



Big Data et Modélisation des données

Un cours de Yann Fornier

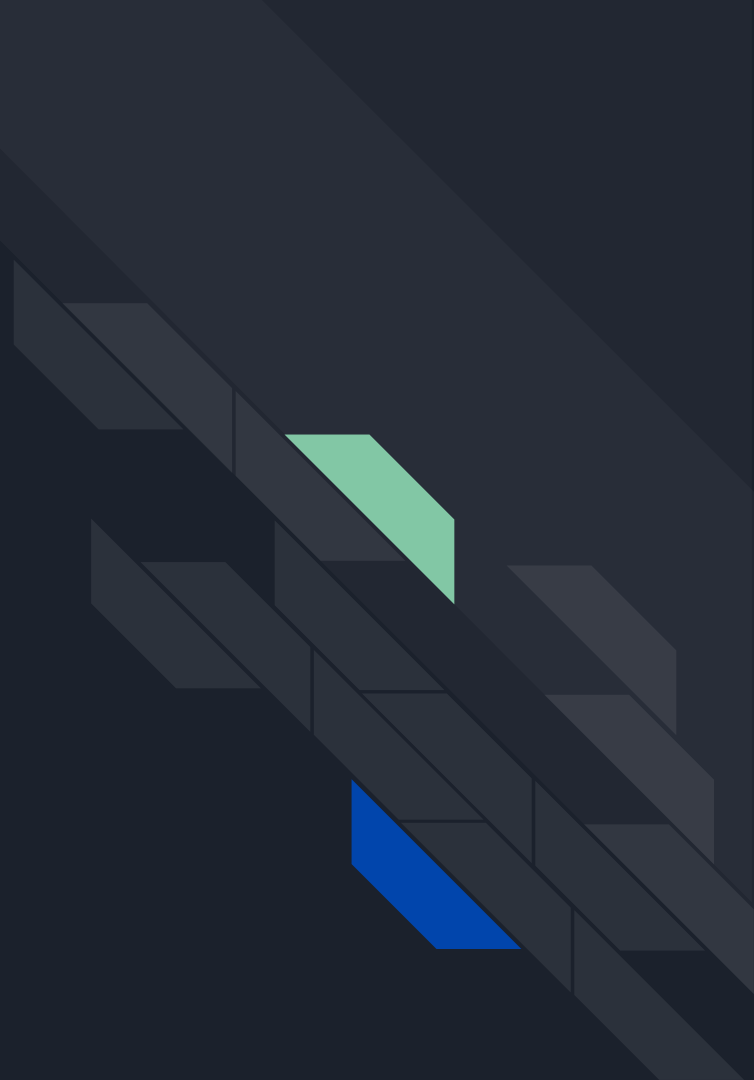


BIG DATA

C'est quoi ?

Un cours de Yann Fornier

C'est quoi une "data" ?





Les types de données

Données Quantitatives

Ces données peuvent être mesurées et exprimées en chiffre.

Revenus, chiffres, âges...

Données Qualitatives

Ces données décrivent des caractéristiques qui ne peuvent pas être mesurées.

Genre, statut matrimonial, préférences...

Données Catégorielles

Ces données sont organisées en catégories, souvent par couleur ou symbole

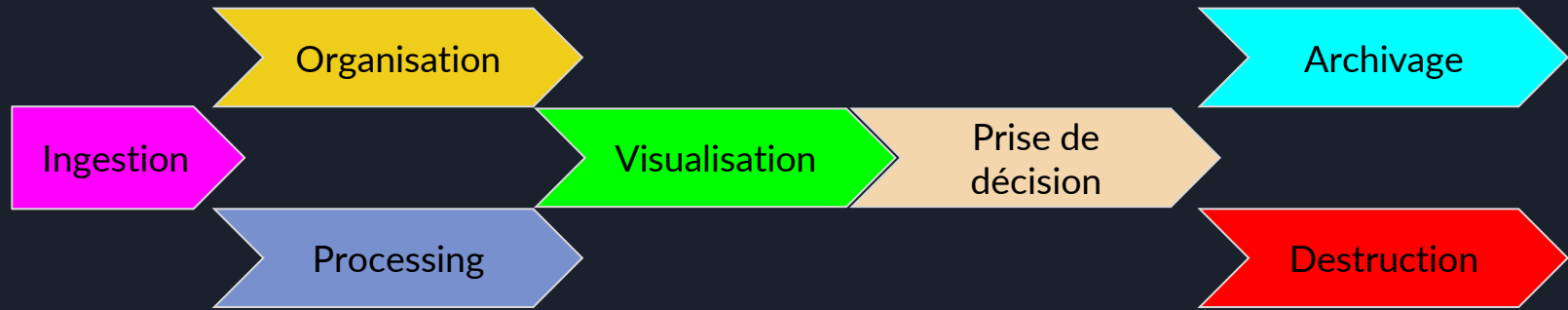
Classement par couleur, symboles...

Données Textuelles

Ces données sont composées de phrases, paragraphes ou textes entiers.

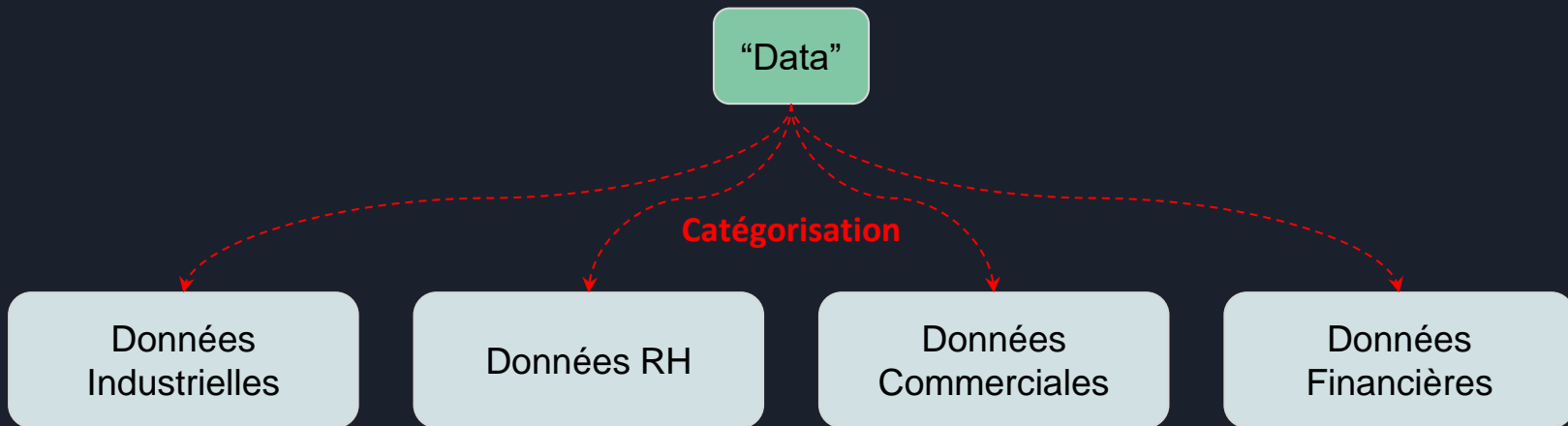
Commentaires, critiques, articles de journaux...

Le cycle de vie de la donnée



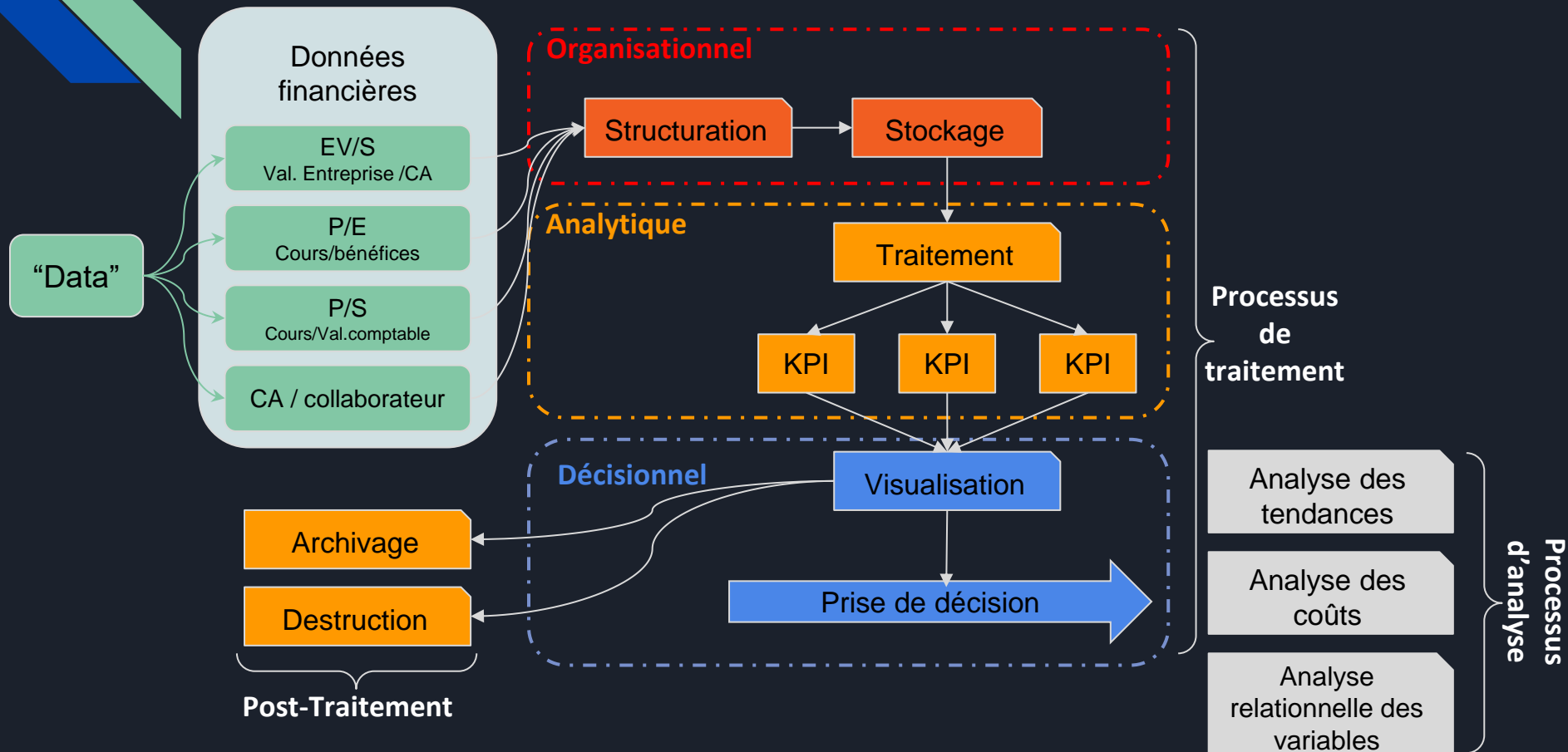
Les processus liés aux données

Catégorisation des données

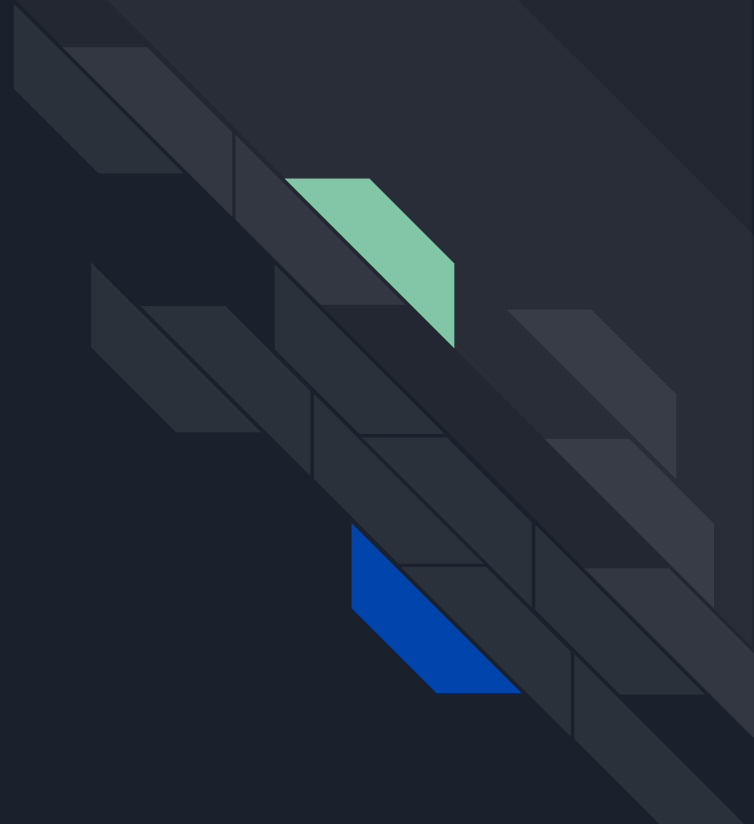


Et plus encore...

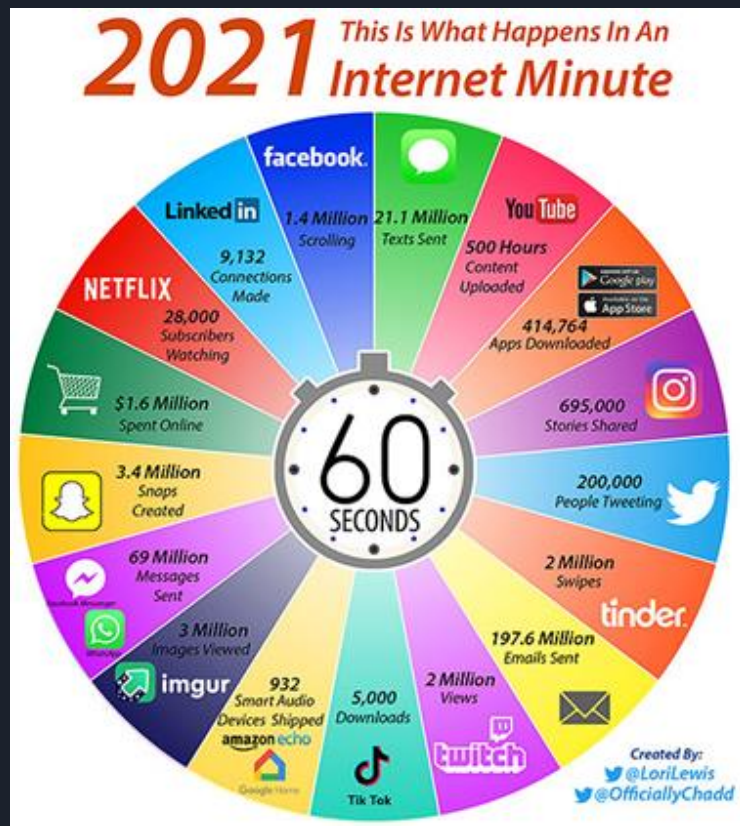
Les processus liés aux données



Pourquoi “Big” ?



Internet en 1 minute



Le “déluge de données”

“La quantité de données massive rend son traitement obsolète par les méthodes scientifiques traditionnelles.”



Chris Anderson - Ex editor in chief of Wired

<https://www.wired.com/2008/06/pb-theory/>



Les “6 V” du Big Data

Big Data

Volume

Valeur

Variété

Big Data +

Variabilité

Véracité

Vélocité

Introduction

Data Lake

Volume
Vélocité
Variété
Véracité
Valeur
Variabilité

Surcharge



Ordinateur
classique

Introduction

Data Lake

Volume
Vélocité
Variété
Véracité
Valeur

Distribution

Cluster (Hadoop, Spark..)

