



# BIG DATA

# Big Data: Pourquoi ?



Données



Frameworks



Plateformes



Analyse



Intelligence  
artificielle

# Big Data: Contexte



Nouvelles solutions « open source »



Nouveaux challenges autours de la « data »



Nouveaux besoins IT :

Data engineer

Data analyste

Data scientist

DevOps engineer

# Big Data: Problématiques et challenges !

- Des données hétérogènes : structurées, non structures
- Différentes sources de données : base de données, fichiers, streaming
- Données non fiables et manquantes
- Sécurisation des données : droit sur les données, droit d'accès, intégrité des données, données anonymisés...
- Prétraitement des données :
  - Extraction, chargement, transformation, visualisation → Mieux comprendre ses données
- Analyse prédictive : prédiction, classification, clustering
  - Trouver des patterns dans des données métier
  - Automatiser des process : Détection d'anomalie..
  - Créer des nouveaux modèles pour répondre aux nouveaux besoins

# Big Data: Définition

- « Big data » comme le ‘small data’, mais en plus grand et plus large
- Avoir des données plus grandes nécessite des approches différentes de résolution : technologies et architectures
- De nouveaux paradigmes pour une résolution des problèmes d'une manière plus efficace
- « Big data » génère de la valeur à partir du stockage et du calcul de grandes quantités de données numériques
- « Big data » c'est l'analyse et le calcul d'une sur des données massives

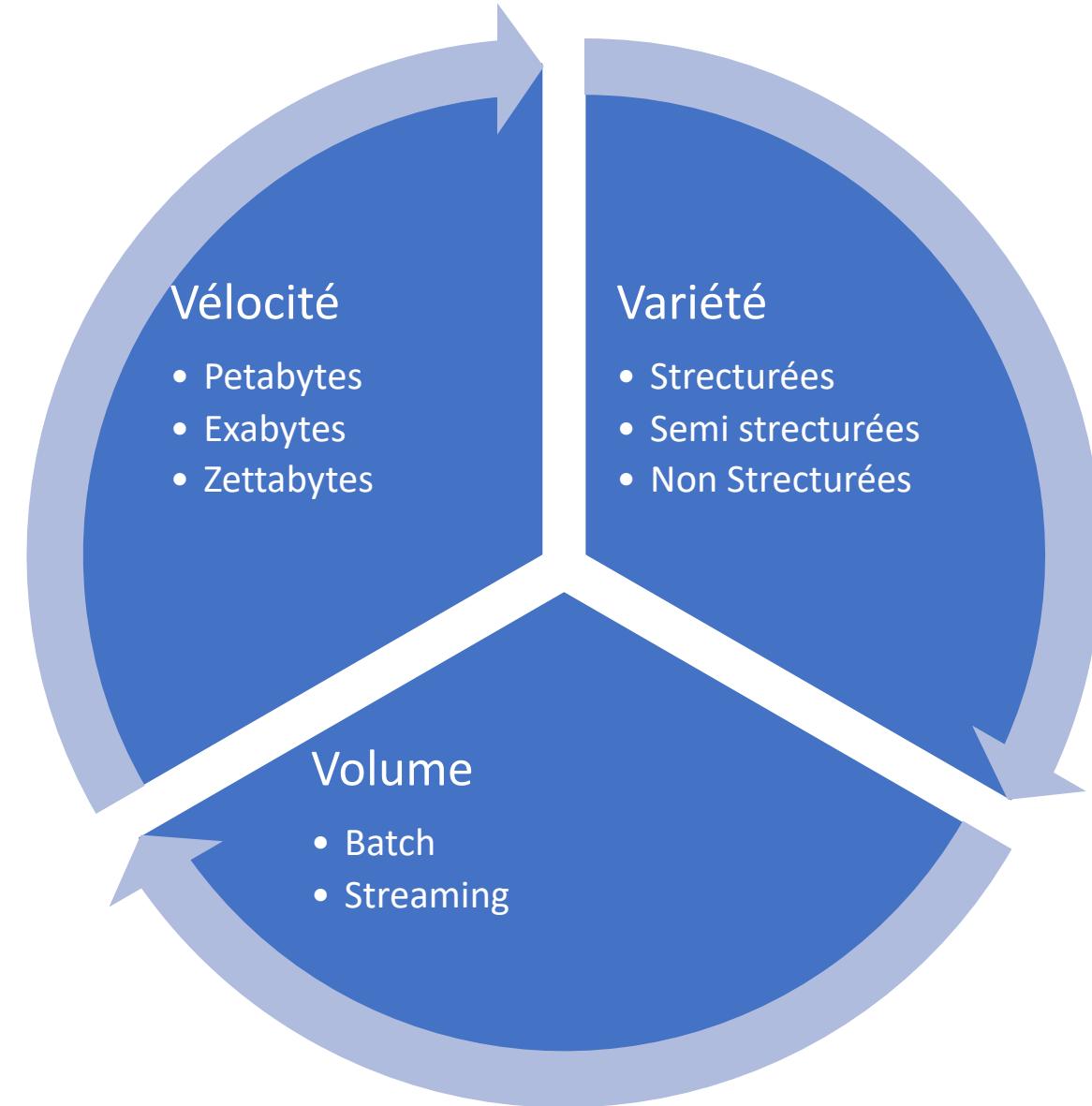
# Big Data: Objectif

Comme d'autres paradigme et technologies d'informations, le « big data » a pour but:

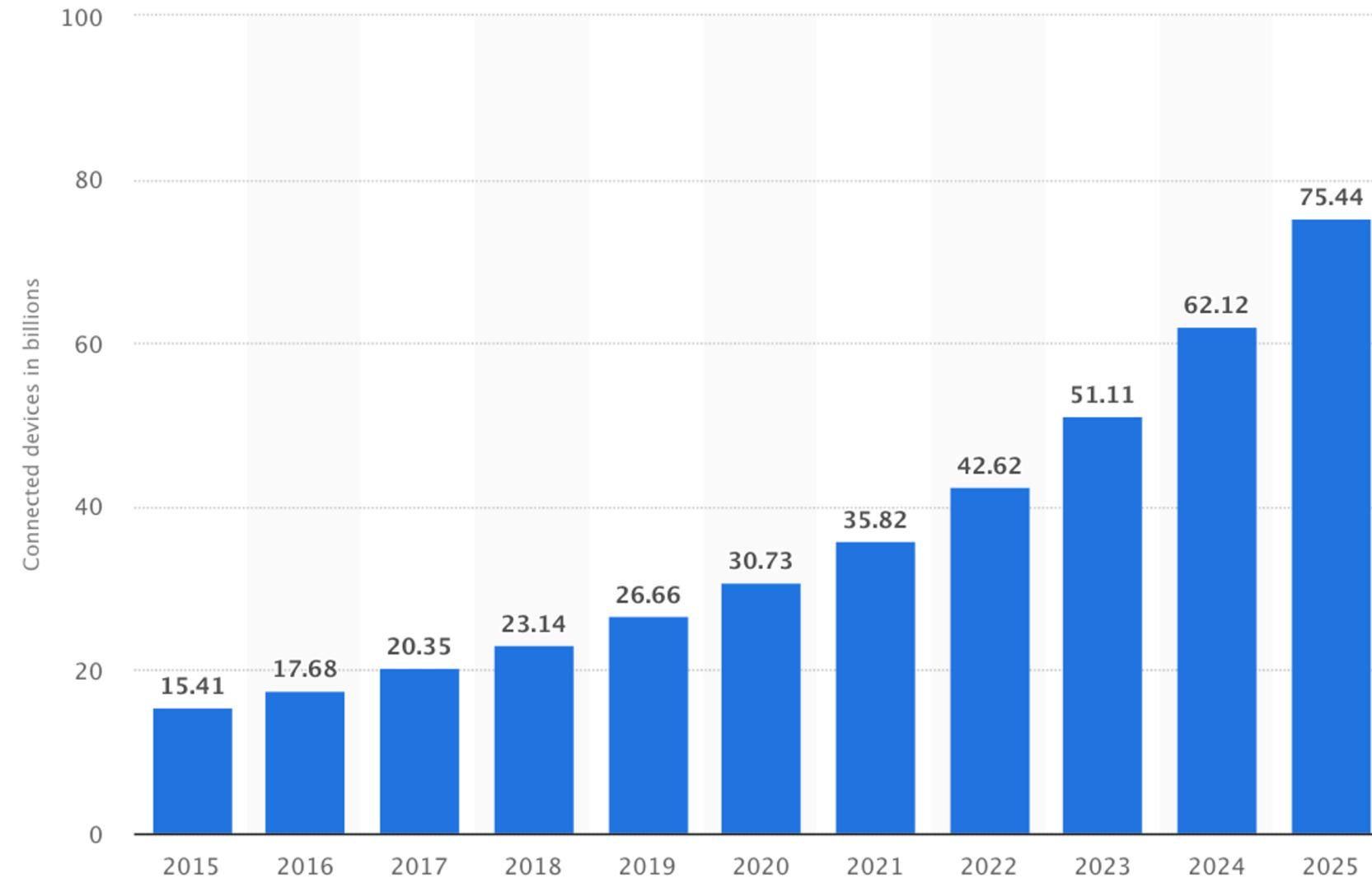
- Réduire les coûts de la gestion des données: sauvegarde et traitement
- Améliorer le temps d'exécution nécessaire au traitement
- Trouver et identifier de nouveaux services

data

# Big Data: 3V

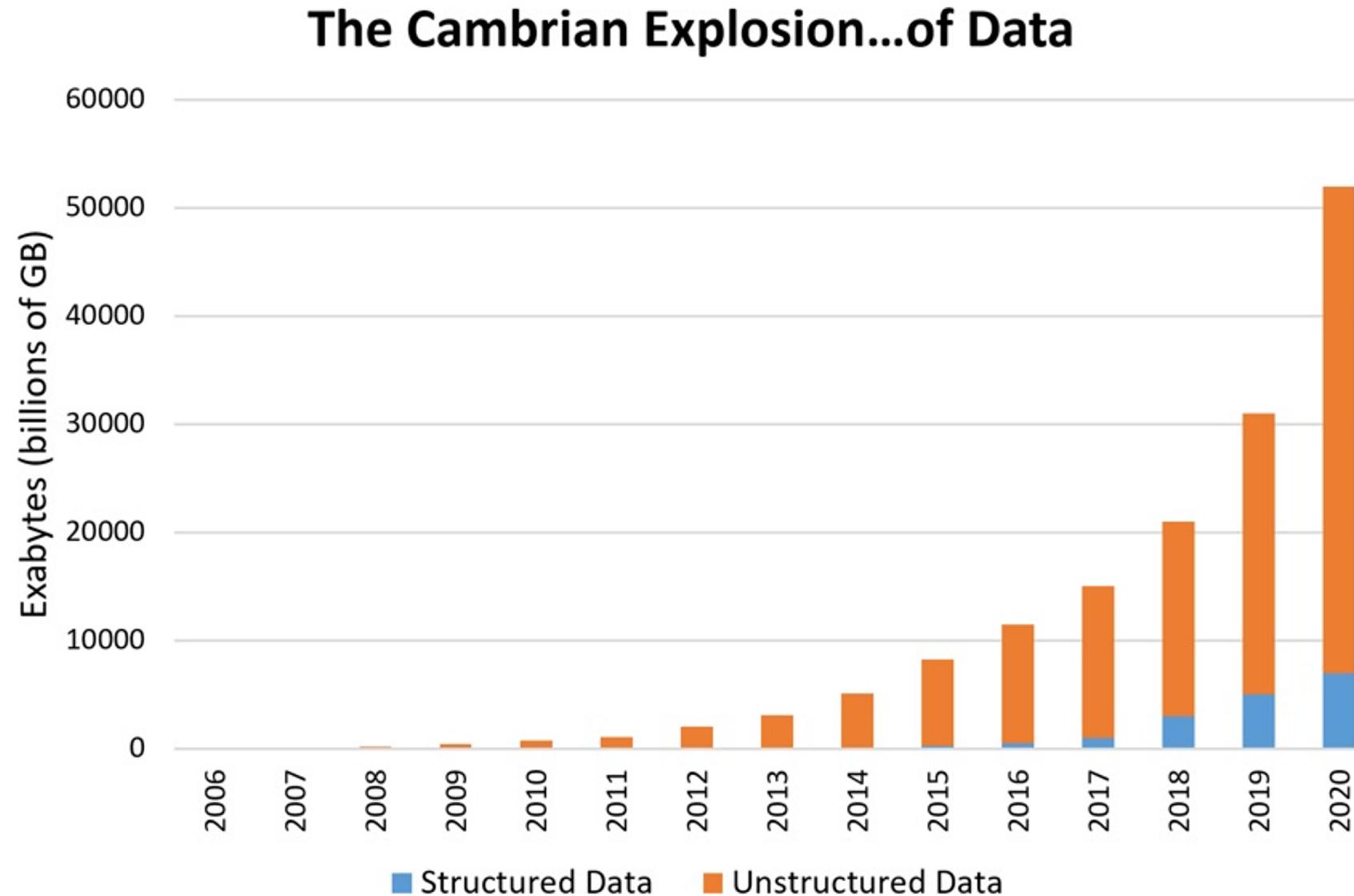


# Big Data: Evolution des objets connectés



Source: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>

# Big Data: Evolution des objets connectés



Source : [https://www.eetimes.com/author.asp?section\\_id=36&doc\\_id=1330462](https://www.eetimes.com/author.asp?section_id=36&doc_id=1330462)

# Big Data: Volume

- Coût moins élevé du stockage: disques moins chères, solution de stockage open source : stockage objet, NFS..
- Coût de la maintenance de l'infrastructure moins élevé: automatisation de déploiement et du maintien en condition opérationnelle : sauvegarde
- Accessibilité des appareils numériques : plus de photos et vidéos, meilleure qualité de photos et des vidéos  
→ plus de stockage
- Émergence des objets connectés : montres connectés, domotiques, voitures connectées, circuit RFID, smartphones

# Big Data: Vélocité

- Evolution exponentielle de la quantité des données  
(utilisateurs facebook upload plus de 900 millions de photos par jour)
- Flux de données issue de navigations web: user tracking et moteur de recommandation
- Communication et échanges machine à machines, entre des milliards d'objets
- L'infrastructure IT et les capteurs génèrent des données massives en temps réel
- Les jeux en ligne et les systèmes supportant des millions de connexions parallèles

# Big Data: Variété

- Big data, il ne s'agit pas que de données structurées. Big data est aussi des données géospatiales, données 3D, vidéos, logs, streaming...
- Plusieurs types de données
  - données structurées : base de données, csv, parquet
  - données semi-structurées : xml, json, log
  - données non-structurées : documents, vidéos..



