Université Constantine 2
جامعة قسنطينة 2

Big Data & NoSQL (BDNS)

– Chapitre 3–

Stockage , traitement et architectures Big Data

Dr. GHEMMAZ W

Faculté NTIC

Wafa.ghemmaz@univ-constantine2.dz

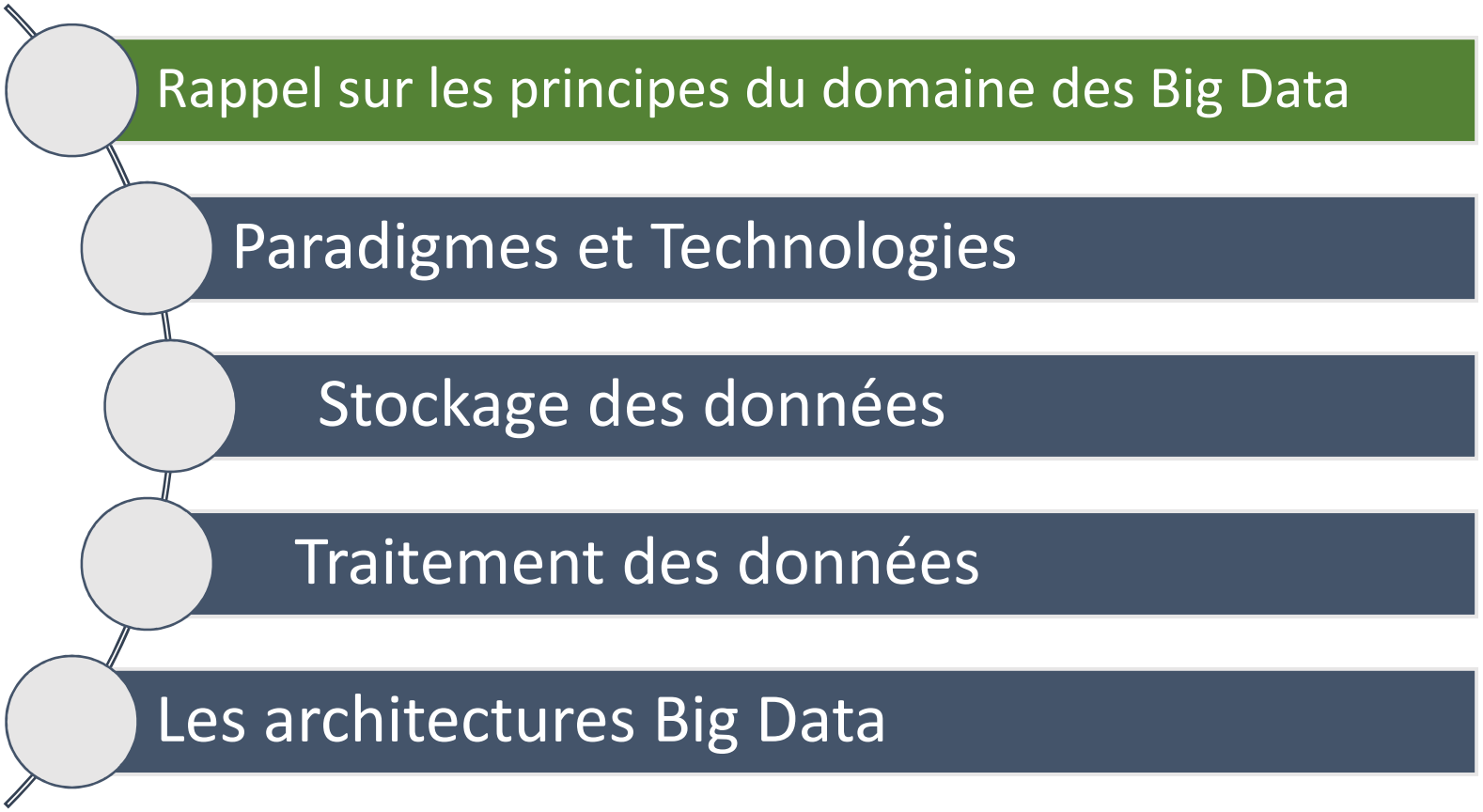
Etudiants concernés

Faculté/Institut	Département	Niveau	Spécialité
Nouvelles technologies	TLSI	Master 1	SDSI

Plan


Rappel sur les principes du domaine des Big Data
Ingestion des données
Stockage des données
Traitement des données
Les architectures Big Data

Plan


Rappel sur les principes du domaine des Big Data
Paradigmes et Technologies
Stockage des données
Traitement des données
Les architectures Big Data

Principes dans Big Data

Principe 1: Stocker d'abords, Réfléchir ensuite

- A cause de la vélocité, il est crucial de reconnaître qu'il peut être difficile, voire impossible, de nettoyer ou de traiter les données avant de les stocker.

- Mettre en place des systèmes de stockage pour les données brutes (Raw Data), qui ne sont pas nettoyées, afin de pouvoir ensuite les soumettre à des traitements.

Principes dans Big Data

Principe 2: Absolument TOUTES les données sont importantes!

Il est largement plus bénéfique de stocker des données qu'on n'utilisera peut-être jamais, plutôt que de gagner de la place et perdre un potentiel pouvoir concurrentiel.

Principes dans Big Data

Principe 3: Ce sont les données qui pilotent le traitement

Les données sont collectées tout d'abord à partir de toutes les sources possibles; des traitements de fouille et d'exploration de ces données sont lancés ensuite, pour extraire de la valeur à partir de ces données.

Définir le traitement à réaliser dépend des données que nous avons réussi à collecter, et pas le contraire. Cela implique donc l'utilisation d'autres types de systèmes de traitement et d'algorithmes d'analyse.

Principes dans Big Data

Principe 4: Co-localisation des données et du traitement

Réaliser les traitements en un temps raisonnable et sans avoir à trimballer les données sur le réseau, il est question dans les systèmes Big Data de déplacer le traitement vers les données massives, au lieu de déplacer les données vers le traitement.

Principes dans Big Data

Principe 5: La redondance

Dans un système Big Data caractérisé par un gros volume de données et une grande vélocité, on tolère à un certain point les risques dus à la redondance pour gagner la disponibilité.

Un système Big Data est un système réparti par excellence, et dans un système réparti, il est primordial d'assurer une bonne tolérance aux fautes en créant des répliques des données, disséminées partout sur le cluster.