Ordinateur
classique

Map-Reduce

Ordinateur
classique

Ordinateur
classique

Map-Reduce

Ordinateur
classique

⋮

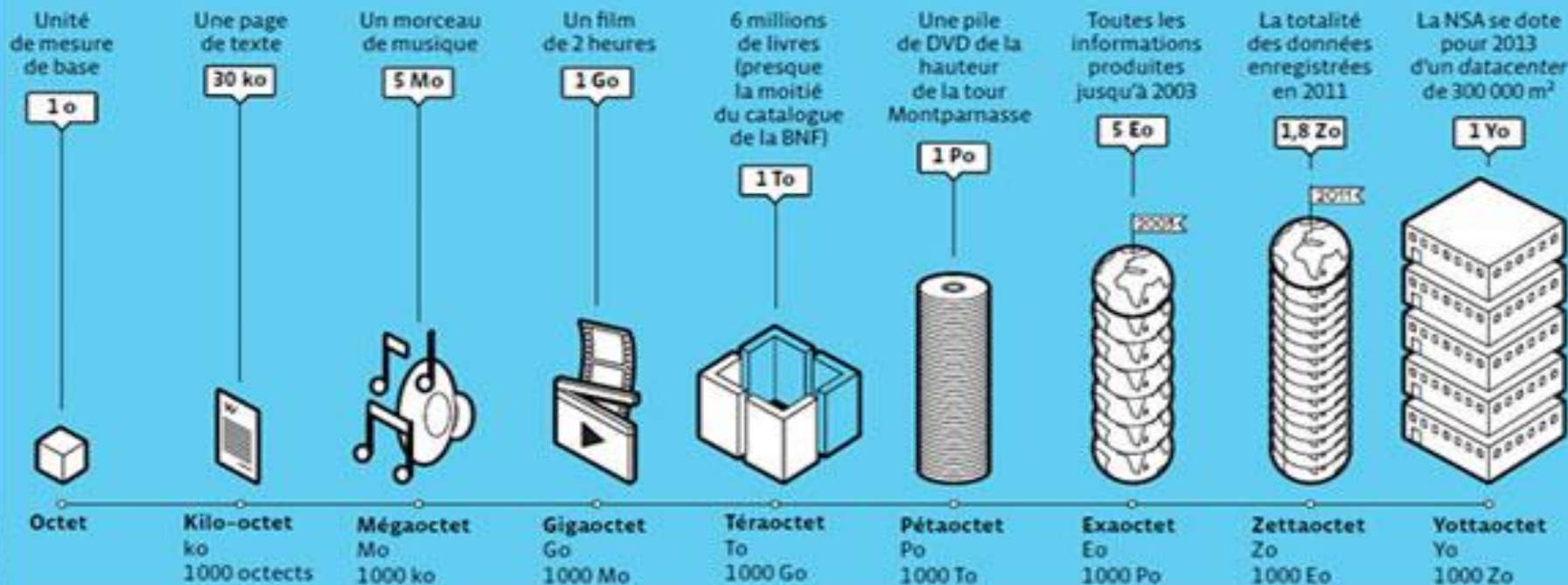
⋮

Ordinateur
classique

Map-Reduce

Ordinateur
classique

L'ÉCHELLE DES OCTETS



NSA Data Center - Utah, USA

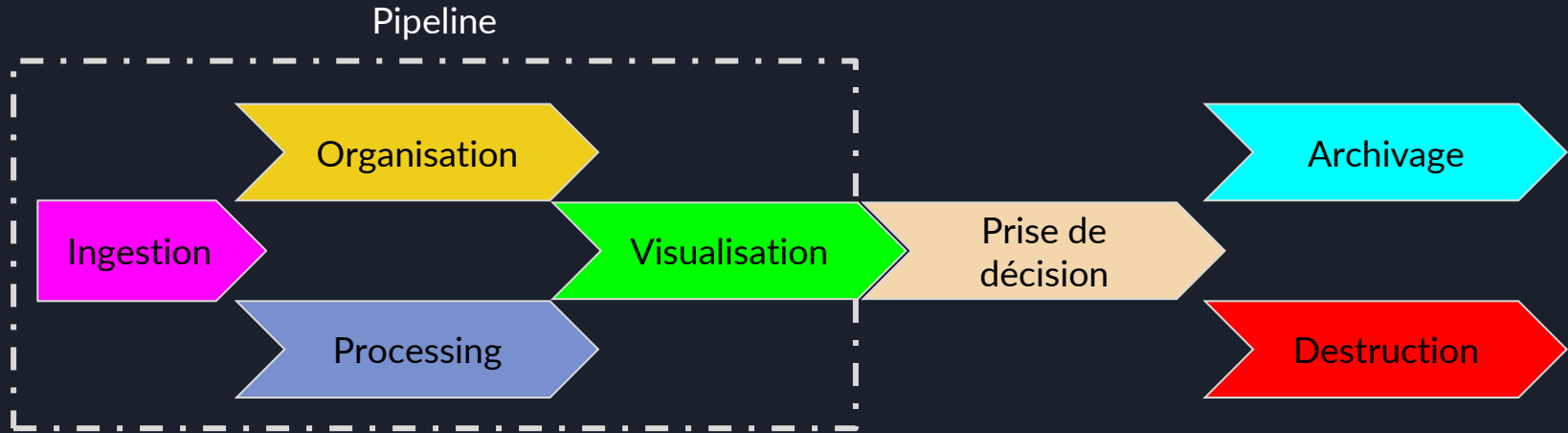




Kolos (?) - Cercle Arctique - Norvège



Le Pipeline du Big Data



Introduction

Data Lake

Volume
Vélocité
Variété
Véracité
Valeur

Surcharge



Ordinateur
classique

Introduction

Data Lake

Volume
Vélocité
Variété
Véracité
Valeur

Distribution

Cluster (Hadoop, Spark..)

Ordinateur
classique

Map-Reduce

Ordinateur
classique

Ordinateur
classique

Map-Reduce

Ordinateur
classique

⋮

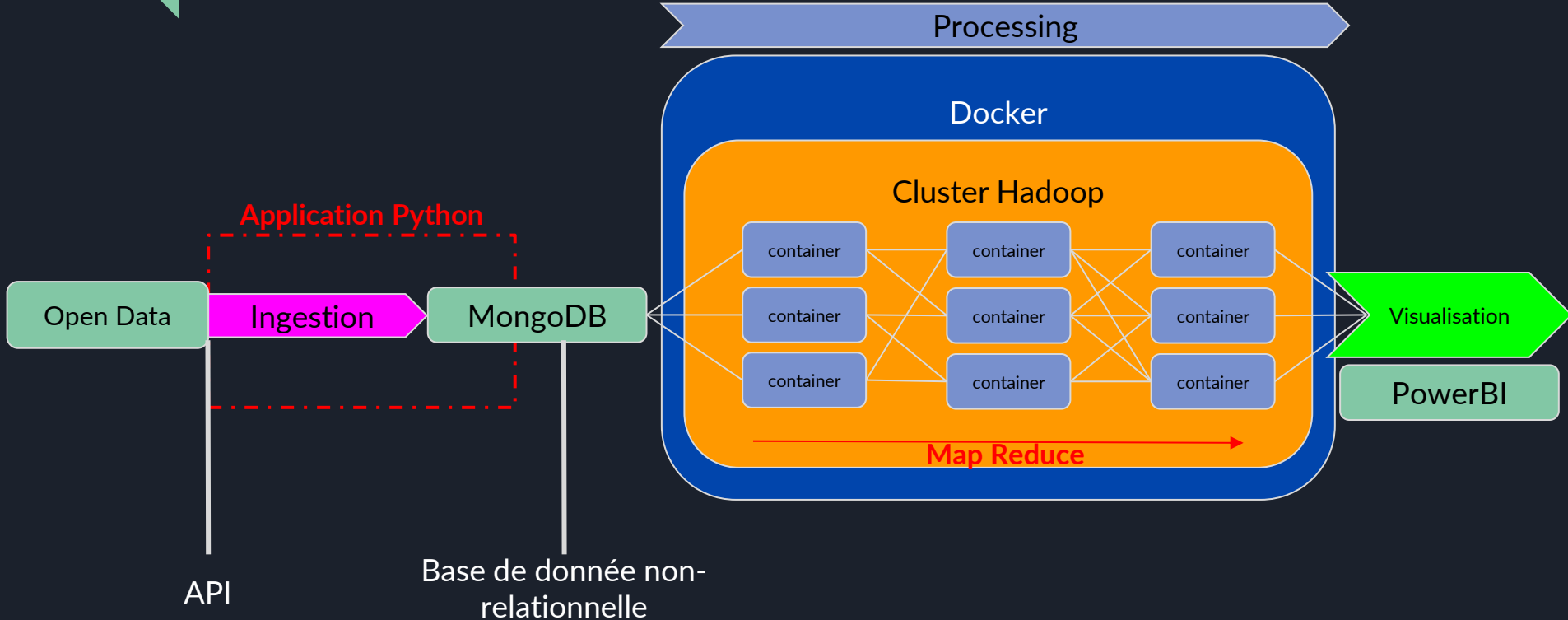
⋮

Ordinateur
classique

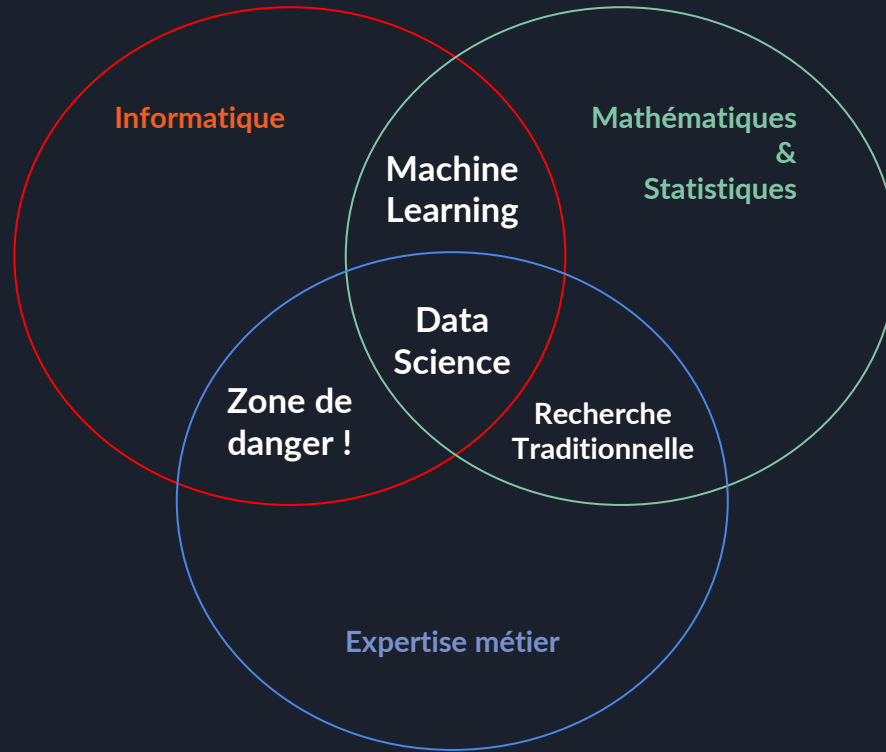
Map-Reduce

Ordinateur
classique

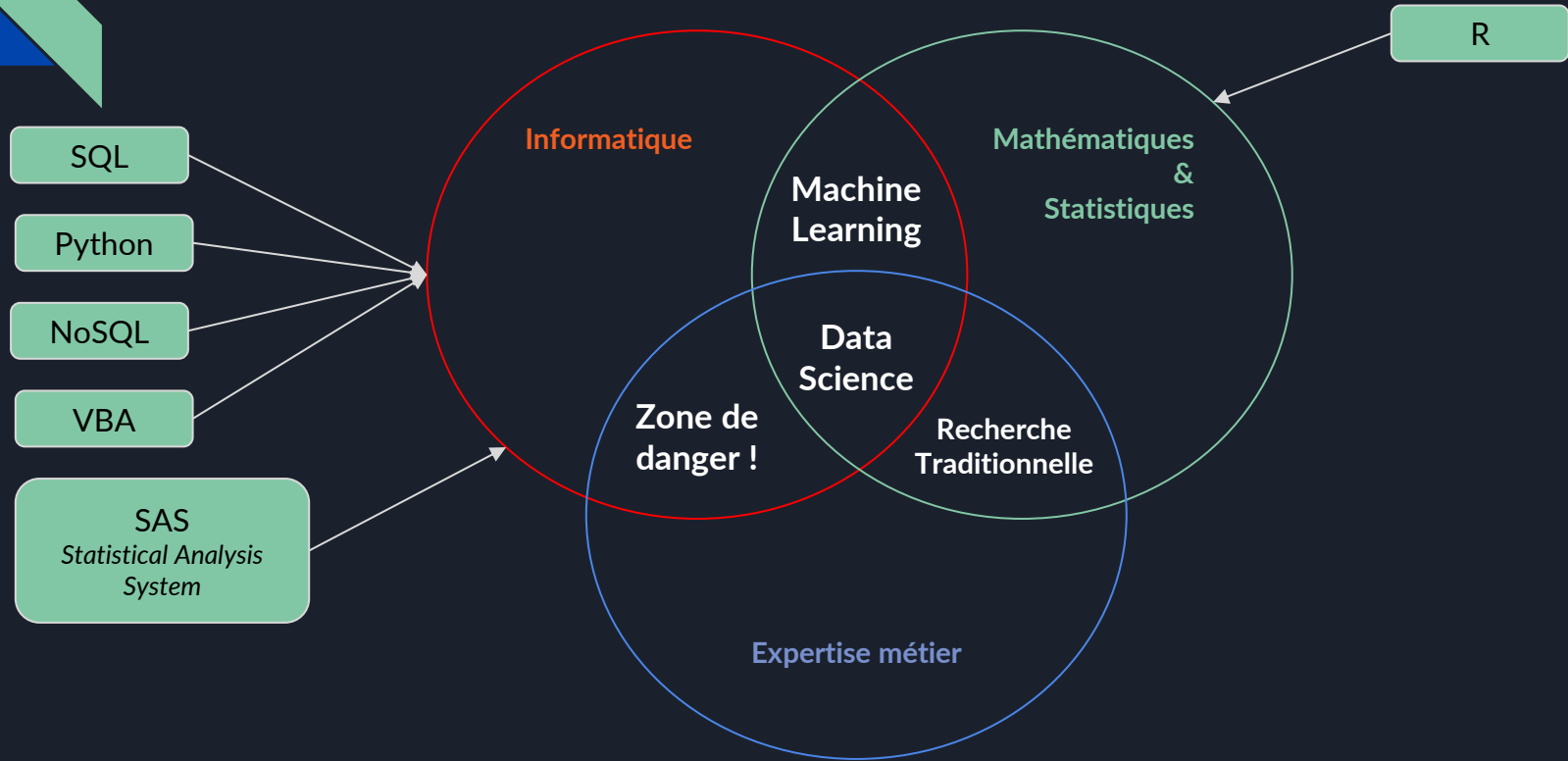
Architecture d'un projet Big Data (Ingénieur)



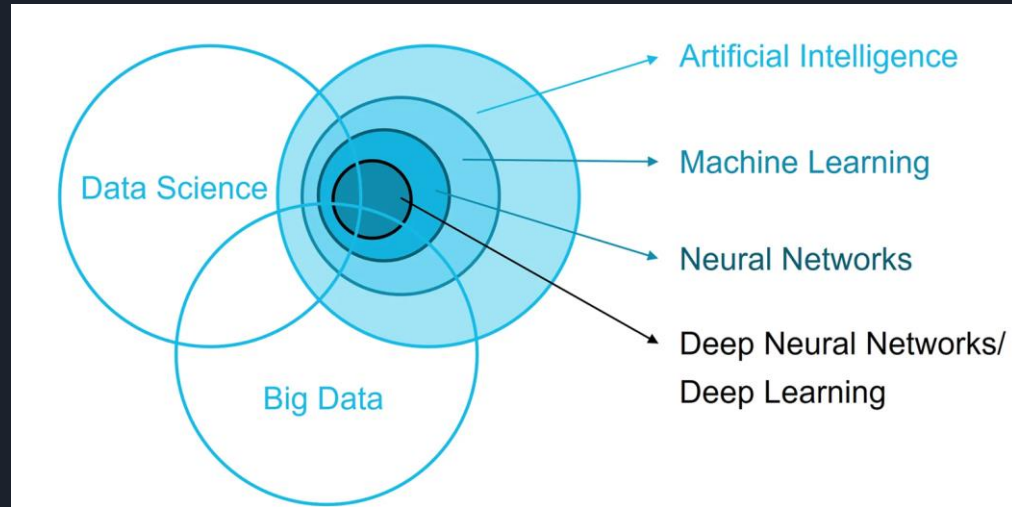
Les domaines du Big Data



Les domaines du Big Data

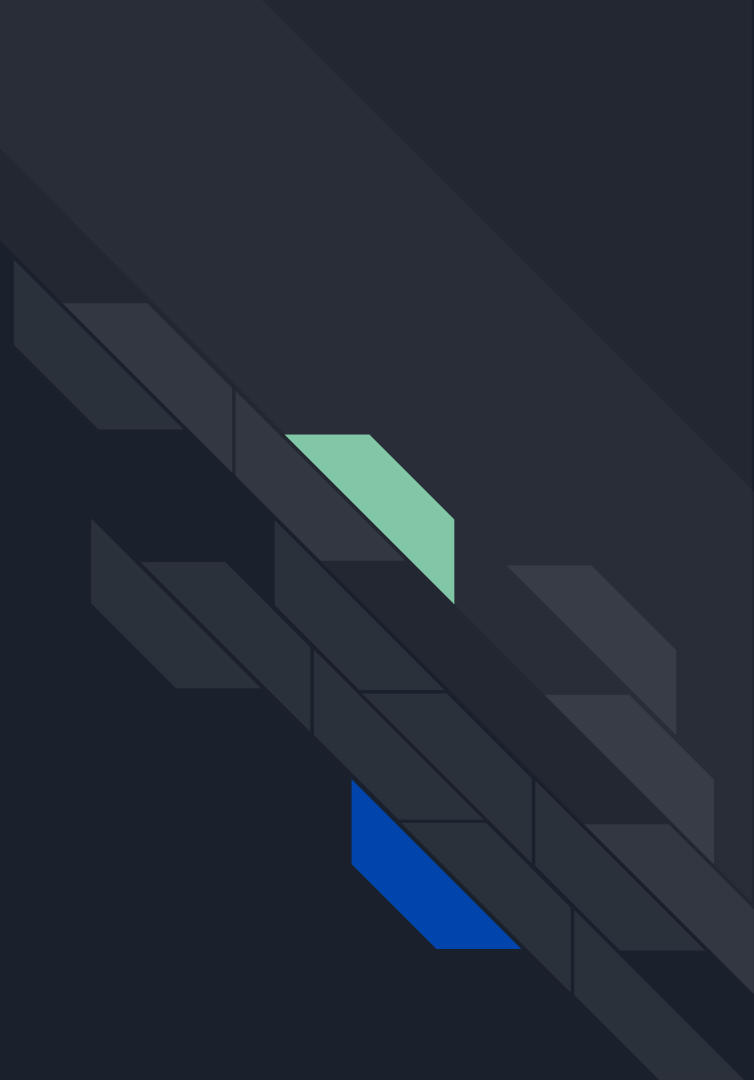


Big Data & Data Science




<https://towardsdatascience.com/role-of-data-science-in-artificial-intelligence-950efedd2579>

Les métiers du Big Data



Les métiers du Big Data

DATA/IA	RÉMUNÉRATION ANNUELLE BRUTE EN K€				
	0-2 ans	2-5 ans	5-10 ans	10 ans et +	
Développeur BI	35 - 45	45 - 60	60 - 70	70 - 80	★ ★ ★
Data engineer/Data scientist	38 - 50	50 - 60	60 - 70	70 - 85	★ ★ ★
DataOps engineer	38 - 50	50 - 65	65 - 75	75 - 85+	★ ★ ★
Analyst BI/Data analyst/Data steward	38 - 48	45 - 60	60 - 70	70 - 80	★ ★ ★
Chef de projet BI/Big data	40 - 45	45 - 60	60 - 70	70 - 80	★ ★ ★
Architecte data	80 - 90	90 - 110	110 - 130	130 - 150+	★ ★ ★
DBA	40 - 45	45 - 55	50 - 60	60 - 70	★ ★ ★
Ingénieur IA/Développeur IA	40 - 45	45 - 60	60 - 70	70 - 80+	★ ★ ★
Data protection officer	38 - 45	40 - 60	55 - 75	75 - 85+	★ ★ ★
Machine learning engineer	40 - 50	50 - 60	60 - 75	75 - 90+	★ ★ ★
Data visualisation consultant	38 - 48	50 - 60	60 - 70	70 - 90+	★ ★ ★

"Etude des salaires en 2023 des métiers du Big Data" - Michael Page

<https://www.lepont-learning.com/fr/cartographie-metiers-data-demandes/>



Avez vous vu les notions suivantes ?