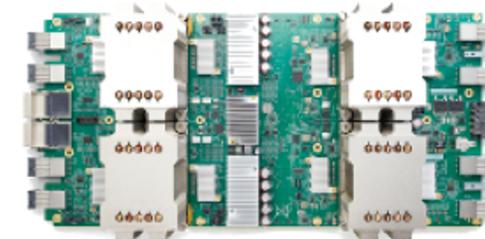
# Big Data: Computing



CPU

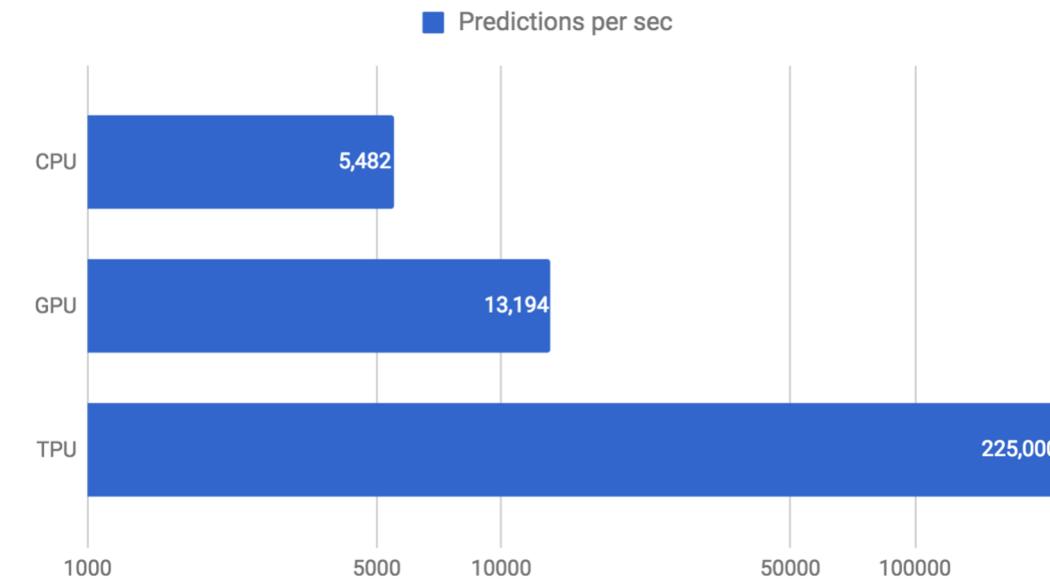
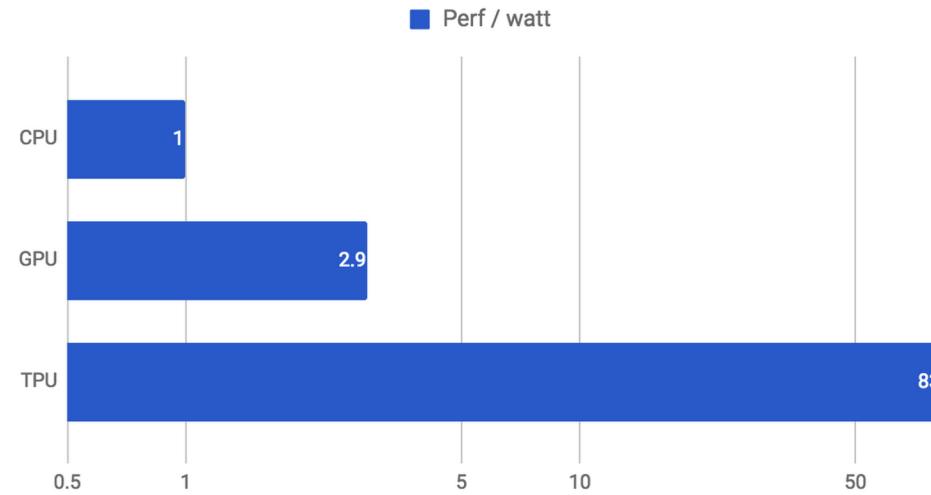


GPU



TPU

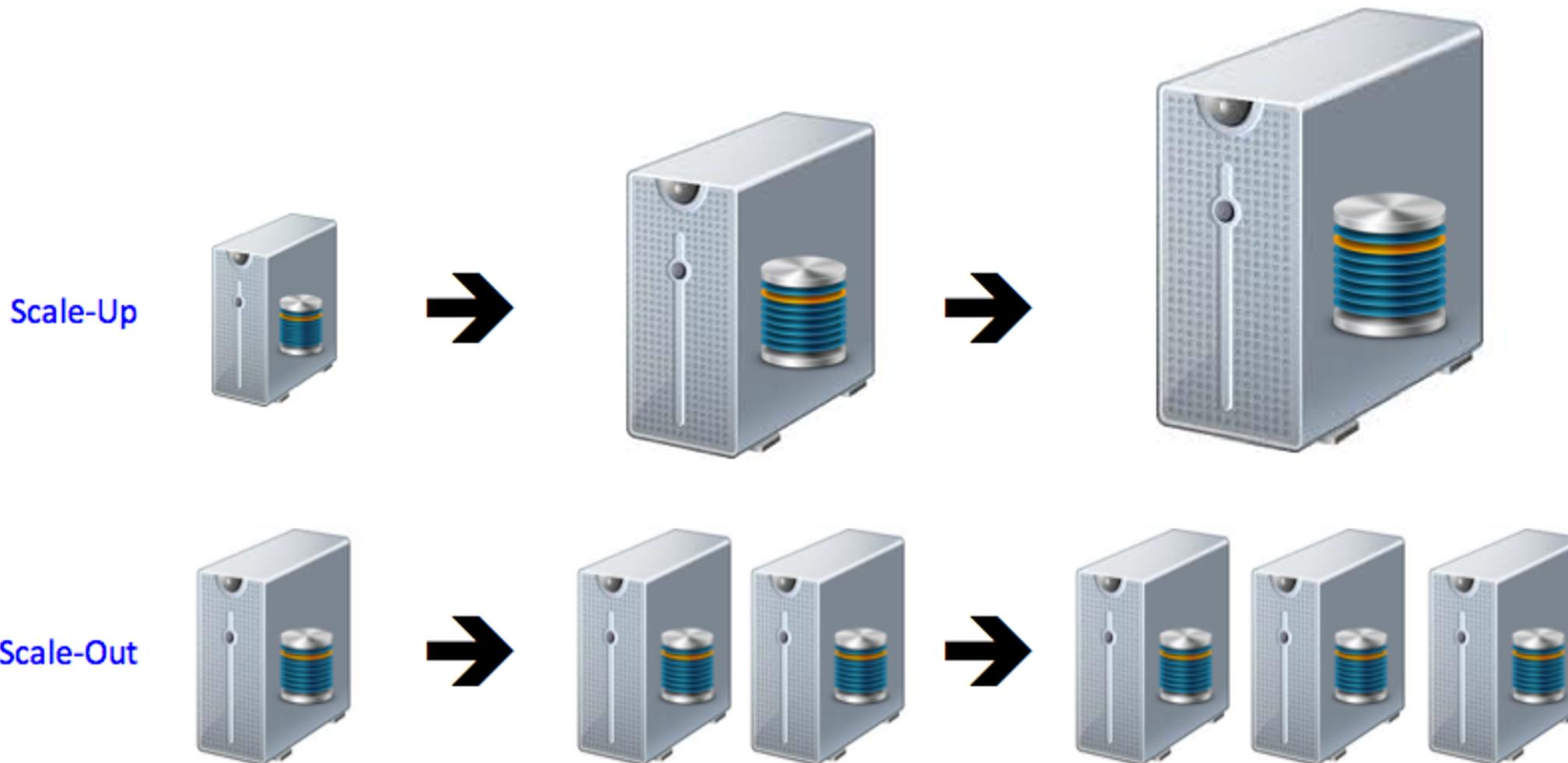
# Big Data: Computing



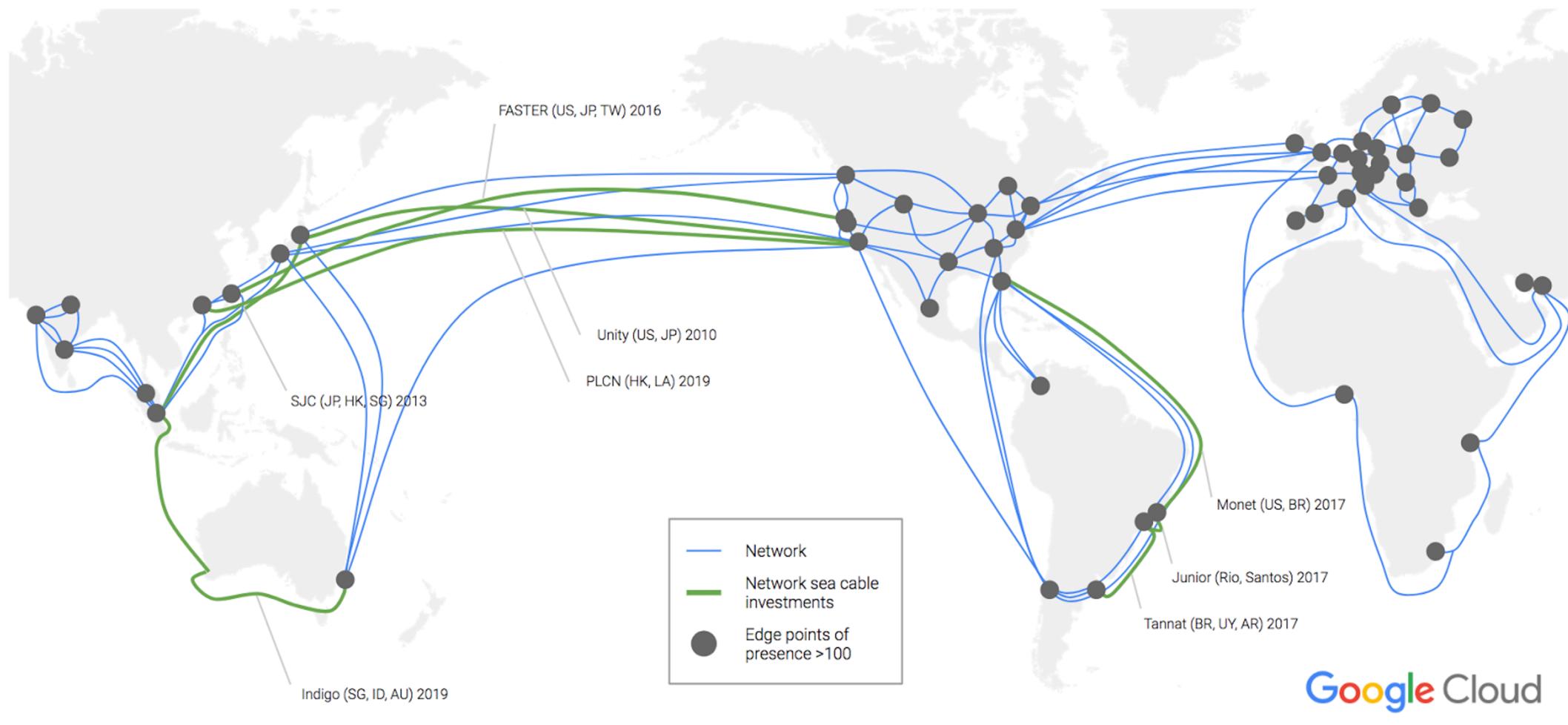
Source:

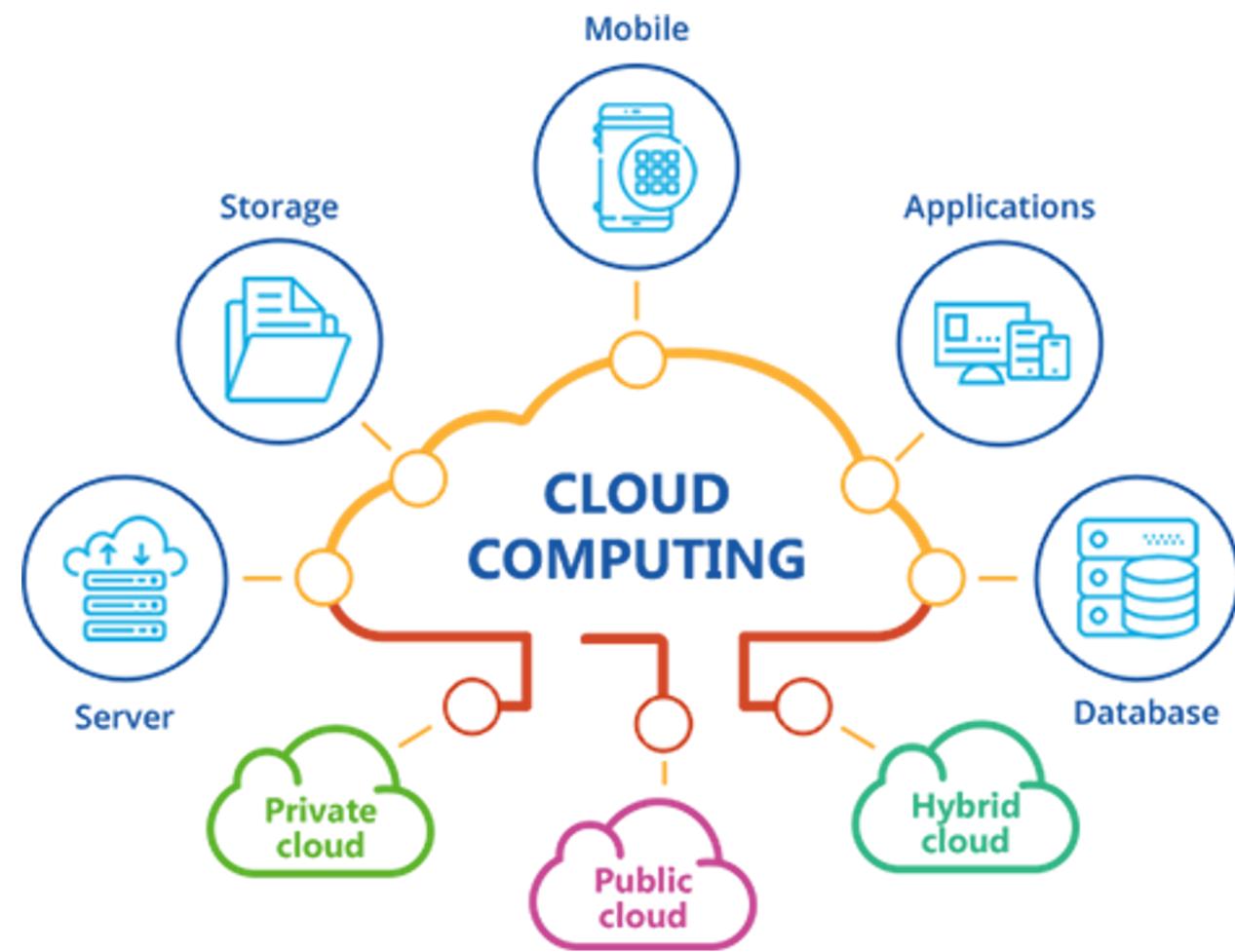
<https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu>

# Big Data: Scale



# Big Data: Réseau



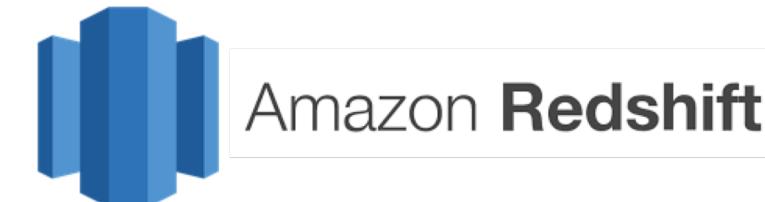


# Big Data: Cloud Computing

- Infrastructure disponible immédiatement
- Infrastructure as a service
- Des services:
  - Managés
  - Sécurisés
  - Résilients
  - Avec haute disponibilités
- « Pay as you go »