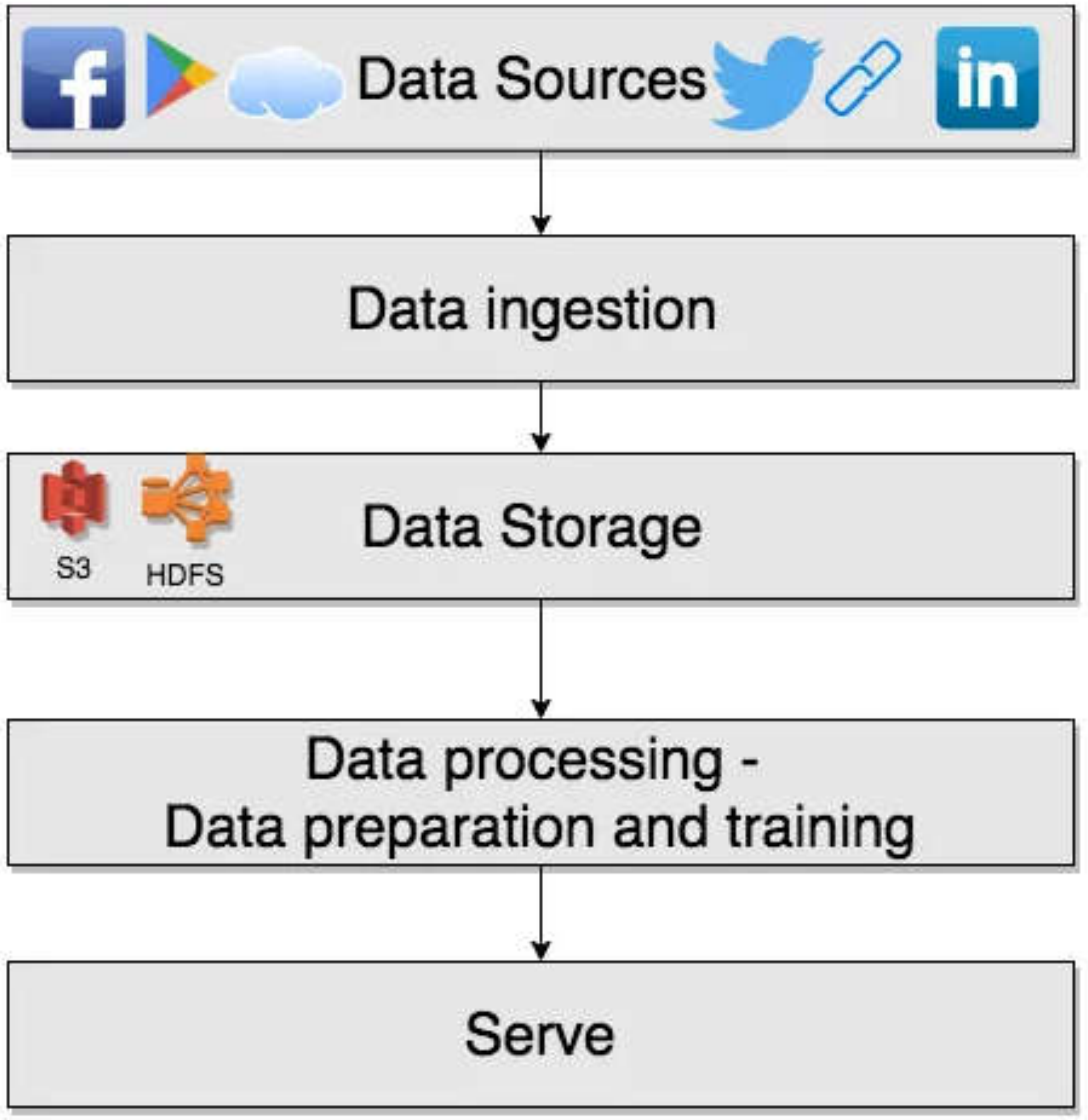
Principes dans Big Data

Principe 5: Le Polyglottisme

Les systèmes Big Data, qui impliquent en général plusieurs traitements et plusieurs types de données différentes (données brutes, données nettoyées, données traitées), encouragent le polyglottisme.

- ***Polyglot Programming***: Plusieurs langages et paradigmes de programmation.
- ***Polyglot Persistence*** : Plusieurs systèmes de stockage différents (relationnels, NOSQL, systèmes de fichiers, etc.).

Paradigmes: Architecture Générale



Paradigmes:

Les opérations à réaliser sur les systèmes Big Data consistent principalement en :

Ingestion des données

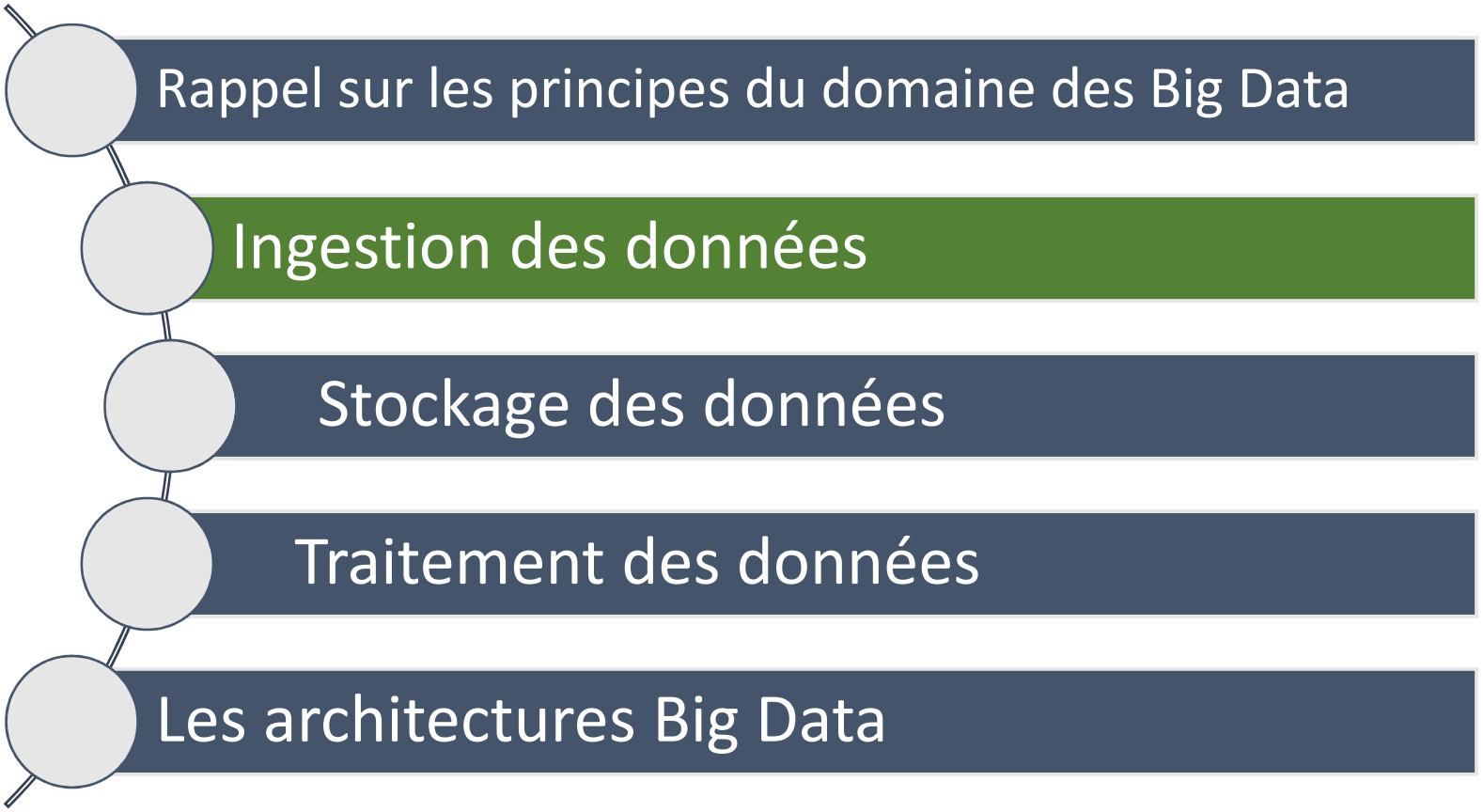
↓

Stockage des données

↓

Traitement des données

Plan


Rappel sur les principes du domaine des Big Data
Ingestion des données
Stockage des données
Traitement des données
Les architectures Big Data

Ingestion des données

Ingestion des données

- Représente les phases de collecte et d'importation des données pour être stockées ou traitées à la volée.
- L'ingestion est crucial car il permet de rassembler des volumes massifs de données, souvent non structurées ou semi-structurées, à partir de sources hétérogènes telles que des bases de données relationnelles, des fichiers de logs, des flux de capteurs, des médias sociaux, des transactions en ligne.

Ingestion des données

Ingestion des données

L'ingestion des données peut se faire:

1. « **En temps réel** »**:** Les données sont importées au moment où elles sont émises par leur source.

Par exemple: Système de surveillance de la santé dans un hôpital. Les capteurs des patients envoient en continu des données sur leur fréquence cardiaque, leur tension artérielle, etc. Ces données sont ingérées en temps réel dans un système informatique qui surveille les signes vitaux des patients. En cas de fluctuations anormales, des alertes peuvent être déclenchées instantanément pour informer le personnel médical.

