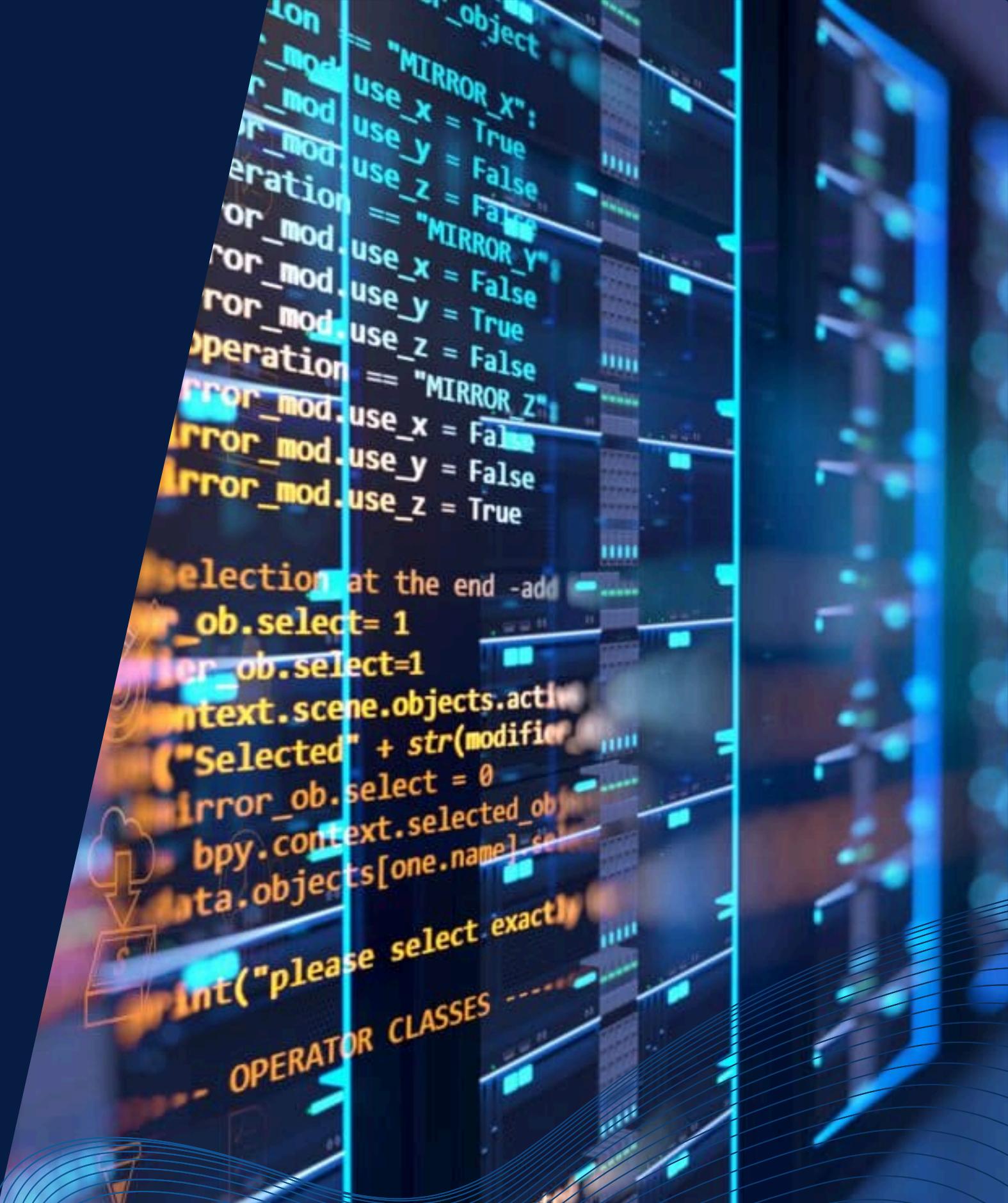


# ANALYSE ET PREDICTION DES DONNEES METEOROLOGIQUES

RÉALISÉ PAR:

Akhatar Mourad  
Bennis Sanae  
Cheikh Yassir  
Jari Mohammed Amine  
Maghraoui Sana



# TABLE OF CONTENT

01

Introduction

02

5V

03

Collecte des  
données

04

Traitemennt des  
données

05

Stockage avec  
hadoop

06

Visualisation

# Introduction

Notre projet vise à collecter et analyser en temps réel des données météorologiques provenant de multiples sources à l'aide des techniques avancées pour collecter, stocker et analyser des données massives.

L'objectif est de fournir des prévisions météorologiques précises, exploitant des algorithmes avancés pour anticiper les conditions climatiques futures.

Cette approche permettra des décisions éclairées dans des domaines tels que l'agriculture et la gestion des risques environnementaux.



# Les 5V

## Volume

Les données provenant de multiples stations météorologiques pour différentes villes (200) génèrent un volume massif de données.

L'agrégation de ces données accumule rapidement une quantité substantielle d'informations météorologiques.

## Variété

Chaque station météorologique offre une variété de données : température, humidité, pression atmosphérique, précipitations, vents, etc.

Ces données sont collectées dans des formats différents, exigeant une intégration harmonieuse pour former un modèle uniifié.

## Vérité

La validation constante des données de chaque station garantit la précision des prévisions météorologiques.

Une surveillance continue assure l'élimination des erreurs et garantit la fiabilité des sources.

## Vélocité

Le streaming en continu des données de multiples stations météorologiques pour différentes villes demande un traitement rapide.

L'analyse quasi instantanée des informations est cruciale pour des prévisions météorologiques en temps réel.

## Valeur

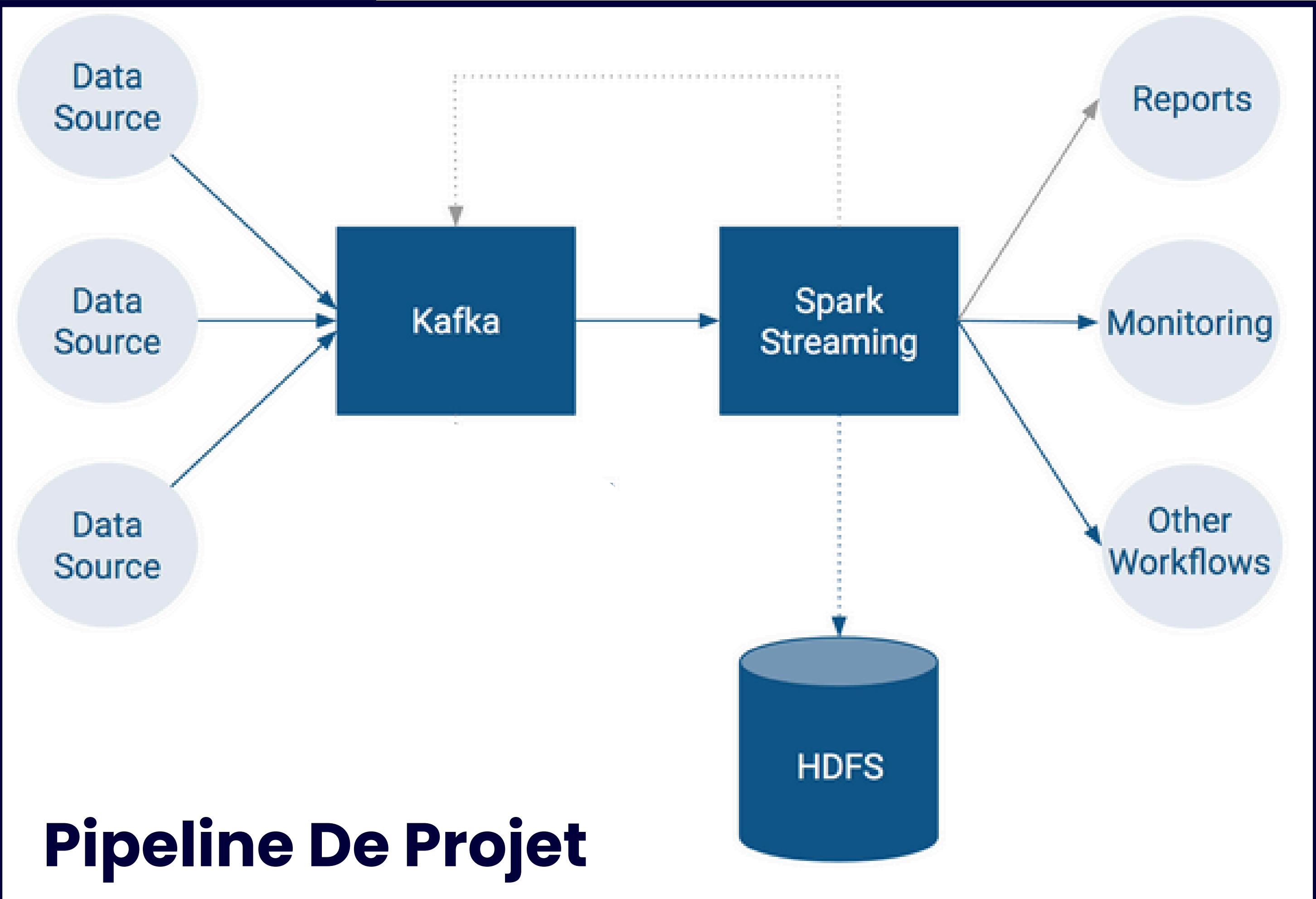
L'analyse des données massives provenant de diverses stations permet d'extraire des modèles et des tendances météorologiques.