Google  
Big Query



Amazon **Redshift**

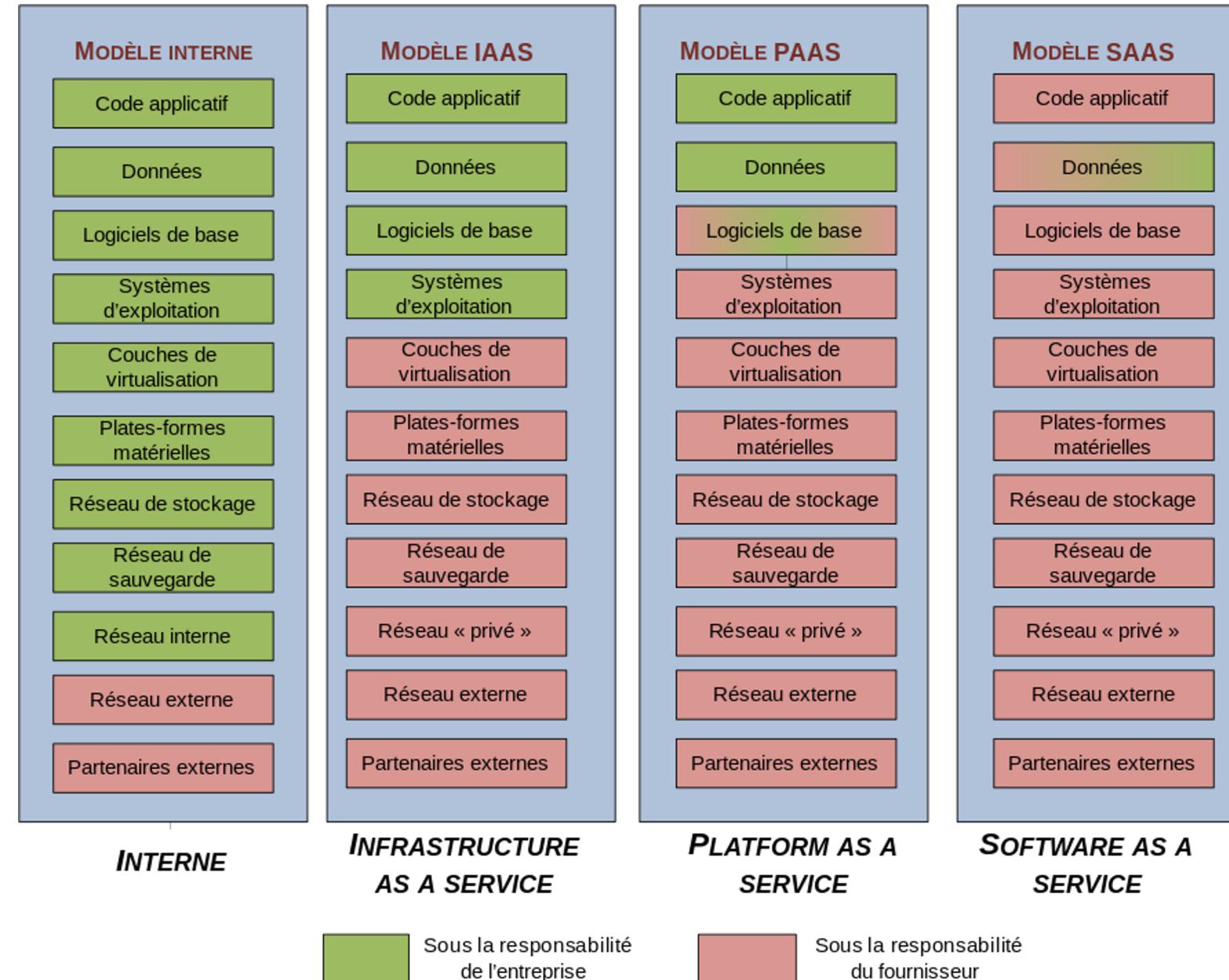


amazon  
S3



AWS Lambda

# Big Data: Cloud Computing



# Big Data: Stockage

- Stockage type objet <clé, objet>
- Serverless : aucune infrastructure à gérer
- Objet sauvegardé sous des « buckets »
  - Structure plate de données
  - Données non modifiables
  - Données non lisible
  - Opérations acceptées : Get, Put
- Exemples de services:
  - Google Cloud Storage : GCS
  - Amazon Simple Storage Services : S3



# Big Data: Analytique

## Google BigQuery :

BigQuery est l'entrepôt de données d'entreprise sans serveur et hautement adaptable de Google, conçu pour augmenter la productivité de des analystes de données

Il permet d'analysez toutes les données en créant un entrepôt logique sur un stockage en colonnes géré, ainsi que des données issues d'espaces de stockage d'objets et de feuilles de calcul

The screenshot shows the Google BigQuery web interface. On the left, there's a sidebar with 'COMPOSE QUERY' buttons for 'Query History' and 'Job History'. Below that is a tree view under 'funnel\_demo' containing numerous tables like 'funnel\_aggregated\_all\_dates', 'funnel\_overview\_2015\_01', etc. The main area is titled 'New Query' and contains a SQL code block:

```
1 WITH ad_platforms AS
2   (SELECT "twitter-demo" AS sourceTypeId, "Twitter" AS sourceTypeName
3    UNION ALL
4    SELECT "bing-demo" AS sourceTypeId, "Bing" AS sourceTypeName
5    UNION ALL
6    SELECT "adwords-demo" AS sourceTypeId, "Google" AS sourceTypeName)
7 SELECT date, if(sourceTypeName is not null, sourceTypeName, sourceType), sourceName, common_campaign, common_cost/1000
8 FROM `firebase-funnel.funnel_demo.funnel_overview_*`
9 LEFT OUTER JOIN ad_platforms a on a.sourceTypeId = sourceType
```

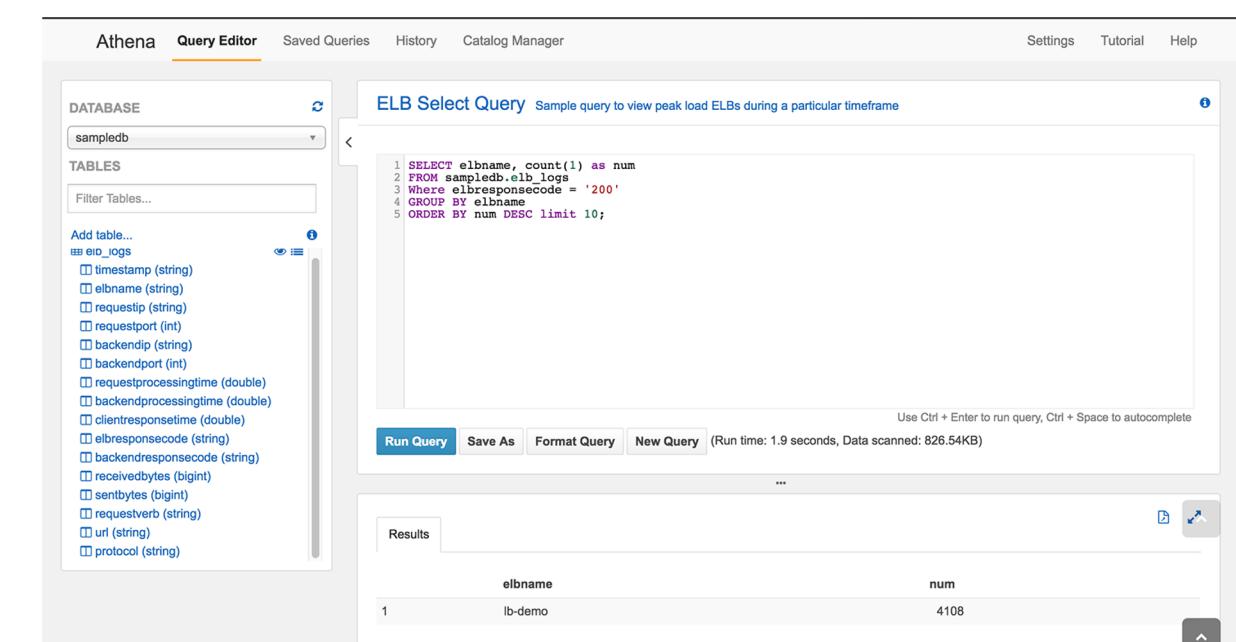
Below the query editor are sections for 'Destination Table' (with 'Select Table' and 'No table selected' options), 'Write Preference' (radio buttons for 'Write if empty', 'Append to table', and 'Overwrite table'), 'Results Size' (checkbox for 'Allow Large Results'), 'Results Schema' (checkbox for 'Flatten Results'), 'Query Caching' (checkbox for 'Use Cached Results'), 'Query Priority' (radio buttons for 'Interactive' and 'Batch'), 'UDF Source URIs' (button for 'Edit'), 'Maximum Billing Tier' (dropdown for 'Project Default'), 'Maximum Bytes Billed' (dropdown for 'Project Default'), and 'SQL Dialect' (checkbox for 'Use Legacy SQL'). At the bottom are buttons for 'RUN QUERY', 'Save Query', 'Save View', 'Format Query', and 'Hide Options'. A status message says 'Query complete (8.8s elapsed, 42.6 MB processed)'. The results table at the bottom has columns: Row, date, f0\_, sourceName, common\_campaign, cost, common\_clicks, common\_impressions. One row is shown: 1 | 2016-02-29 | Bing | Fashion Retailer DE | [S] Brand | 6.264735 | 170.0 | 689.0.

# Big Data: Analytique

## Amazon Athena

Amazon Athena est un service de requête interactif qui facilite l'analyse des données dans Amazon S3 à l'aide de la syntaxe SQL standard

Athena fonctionne sans serveur. Il n'existe aucune infrastructure à gérer et vous ne payez que pour les requêtes que vous exécutez.



The screenshot shows the Amazon Athena Query Editor interface. At the top, there are tabs for 'Athena' and 'Query Editor' (which is currently selected), along with links for 'Saved Queries', 'History', and 'Catalog Manager'. On the far right, there are 'Settings', 'Tutorial', and 'Help' links. The main area is divided into two sections: 'DATABASE' and 'TABLES'. Under 'DATABASE', 'sampledb' is selected. Under 'TABLES', there is a list of columns from a table named 'elb\_logs': timestamp (string), elbname (string), requestip (string), requestport (int), backendip (string), backendport (int), requestprocessingtime (double), backendprocessingtime (double), clientresponsetime (double), elbresponsecode (string), backendresponsecode (string), receivedbytes (bigint), sentbytes (bigint), requestverb (string), url (string), and protocol (string). To the right of this is the 'ELB Select Query' section, which contains a sample query:

```
1 SELECT elbname, count(1) as num
2 FROM sampledb.elb_logs
3 Where elbresponsecode = '200'
4 GROUP BY elbname
5 ORDER BY num DESC limit 10;
```

Below the query is a status bar with 'Run Query', 'Save As', 'Format Query', and 'New Query' buttons, and a note '(Run time: 1.9 seconds, Data scanned: 826.54KB)'. At the bottom, the 'Results' section displays the query results:

	elbname	num
1	lb-demo	4108

# Big Data: Machines virtuelles

Une machine virtuelle, ou VM (Virtual Machine), est un environnement d'application ou de système d'exploitation (OS, Operating System) installé sur un logiciel qui imite un matériel dédié

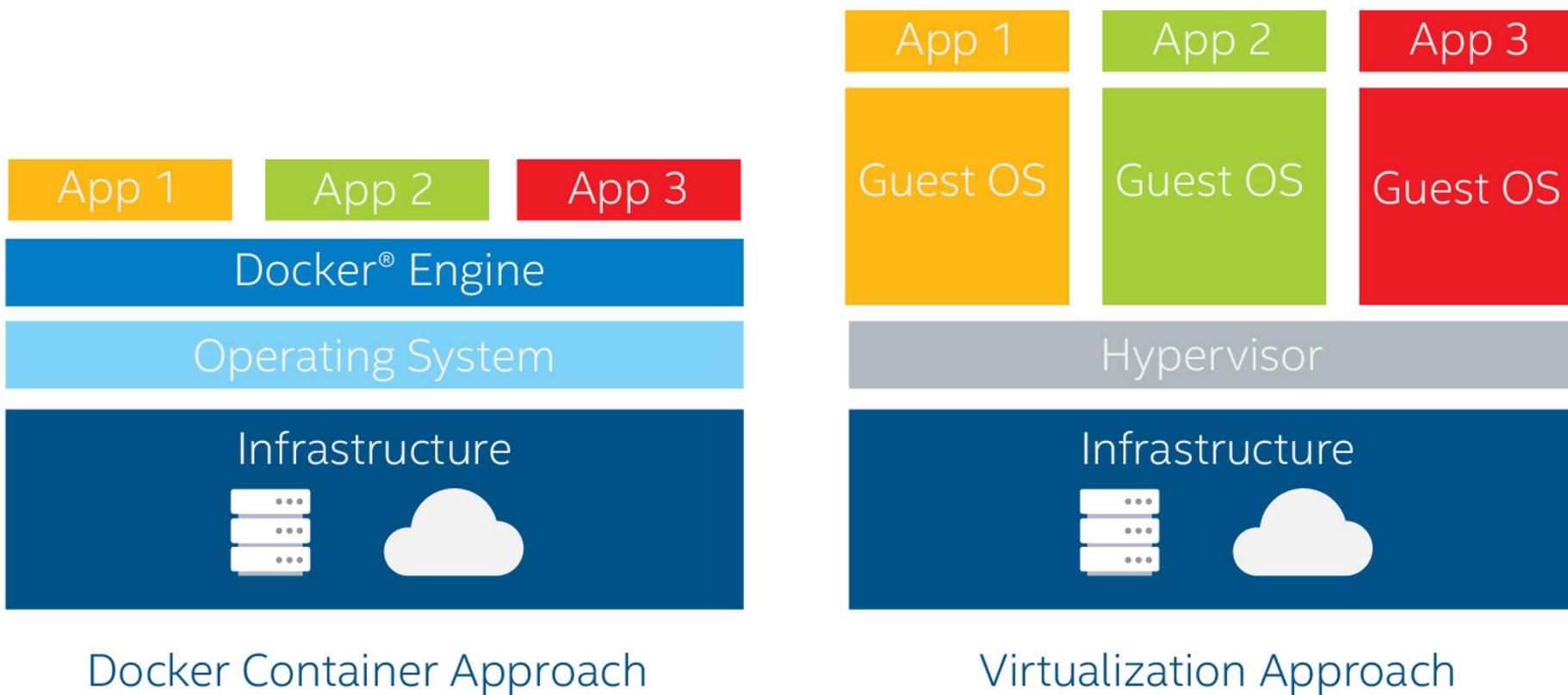
- Haute disponibilité & Accessibilité
- Infrastructure as a code
- Caractérisé par : Taille vCpu, memoire
- Sécurité: Attaque, réseau

Exemples de services:

- Amazon EC2
- Google GCE

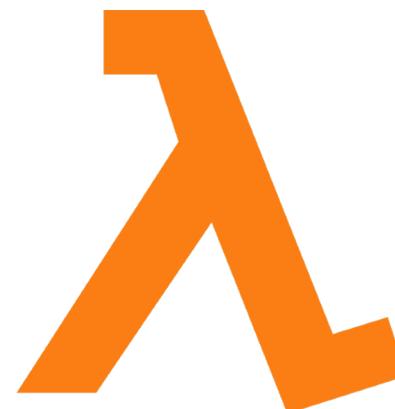


# Big Data: Docker



# Big Data: Lambda & Cloud functions

- « AWS Lambda » est un service de calcul sans serveur qui exécute votre code en réponse à des événements et gère automatiquement les ressources de calcul sous-jacentes.
- « Google Cloud Function » est une solution de calcul qui permet aux développeurs de créer des fonctions autonomes à but unique qui répondent aux événements, sans qu'il soit nécessaire de gérer un serveur ou un environnement d'exécution.





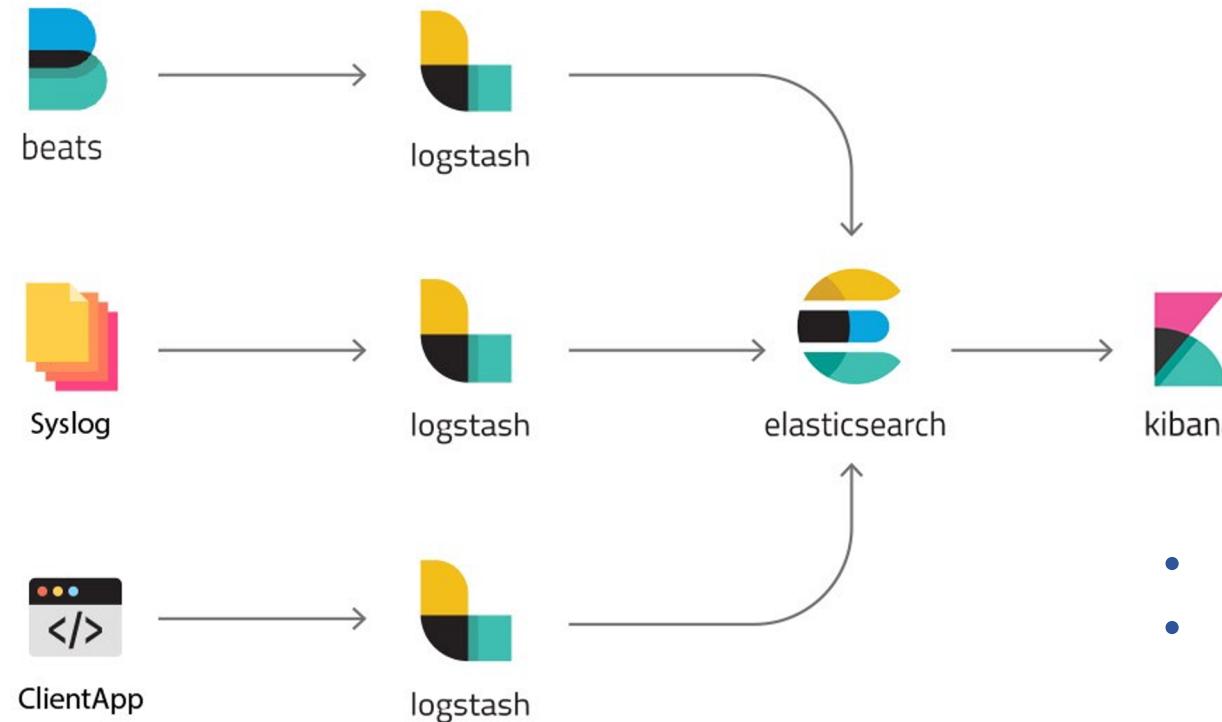
# Big Data: Base de données NoSQL

- Pas de structure en écriture
- Structure en lecture
- Haute disponibilité
- Faible latency
- Format Json
- Scalabilité horizontale
- Résilientes aux pannes hardware
- !! Ne permettes pas de faire des jointures
- !! Opération unitaire et pas opération ensembliste