Ingestion des données

Ingestion des données

APACHE kafka®

APACHE nifi

StreamSets

amazon KINESIS

Ingestion des données

Ingestion des données

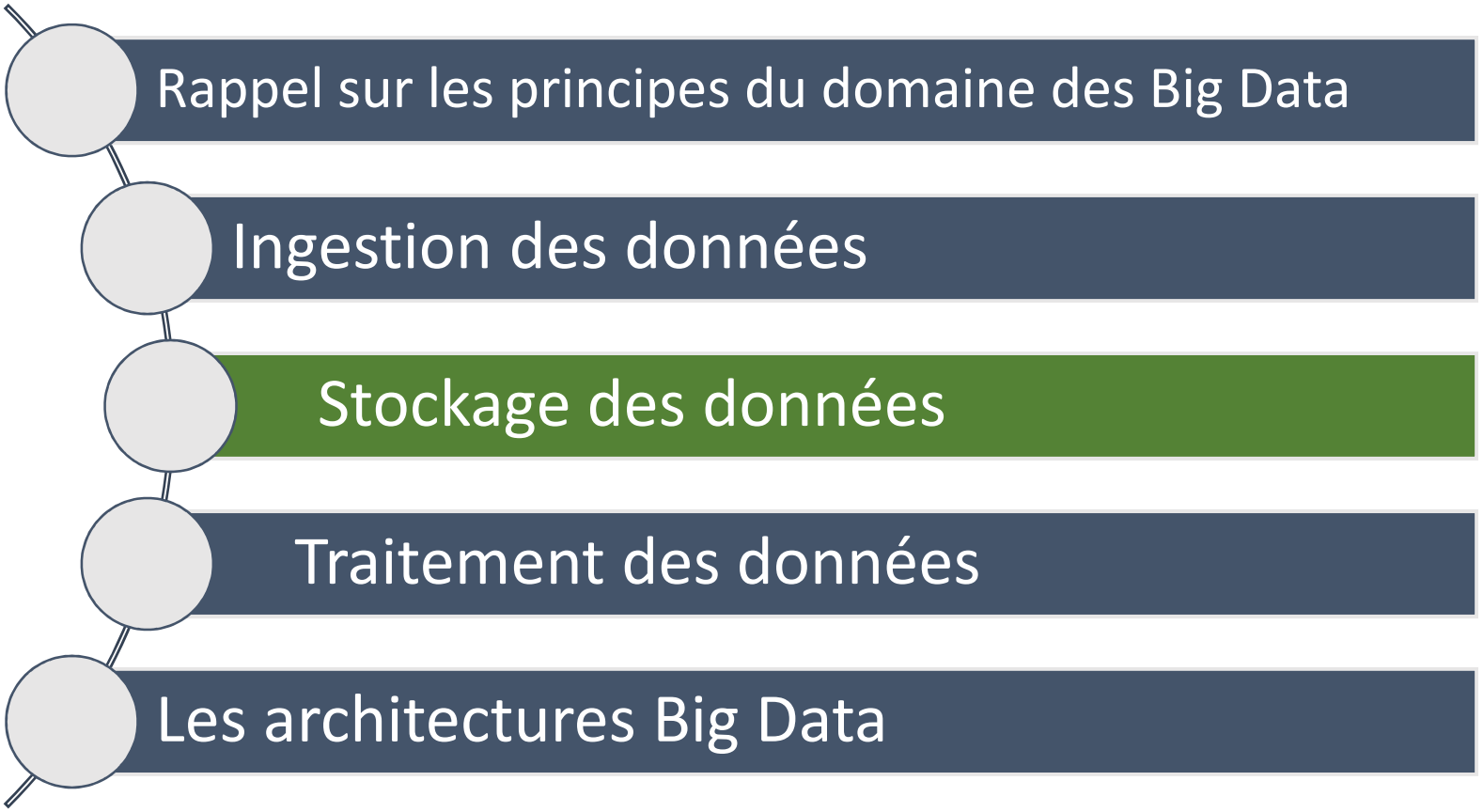
2. « Par lots »: Les données sont importées par portions (ou groupes) à intervalles réguliers.

Par exemple: Une entreprise de vente en ligne peut collecter des données sur les transactions de ses clients tout au long de la journée. Plutôt que de traiter chaque transaction individuellement, elle peut choisir d'ingérer les données par lots à intervalles réguliers, par exemple toutes les heures.

Ingestion des données

Ingestion des données

Plan


Rappel sur les principes du domaine des Big Data
Ingestion des données
Stockage des données
Traitement des données
Les architectures Big Data

Stockage des données

Les systèmes de stockage de données respectant les propriétés de Big Data se distinguent principalement en:

- **Systèmes de fichiers distribués,** tel que Hadoop HDFS ou Google GFS
- **BDs NOSQL**, tel que MongoDB, Cassandra, Redis ou Neo4J.

Stockage des données

Dans Big Data, Les données des BDs NoSQL Doivent être distribuées sur plusieurs nœuds.