Cette transformation des données offre une valeur ajoutée significative aux utilisateurs finaux pour des décisions éclairées.

# COLLECTE DES DONÉES

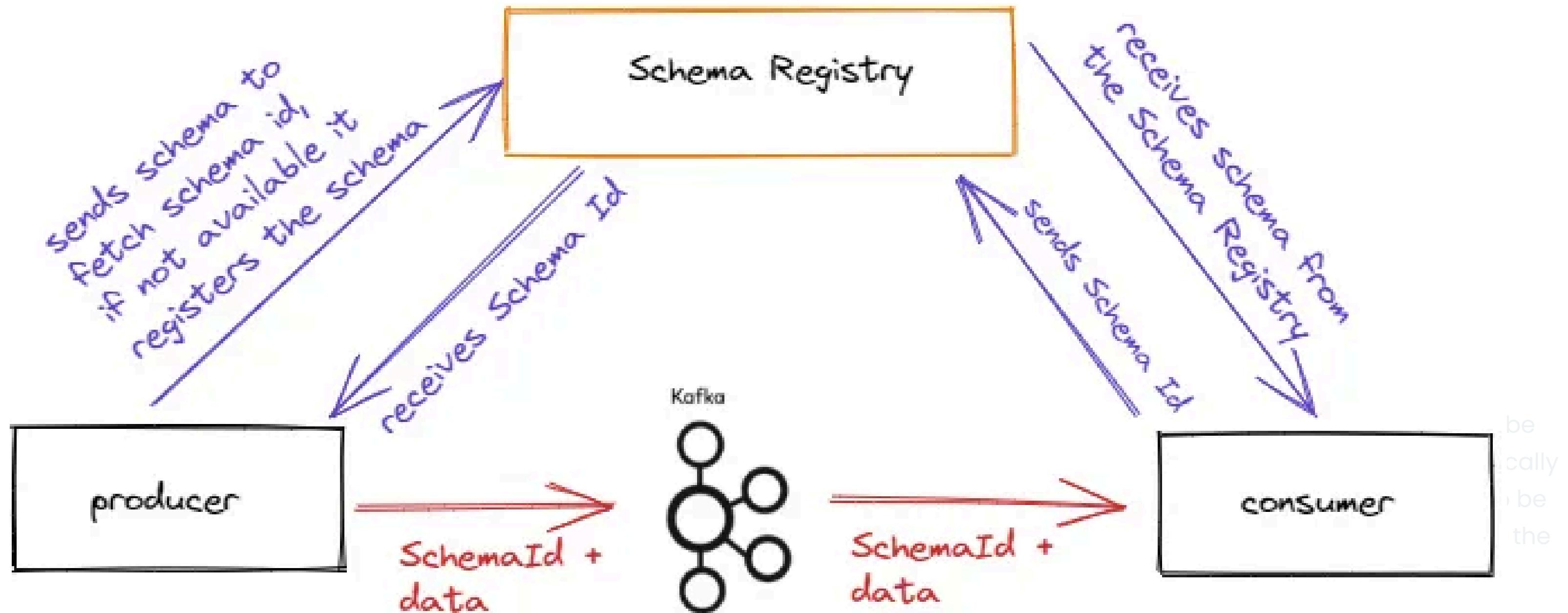
## Open-Weather-Api

```
{  
  "timezone":"America\\New_York",  
  "state_code":"NC",  
  "lat":35.7721,  
  "lon":-78.63861,  
  "country_code":"US",  
  "station_id":"723060-13722",  
  "sources":["723060-13722", "USC00445050", "USW00013732"],  
  "data": [  
    {  
      "rh":32,  
      "wind_spd":6.7,  
      "wind_gust_spd": 9.4,  
      "slp":1020.3,  
      "h_angle":15,  
      "azimuth":25,  
      "dewpt":-7.5,  
      "snow":0,  
      "uv":0,  
      "wind_dir":220,  
      "weather":{  
        "icon":"c01n",  
        "code":"800",  
        "description":"Clear sky"  
      },  
      "pod":"n",  
      "vis":1.5,  
      "precip":0,  
      "elev_angle":-33,  
      "ts":1483232400,  
      "pres":1004.7,  
      "datetime":"2018-05-01:06",  
      "timestamp_utc":"2015-05-01T06:00:00",  
      "timestamp_local":"2015-05-01T02:00:00",  
      "revision_status":"final",  
      "temp":8.3,  
      "dhi":15,  
      "dni":240.23,  
      "ghi":450.9,  
      "solar_rad":445.85,  
      "clouds":0  
    }, ...  
  ],  
  "city_name":"Raleigh",  
  "city_id":"4487042"  
}
```



# Utilisation de KAFKA

- Mise en place d'un pipeline Kafka pour recevoir et traiter les données en temps réel.
- Développement de producteurs pour l'alimentation du pipeline Kafka avec les données météorologiques.
- Configuration des consommateurs PySpark Streaming pour traiter les flux de données depuis Kafka.



# TRAITEMENT DES DONNÉES

## Nettoyage des Données

Eliminer les valeurs aberrantes, les données manquantes ou corrompues.

## Filtrage et Normalisation

Les données brutes peuvent contenir du bruit ou des redondances. Le filtrage est effectué pour éliminer le bruit, tandis que la normalisation est utilisée pour ramener les différentes échelles de mesure (température, pression, humidité, etc.) à une échelle uniforme, ce qui facilite la comparaison et l'analyse.

## Agrégation Temporelle

Pour réduire la dimensionnalité des données et rendre l'analyse plus efficace, une agrégation temporelle peut être appliquée. Cela peut inclure des moyennes par heure, par jour ou par saison, selon les besoins spécifiques du projet.