# Big Data: ElasticSearch ELK

- Moteur de recherche et d'indexation
- Distribué et Répliqué
- Restful



- Visualisation
- Requêteage

- Collecte des données
- Sérialisation

# Big Data: ElasticSearch ELK



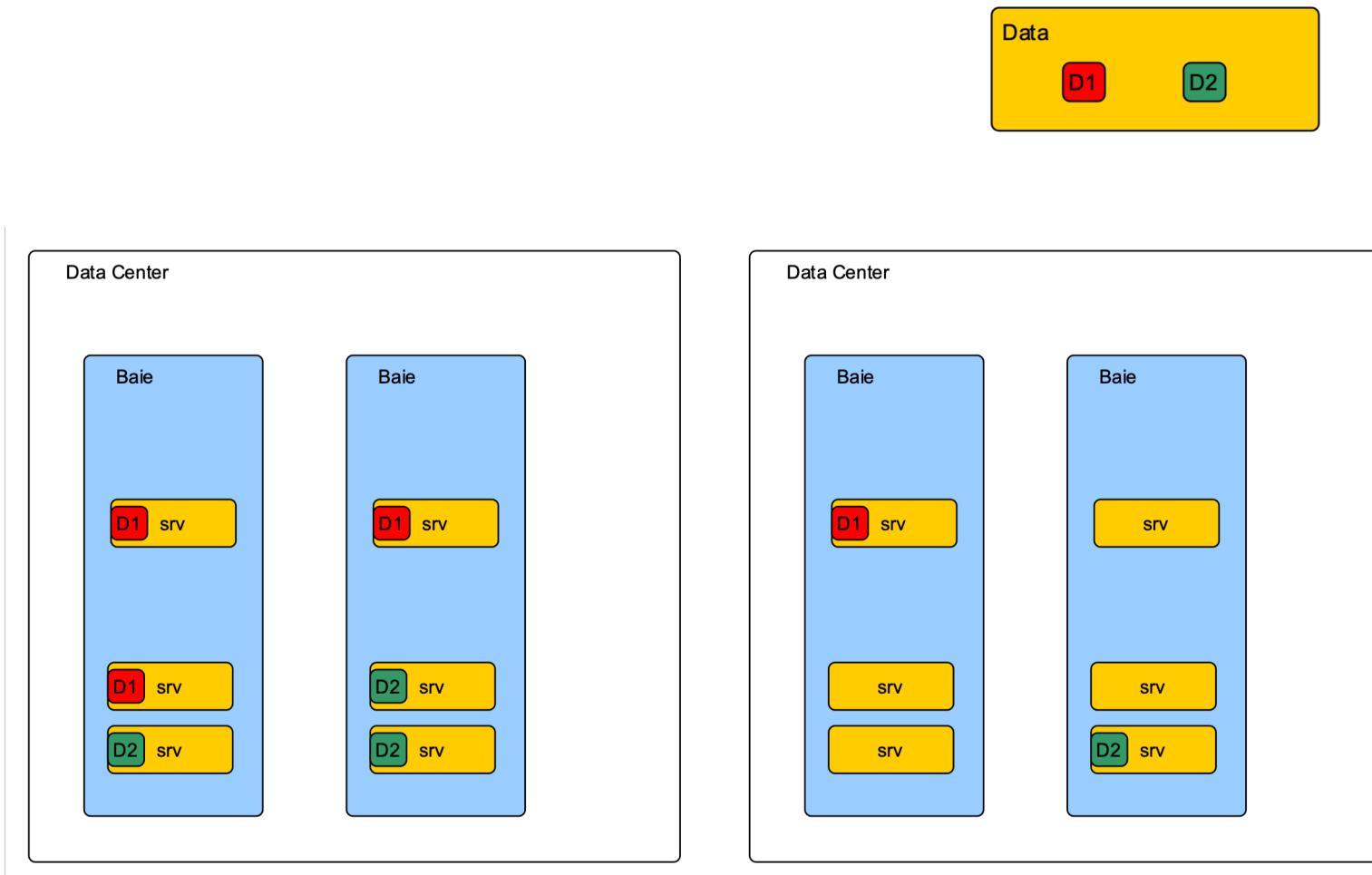
# Big Data: Hadoop

- Hadoop est un framework de stockage et de calcul de données distribuées
- Hadoop utilise des serveurs à usage générique
- Le core de hadoop est composé de :
  - HDFS hadoop distributed file system : est système de stockage de données distribués et réparties
  - MapReduce : Nouveau paradigme de développement, qui consiste à lancer des mapper, reducer
    - Mapper : charger les données et construire un dictionnaire à clé valeur.
    - Reducer : utiliser la sortie du mapper pour agréger les données

# Big Data: Hadoop Histoire

- Google file system introduit par google en 2003
- Paradigme MapReduce introduit par Google en 2004
- Première version communautaire du projet Apache Hadoop en 2006 utilisant :
  - MapReduce
  - HDFS
- Première implémentation dans un data center par yahoo en 2006

# Big Data: Hadoop HDFS



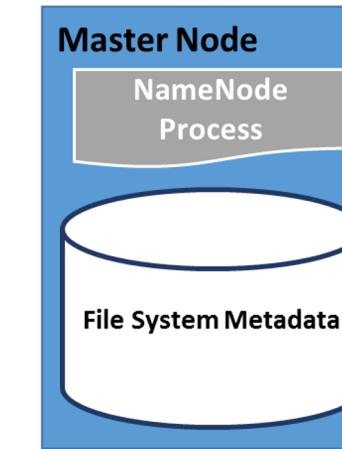
# Big Data: Hadoop

- MapReduce est un paradigme de programmation
- Fonctionne exclusivement sur le couple <clé, valeur>
- L'entrée et la sortie d'un job sont toujours sous la forme <clé, valeur>
- La clé et la valeur doivent être sérialisables
- La séquence d'un job MapReduce :

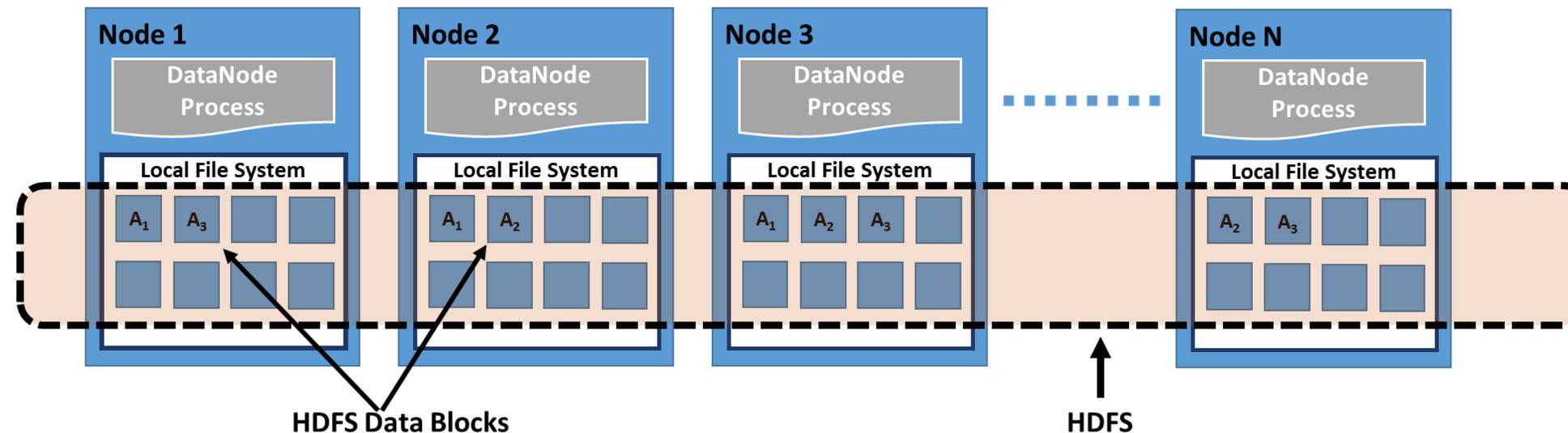
(input) <k1,v1> → map → <k2,v2> → shuffle → <k2,v2> → reduce → <k3,v3> (output)

# Big Data: Hadoop Architecture

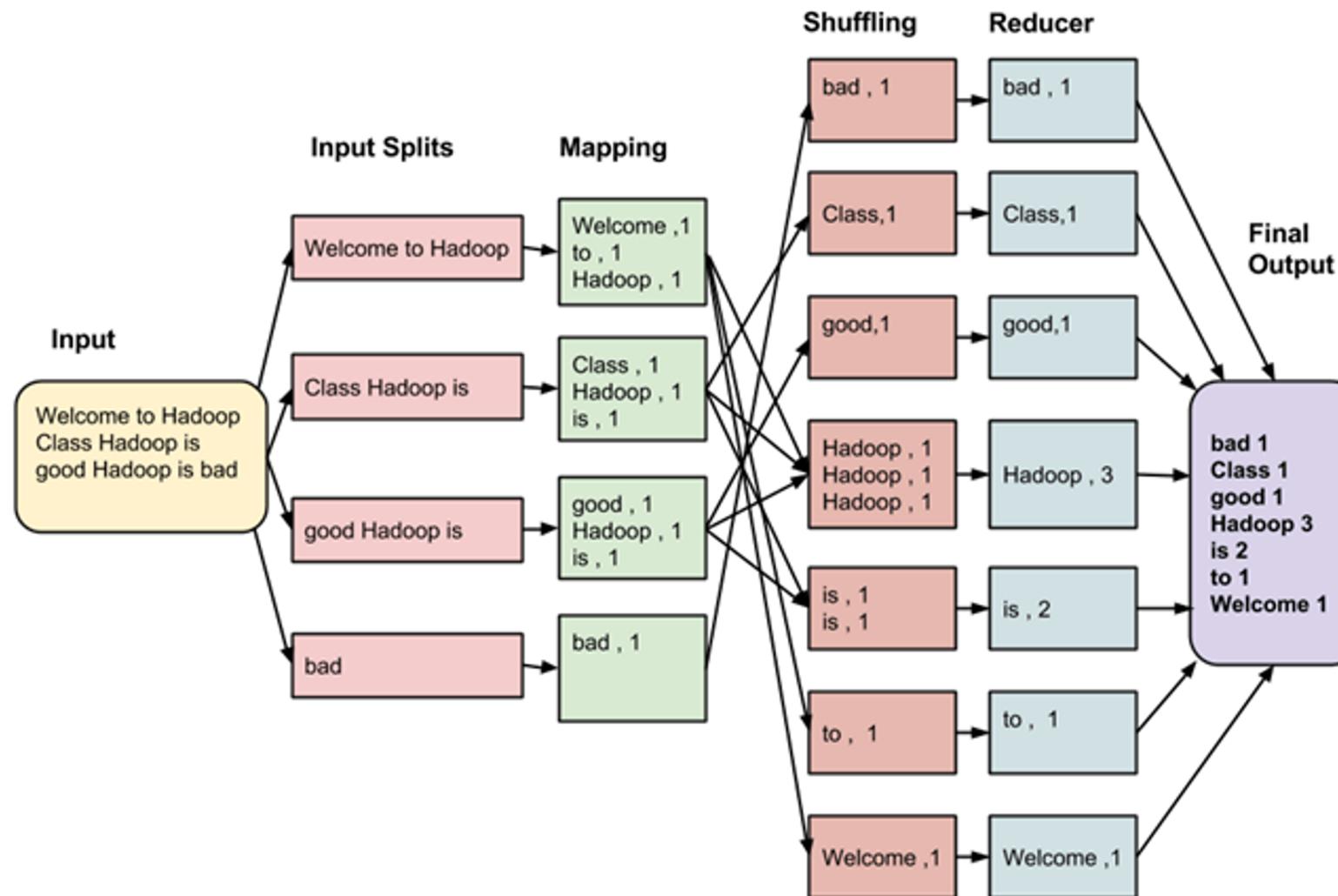
**MapReduce** est un framework de calcul distribué  
Le calcule se fait au plus près de la données