Big Data et concepts généraux ? Les 5 V du Big Data ?

Pipeline Data : Ingestion, Organisation, Processing, Visualisation ?

Open Data ?

Bases de données non relationnelles NoSQL ?

Bases de données en graphe ?

Concepts d'ETL et d'ELT ?

Ingestion par batch ? par stream ?



Open Data

L'Open data ou donnée ouverte est une donnée numérique dont l'accès et l'usage sont laissés libre aux usagers. Elle peut être d'origine publique ou privée.

<https://www.axysweb.com/top-15-des-sources-open-data/#data.gouv.fr>

Open Data



MINISTÈRE
DE L'ÉCONOMIE
DES FINANCES
ET DE LA SOUVERAINETÉ
INDUSTRIELLE ET NUMÉRIQUE
*Liberté
Égalité
Fraternité*

data.economie.gouv.fr

le site des données ouvertes du ministère

Connexion

Données

Visualisations

Création de carte

Création de graphique

API

Démarche

33 585 enregistrements

Aucun filtre actif

Filtres

La recherche texte n'est pas appliquée pendant l'utilisation de la Console d'API.

Rechercher...

id

10410003

6

11110008

6

13545001

6

Prix des carburants en France - Flux instantané

Informations

Tableau

Carte

Communauté

Export

API

Ce jeu de données peut être utilisé via une API qui autorise la recherche et le téléchargement d'enregistrements avec plusieurs paramètres.
Jetez un oeil à la [documentation de l'API](#) et utilisez la [console d'API complète](#) pour essayer les autres API!

La console ci-dessous utilise le [point d'entrée "Query dataset records"](#) de l'API Explore 2.1.

Requête pour l'appel API

dataset

prix-carburants-fichier-i

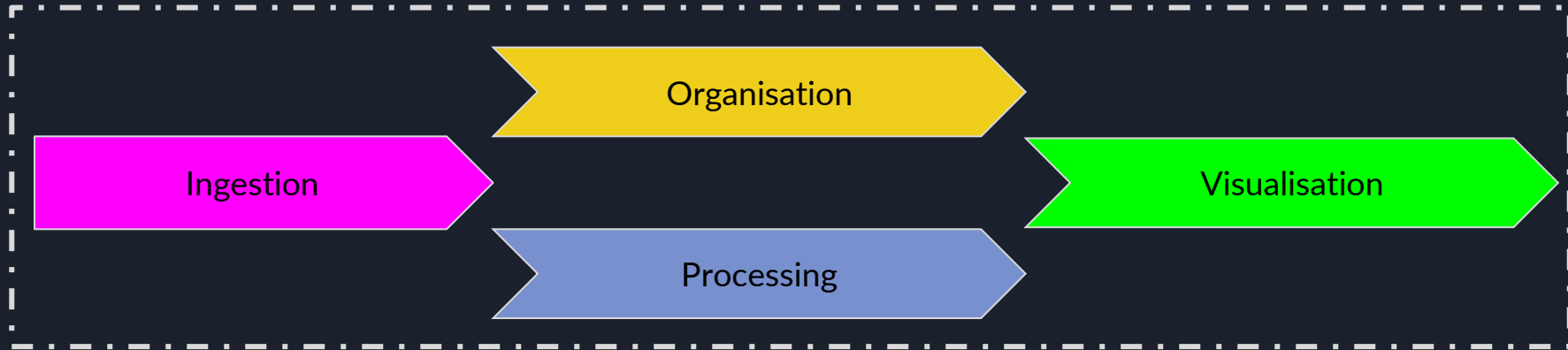
Résultats

```
{  "total_count": 33585,  "results": [
```

<https://data.economie.gouv.fr/explore/dataset/prix-carburants-fichier-instantane-test-ods-copie/api/>

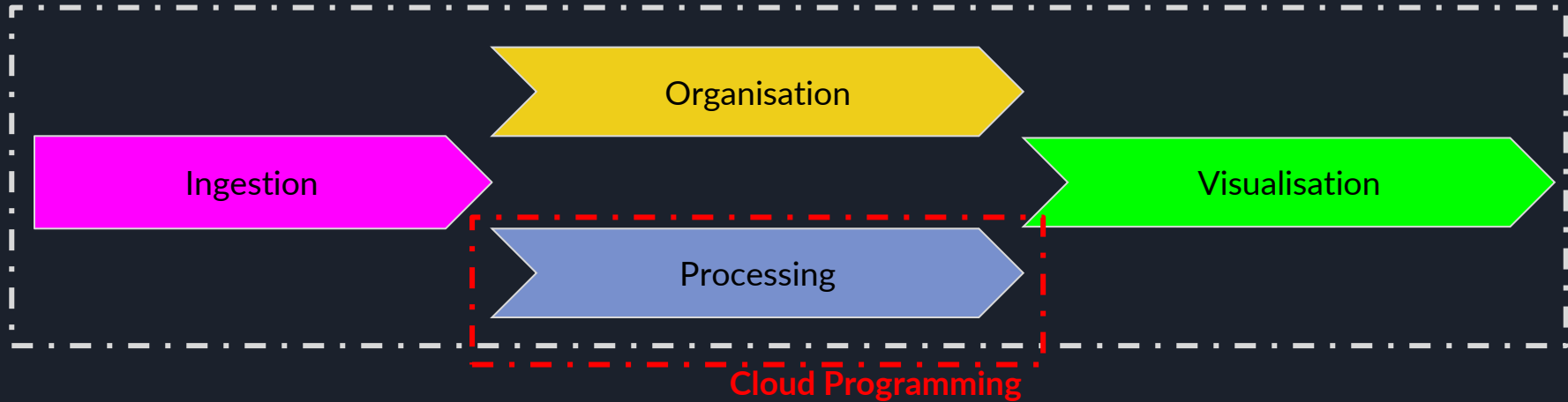
Introduction

Pipeline



Introduction

Pipeline





Présentation des bases de données

Il existe aujourd'hui 2 grands types de bases de données : **Relationnelles** et **Non Relationnelles**.



The diagram consists of a large light green rounded rectangle. Inside this rectangle, at the top center, is the text 'Bases de données'. Below this text, there are two yellow rounded rectangles positioned side-by-side. The left yellow rectangle contains the text 'Bases de données Relationnelles' and the right yellow rectangle contains the text 'Bases de données Non Relationnelles'.

Bases de données

Bases de données
Relationnelles

Bases de données
Non Relationnelles

Bases de données relationnelles

Bénéfices :

Faciles à utiliser

Intégrité de la donnée

Stockage de la donnée réduit

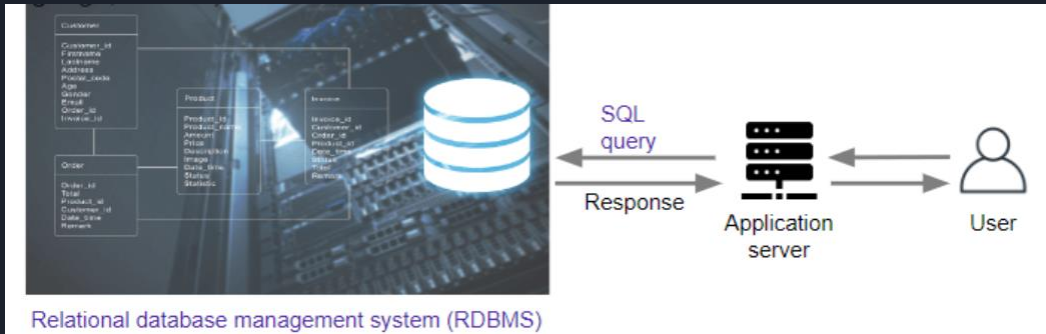
Langage commun (Structured Query Language ou SQL)

Les bases de données relationnelles sont idéales quand vous :

Avez besoin d'avoir un schéma de règle strict, une compliance ACID et une haute qualité de la donnée

Vous n'avez pas besoin d'une capacité en lecture/écriture extrême (plusieurs milliers par seconde)

Quand vous n'avez pas besoin d'une extrême performance



Avantage des bases de données non relationnelles

Bénéfices :

Données non structurées qui peuvent être stockées et traitées

Les bases de données sont plus facilement évolutives et peuvent gérer des quantités de données plus importantes

Les données peuvent être stockées dans différents formats tels que des documents, des graphes ou des paires clé-valeur

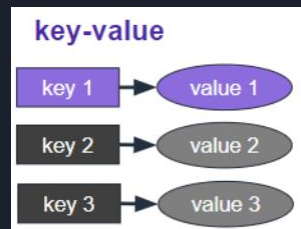
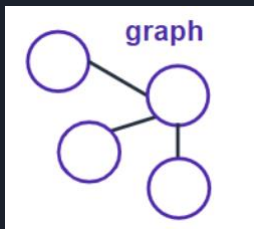
Les bases de données non relationnelles sont plus flexibles et peuvent être adaptées à différents formats de données.

Les requêtes peuvent être exécutées sur plusieurs noeuds ce qui améliore les performances

La base de données peut s'étendre horizontalement pour pouvoir gérer la quantité de données

La donnée ne peut pas être contenue dans des modèles traditionnels (Entité Association)

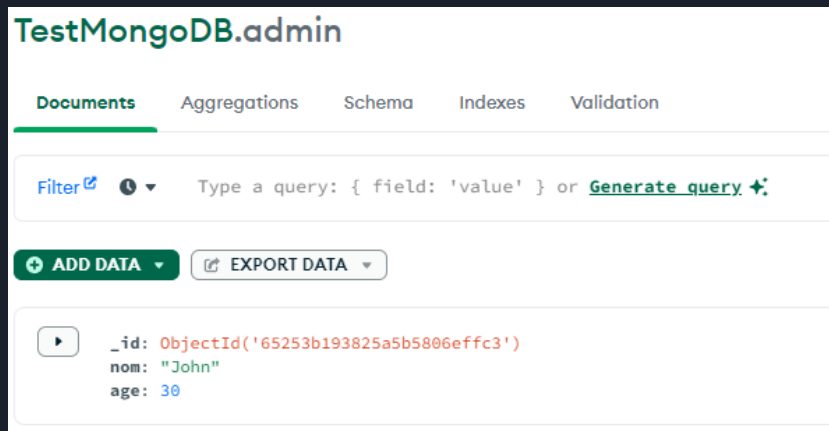
La lecture/écriture dépasse ce que peut supporter un SGBD traditionnel



Introduction

Les bases de données NoSQL sont conçues pour gérer des volumes massifs de données non **structurées** ou **semi-structurées**, ainsi que pour offrir une évolutivité horizontale.

La représentation et l'optimisation des données dans un contexte NoSQL nécessitent une compréhension approfondie des modèles de données, des requêtes et des besoins spécifiques de l'application.





Modèles de Données en NoSQL

Les bases de données NoSQL prennent en charge divers modèles de données, notamment :

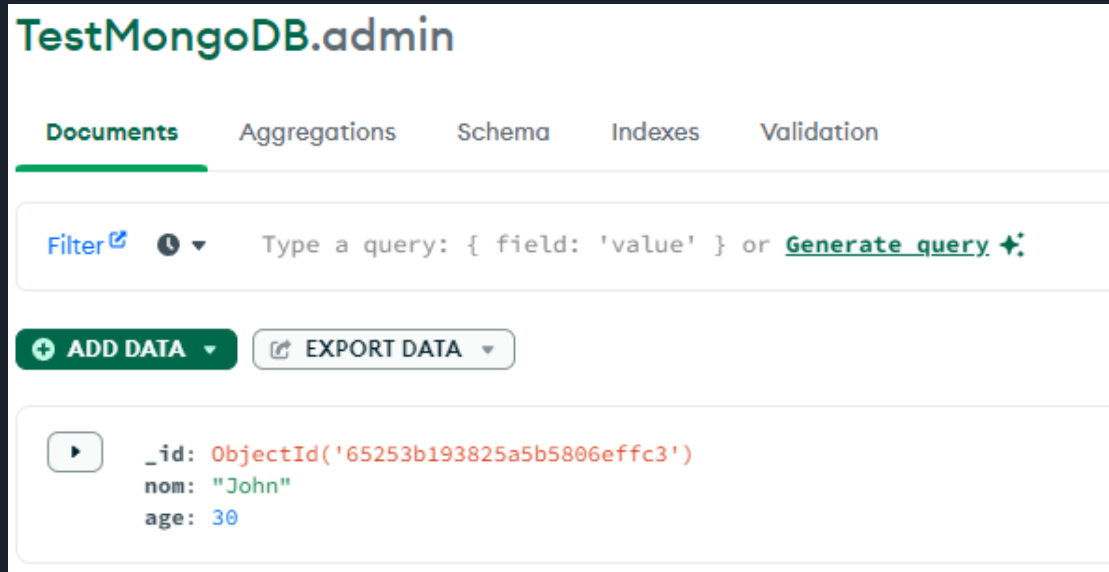
Document : Stocke les données dans des documents (par exemple, JSON ou BSON) pouvant être hiérarchiques.

JSON : JavaScript Object Notation

BSON : Binary JSON (développé par MongoDB)

Modèle de données en NoSQL

Clé-Valeur : Stocke les données sous forme de paires clé-valeur.



The screenshot displays the TestMongoDB.admin web interface. At the top, the title "TestMongoDB.admin" is shown in green. Below it, a navigation bar contains five tabs: "Documents" (highlighted with a green underline), "Aggregations", "Schema", "Indexes", and "Validation". A search bar with a "Filter" button and a clock icon is present, followed by the text "Type a query: { field: 'value' } or [Generate query](#) with a plus icon". Below the search bar are two buttons: a green "ADD DATA" button with a plus icon and a grey "EXPORT DATA" button with a document icon. The main content area shows a single document with a play button icon on the left and the following fields: "_id: ObjectId('65253b193825a5b5806effc3')", "nom: 'John'", and "age: 30".

TestMongoDB.admin

Documents Aggregations Schema Indexes Validation

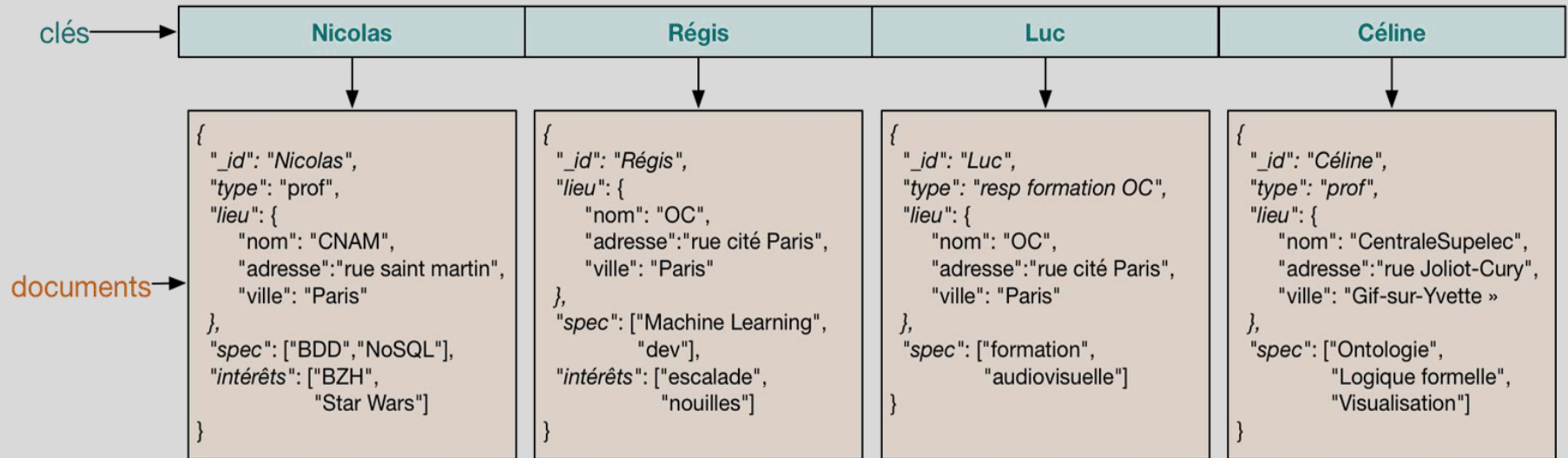
Filter ⌚ Type a query: { field: 'value' } or [Generate query](#) ➕

+ ADD DATA EXPORT DATA

▶ `_id: ObjectId('65253b193825a5b5806effc3')`
`nom: "John"`
`age: 30`

Modèles de données en NoSQL

Colonnes Familles : Stocke les données sous forme de colonnes plutôt que de lignes, adapté aux cas d'utilisation de type base de données en colonnes.



