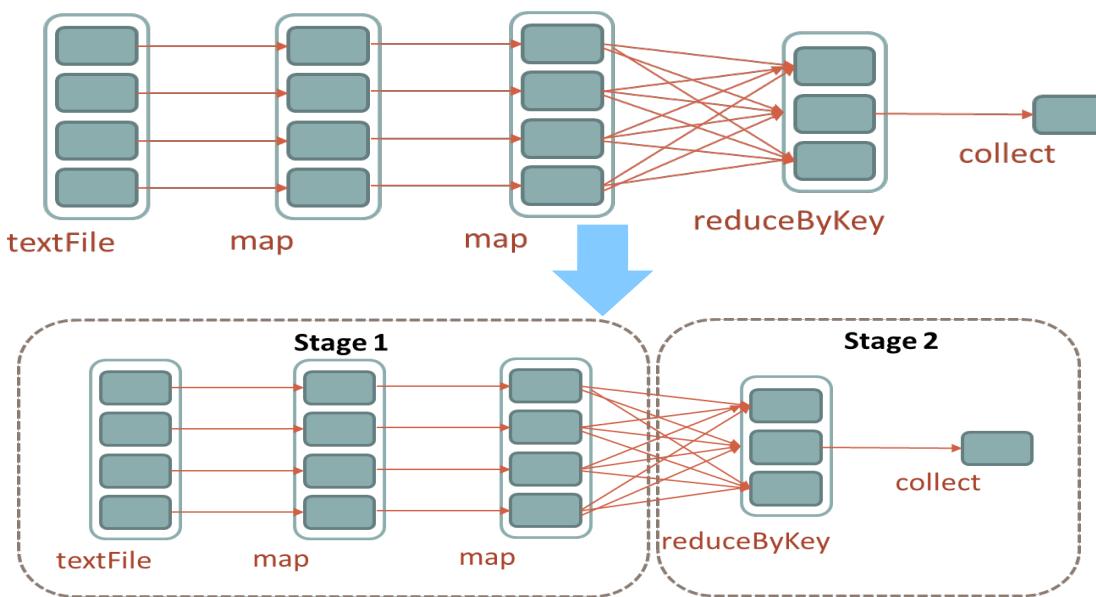
**Actions**

  - Retourne au driver programme une valeur ou exporte les données vers un système de stockage
  - Exemple:
    - Count()
    - Reduce(func)
    - Collect
    - Take()



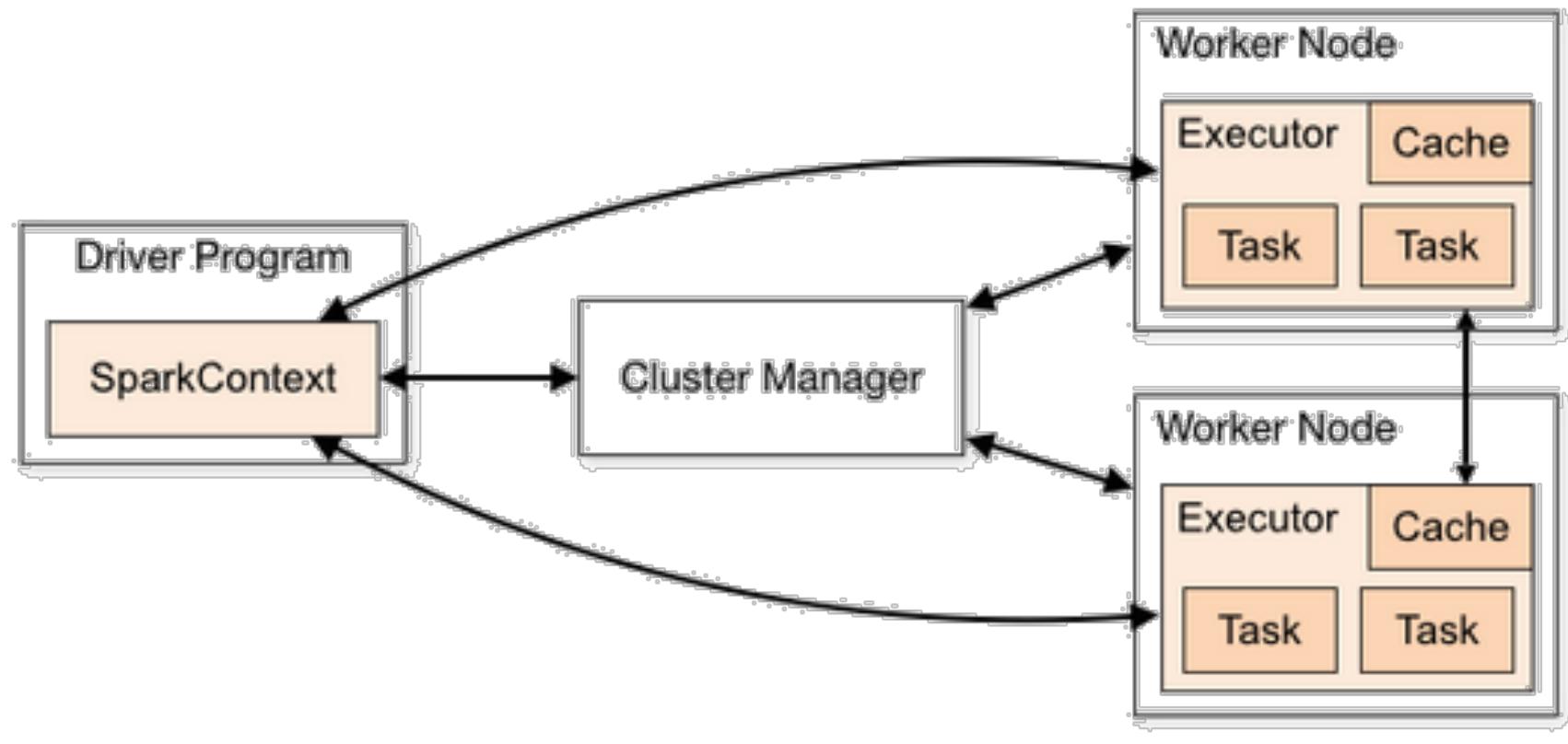
# PLAN D'EXÉCUTION DE SPARK

- Les **taches** sont les unités fondamentales d'exécution
- Les **stages**
  - ensemble de taches qui peuvent être exécutés en parallèles
  - ensemble de séquences de RDD sans Shuffle (tri par clé)
- Le **shuffle** est appliqué entre les stages





# ÉXECUTION DE SPARK



# SOMMAIRE

- 1 Big Data & son écosystème
- 2 Hadoop
- 3 Spark
- 4 Conclusion & Questions

# CONCLUSION

---

## Constat

- *Hadoop* a grandi avec les **géants du web**
- **Écosystème très riche** avec éclosion de nouvelles technologies
- Marché dynamisé par **l'open source & les startups**

## Nouveaux challenges

- Les **distributions** vs **Hadoop-as-a-Service**
- La **gouvernance** des données et la **sécurité**
- Démocratisation des outils **Machine Learning**