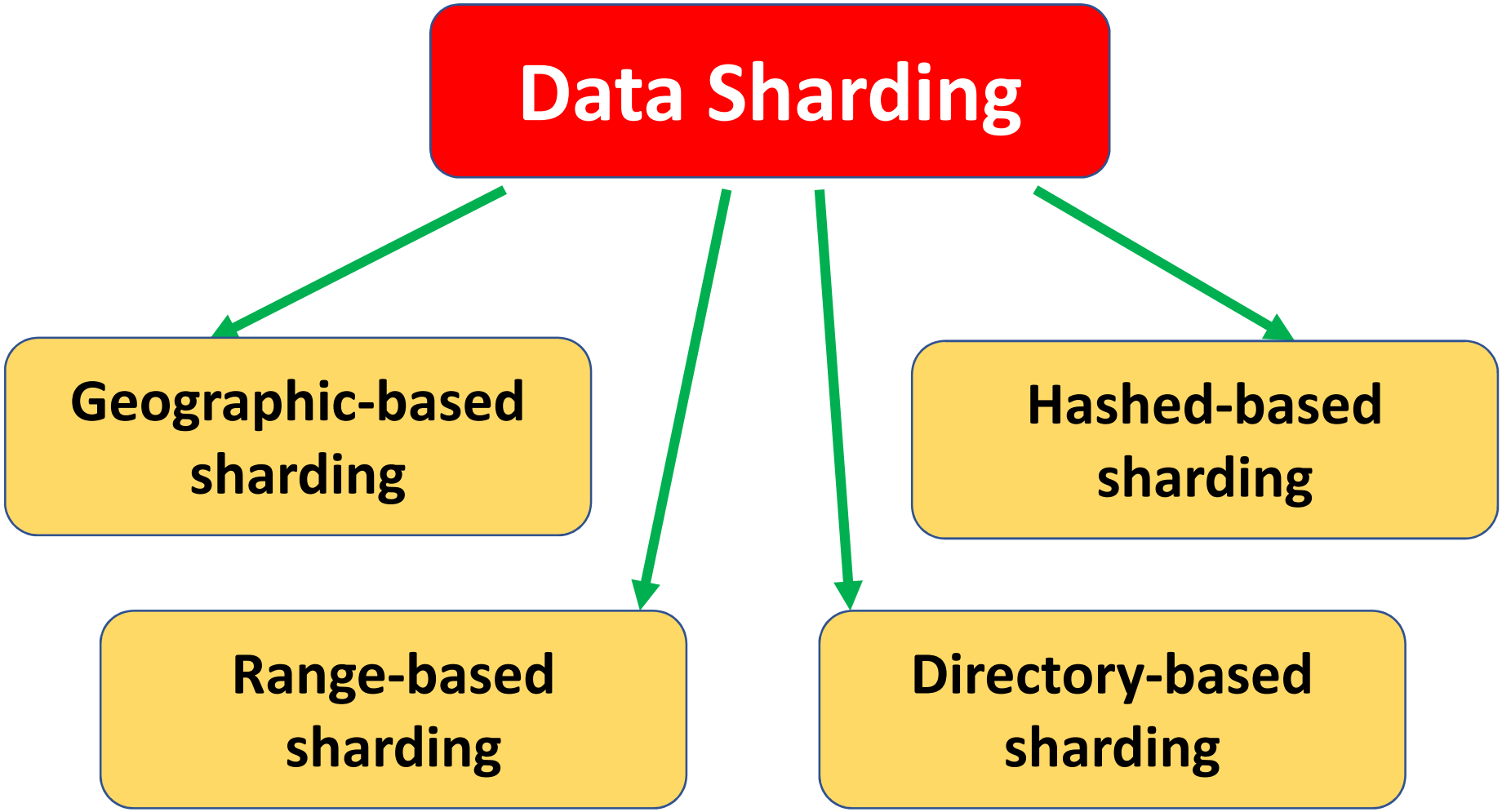
Stockage des données

Dans Big Data, Les données des BDs NoSQL Doivent être distribuées sur plusieurs nœuds.

Sharding (Partitionnement)

Stockage des données

Types de partitionnement



Clé de partitionnement (Shard Key)

- Le partitionnement s'effectue toujours en fonction d'une **clé** appelée « **Shard Key** ».

- Cette clé est utilisée dans les systèmes de bases de données distribuées pour déterminer comment les données sont réparties (shardées) entre différents nœuds dans un cluster.

Types de partitionnement

Clé de partitionnement (Shard Key)

- Chaque nœud est responsable d'un sous-ensemble des données, et la clé de partitionnement est utilisée pour router les requêtes vers le nœud approprié.

- L'objectif principal de la clé de partitionnement est de garantir une distribution équilibrée des données à travers le cluster, afin d'optimiser les performances et la scalabilité du système.

Stockage des données

Types de partitionnement: Range –based sharding

Student	Marks
Adam	89
Ben	95
Catherine	54
David	33
Elizabeth	68
Ali	76

Range-based sharding

Shard 1. Students with marks between 0 and 60

Student	Marks
Catherine	54
[illegible]	33

Shard 3. Students with marks between 85 and 100

Student	Marks
Adam	89
Ben	95

Shard 2. Students with marks between 60 and 85

Student	Marks
Elizabeth	68
Ali	76

Répartition ordonnée

Stockage des données

Types de partitionnement: Hash –based sharding

Column 1
MN
OP
AB
CD

Column 1	Hash Values
MN	2
OP	1
AB	2
CD	1

Hash-based sharding

Shard 1

Column 1	Column 1
OP	Value 1
CD	Value 2

Shard 2

Column 1	Column 1
MN	Value 3
AB	Value 4

Stockage des données

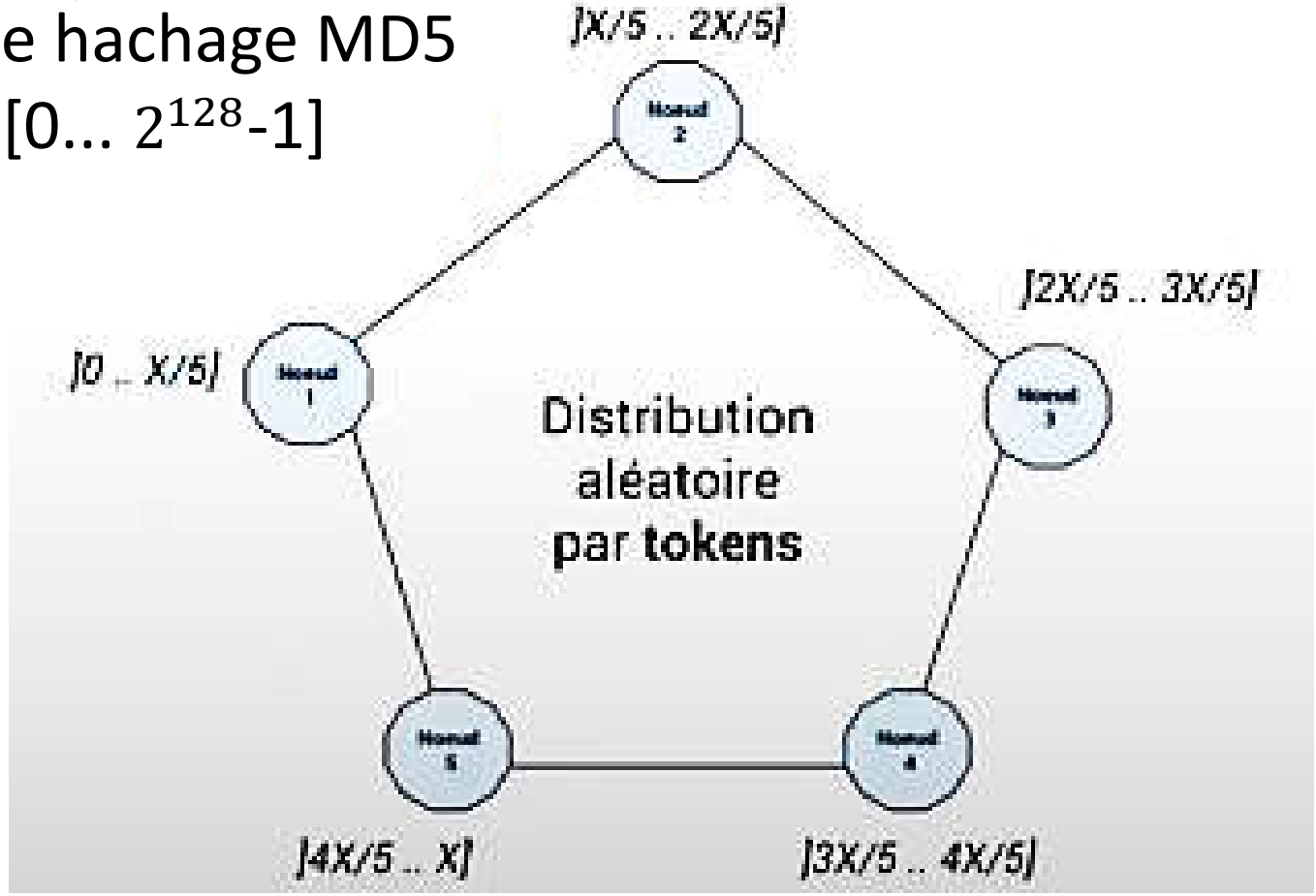
Types de partitionnement: Hash –based sharding

Exemple:

Fonction de hachage MD5

Intervalle: [0... 2^{128}-1]

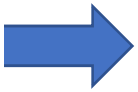
X= 2^{128}-1



Stockage des données

Types de partitionnement: Directory – based sharding

Section	Student First Name	Student Last Name
C	Alex	Hales
B	Ben	Davies
A	Craig	Hall
D	David	Copperfield