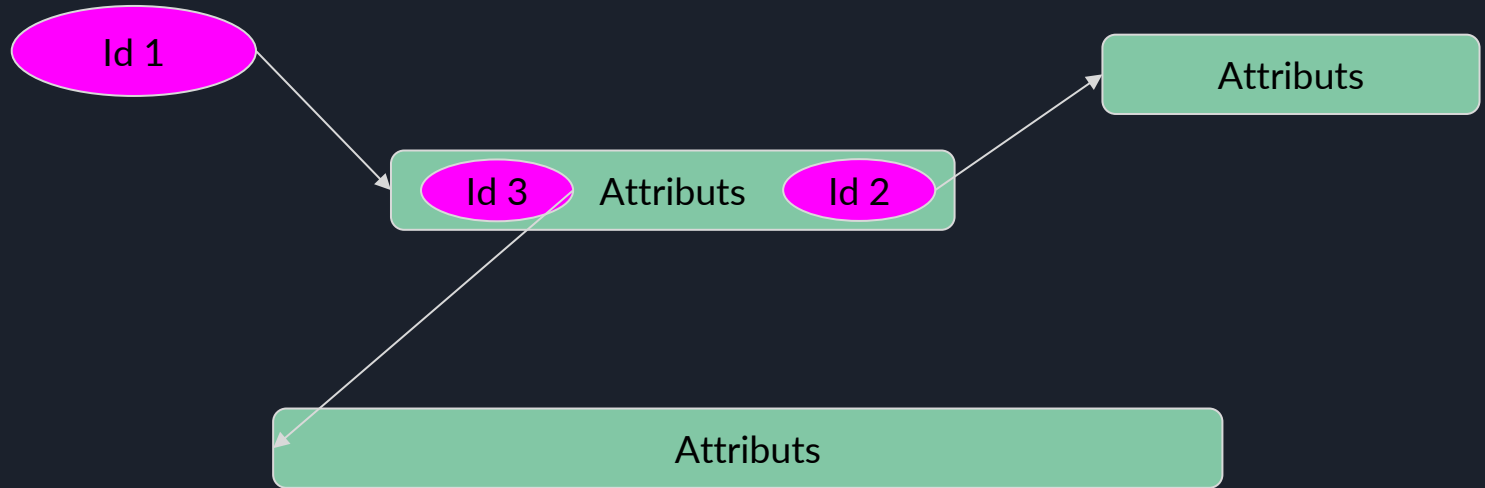
Conception de Schéma Flexible

L'un des avantages clés des bases de données NoSQL est la flexibilité du schéma. Vous pouvez ajouter de nouveaux champs ou modifier la structure des données sans avoir à migrer une base de données complexe. Cette flexibilité permet de s'adapter rapidement aux besoins changeants de l'application.



Modèle de données en NoSQL

Grappe : Stocke les données sous forme de graphes, adapté aux données interconnectées.



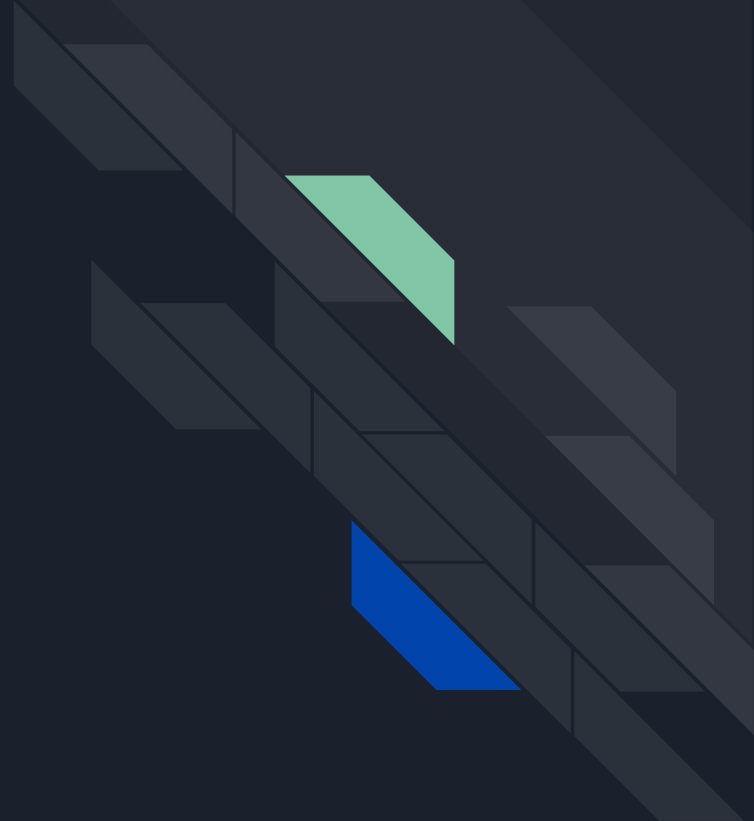


Etude de cas

Trouvez des cas d'application au Big Data et des domaines où une optimisation serait possible.

Réalisez une courte présentation sur ce cas d'application

Modélisation des bases de données industrielles





Le modèle Conceptuel de Données

Le modèle conceptuel de données sert à la représentation abstraite des concepts et relations dans l'industrie. Il permet la description des entités impliquées dans les processus industriels et de les associer. En général, on représente ce modèle par des diagrammes entité-association ou des diagrammes de classes.

Entités

Représentation des objets
intervenant dans les
processus industriels

Relations

Liens entre les entités pour
construire un modèle complet
et logique



Le modèle logique de données

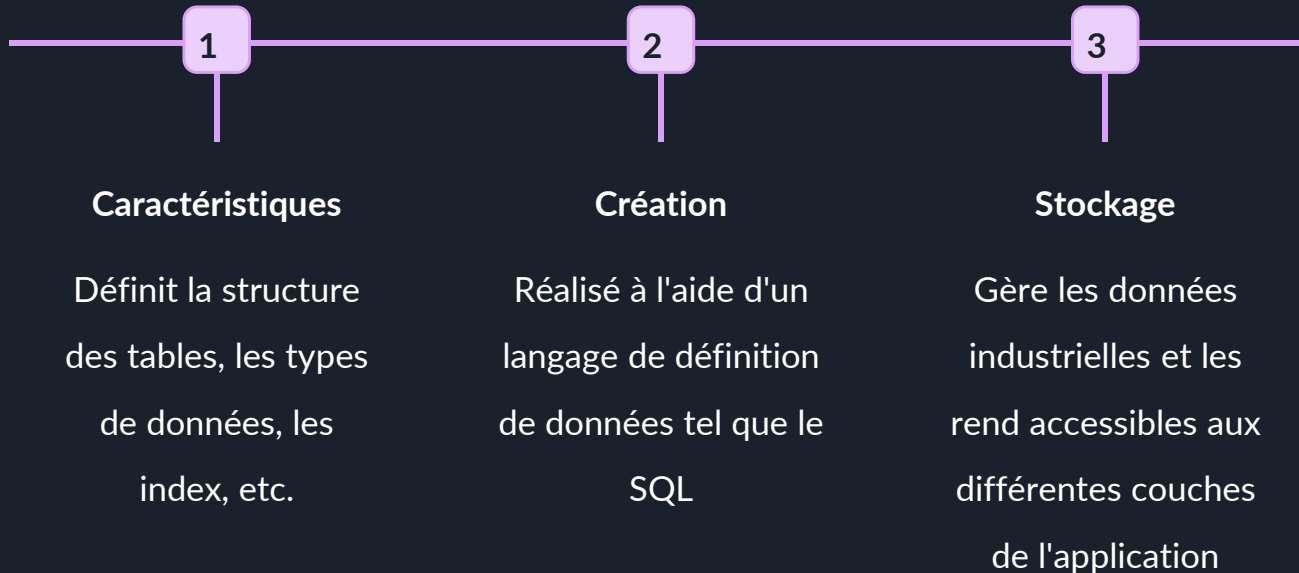
Le modèle logique traduit le modèle conceptuel en une structure plus concrète définissant les tables, les attributs et les relations pour représenter les données industrielles. Il permet de normaliser les données, assurer leur intégrité et cohérence, et faciliter leur manipulation.

Table	Attributs	Relations
Représentation d'une entité impliquée dans les processus industriels	Caractéristiques d'une table définissant la structure des données	Liens établis entre les tables à l'aide de clés primaires et clés étrangères



Schéma de base de données

Le schéma de base de données décrit la structure concrète et physique du modèle logique de données. Il permet de créer la base de données effective stockant les données industrielles, en définissant les tables, les types de données, les contraintes d'intégrité, les index, etc.





La normalisation des bases de données

La normalisation des données est une technique employée pour organiser les données de manière optimale en éliminant les redondances et les anomalies de mise à jour. Elle permet une meilleure efficacité de la base de données, évite les incohérences et facilite les opérations de requête et de manipulation des données.

1

Technique

Décomposition des tables en plusieurs tables plus petites et mieux structurées

2

Optimisation

Réduire les duplications et les anomalies de mise à jour dans les données



Méthodes de modélisation spécifiques

Dans l'industrie, différentes méthodes de modélisation peuvent être utilisées en fonction des caractéristiques propres à chaque industrie. Par exemple, la modélisation orientée objet peut être utilisée pour représenter les objets complexes et les relations dans un environnement de fabrication avancé.

Spécifique

Utilisée en fonction des caractéristiques propres à chaque industrie

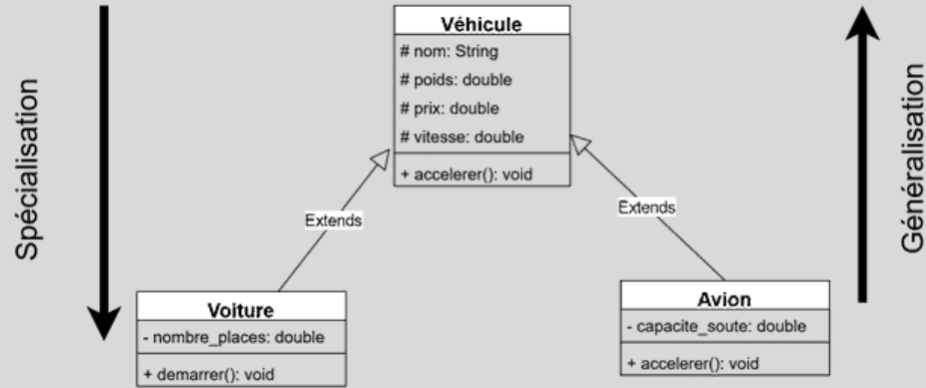
Orientée Objet

Utilisée pour représenter les objets complexes et les relations dans les environnements avancés

Business Process Modeling

Utilisée pour représenter les flux de travail et les interactions entre les différentes étapes des processus industriels

La modélisation orientée Objet




La modélisation orientée objet (MOO) est une méthode de modélisation de données qui utilise des objets pour représenter les entités et les relations entre elles. Dans une MOO, chaque objet est une instance d'une classe, qui définit les propriétés et les comportements de l'objet. Les propriétés sont les caractéristiques de l'objet, telles que son nom, son âge ou sa taille. Les comportements sont les actions que l'objet peut effectuer, telles que se déplacer, changer d'état ou interagir avec d'autres objets. Les outils de MOO les plus courants sont UML (Unified Modeling Language) et SysML (Systems Modeling Language). Ils fournissent des notations standardisées pour représenter les classes, les objets, les associations, les agrégations et les compositions, ainsi que d'autres concepts de la MOO. La MOO est largement utilisée dans le développement de logiciels, mais peut également être appliquée à la modélisation de données industrielles.

La modélisation des processus métiers



La **modélisation des processus métier** (BPM) est une méthode utilisée pour représenter visuellement les processus et les flux de travail d'une entreprise. Elle aide les organisations à améliorer leur efficacité et leur efficience en identifiant des opportunités d'optimisation et d'automatisation. Les diagrammes BPM peuvent être utilisés pour communiquer des informations sur les processus aux parties prenantes et aider à la prise de décision. Les notations BPM courantes comprennent **BPMN** (Business Process Model and Notation) et **EPC** (Event-driven Process Chain). Grâce à la BPM, les organisations peuvent mieux comprendre leurs processus, réduire les coûts et améliorer la satisfaction client.




L'ingénierie système basée sur des modèles (MBSE)

La MBSE est une méthodologie utilisée pour concevoir et construire des systèmes complexes. Elle implique l'utilisation de modèles pour représenter le système et ses composants, et l'utilisation de ces modèles pour analyser, simuler et optimiser le système. Cette approche aide à s'assurer que le système répond à ses exigences et fonctionne comme prévu.



ARCADIA

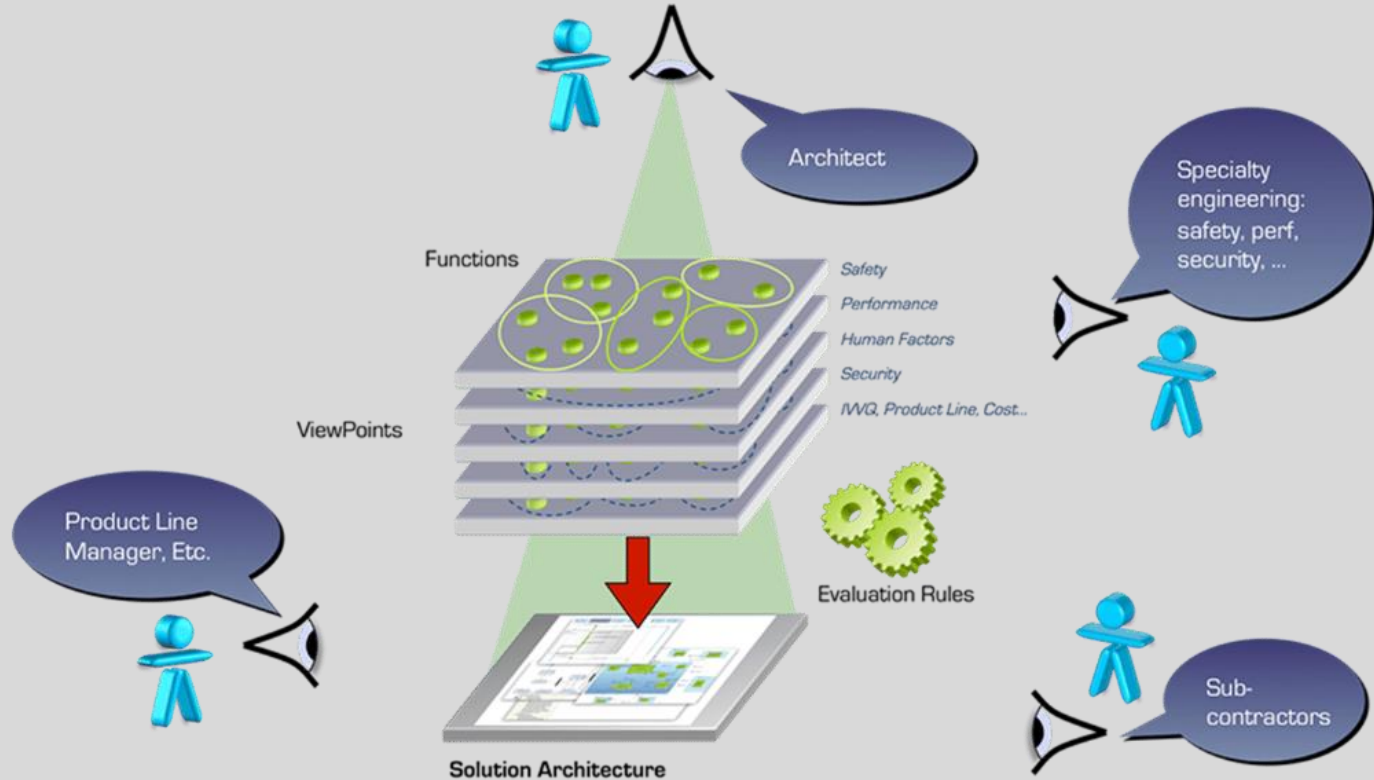


L'ingénierie système basée sur des modèles (MBSE)

La MBSE est souvent utilisée dans des industries comme l'aérospatiale et la défense, où la complexité et la sécurité sont des enjeux majeurs. Elle permet aux ingénieurs de mieux comprendre les interactions entre les différents composants du système, de prédire les performances du système et de détecter les problèmes potentiels plus tôt dans le processus de conception.



L'ingénierie système basée sur des modèles (MBSE)





Big Data et Industrie 4.0

Le Big Data joue un rôle clé dans l'industrie 4.0 en collectant, analysant et utilisant des données en temps réel pour améliorer l'efficacité, réduire les coûts, faciliter la maintenance et la prévention des pannes, et optimiser les performances des processus industriels. Il permet également l'intégration de systèmes d'automatisation avancés, de l'Internet des Objets et de la réalité augmentée pour maximiser les performances.