# PySpark

- **Implémentation des opérations de nettoyage et de validation des données en temps réel avec PySpark Streaming.**
- **Transformation des données brutes pour les adapter aux besoins de notre modèle de prédiction météorologique.**
- **Utilisation de PySpark pour développer et entraîner un modèle de prédiction météorologique.**



# Stockage avec hadoop



## Distribution des données

HDFS est conçu pour gérer la distribution des données sur un cluster de nœuds. Il divise les fichiers en blocs de taille fixe (par défaut, 128 Mo ou 256 Mo) et distribue ces blocs sur différents nœuds du cluster. Cette approche permet d'exploiter la puissance de calcul parallèle en traitant simultanément plusieurs blocs sur différents nœuds, améliorant ainsi les performances globales du système.

## Le choix du format de stockage des données

sur Hadoop Distributed File System (HDFS) est crucial pour optimiser les performances de lecture et d'écriture, ainsi que pour économiser de l'espace de stockage.

### Format de fichier texte séquentiel (JSON)

**Avantages :** Simplicité, lisibilité, et compatibilité avec de nombreux outils.

**Inconvénients :** Moins efficace en termes d'espace de stockage et de performances lors de traitements complexes nécessitant la lecture de colonnes spécifiques.



# Visualisation des Données avec Grafana

- Intégration de Grafana pour visualiser les données stockées dans Hadoop.
- Création de tableaux de bord interactifs pour présenter les prévisions météorologiques et les tendances temporelles.