**MapReduce → YARN**  
Mapper, Shuffle, Reduce



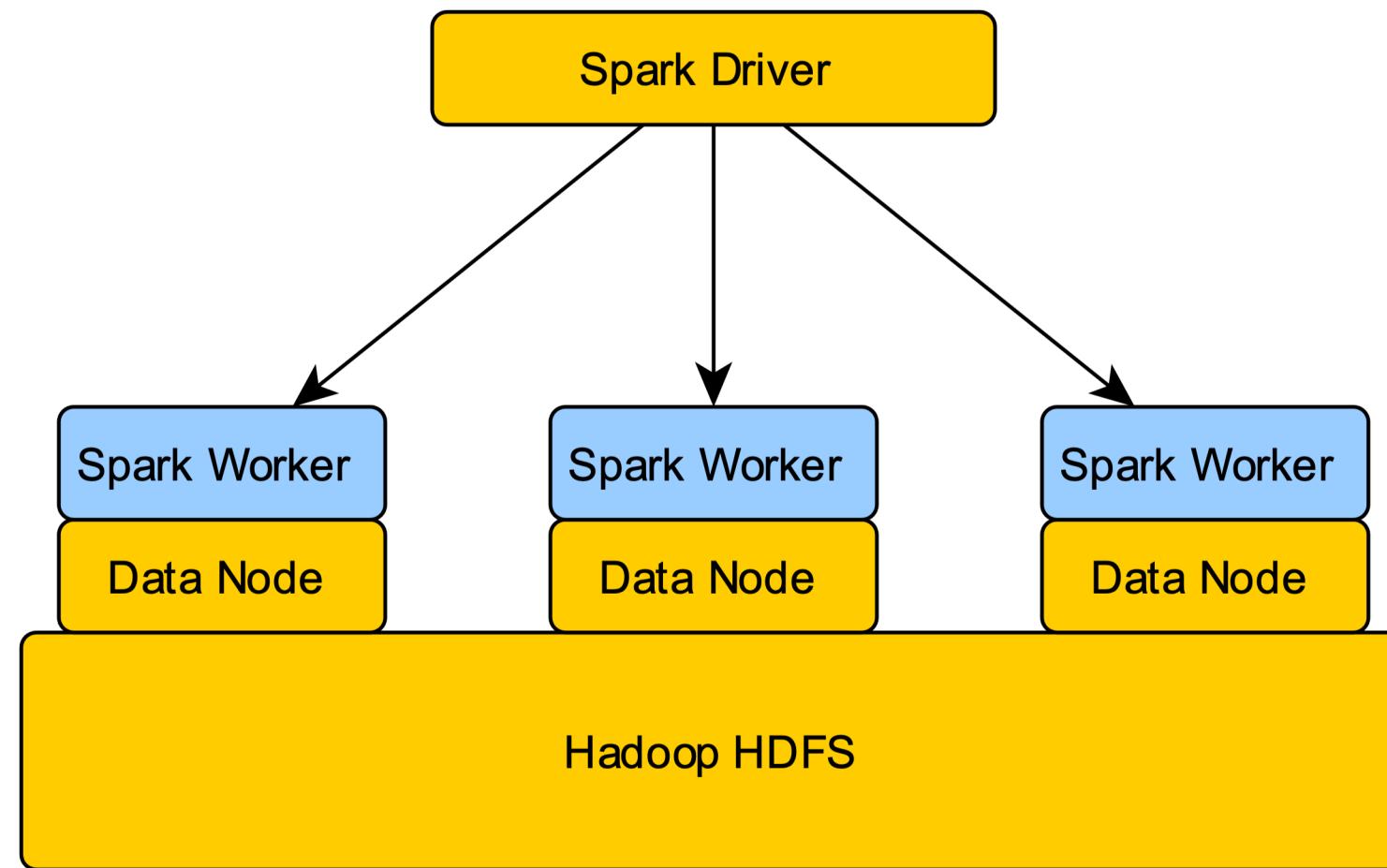
# Big Data: Hadoop MapReduce



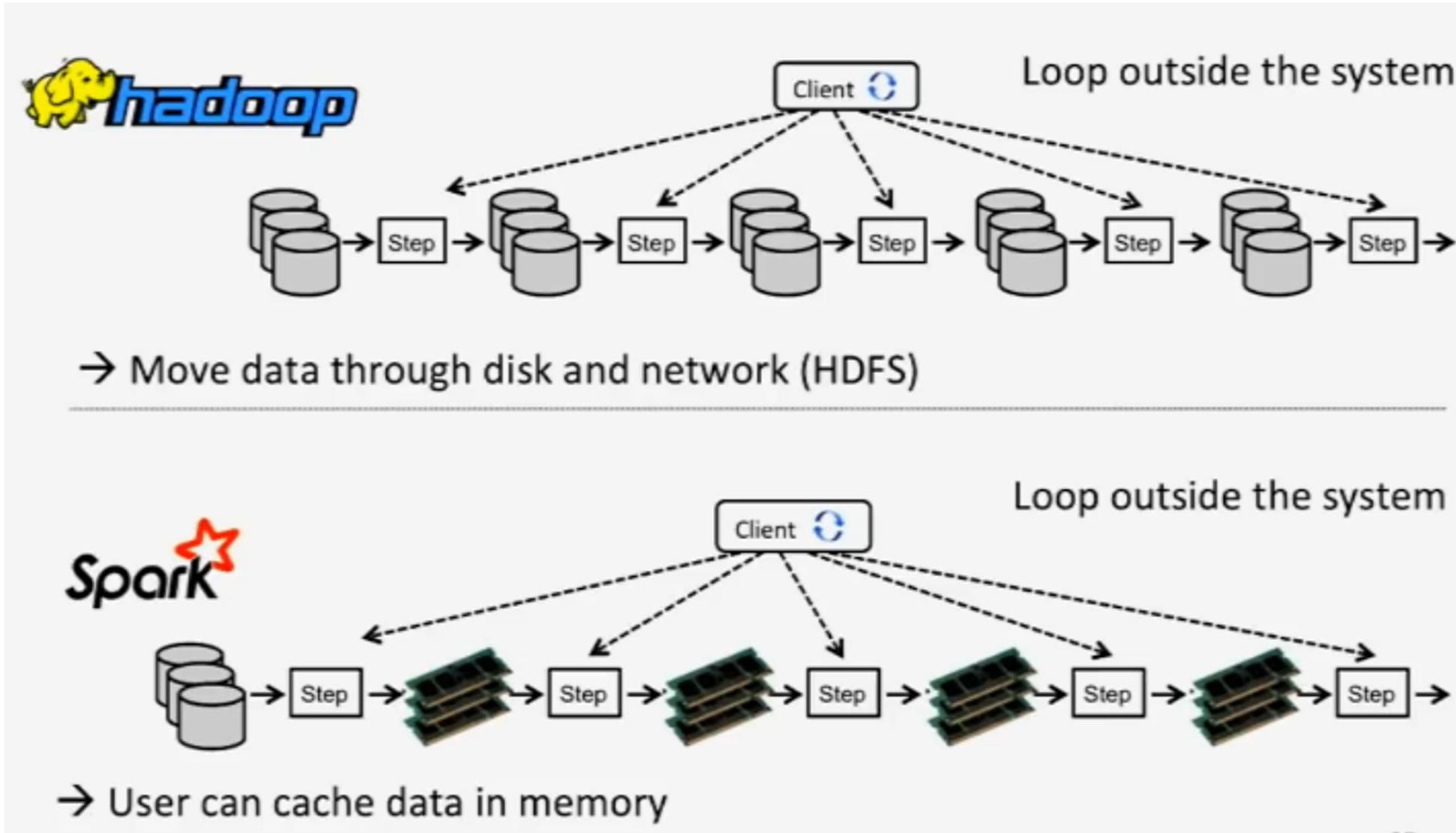
# Big Data: Spark

- Est un framework de développement :
  - De calcul : MapReduce
  - Distribué
  - Exécution en mémoire
  - Fonctionnel
  - Lazy Execution « paresseux »
- Spark peut être jusqu'à 100 fois plus rapide que Hadoop
- Permet de lancer des requêtes en mode interactif
- Spark offre plusieurs API : Java, Python, Scala, R et SQL

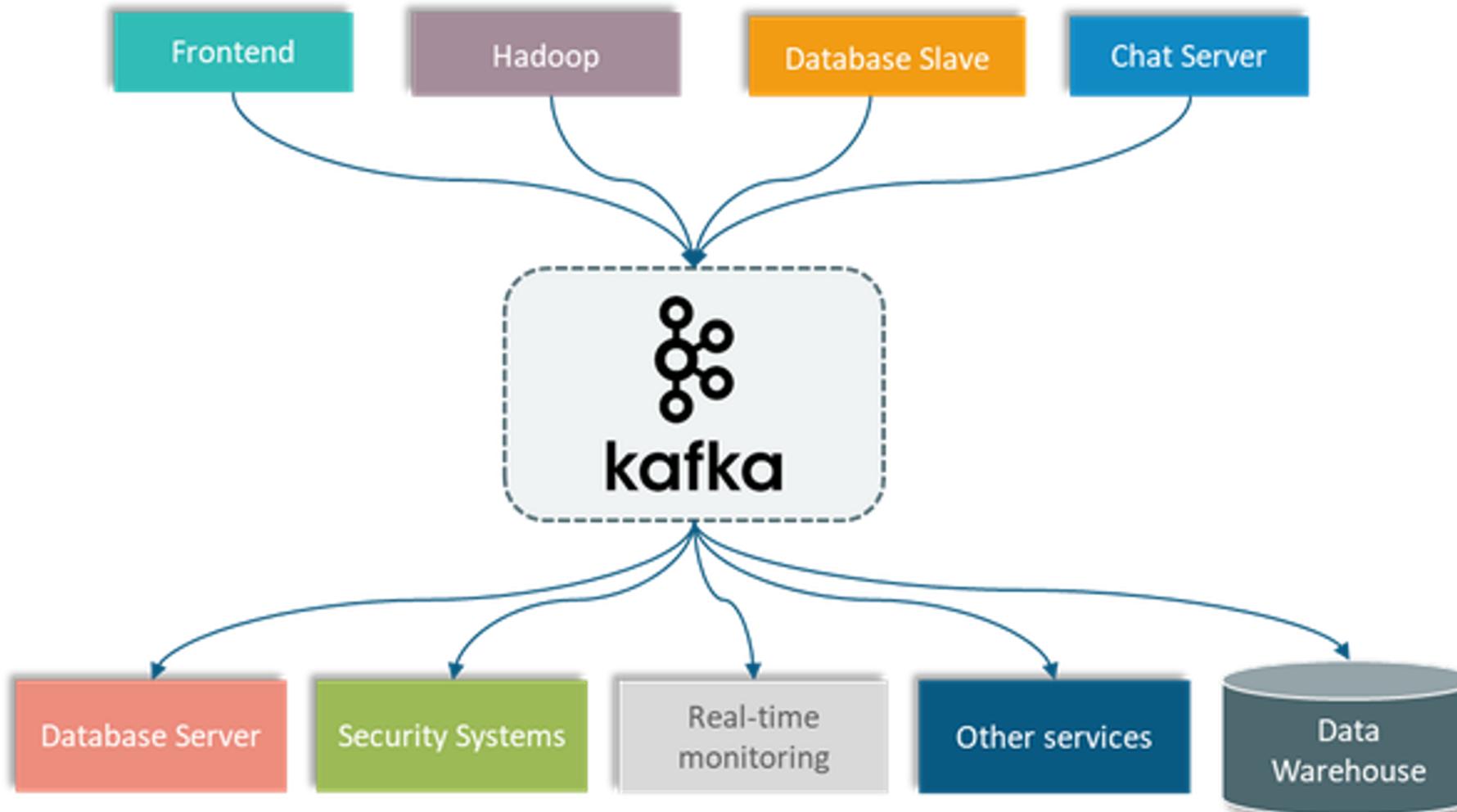
# Big Data: Spark Architecture



# Big Data: Spark vs Hadoop

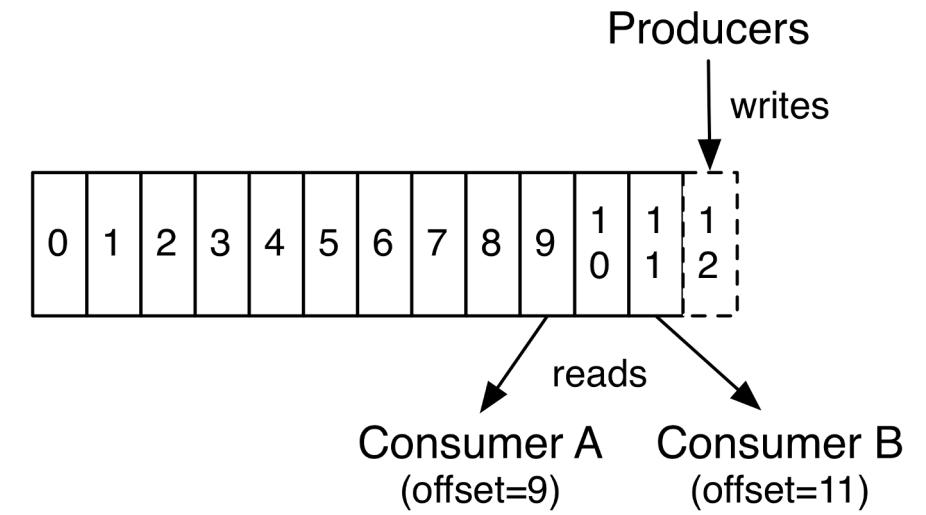


# Big Data: Kafka

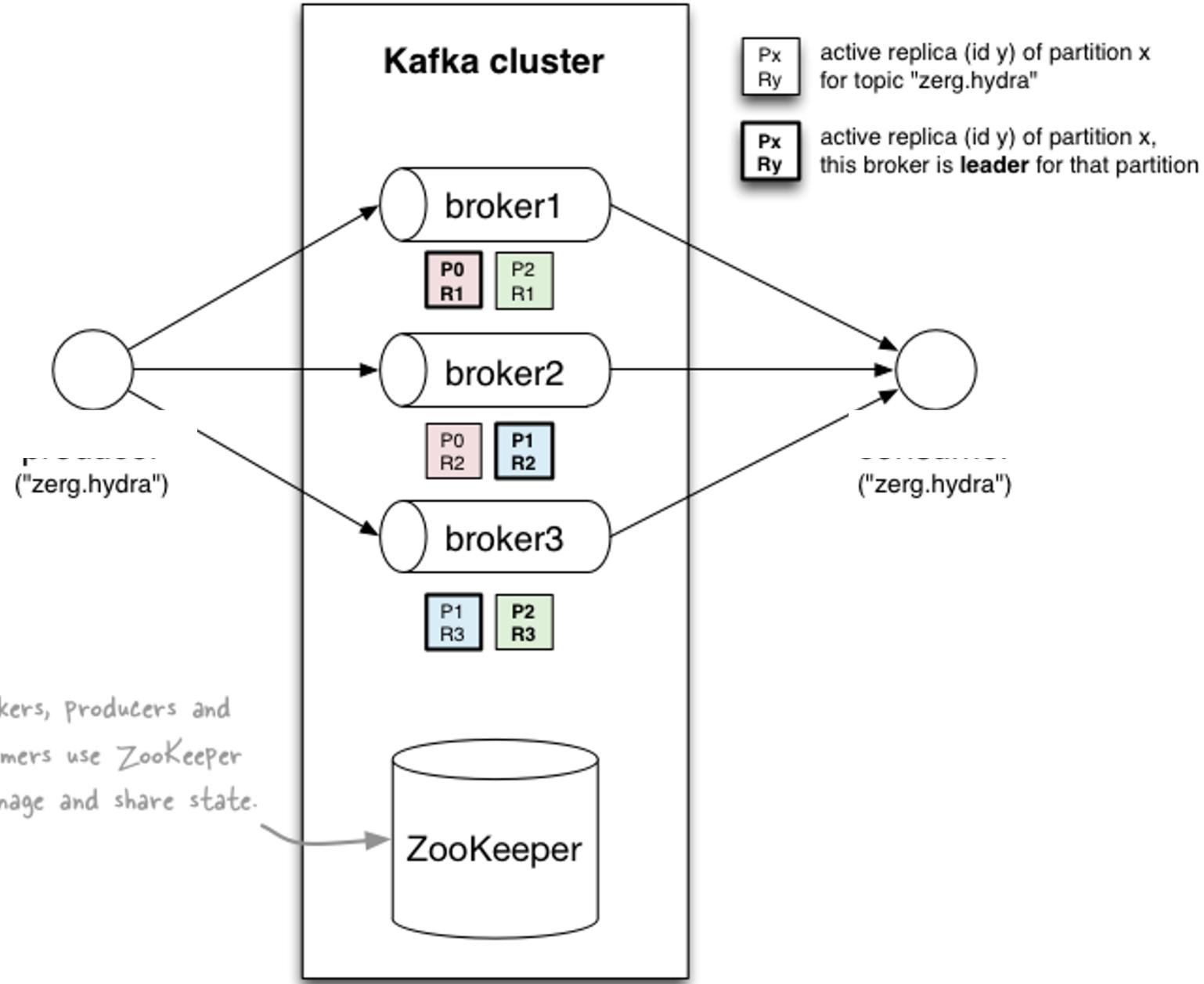


# Big Data: Kafka

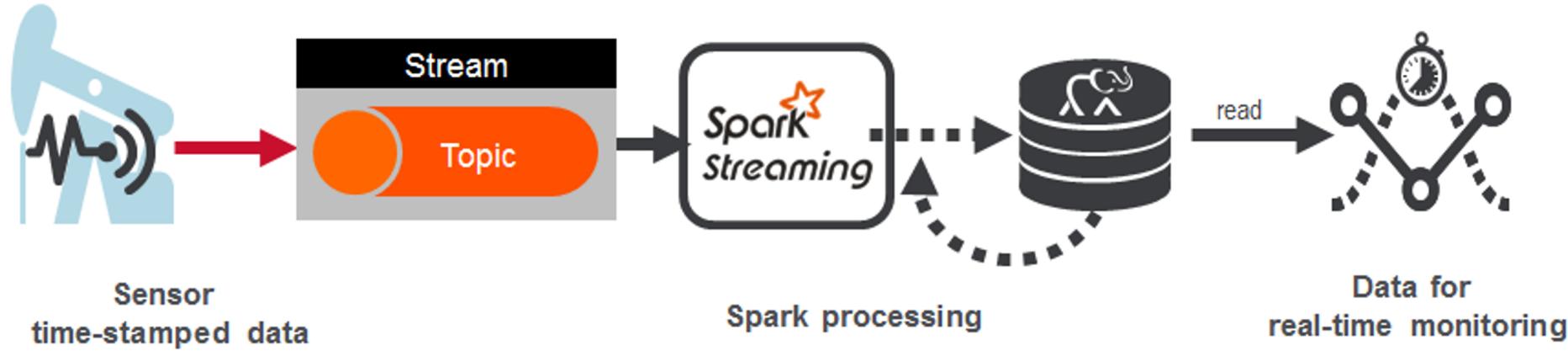
- Un bus de message
- Performant : millions de messages par seconde
- Résilient : distribué et réPLICATION
- Une base de données bas niveau
- Un message est composé d'une valeur, d'une clé et d'un timestamp
- Kafka organise les messages en catégories appelées topics



# Big Data: Kafka



# Big Data: Architecture





# Big Data: DataLake

- Sécurité : Privacy by design
    - Réseau
    - Accès
    - Autorisation
    - Chiffrement et anonymisation
  - Cycle de vie
  - Etat de la donnée
    - En mouvement
    - Statique : Chaude, Tiède et Froide
  - Retracer le cycle de vie de la données
- 
- Exploration de la données
    - Mettre en place les outils nécessaire pour exploiter les données et les analyser. Notebook jupyter ou Tableau software
  - Data Ingestion et discovery :
  - Mettre en place des mécanisme d'ingestion et sauvegarde
  - Découverte de la structure de la données.
  - Data Auditing
  - Garder trace de toutes le transformations et de qui a fait quoi



# RGPD

---

Règlement Général sur  
la Protection des Données

# Big Data: RGPD

Du RGPD se dégagent quatre principes clés :

**le consentement, la transparence, le droit des personnes et la responsabilité.**

- **Le consentement** "le consentement devrait être donné par un acte positif clair par lequel la personne concernée manifeste de façon libre, spécifique, éclairée et univoque son accord au traitement des données à caractère personnel la concernant, par exemple au moyen d'une déclaration écrite, y compris par voie électronique, ou d'une déclaration orale." Le consentement peut être retiré à tout moment par les personnes le demandant.
- **La transparence** les organisations doivent fournir aux individus des informations claires et sans ambiguïté sur la façon dont sont traitées leurs données. Celles-ci doivent être accessibles par tous, via des documents contractuels, des formulaires de collecte ou les pages "privacy" des sites web.