Collecte et analyse en temps réel des données

Optimisation des performances des processus industriels

Systèmes d'automatisation avancées intégrés



Big Data et Industrie 4.0

IA

Cybersécurité

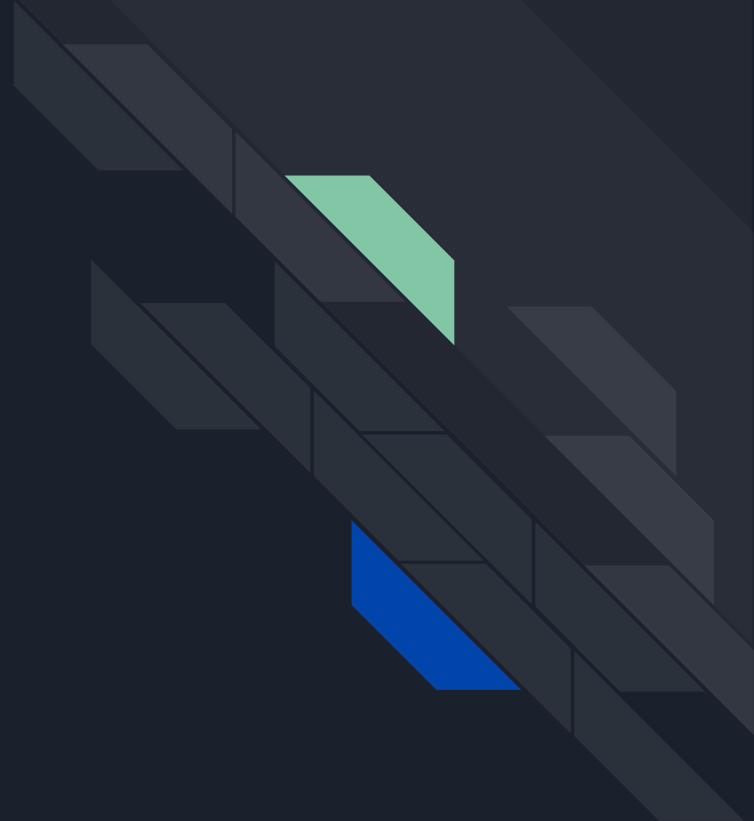
Big Data
Industrie 4.0



Conclusion

La modélisation des données est une étape cruciale dans la mise en place des systèmes d'informations industriels. Elle permet de collecter, stocker, et gérer les données de manière efficace et de les rendre intelligibles aux différents utilisateurs. Avec l'essor des technologies telles que le Big Data, l'Intelligence Artificielle, l'Industrie 4.0, il devient impératif de maîtriser la modélisation des données industrielles pour répondre aux enjeux du futur et de rester compétitifs dans le domaine industriel.

Les techniques de gestion de données massives





L'exploitation de données massives

Les données industrielles contiennent des informations précieuses pour les entreprises.

L'analyse et l'exploitation de ces données peuvent permettre d'améliorer les performances et l'efficacité opérationnelle, ainsi que d'anticiper les pannes.





Le traitement de données complexes

Les données industrielles peuvent être complexes et nécessiter des techniques avancées de traitement et d'analyse telles que l'apprentissage automatique, l'analyse prédictive et la fouille de données. Les systèmes d'informations industriels doivent intégrer ces techniques pour obtenir des informations pertinentes à partir des données.

Apprentissage automatique

Utilisation de l'apprentissage automatique pour étudier les comportements des machines et des équipements.

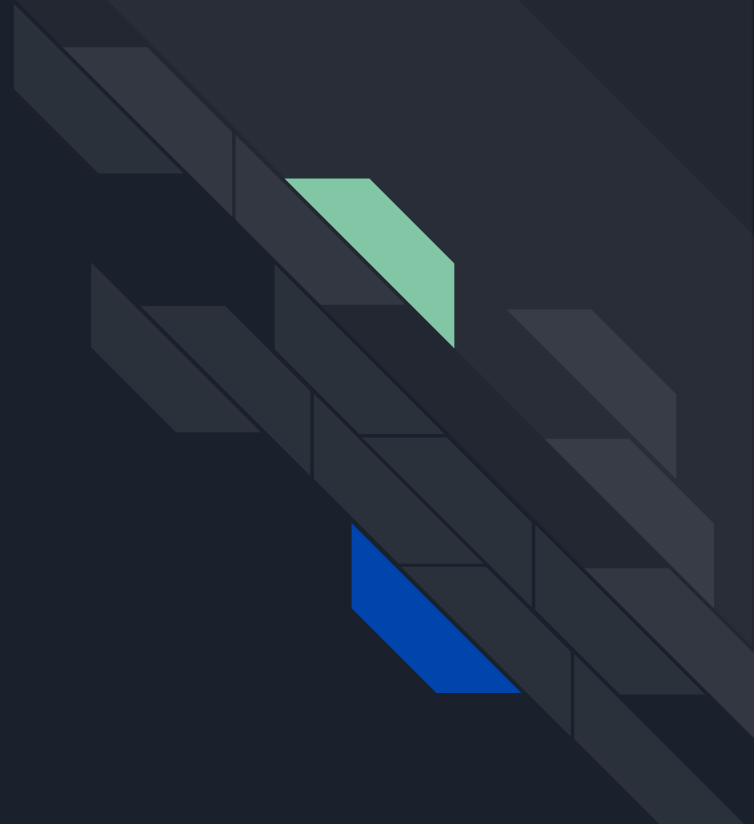
Optimisation des données

Utilisation de l'optimisation de données pour des opérations et prises de décisions plus intelligentes.

Modèles de données adaptés

Utilisation de modèles de données tels que les entrepôts de données et les bases de données NoSQL pour gérer les données industrielles.

Quelles sont les
solutions d'aujourd'hui ?





Google File System (GFS)

Remplacé par Colossus

GFS a été conçu pour répondre aux besoins de stockage de données dans les applications Google. il est optimisé pour la gestion de fichiers de taille importante.

Fault Tolerance and Recovery

Google File System (GFS)



Google File System (GFS)

GFS est un système de fichiers distribué développé par Google pour gérer efficacement de grandes quantités de données.

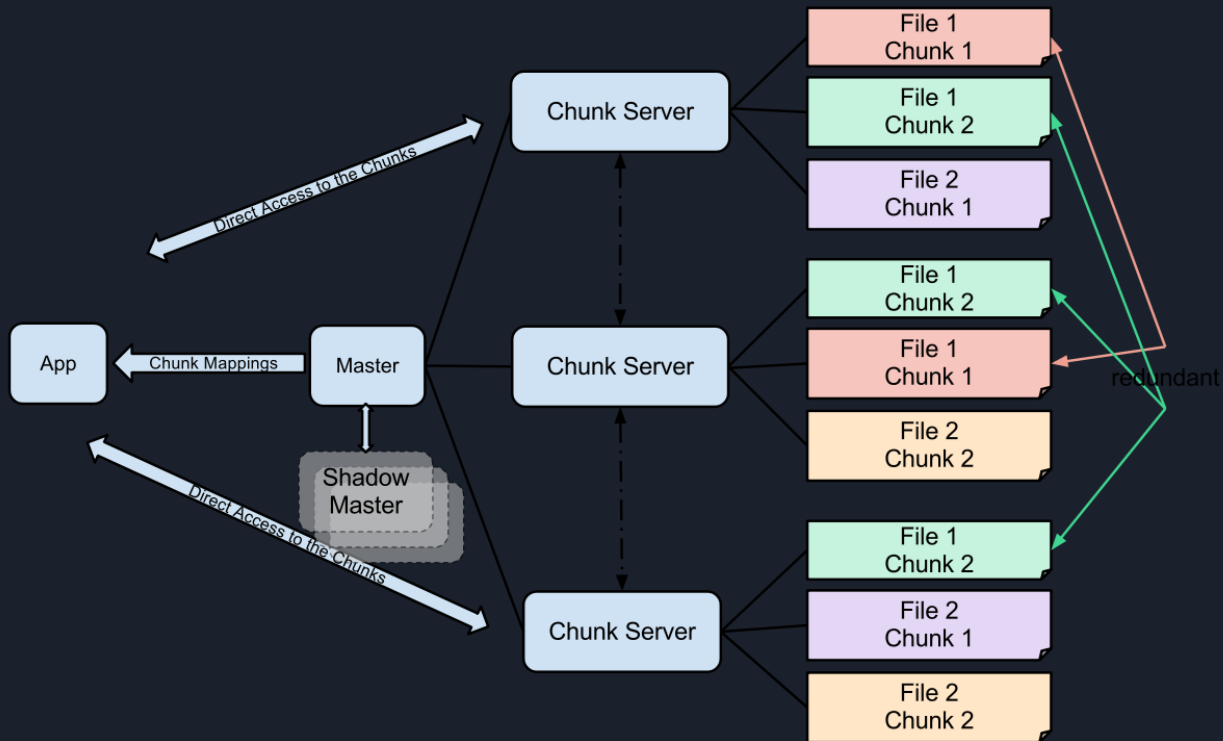
Redondance des données

GFS réplique
automatiquement les
données pour garantir leur
disponibilité en cas de
défaillance matérielle

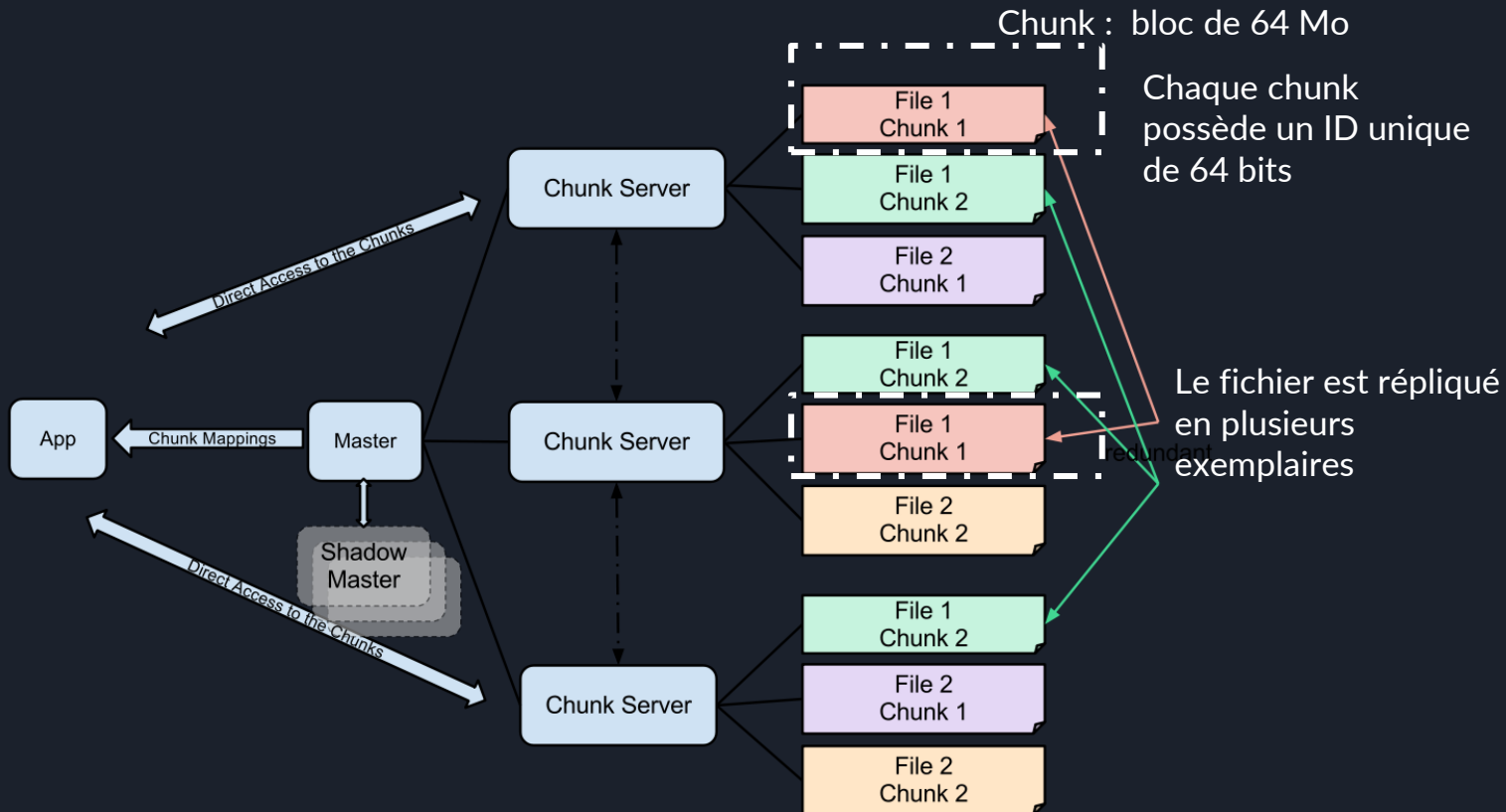
Haute performance

Efficacité dans la gestion
des fichiers volumineux et
les multiples lectures en
parallèle

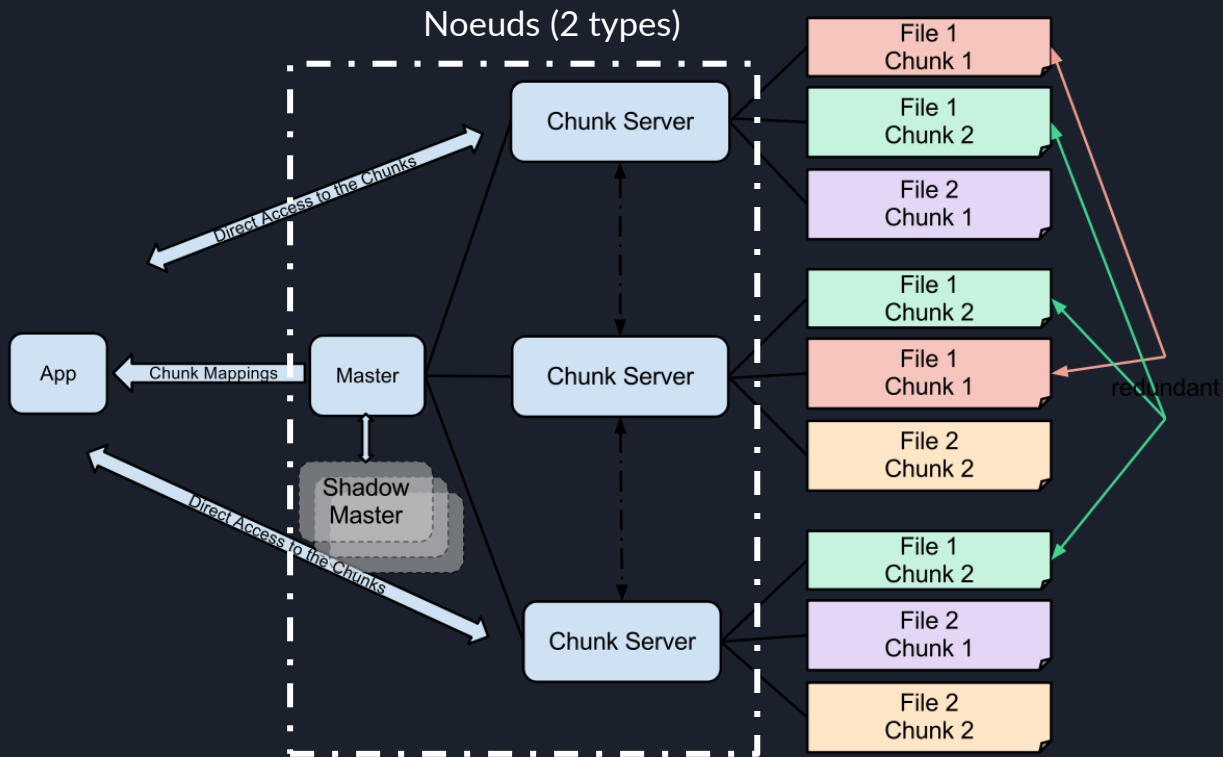
Google File System (GFS)



Google File System (GFS)

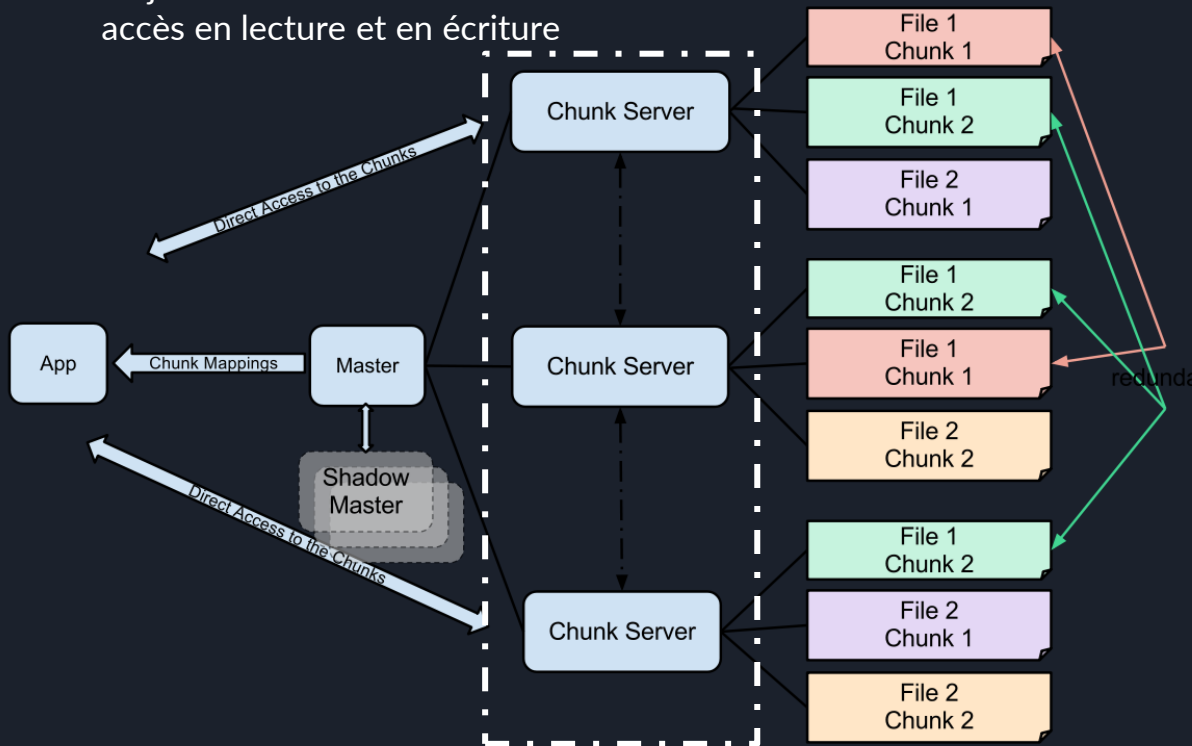


Google File System (GFS)