Grafana

```
[2023-12-21 10:56:00,778] INFO Server environment:os.memory.max=512MB (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,778] INFO Server environment:os.memory.total=512MB (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,778] INFO zookeeper.enableEagerACLCheck = false (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,778] INFO zookeeper.digest.enabled = true (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,778] INFO zookeeper.closeSessionTxn.enabled = true (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,778] INFO zookeeper.flushDelay = 0 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,778] INFO zookeeper.maxWriteQueuePollTime = 0 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,778] INFO zookeeper.maxBatchSize=1000 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,778] INFO zookeeper.intBufferStartingSizeBytes = 1024 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,771] INFO Weighted connection throttling is disabled (org.apache.zookeeper.server.BlueThrottle)
[2023-12-21 10:56:00,772] INFO minSessionTimeout set to 6000 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,772] INFO maxSessionTimeout set to 60000 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,773] INFO getData response cache size is initialized with value 400. (org.apache.zookeeper.server.ResponseCache)
[2023-12-21 10:56:00,773] INFO getChildren response cache size is initialized with value 400. (org.apache.zookeeper.server.ResponseCache)
[2023-12-21 10:56:00,774] INFO zookeeper.pathStats.slotCapacity = 60 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2023-12-21 10:56:00,774] INFO zookeeper.pathStats.slotDuration = 15 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2023-12-21 10:56:00,774] INFO zookeeper.pathStats.maxDepth = 6 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2023-12-21 10:56:00,774] INFO zookeeper.pathStats.initialDelay = 5 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2023-12-21 10:56:00,774] INFO zookeeper.pathStats.delay = 5 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2023-12-21 10:56:00,774] INFO zookeeper.pathStats.enabled = false (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2023-12-21 10:56:00,776] INFO The max bytes for all large requests are set to 104857600 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,776] INFO The large request threshold is set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,776] INFO zookeeper.enforce.auth.enabled = false (org.apache.zookeeper.server.AuthenticationHelper)
[2023-12-21 10:56:00,776] INFO zookeeper.enforce.auth.schemes = [] (org.apache.zookeeper.server.AuthenticationHelper)
[2023-12-21 10:56:00,776] INFO Created server with tickTime 3000 ms minSessionTimeout 6000 ms maxSessionTimeout 60000 ms clientPortListenBacklog -1 datadir /home/jari/kafka_2.13-3.6.0/data/zookeeper/version-2 snapdir /home/jari/kafka_2.13-3.6.0/data/zookeeper/version-2 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,781] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.ServerCnxnFactory)
[2023-12-21 10:56:00,782] WARN maxCnxns is not configured, using default value 0. (org.apache.zookeeper.server.ServerCnxnFactory)