Section (Shard Key)	Shard ID
A	S1
B	S2
C	S3
D	S4

Annuaire

Directory -based sharding

Shard S1

A	Craig	Hall

Shard S3

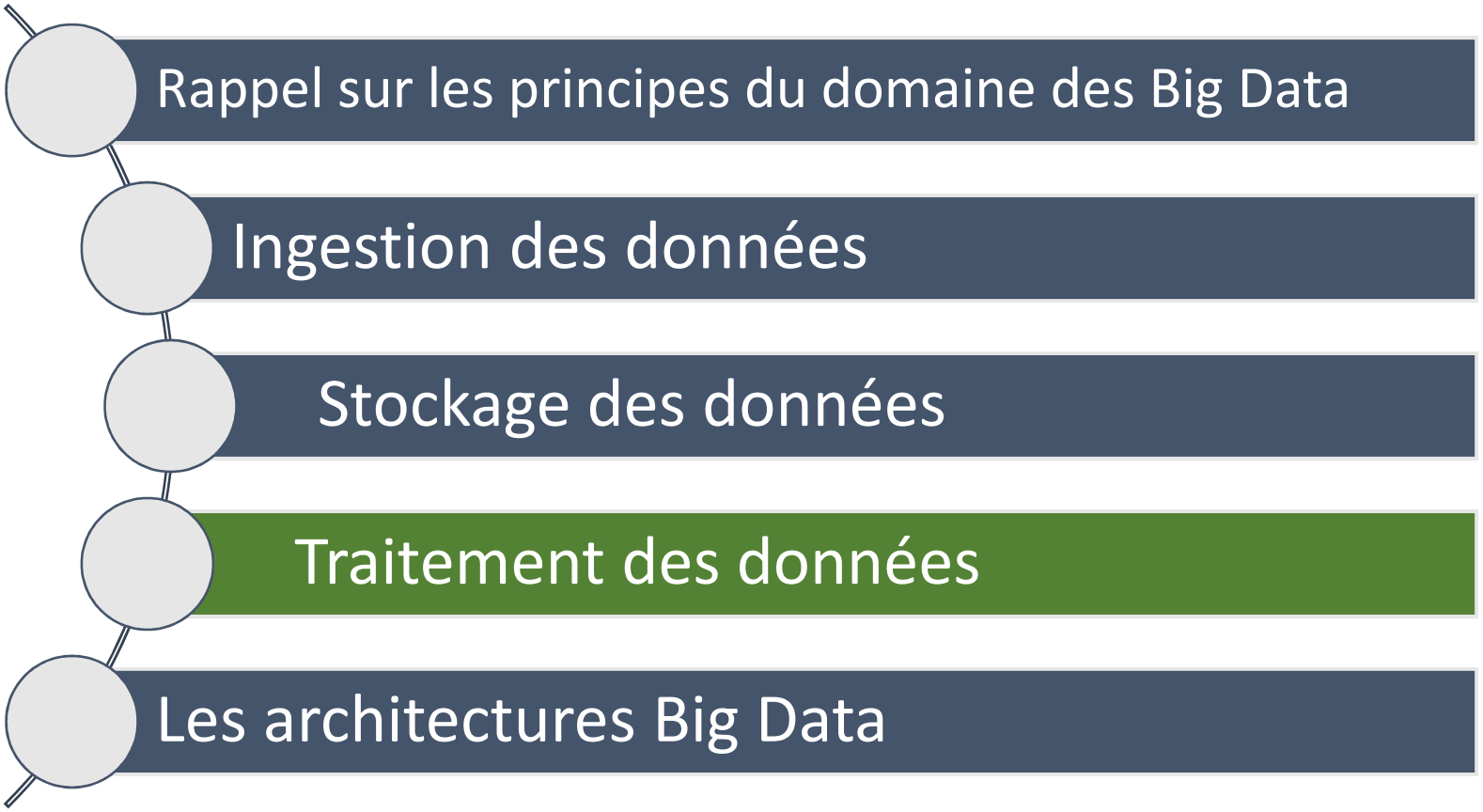
C	Alex	Hales

Shard S2

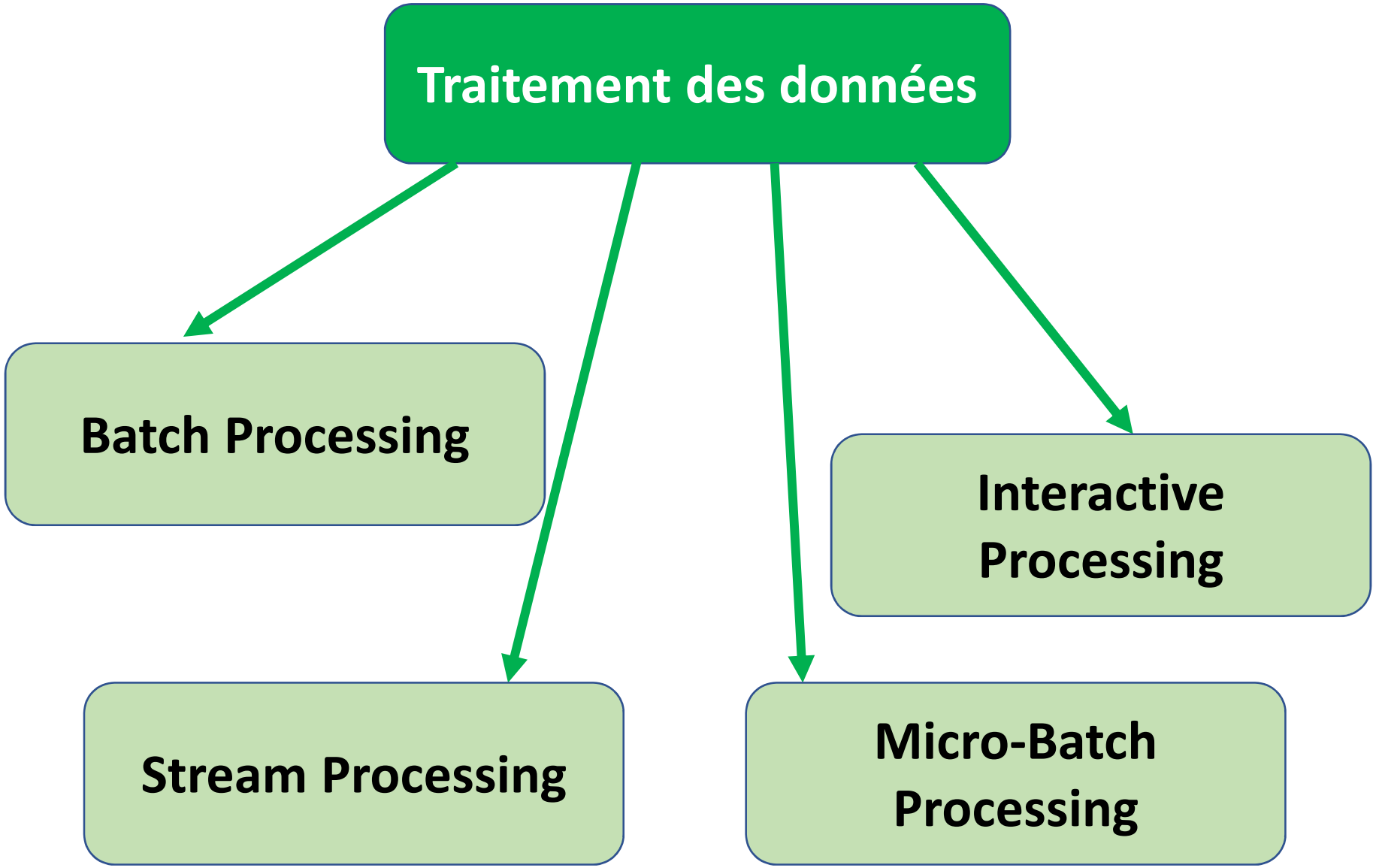
B	Ben	Davies

D	David	Copperfield

Plan


Rappel sur les principes du domaine des Big Data
Ingestion des données
Stockage des données
Traitement des données
Les architectures Big Data