Google File System (GFS)

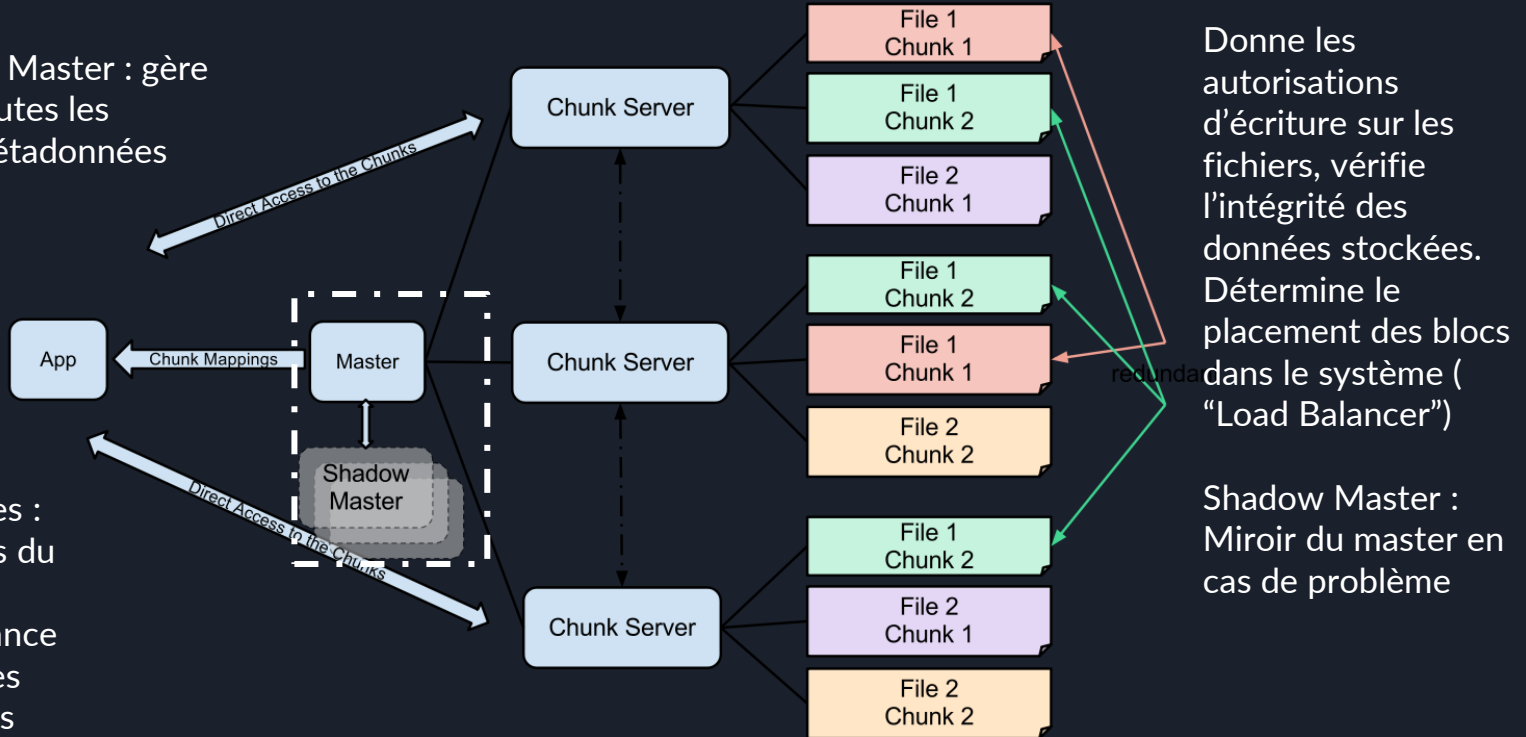
Objectif : stocker les chunks et effectuer les accès en lecture et en écriture



Effectue l'opération d'écriture sur le bloc considéré comme "copie primaire" puis donne l'ordre aux copies secondaires d'effectuer la même opération. Opérations d'écriture effectuées dans le même ordre sur toutes les répliques

Google File System (GFS)

Le Master : gère toutes les métadonnées

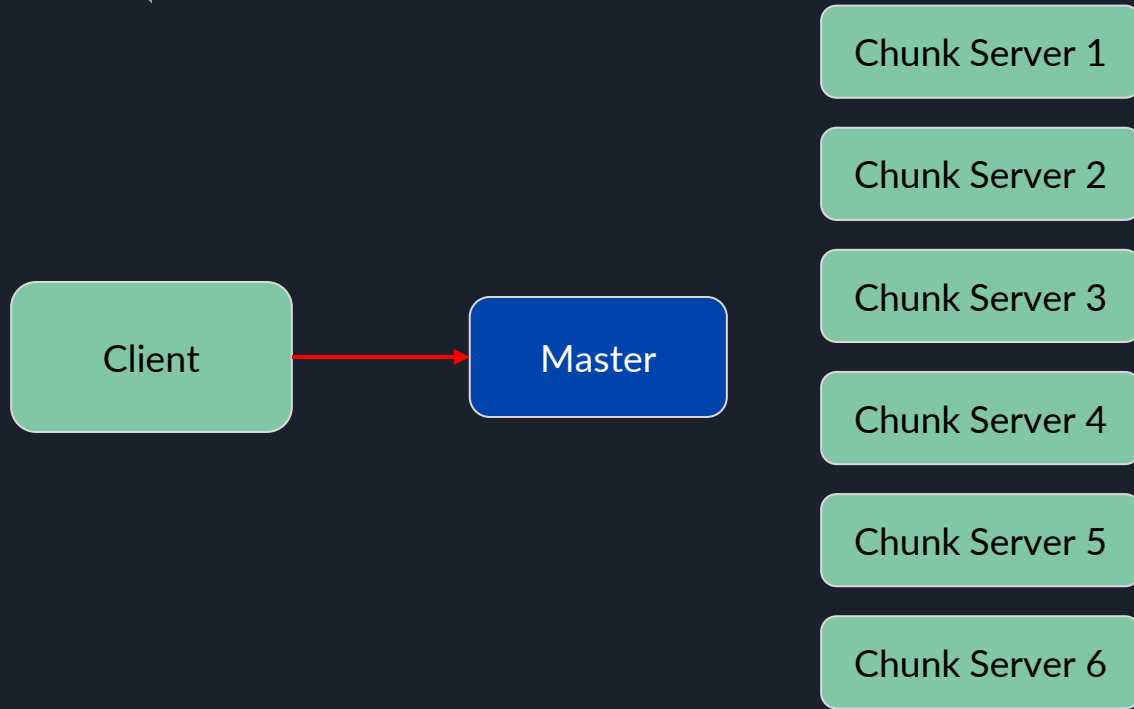


Donne les autorisations d'écriture sur les fichiers, vérifie l'intégrité des données stockées. Détermine le placement des blocs dans le système ("Load Balancer")

Shadow Master : Miroir du master en cas de problème

Métadonnées : Informations du namespace correspondance entre tous les fichiers et les chunks qui le composent

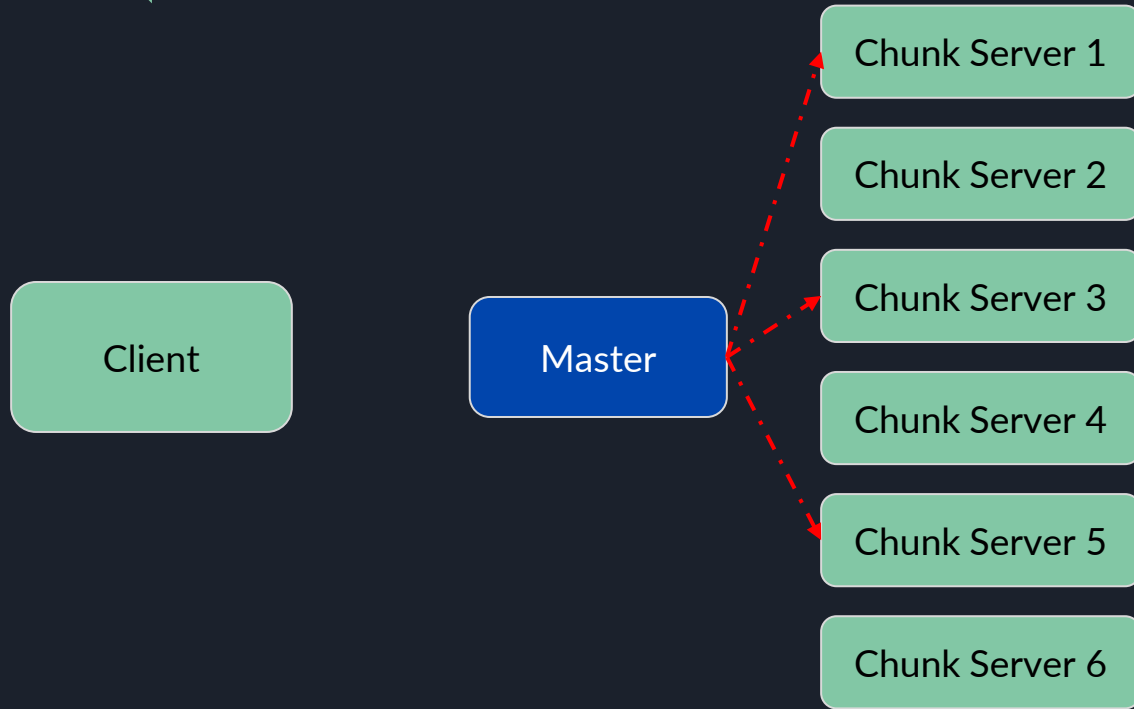
Les opérations dans GFS



Lecture

Pour effectuer une lecture, un client commence par demander au master l'adresse des machines possédant une copie du chunk qui l'intéresse.

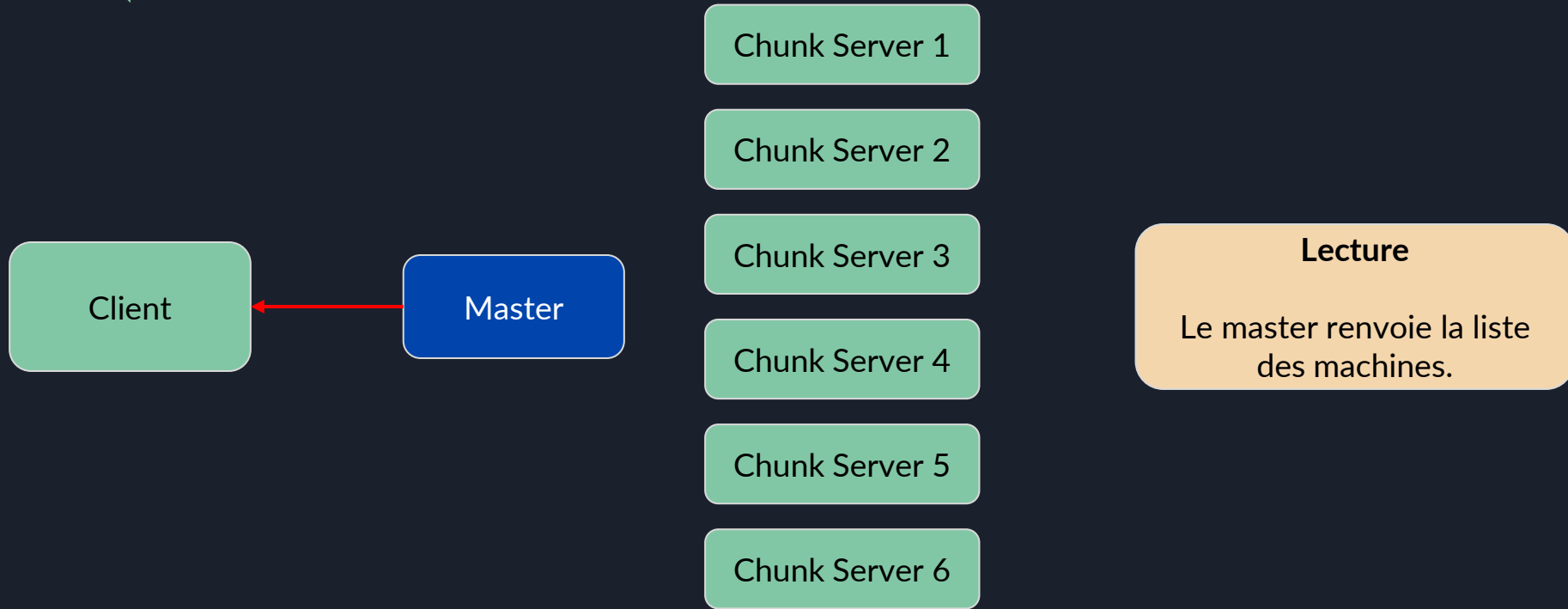
Les opérations dans GFS



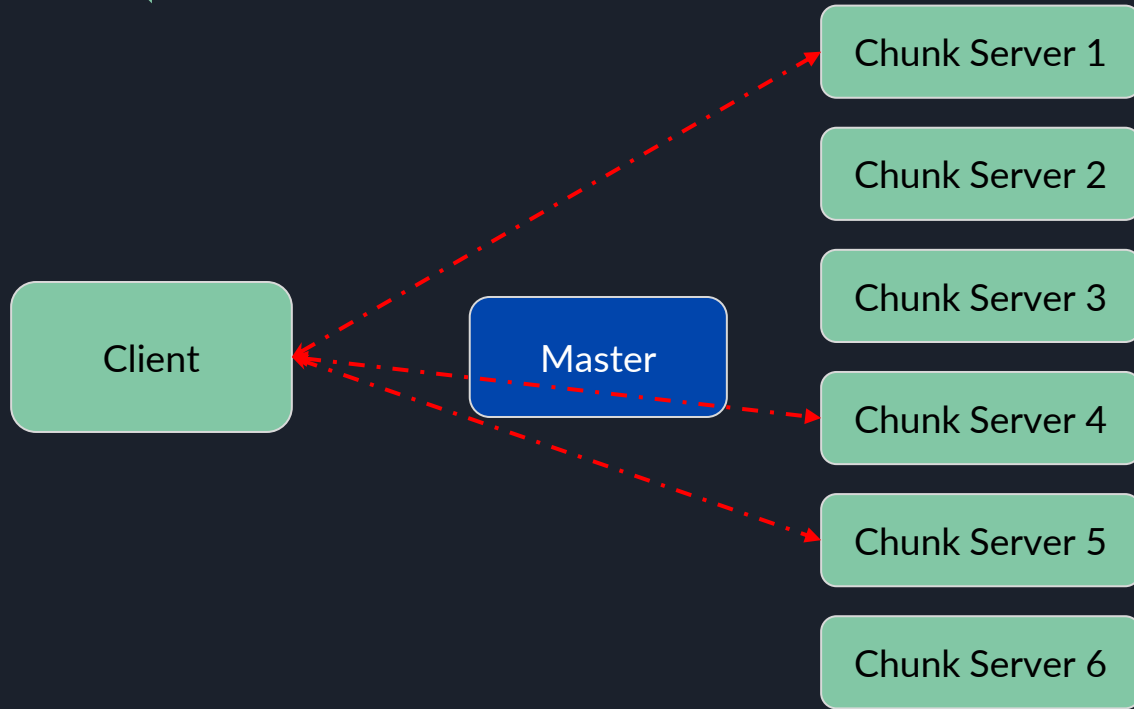
Lecture

Le master interroge les chunkservers qui possèdent le fichier.