[2023-12-21 10:56:00,783] INFO Configuring NIO connection handler with 10s sessionless connection timeout, 2 selector thread(s), 16 worker threads, and 64 kB direct buffers. (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2023-12-21 10:56:00,788] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2023-12-21 10:56:00,810] INFO Using org.apache.zookeeper.server.watch.WatchManager as watch manager (org.apache.zookeeper.server.watch.WatchManagerFactory)
[2023-12-21 10:56:00,810] INFO Using org.apache.zookeeper.server.watch.WatchManager as watch manager (org.apache.zookeeper.server.watch.WatchManagerFactory)
[2023-12-21 10:56:00,811] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.zookeeper.server.ZKDatabase)
[2023-12-21 10:56:00,811] INFO zookeeper.commitLogCount=500 (org.apache.zookeeper.server.ZKDatabase)
[2023-12-21 10:56:00,814] INFO zookeeper.snapshot.compression.method = CHECKED (org.apache.zookeeper.server.persistence.SnapStream)
[2023-12-21 10:56:00,815] INFO Reading snapshot /home/jari/kafka_2.13-3.6.0/data/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileSnap)
[2023-12-21 10:56:00,816] INFO The digest value is empty in snapshot (org.apache.zookeeper.server.DataTree)
[2023-12-21 10:56:00,840] INFO ZooKeeper audit is disabled. (org.apache.zookeeper.audit.ZKAuditProvider)
[2023-12-21 10:56:00,856] INFO 183 txns loaded in 36 ms (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2023-12-21 10:56:00,856] INFO Snapshot loaded in 45 ms, highest zfid is 0xb7, digest is 288879311623 (org.apache.zookeeper.server.ZKDatabase)
[2023-12-21 10:56:00,858] INFO Snapshotting: 0xb7 to /home/jari/kafka_2.13-3.6.0/data/zookeeper/version-2/snapshot.b7 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2023-12-21 10:56:00,862] INFO Snapshot taken in 3 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:00,872] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2023-12-21 10:56:00,873] INFO zookeeper.request_throttler.shutdownTimeout = 10000 ms (org.apache.zookeeper.server.RequestThrottler)
[2023-12-21 10:56:00,890] INFO Using checkIntervalMs=60000 maxPerMinute=10000 maxNeverUsedIntervalMs=0 (org.apache.zookeeper.server.ContainerManager)
[2023-12-21 10:56:21,123] INFO Expiring session 0x1000011382100002, timeout of 10000ms exceeded (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 10:56:21,129] INFO Creating new log file: log.b8 (org.apache.zookeeper.server.persistence.FileTxnLog)
[2023-12-21 11:44:26,289] INFO Unable to read additional data from client, it probably closed the socket: address = /127.0.0.1:66470, session = 0x1000009653d0000 (org.apache.zookeeper.server.NIOServerCnxn)
[2023-12-21 11:44:57,194] INFO Expiring session 0x1000009653d0000, timeout of 10000ms exceeded (org.apache.zookeeper.server.ZooKeeperServer)
[2023-12-21 11:44:58,818] INFO Invalid session 0x1000009653d0000 for client /127.0.0.1:43412, probably expired (org.apache.zookeeper.server.ZooKeeperServer)
```

# CONCLUSION

Le projet d'analyse de données météorologiques en temps réel, basé sur l'intégration de PySpark Streaming, Kafka, Hadoop (HDFS), et Grafana, représente une avancée significative dans la capacité à traiter, stocker et visualiser les informations météorologiques critiques.