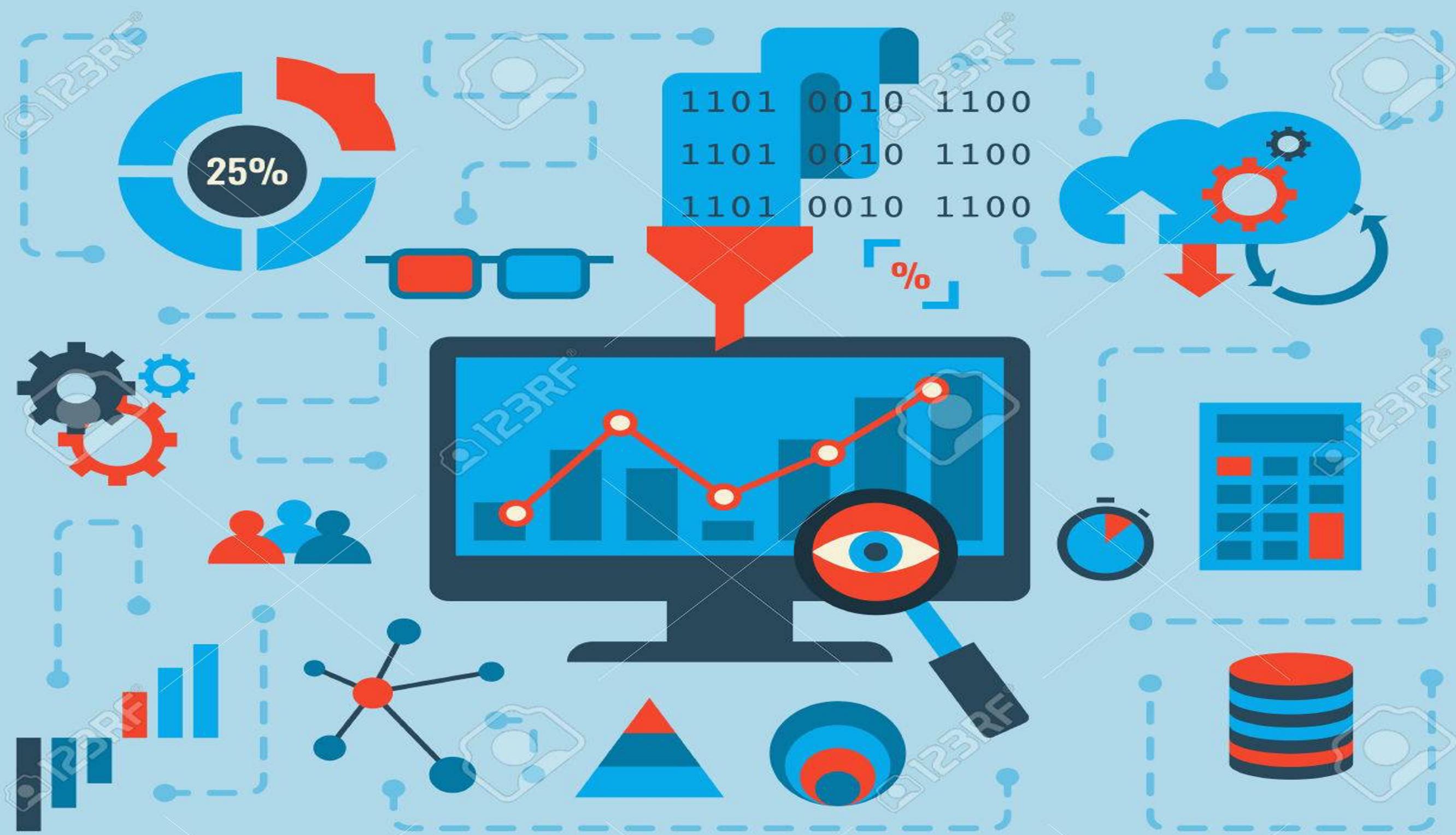
# Big Data: RGPD

## Le droit des personnes

- le droit à l'oubli pour tous les utilisateurs : les organisations n'ont plus qu'un mois pour supprimer les données suite à une demande.
- Le droit à la portabilité des données: il permet à un individu de récupérer les informations qu'il a fournies sous une forme réutilisable pour, le cas échéant, les transférer à un tiers.

## Le principe de responsabilité

- Il regroupe toutes les mesures qui visent à responsabiliser d'avantage les entreprises dans le traitement des données à caractère personnel. Les organismes doivent par exemple mettre en place des mesures adéquates pour garantir la sécurité des données. Elles doivent également appliquer le "privacy by design", un concept qui impose de réfléchir à la protection des données personnelles en amont de la conception d'un produit ou d'un service. Elles doivent aussi choisir des sous-traitants qui soient conformes au RGPD ou encore désigner un data protection officer (DPO), chargé de contrôler la conformité de l'organisme avec le RGPD.



# Big Data: Analyse de données

## Analyse qualitative

- Nettoyage des données :
- Suppression des champs vides ou mal formaté
- Suppression des valeurs aberrantes
- Reformatage des données
- Analyse des distributions des données
- Définition de la variables cible
- Analyse des corrélation entre variables
- Construction de Dashborad de visualisation
- Etude de l'impact des variables sur la cible
- Normalisation des données
- Analyse de la composante principale PCA

## Analyse quantitative

- Analyse des distribution des données par rapport à la cible
- Vérification du nombre d'entrée pour chaque catégorie de la cible

# Big Data: Analyse de données

## Développeur Python:

- Pandas: est une bibliothèque Open Source fournissant des structures de données adaptées à l'analyse de données.
- Numpy: est une bibliothèque de calcul scientifique
- Matplotlib : est une bibliothèque de visualisation 2D
- Jupyter: est un notebook de développement dans une interface web

## Recommandation :

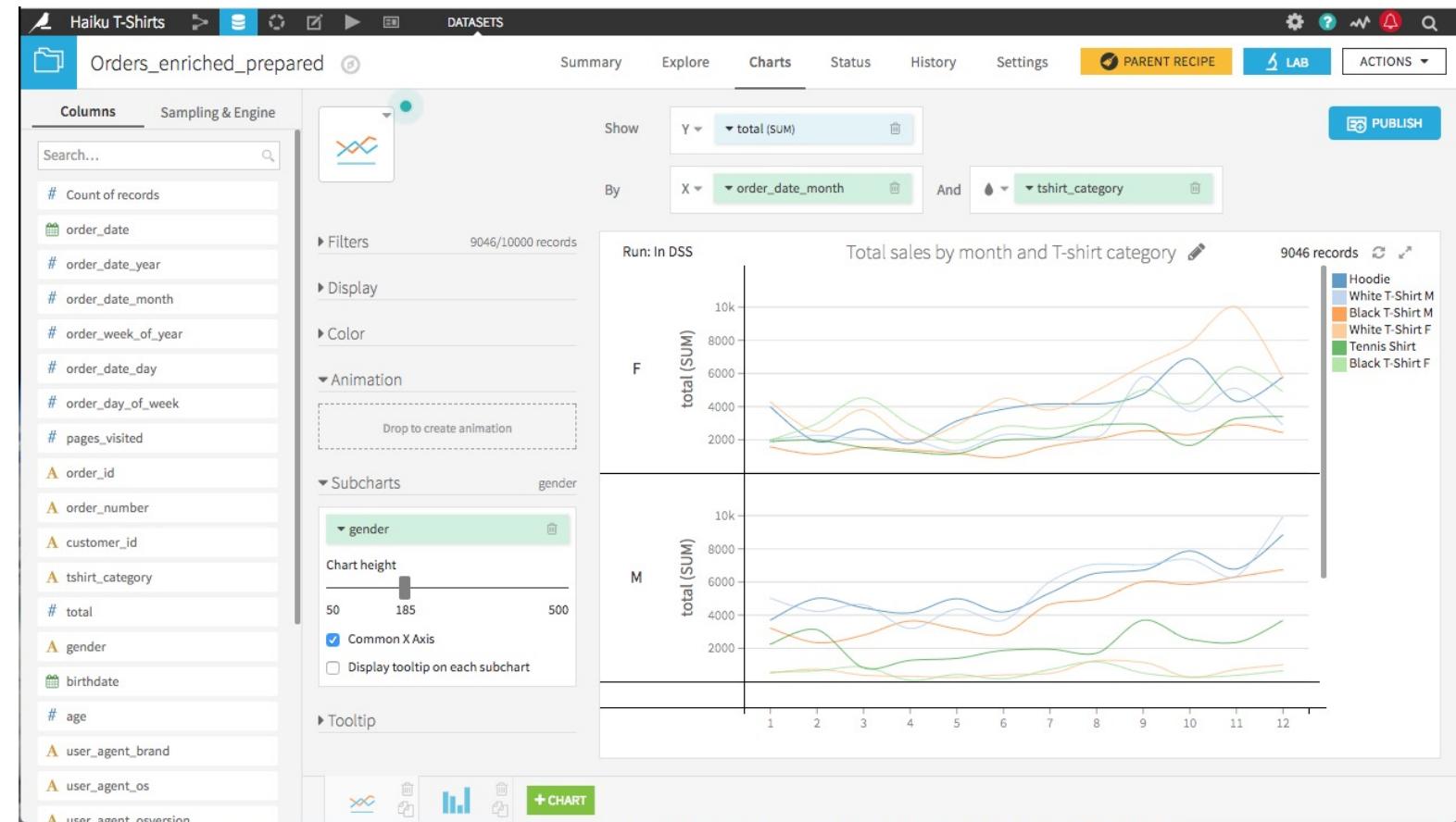
- Utiliser toujours un environnement virtuel python
- Utiliser un système de versionning : git, svn

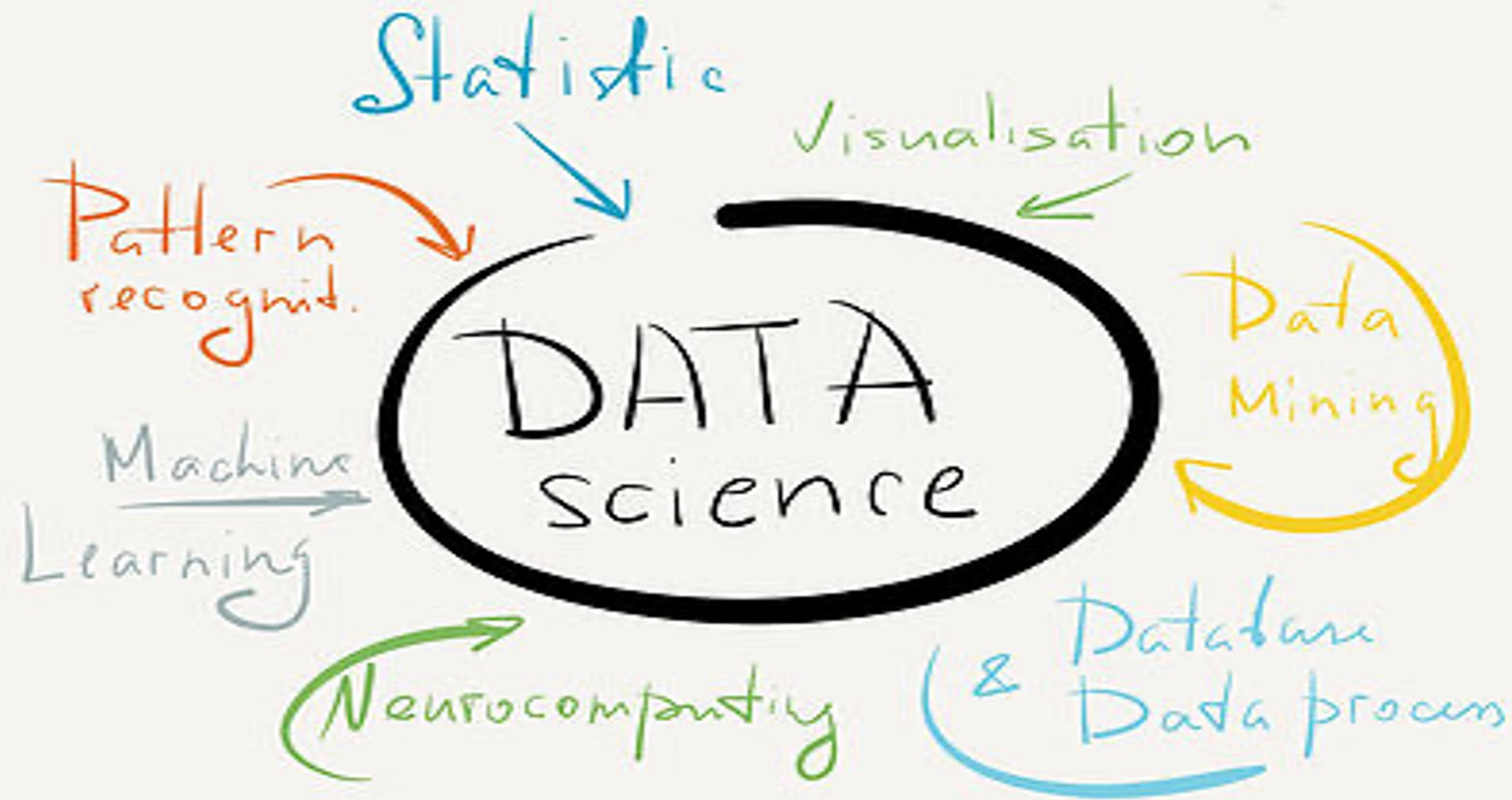
# Big Data: Analyse de données

Utilisation d'outil orienté analyse de données

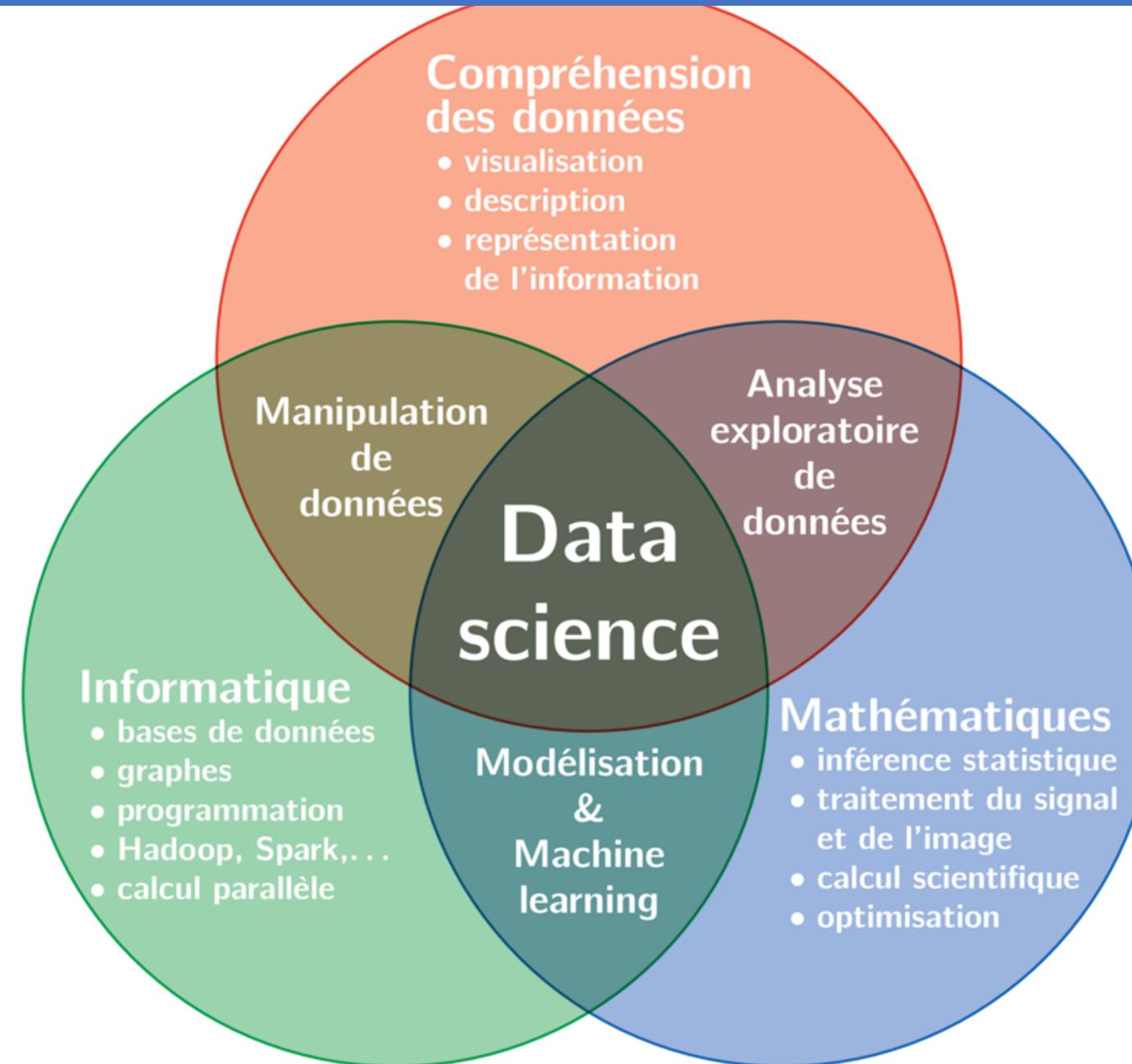
Tableau server, Dataiku..