Les opérations dans GFS



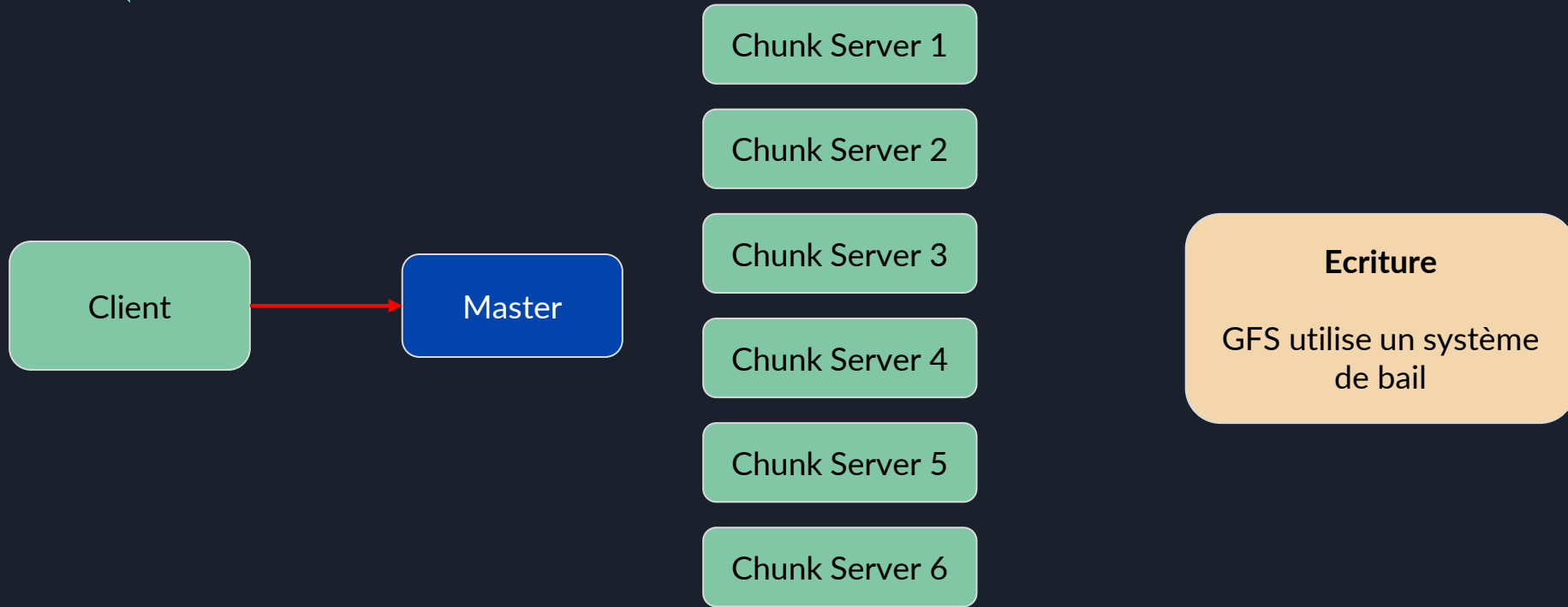
Les opérations dans GFS



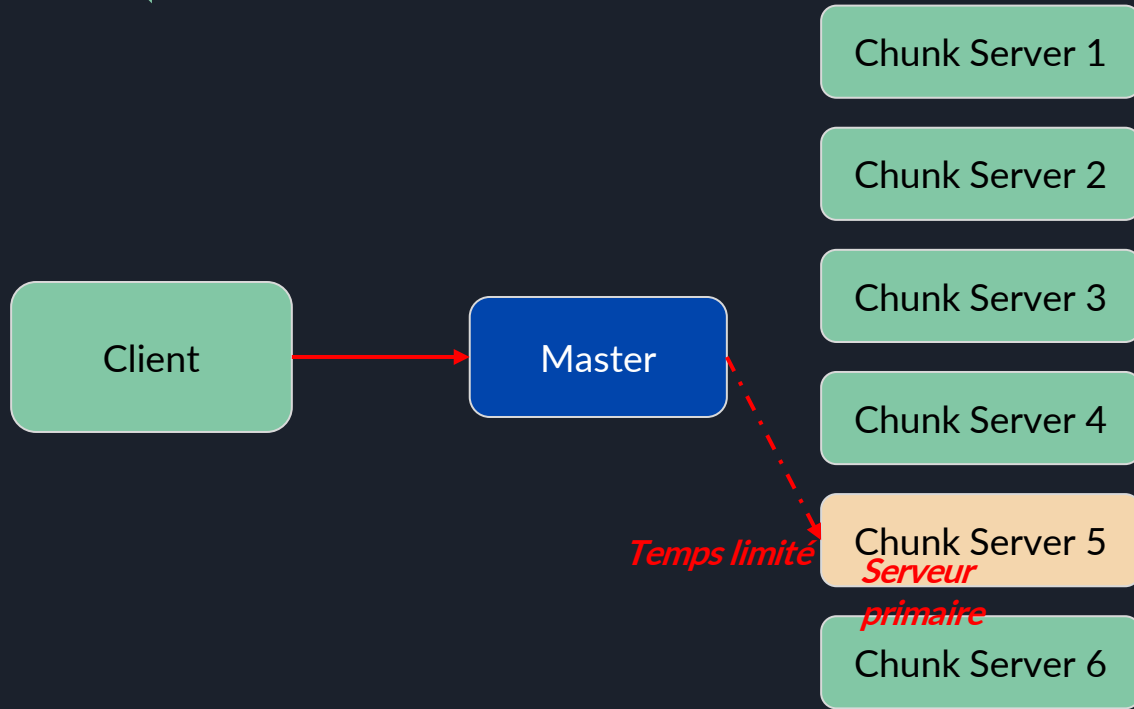
Lecture

Le client s'adresse ensuite directement à l'un des chunkservers qui lui envoie les données désirées.

Les opérations dans GFS



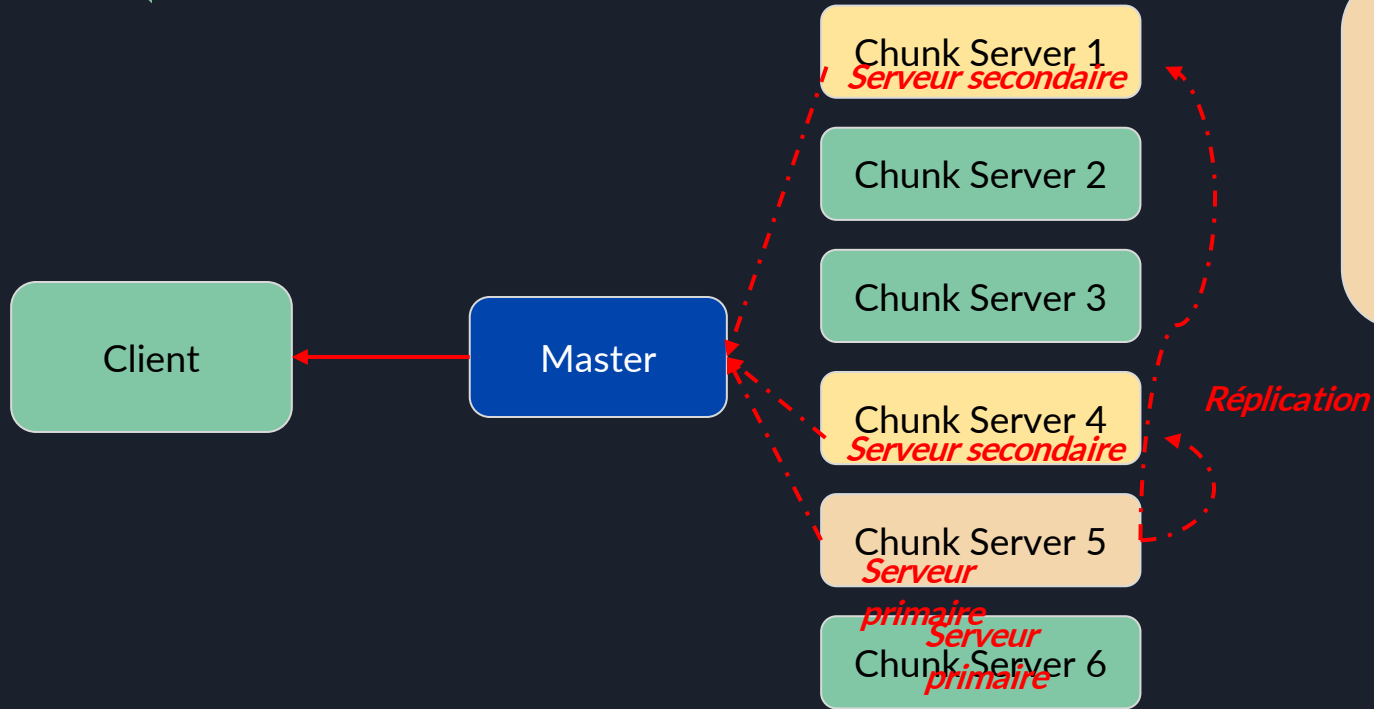
Les opérations dans GFS



Ecriture

Il choisit l'un des chunkservers possédant une copie du bloc et lui accorde un temps limité pour effectuer des écritures, ce serveur est alors appelé "primaire".

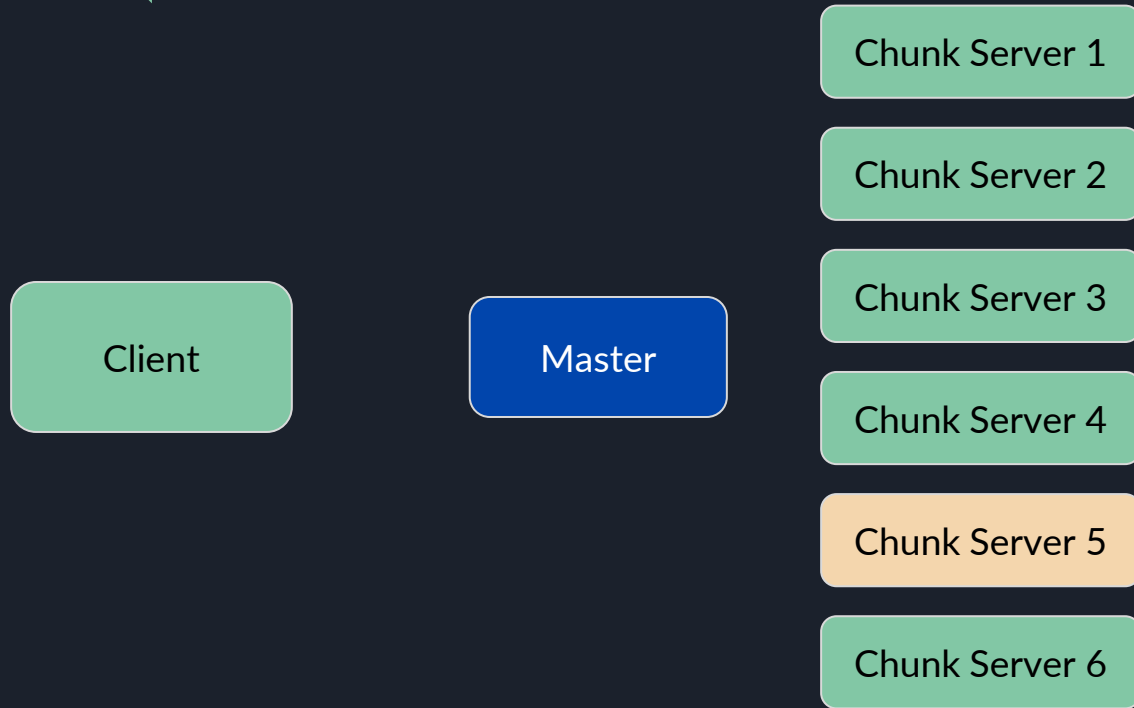
Les opérations dans GFS



Ecriture

Le master envoie ensuite au client l'adresse du primaire et des copies secondaires

Les opérations dans GFS

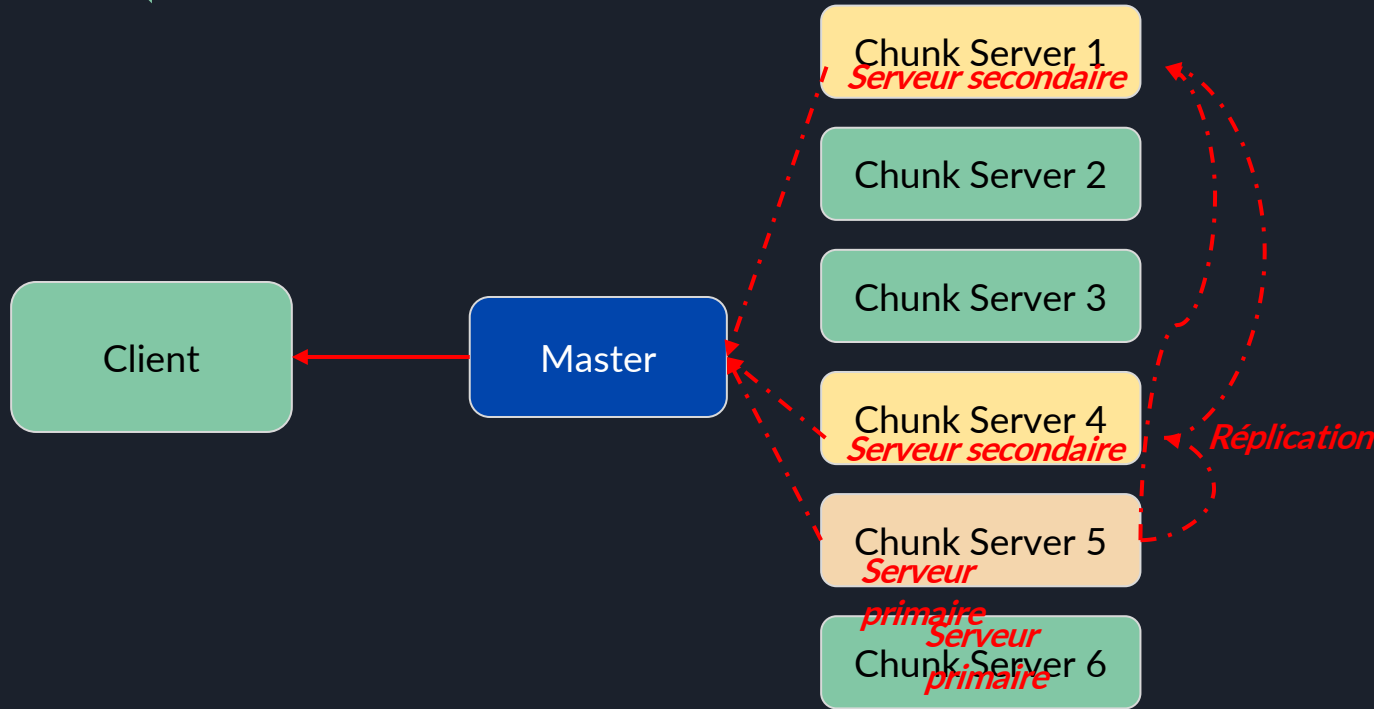


Mise à jour des checksum

Lors d'une écriture, les clients doivent mettre à jour la checksum des blocs (64ko) concernés.

Checksum

Les opérations dans GFS



Flux de données :

Lors d'une écriture, les données sont transmises à un pipeline : le client envoie ses données au chunkserver primaire. Ce dernier tente de trouver la machine la plus proche de lui (par l'adresse IP) puis envoie l'information le réseau étant en mode "full-duplex" un serveur peut transmettre les données dès qu'il commence à les recevoir.



HADOOP

Hadoop est un framework open source écrit en Java pour le traitement distribué de données volumineuses.





HADOOP

Le framework d'Hadoop se décompose en 4 modules : le système de stockage HDFS (Hadoop Distributed File System) et le framework de traitement MapReduce.

HDFS

*Hadoop Distributed File
System*

Hadoop Map Reduce

Hadoop YARN

*Yet Another Resource
Negotiator*

Hadoop Common



HDFS

HDFS
*Hadoop Distributed File
System*

Le noyau d'Hadoop est constitué d'une partie de stockage que l'on nomme HDFS et d'une partie de traitement appelée "Map-Reduce". Hadoop fractionne les fichiers en gros blocs et les distribue à travers les noeuds du cluster.

Pour traiter les données, il transfère le code à chaque noeud qui va traiter les données dont il dispose.

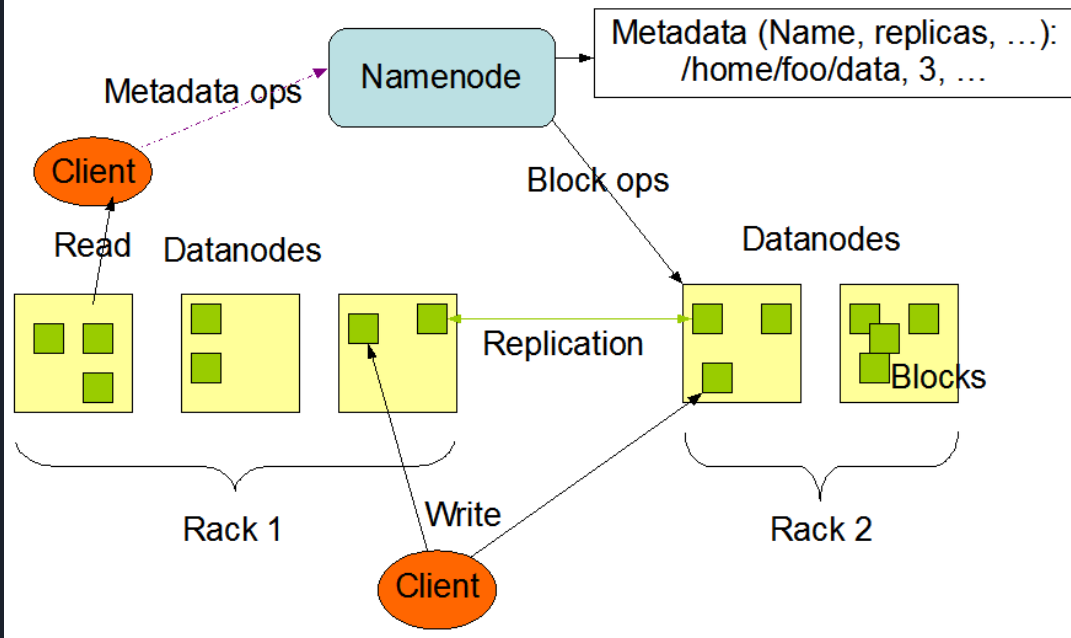
HDFS

HDFS développé
par Hadoop à
partir de Google FS

NameNode

DataNode

HDFS Architecture





HDFS

NameNode

Le NameNode gère l'espace de noms, l'arborescence du système de fichiers et les métadonnées des fichiers et des répertoires.

Il centralise la localisation des blocs répartis dans le cluster.

Il est unique et dispose d'une instance back-up secondaire qui gère l'historique des modifications dans le système.

Ce NameNode "back-up" permet la continuité du fonctionnement du cluster en cas de panne du NameNode d'origine



HDFS

DataNode

Le DataNode est un composant qui stocke et restitue les blocs de données. Lors du processus de lecture d'un fichier, le NameNode est interrogé pour localiser l'ensemble des blocs de données. Pour chacun d'entre eux, le NameNode renvoie l'adresse du DataNode le plus accessible, c'est à dire que le Data Node qui dispose de la plus grande bande passante.