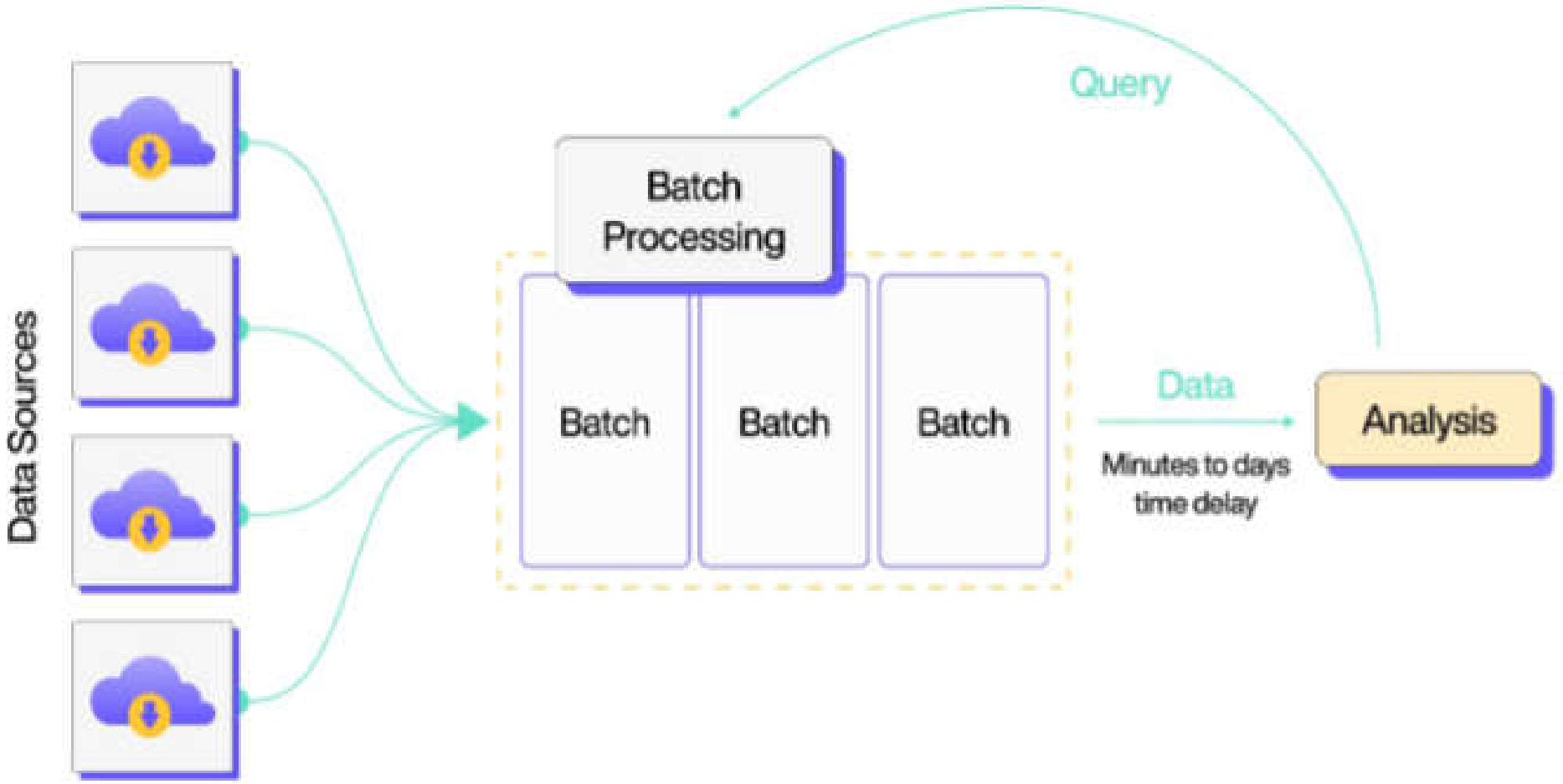
Traitement des données



Traitement des données : Batch Processing

- C'est le traitement des *données au repos (data at rest)* qui se fait sur l'ensemble des données stockées, sans avoir besoin d'une interaction avec l'utilisateur.
- Le traitement par lot est adapté principalement aux opérations permettant d'avoir une vision globale sur la totalité des données, par exemple pour avoir un rapport global ou une analyse mensuelle.
- Les opérations de traitement par lots sont en général lancées à des périodes régulières, car elles sont connues pour avoir une grande latence (temps total de traitement).

Traitement des données : : Batch Processing

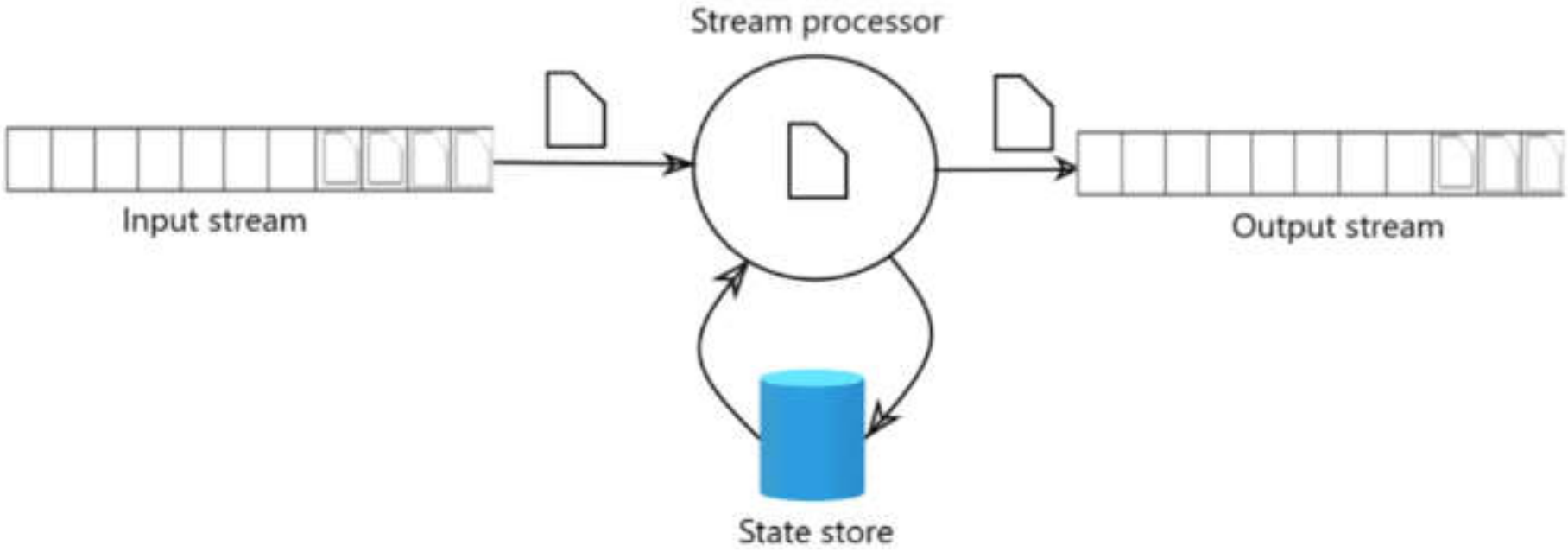


Traitement des données : : Batch Processing

Traitement des données : Stream Processing

- c'est le traitement des données *en transit (data in motion)*, ou en d'autres termes, le traitement des données pendant qu'elles sont produites ou reçues.
- Les données étant en général créées en tant que flux continu (évènements de capteurs, activité des utilisateurs sur un site web, flux vidéo, etc.)
- Avant la création des traitements en streaming, ces données étaient stockées dans une base de données, un système de fichier ou tout autre forme de stockage en masse.
- Grâce à ce nouveau paradigme, les données peuvent maintenant être traitées à la volée, ce qui permet à la couche applicative d'être toujours sur écoute et à jour.
- La latence est faible