Des outils plus adaptés pour le métier de data-analyste

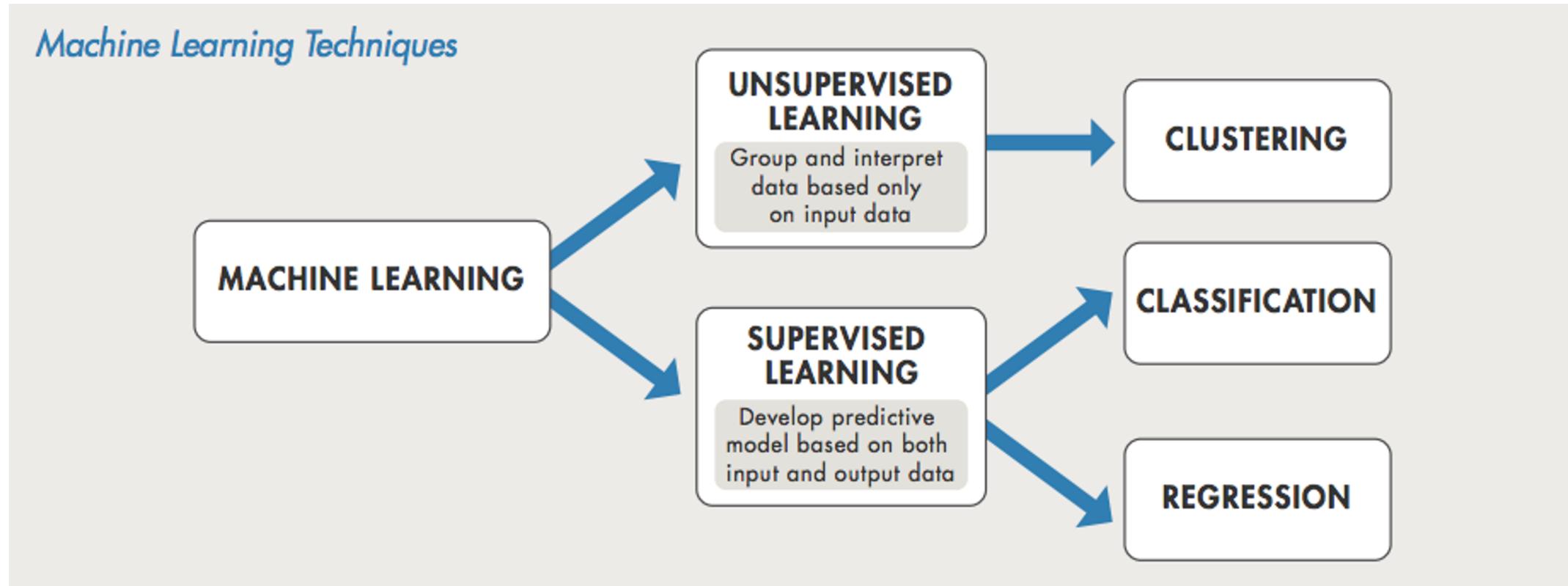




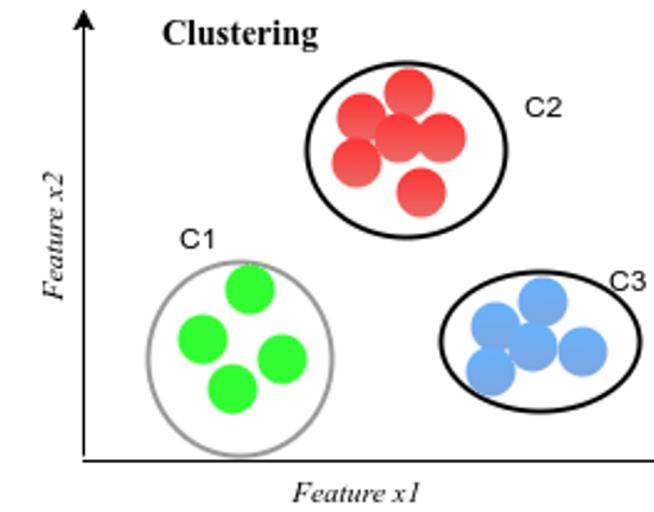
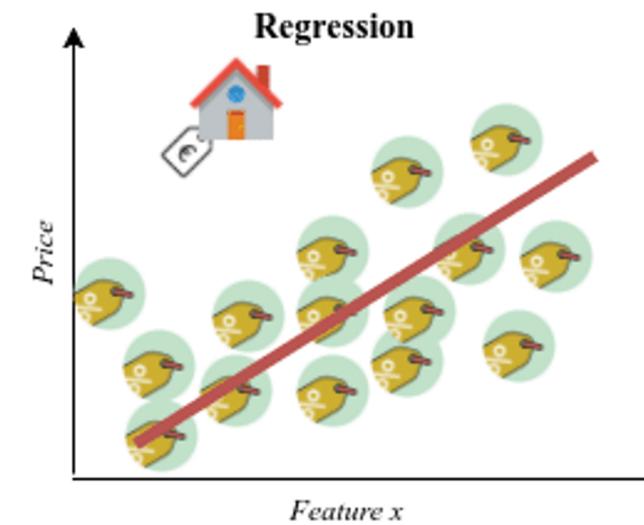
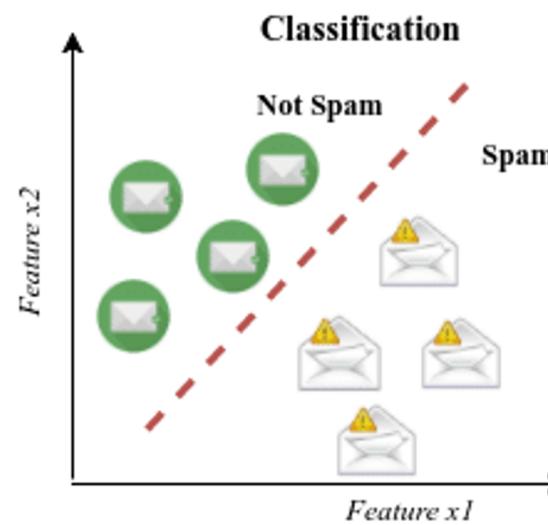
# Big Data: Data Science



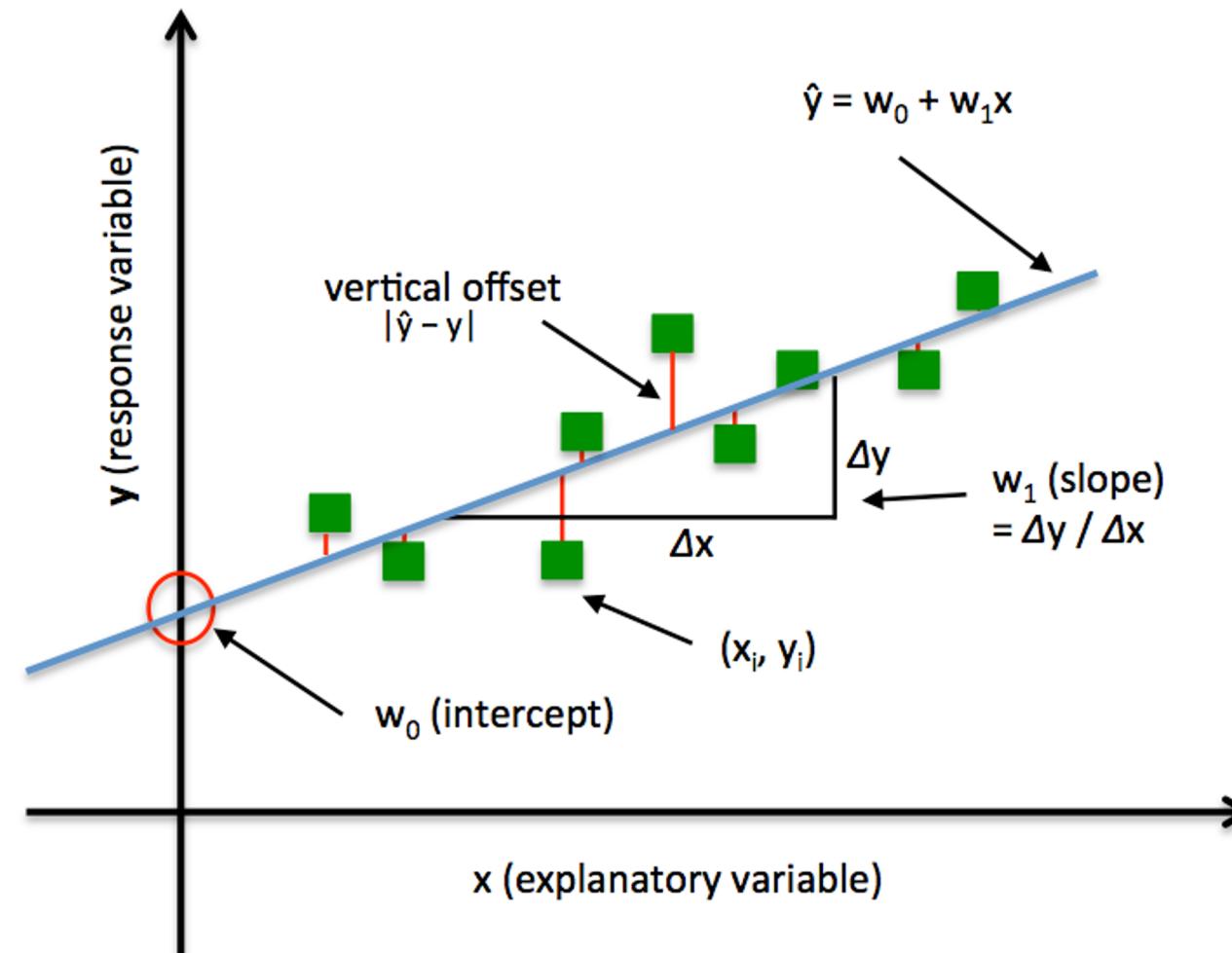
# Big Data: Data Science



# Big Data: Data Science

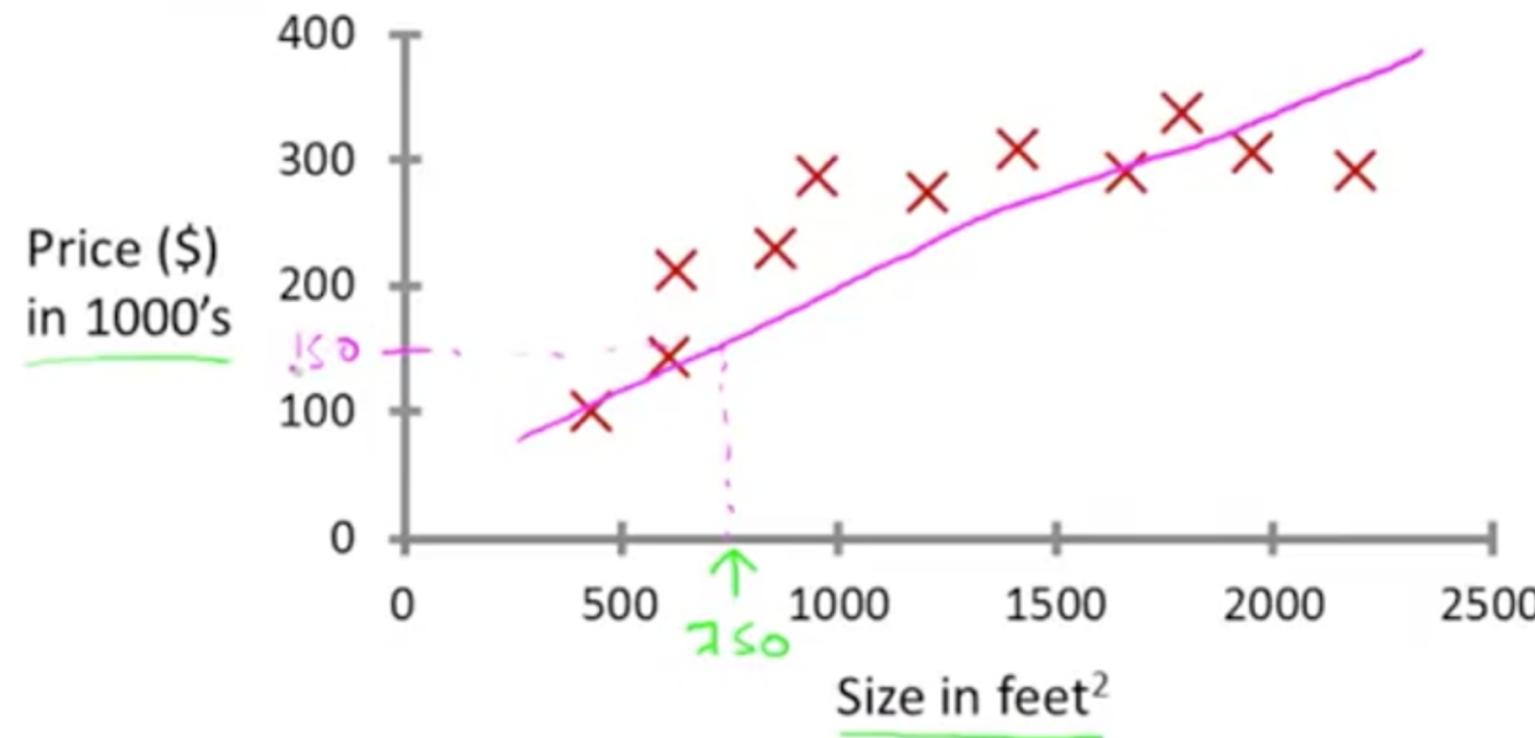


# Big Data: Régression linéaire

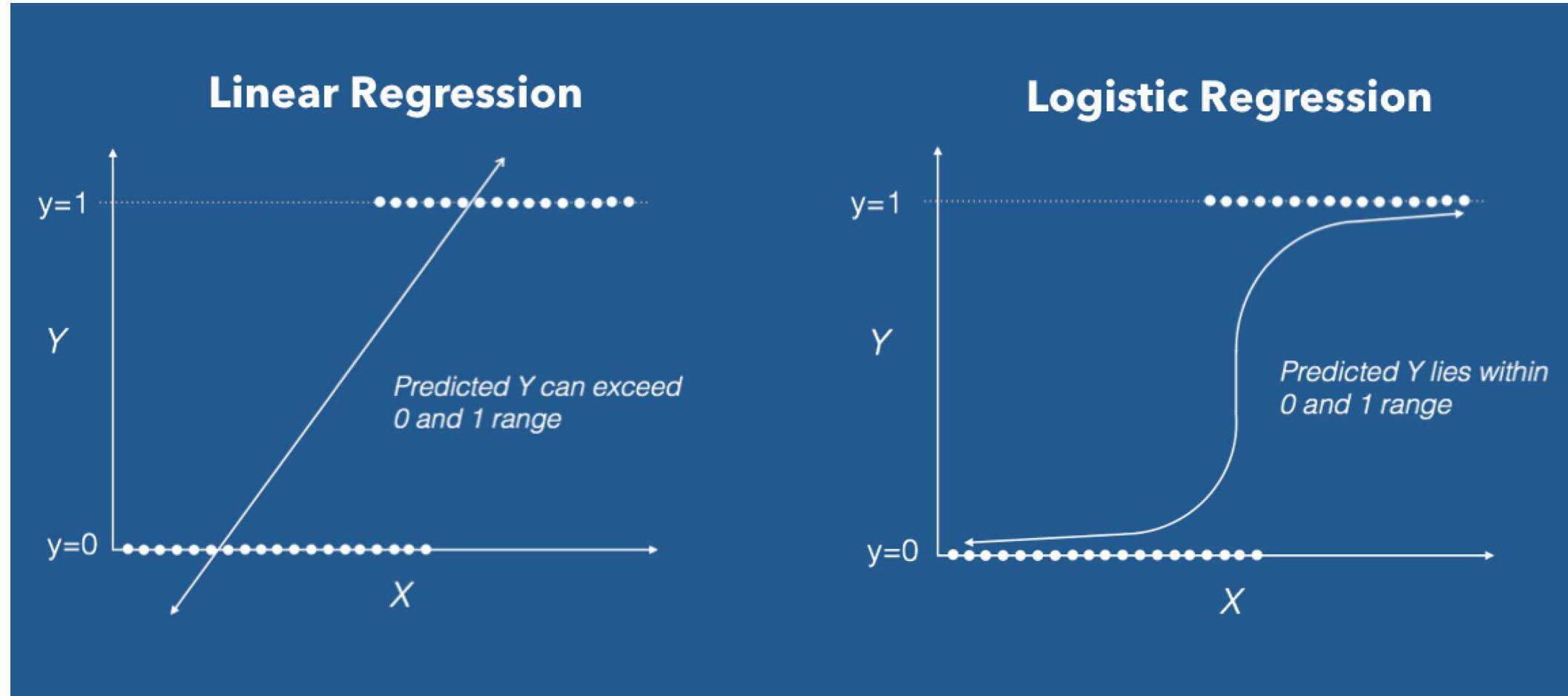


# Big Data: Régression linéaire

Housing price prediction.