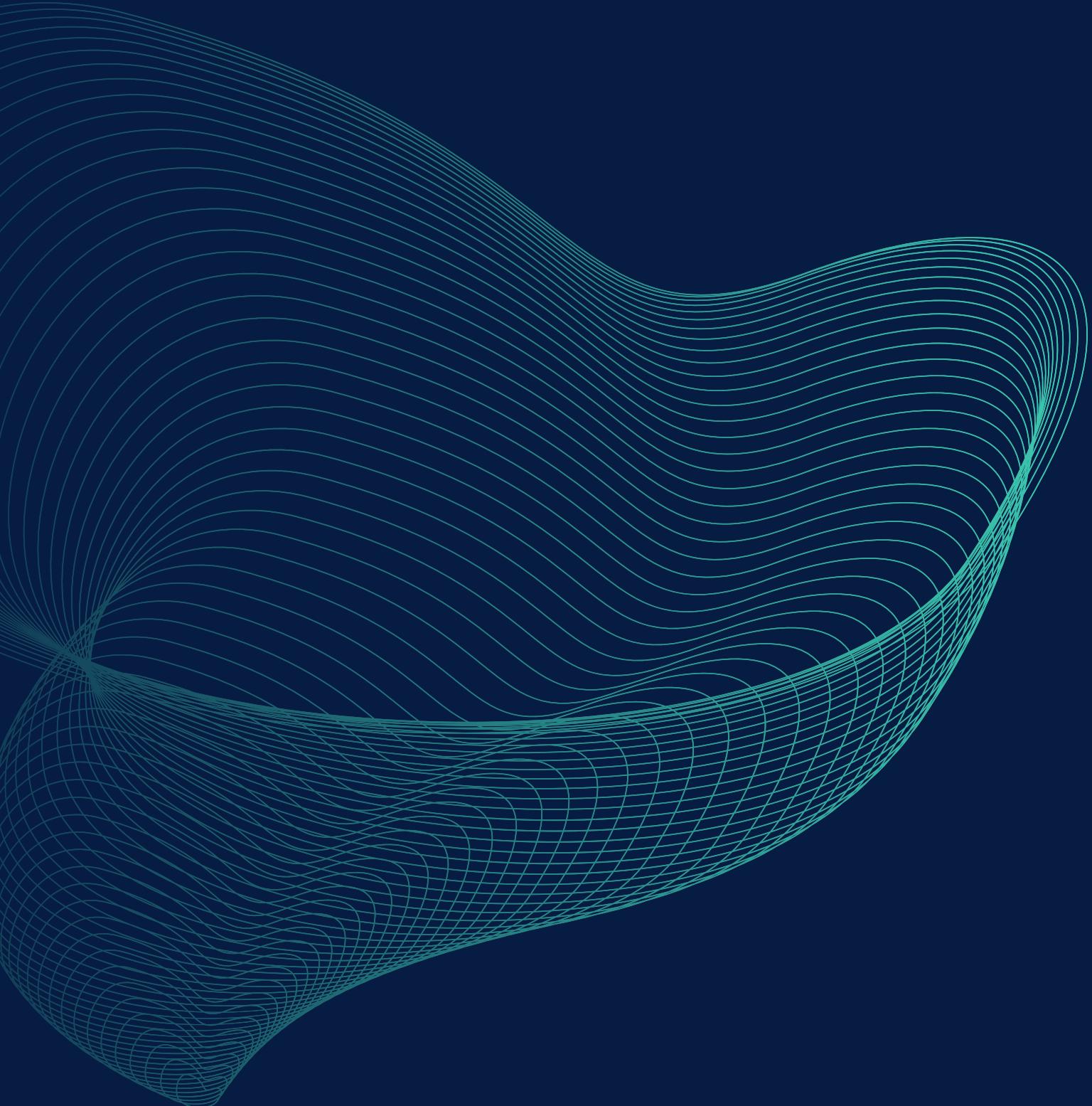
**Évolutivité et Gestion des Données Massives**

**Rapidité d'Analyse**

**Efficacité dans le Traitement des Flux de Données**

**Précision Améliorée des Prévisions**

---



**MERCI POUR  
VOTRE  
ATTENTION**