Traitement des données : Stream Processing



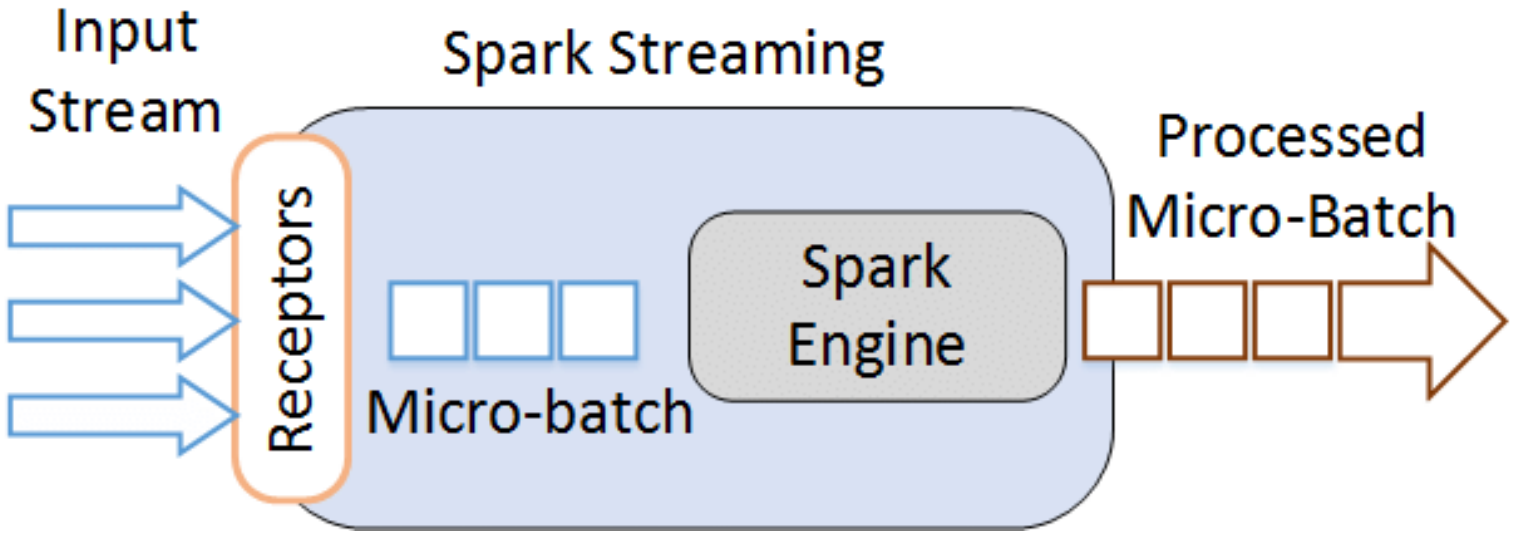
Traitement des données : Stream Processing

Apache Flink

APACHE STORM™

Traitement des données : Micro-Batch Processing

- Hybridation entre le batch processing et le stream processing
- Les données sont en micro lots.
- Traite un flux avec le principe de traitement en batch
- Exemple:



Traitement des données : Micro-Batch Processing

Traitement des données : Interactive Processing

- En Big Data, on parle donc rarement de traitement transactionnel, mais de traitements plutôt interactifs : une requête est envoyée par le client, traitée immédiatement par le système qui renverra un résultat dans un temps raisonnable.
- On parle alors d'interaction entre l'utilisateur et le système.
- Les traitements en batch et en streaming ne sont pas censés communiquer avec un utilisateur de l'autre côté.
- Le traitement interactif est donc le résultat d'une requête de l'utilisateur, faite en général sur une base de données (relationnelle ou NOSQL).
- Batch à la demande de l'utilisateur
- Les traitement synchrone : les données dont traitées immédiatement à la demande de l'utilisateur. L e système renverra un résultat dans un temps raisonnable.

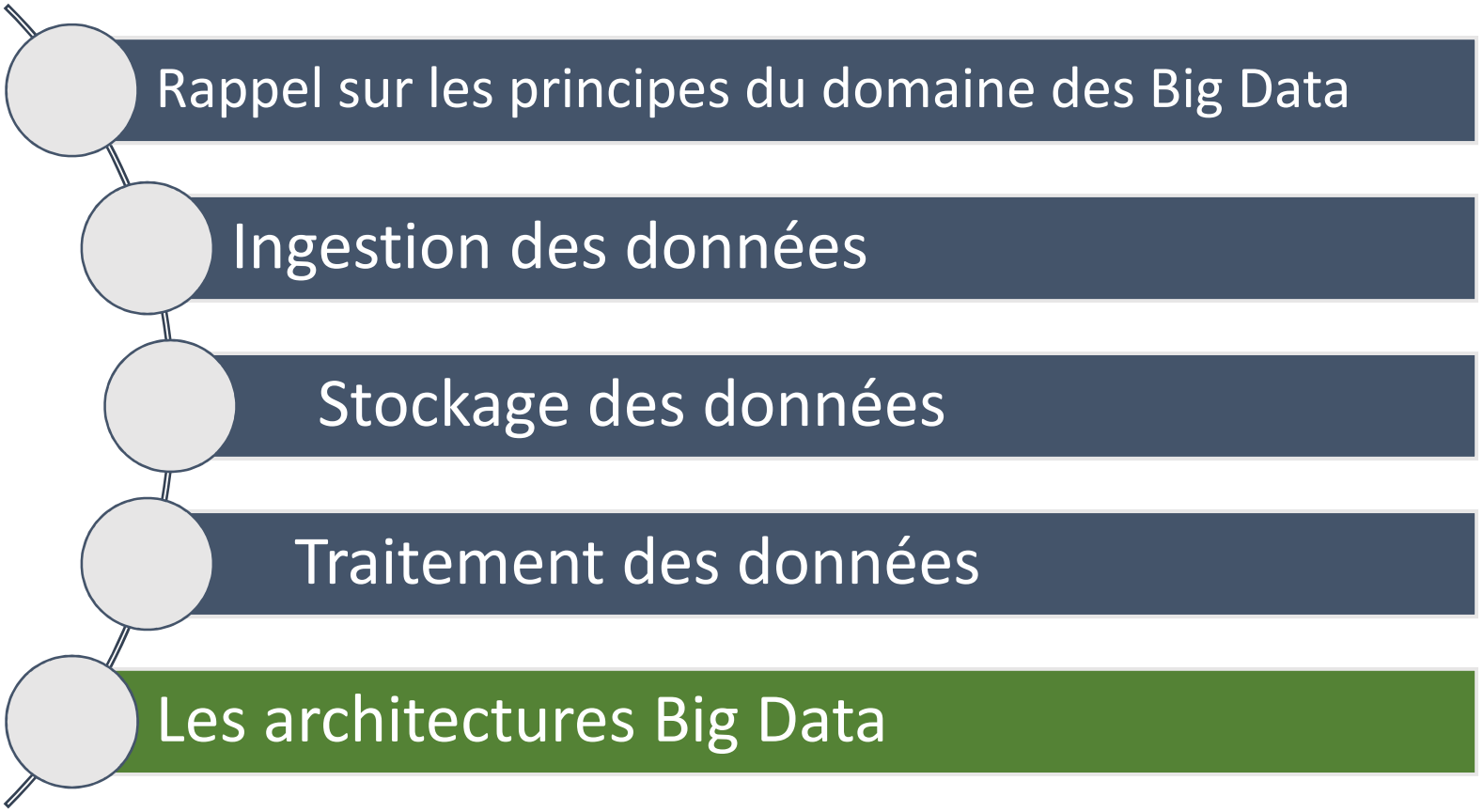
Traitement des données : Interactive Processing