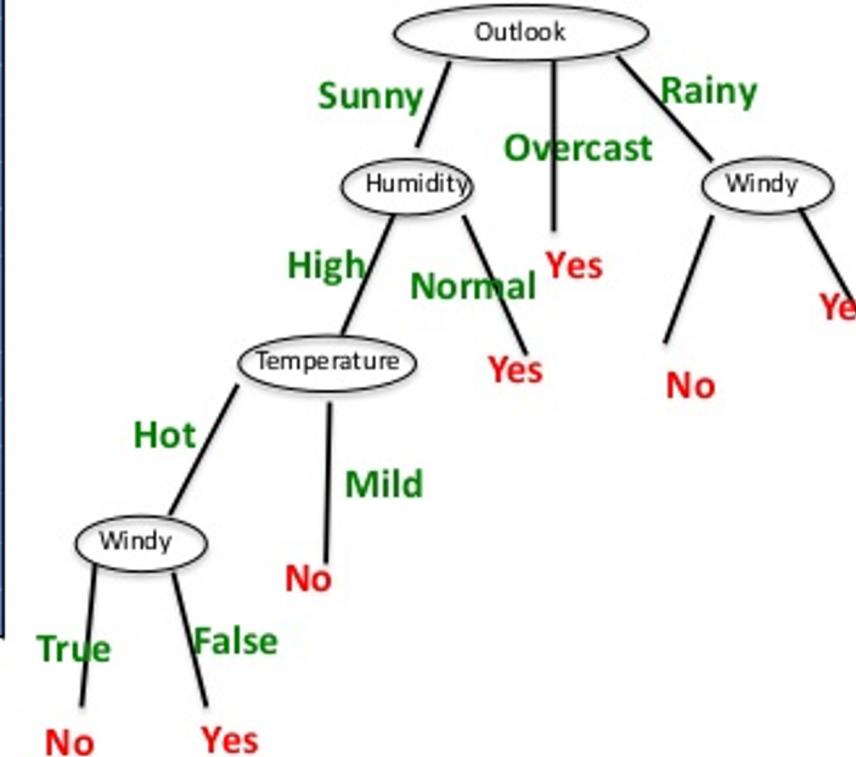
# Big Data: Régression logistique



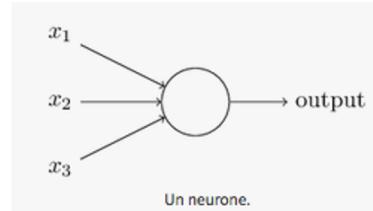
# Big Data: Arbre de décision

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	Yes
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

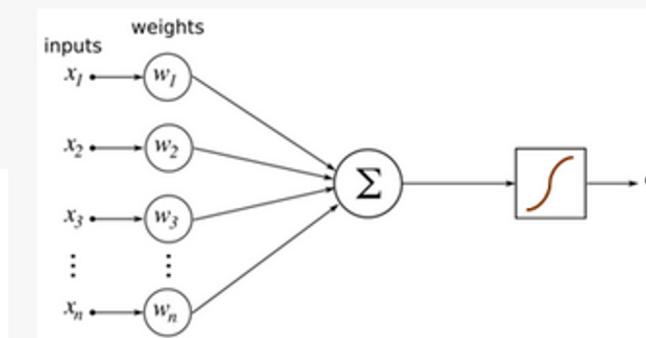
```
outlook = sunny  
humidity = high  
temperature = hot  
| windy = TRUE: no  
| windy = FALSE: yes  
temperature = mild: no  
temperature = cool: null  
humidity = normal: yes  
outlook = overcast: yes  
outlook = rainy  
| windy = TRUE: no  
| windy = FALSE: yes
```



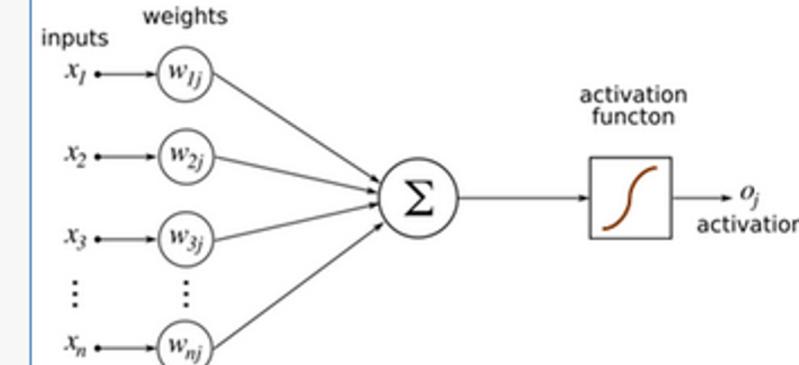
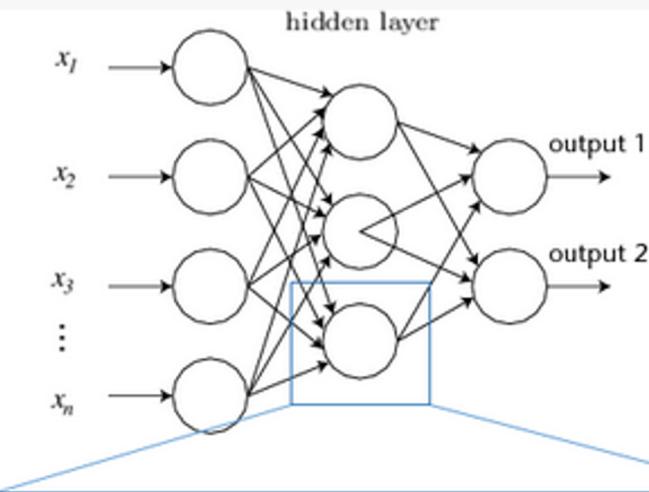
# Big Data: Réseau de neurones



Un neurone.



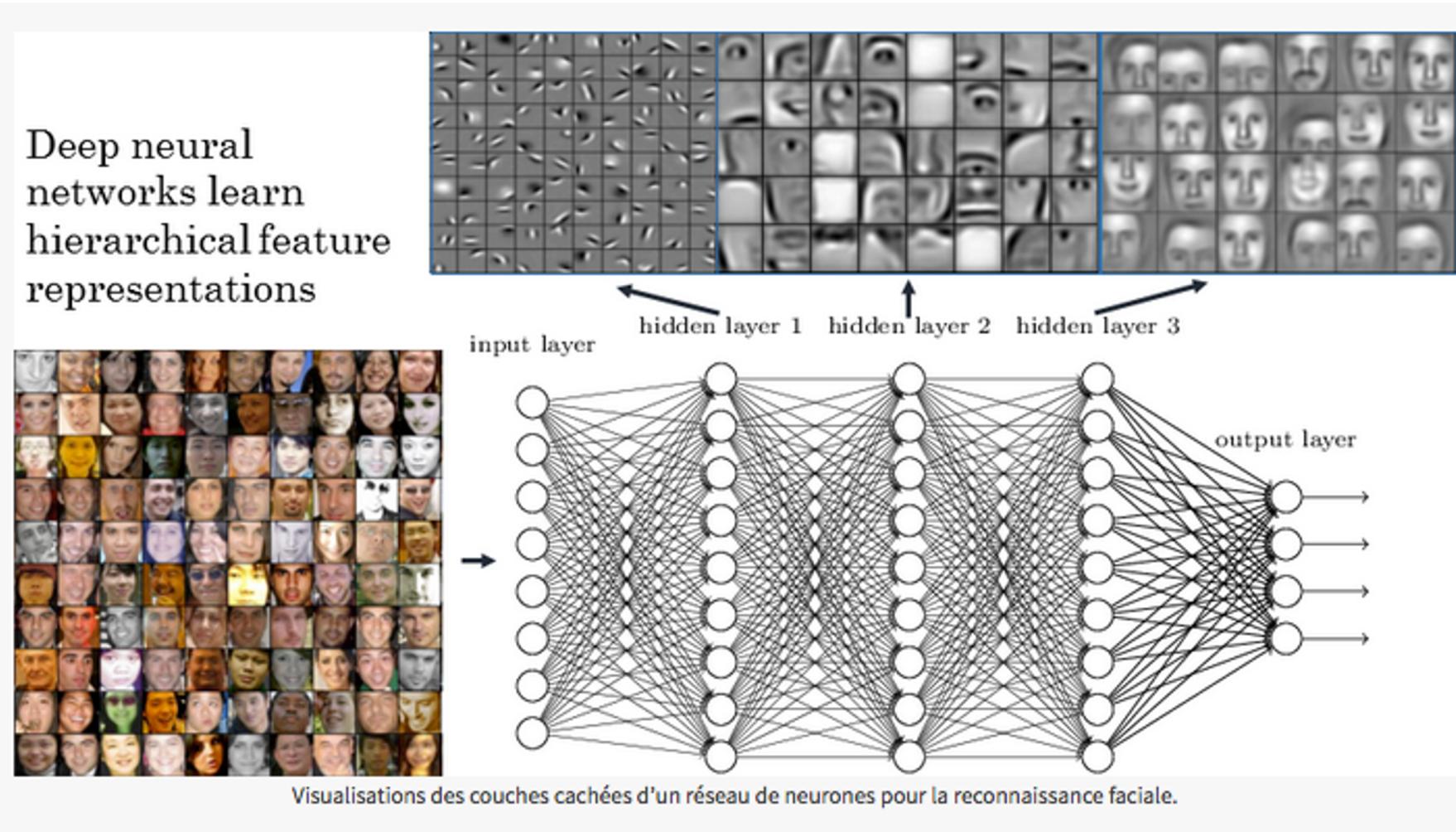
Un réseau de un neurone.



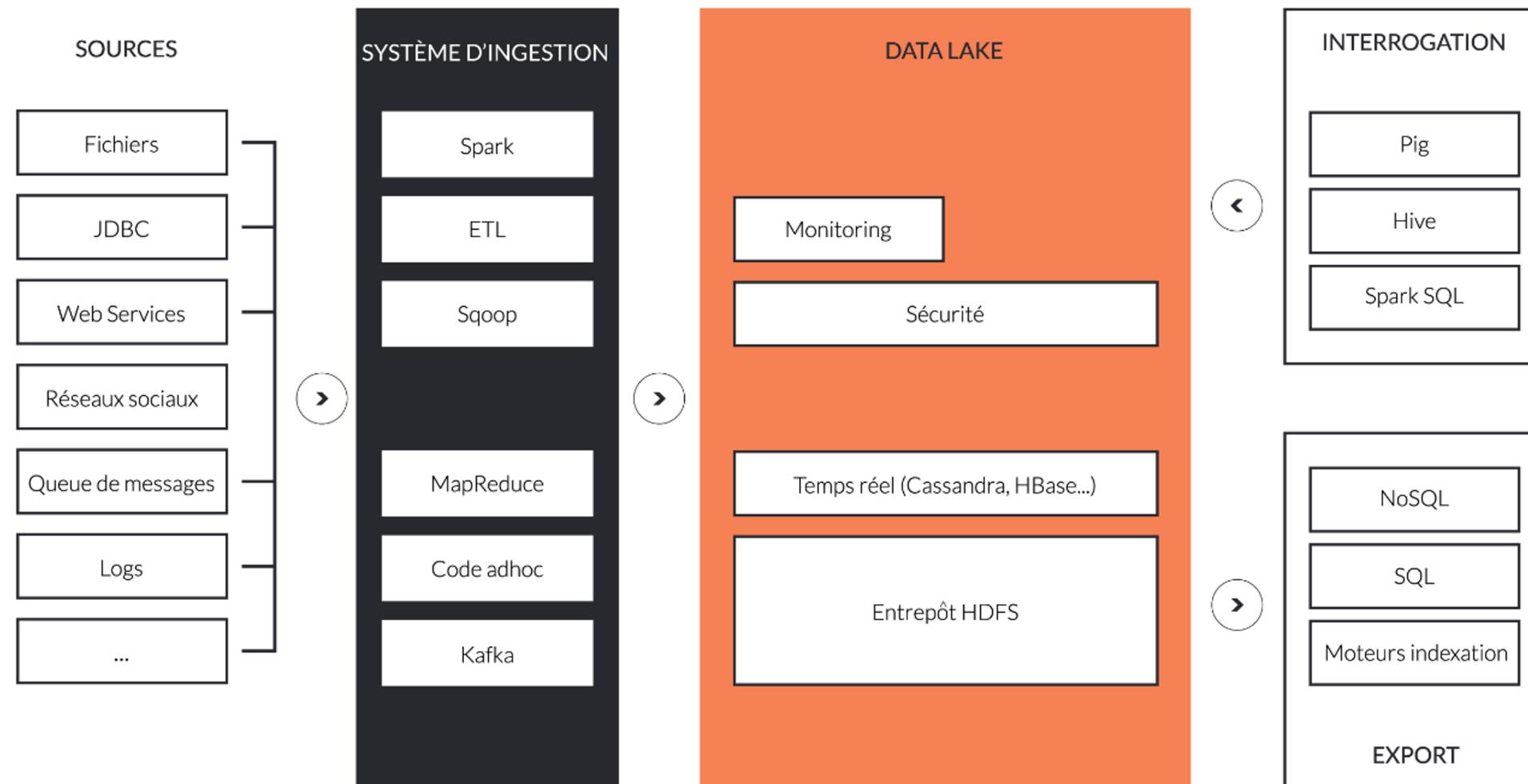
Réseau de neurones multi-couches ou *deep neural network*.

- <https://playground.tensorflow.org>

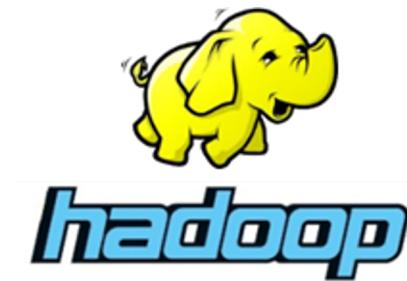
# Big Data: Deep learning



# Big Data: Data pipeline



# Big Data: Tools



elasticsearch