Les DataNodes communiquent de manière périodique au NameNode la liste des blocs de données qu'ils hébergent. Si certains de ces blocs ne sont pas assez répliqués dans le cluster, l'écriture de ces blocs s'effectue en cascade par copie sur d'autres



Apache Spark

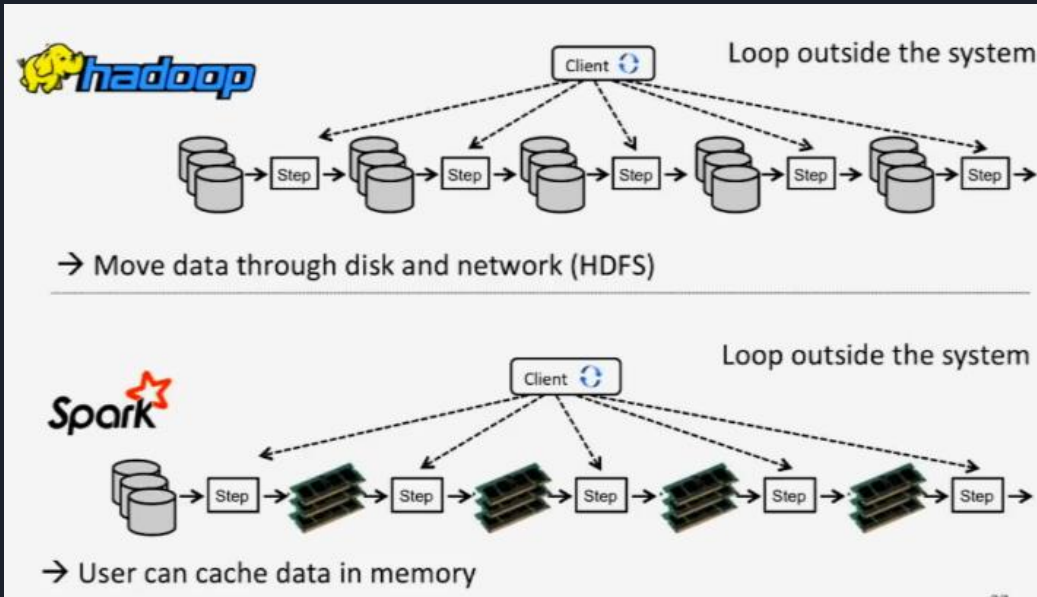
Spark est un moteur de traitement de données open source pour le traitement de données en temps réel et en batch. On l'utilise beaucoup dans le Big Data pour l'apprentissage automatique et les applications d'IA

Le moteur d'analyse de Spark traite les données 10 à 100 fois plus vite que les autres solutions. Il s'adapte en répartissant le travail de manière distribuée avec un parallélisme et une tolérance aux pannes intégrés.



Apache Spark vs Hadoop

Il y a souvent confusion et comparaison entre Hadoop et Spark (en particulier Map-Reduce). La principale différence réside dans le fait que Spark traite et conserve les données en mémoire pour les étapes suivantes, sans écrire ni lire sur le disque. Ce qui permet d'accélérer considérablement les vitesses de traitement.





Apache Cassandra

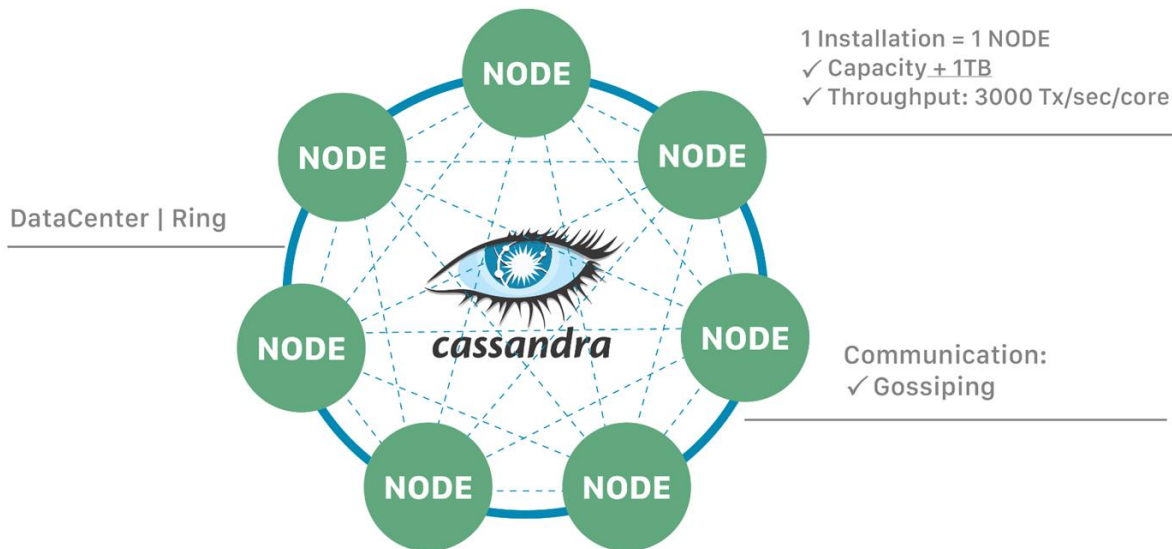
Cassandra est une base de données NoSQL distribuée conçue pour gérer de grandes quantités de données sur plusieurs serveurs.



Apache Cassandra

Apache Cassandra est un SGBD de type NoSQL conçu pour gérer des quantités massives de données sur un grand nombre de serveurs, assurant une haute disponibilité en éliminant les SPoF (Single Point of Failure).

ApacheCassandra™ = NoSQL Distributed Database





Le langage de BDD pour Cassandra

Le langage de requête pour Cassandra s'appelle le CQL (Cassandra Query Language). Des implémentations (SDK) existent pour Java (JDBC), Python (DBAPI2), Node.js (Helenus) etc..

Example query 1:

```
CREATE TABLE playlists(  
  Id uuid,  
  Song_order int,  
  Song_id uuid,  
  title text,  
  album text,  
  artist text,  
  PRIMARY KEY (id, song_order));
```

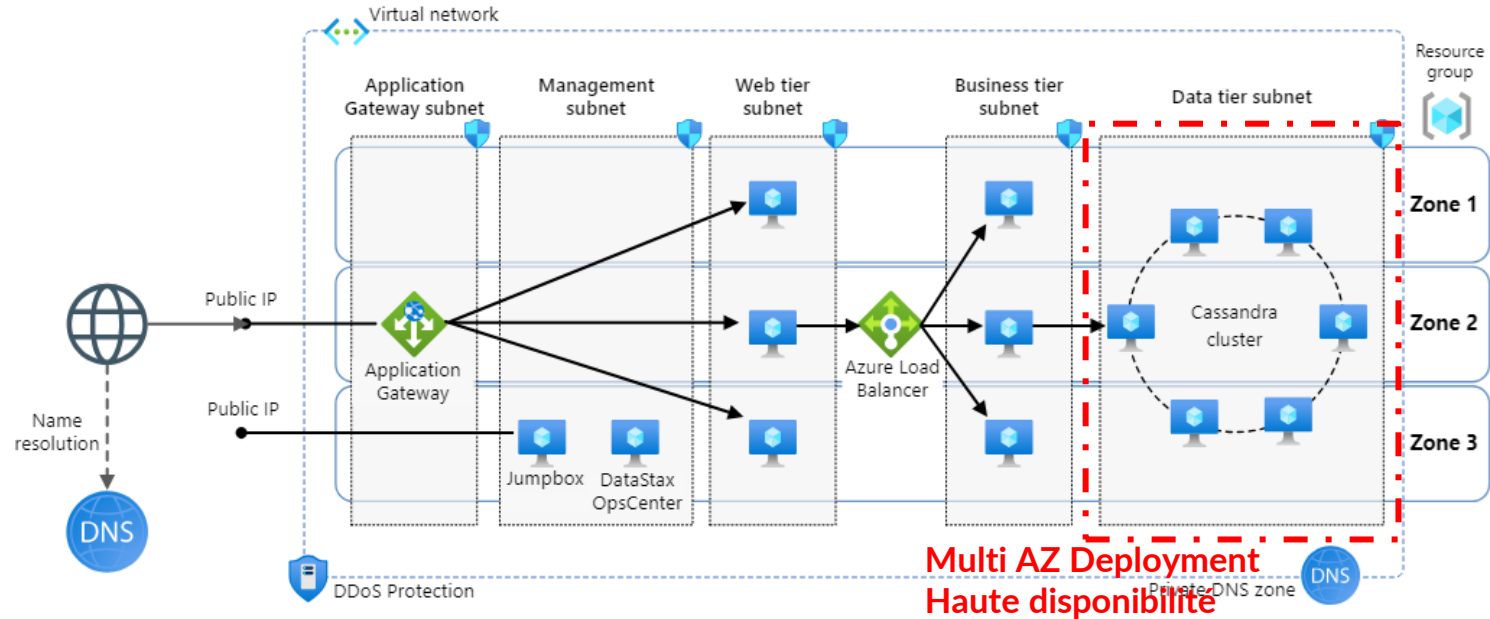
Example query 2:

```
INSERT INTO playlist (id, song_order, song_id, title, artist, album)  
VALUES (62c36092-82a1-3a00-93d1-46196ee77204, 4,  
       7db1a490-5878-11e2-bcfd-o800200c9a66,  
       'ojo Rojo', 'Fu Manchu', 'No One Rides for Free');
```

Example query 3:

```
SELECT * FROM playlists;
```

Apache Cassandra dans la pratique

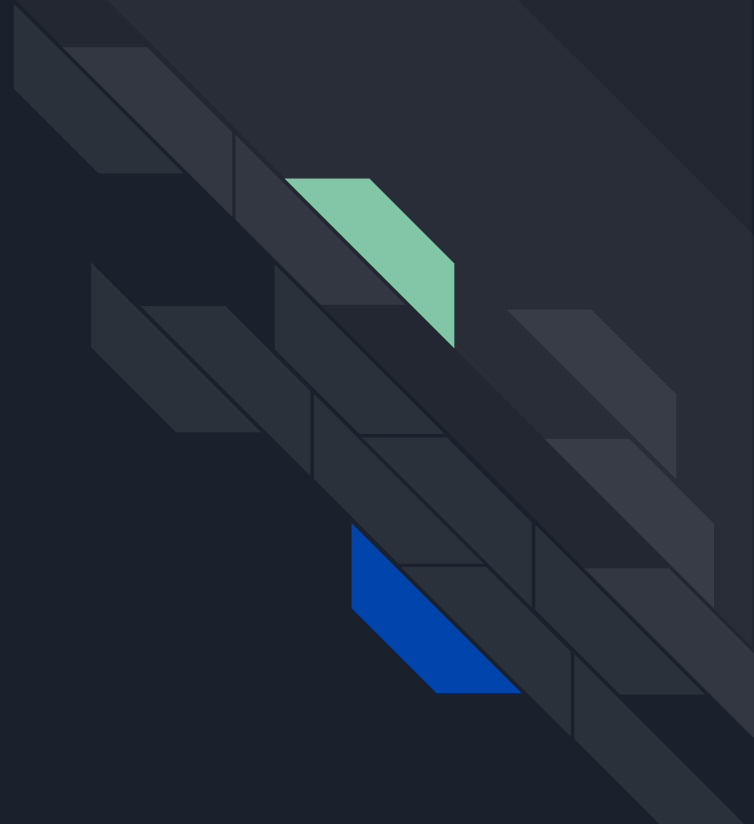




Conclusion

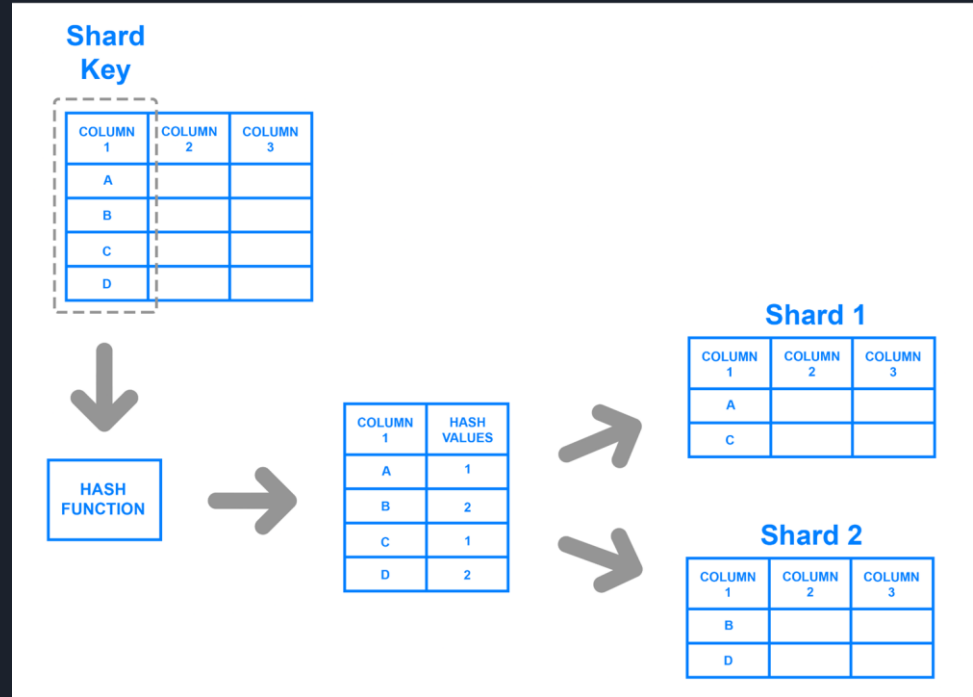
Le Cloud Programming est un domaine en constante évolution avec de nombreuses solutions adaptées à différents besoins. Les solutions telles que MongoDB, GFS, Hadoop, DynamoDB, Spark et Cassandra sont largement utilisées dans le monde entier pour gérer et traiter efficacement les données à grande échelle dans des environnements cloud. Le choix de la solution dépendra des exigences spécifiques de votre projet et de la capacité à répondre à ces exigences.

Réplication & Sharding



Qu'est ce que le sharding ?

Le “sharding” en anglais signifie “éclater”. Dans le Cloud Programming, c’est un concept qui permet de partitionner un ensemble de données venant d’une même base de données. On fractionne ainsi notre base de données en plusieurs sous ensembles appelés “datasets”.





Qu'est ce que le sharding ?

Il existe 2 principales méthodes de sharding : le partitionnement **horizontal** et le partitionnement **vertical**.

Le partitionnement **vertical** permet de séparer les colonnes et de les stocker dans différents serveurs.

Le partitionnement **horizontal** permet de séparer les informations en différentes tables pour une même entrée (Formes normales en BDDR)

Le Sharding Vertical

Le sharding **vertical** permet de séparer les données en plusieurs catégories.



Le Sharding horizontal

Le sharding **horizontal** permet de répartir les données sur plusieurs serveurs.





Avantages et désavantages du sharding

Le Sharding permet de dépasser la capacité d'une seule machine en hébergeant les données sur différents serveurs.

Rapidité

Sécurité

Déséquilibre des
fragments

Gestion de la
synchronicité



Les prérequis pour effectuer du sharding

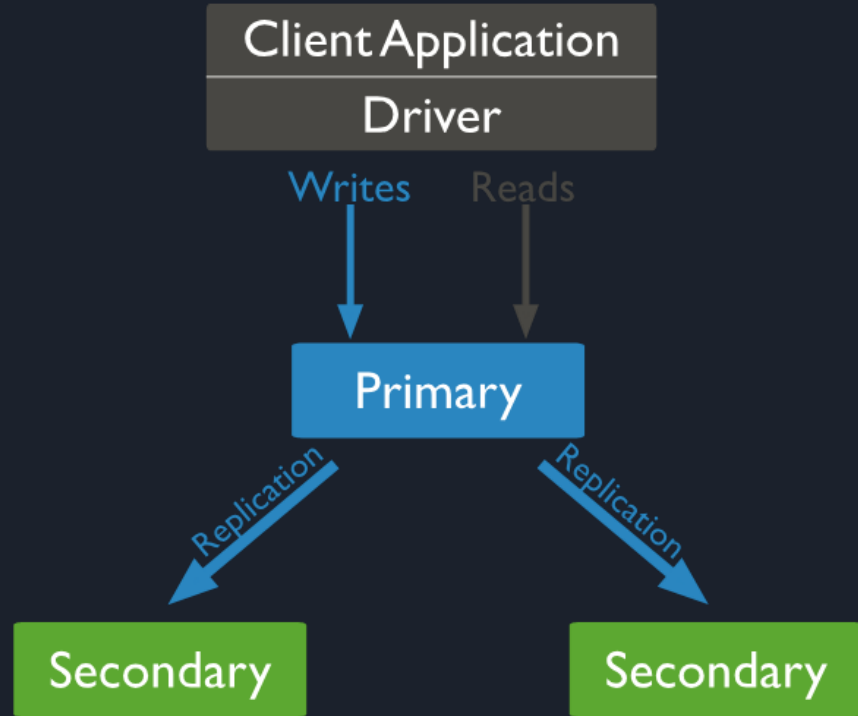
Tous les accès aux données se font via une **clé de répartition** (appelée **Shard Key**)

Chaque table d'une BDD doit posséder une colonne correspondant à la **Shard Key**. Il doit y avoir unicité de toutes les partitions. Aussi, la jointure entre les tables doit être réalisée à partir de la même clé de partition.

La réplication

1. Réplication en MongoDB

La réplication MongoDB permet de maintenir la redondance et la disponibilité des données. Un ensemble de rélicas MongoDB est composé de plusieurs nœuds, dont l'un est le nœud primaire et les autres sont des nœuds secondaires.





Etude de cas : Application du Big Data dans vos SII

Les Systèmes d'Information Industriels (SII) sont de plus en plus confrontés à des défis liés à l'intégration de données provenant de sources multiples et variées. Le Big Data offre une solution pour traiter ces données massives et complexes en temps réel, ce qui permet aux entreprises de mieux comprendre leurs processus, d'optimiser leurs opérations et de prendre des décisions plus éclairées. Dans cette étude de cas, nous explorerons comment le Big Data a été appliqué avec succès dans des projets de SII pour améliorer les performances, la qualité et la fiabilité des systèmes informatiques industriels.