Real time Processing

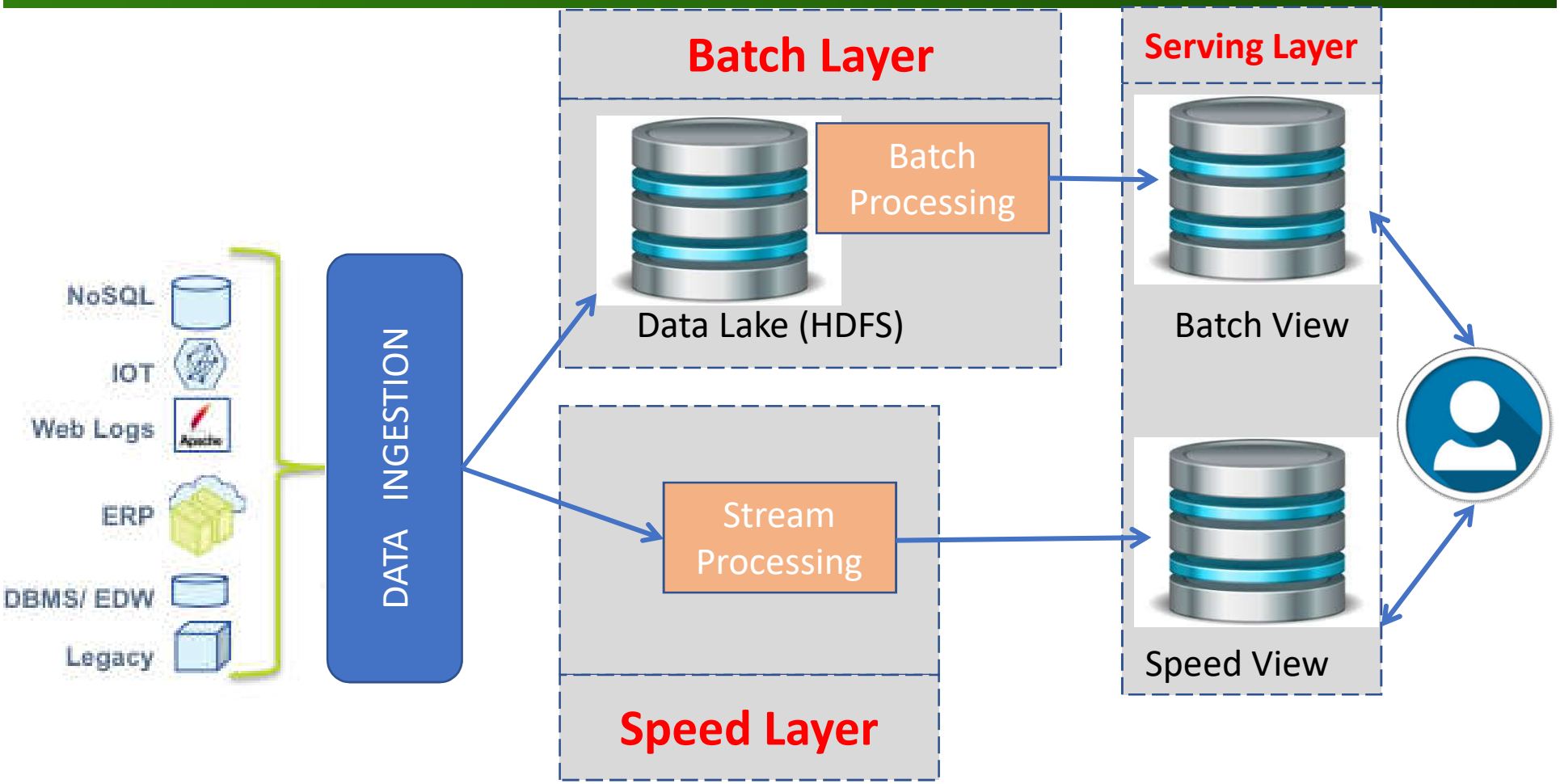
Real time Processing

- Un système en streaming est un système qui réalise une opération au fur et à mesure de l'arrivée des données.
- Par défaut, un système en streaming n'est pas un système en temps réel. Il peut être un système en temps réel si l'opération devra être réalisée dans un temps t (à ne pas dépasser).

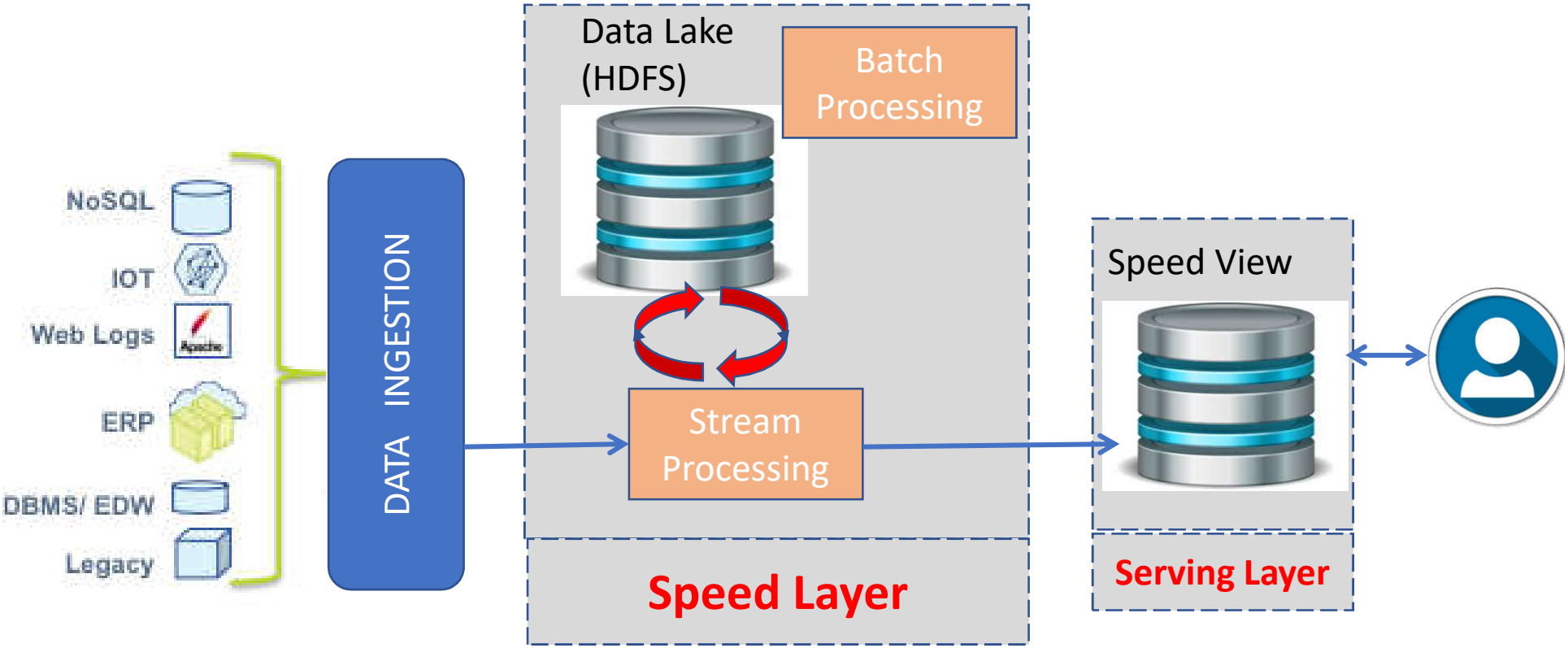
Plan


Rappel sur les principes du domaine des Big Data
Ingestion des données
Stockage des données
Traitement des données
Les architectures Big Data

Architecture Big Data : Lambda



Architecture Big Data : Kappa



Références

- Rudi Bruchez. (2015), « Les bases de données NoSQL et le Big Data », Editeur : Eyrolles, ISBN : 978-2-212-14155-9
- Rudi Bruchez (2021), « Les bases de données NoSQL », Editeur : Eyrolles ISBN : 978-2-212-67866-6
- Juvénal Chokogoue. (2017). « Hadoop - Devenez opérationnel dans le monde du Big Data », Editeur: ENI, ISBN: 978-2409007613