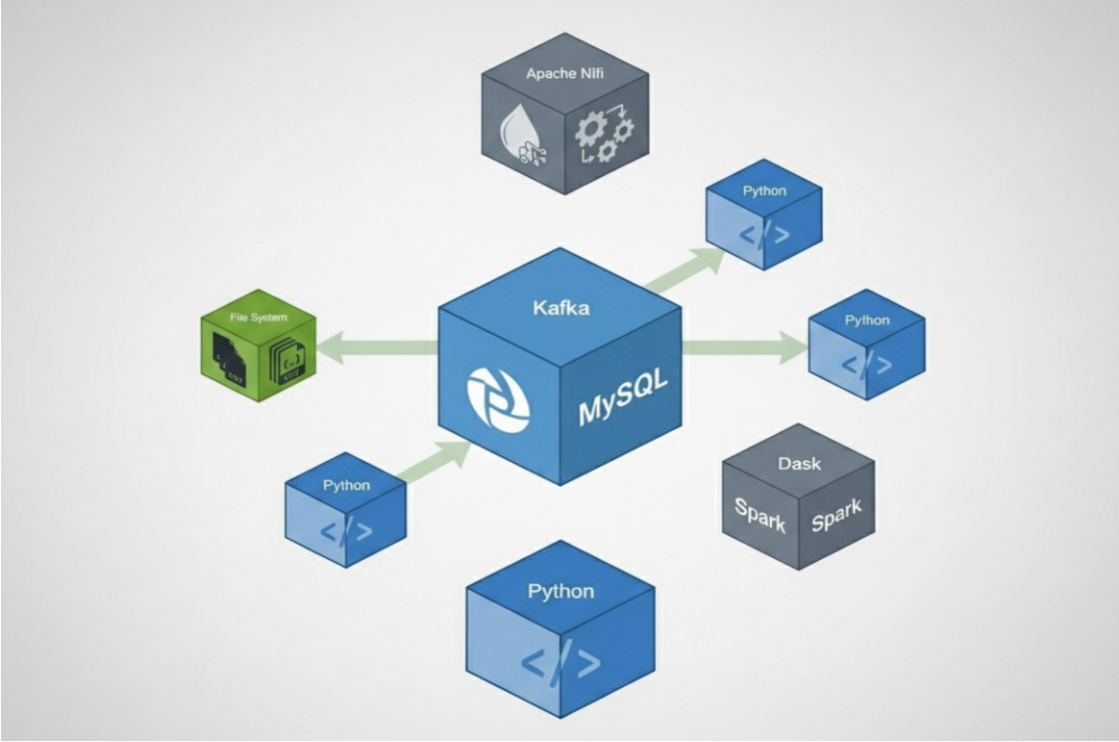
Data Engineering - Projet 2025

Pipeline Données Temps Réel



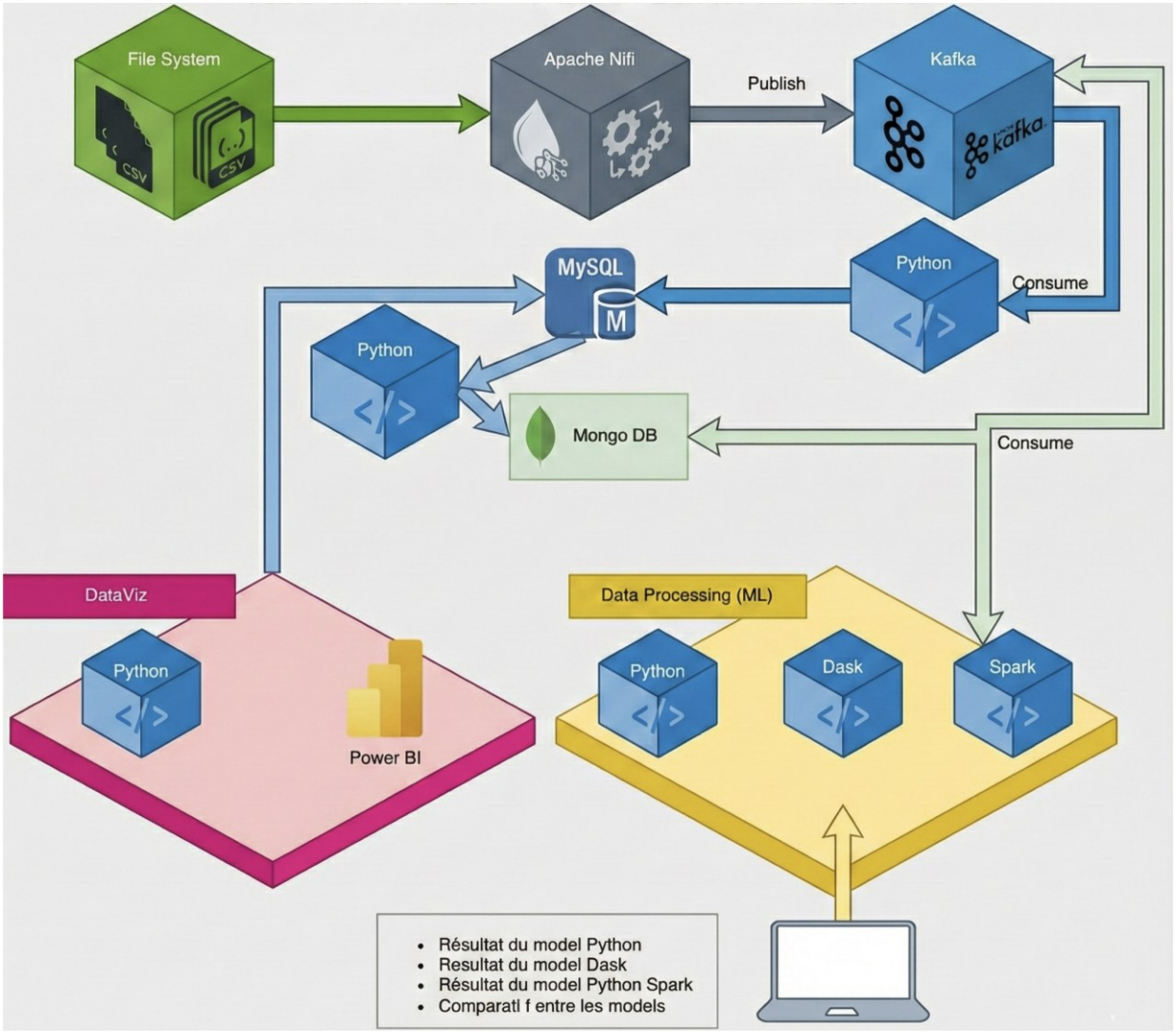
CONCEPTION D'UNE CHAINE
DE TRAITEMENT ET D'ANALYSE EN TEMPS REEL

OBJECTIF GENERAL

L'ambition de ce projet est de maîtriser l'ensemble du cycle de vie des données en temps réel : de la collecte initiale jusqu'à l'analyse prédictive. Vous construirez une architecture complète mobilisant Apache NiFi, Kafka, MySQL, MongoDB, Power BI et Python. Le but ultime est de produire des tableaux de bord interactifs et de mettre en place un système de prédiction qui s'améliore continuellement grâce aux nouvelles données.

Ce projet fera l'objet d'une présentation devant un jury.

ARCHITECTURE DU PIPELINE



PARTIES OBLIGATOIRES

1. DATA INGESTION

- **ÉTAPE 1 : CHOIX DU DATASET**
  Identifier et récupérer un jeu de données adapté à votre problématique. Cette décision orientera l'ensemble de votre démarche analytique. Veillez à sélectionner des données de qualité suffisante pour permettre une exploitation pertinente.
  Une validation préalable par l'encadrant est requise avant de poursuivre.

- **ÉTAPE 2 : GENERATION DU FLUX AVEC NIFI**
  Transformer le fichier statique en flux dynamique grâce à Apache NiFi. En utilisant les capacités No-Code de l'outil, adapter les timestamps et enrichir les enregistrements pour reproduire un comportement temps réel.

2. DATA STREAMING

- **ÉTAPE 3 : MISE EN PLACE DE KAFKA**
  Déployer Apache Kafka comme broker de messages pour acheminer le flux généré par NiFi. Ce composant servira de point central pour distribuer les données aux différents consommateurs.

- **ÉTAPE 4 : PERSISTANCE RELATIONNELLE (SCRIPT PYTHON 1)**
  Développer un consommateur Python qui récupère les messages Kafka et les insère dans MySQL. Concevoir au préalable un schéma relationnel adapté avec plusieurs tables normalisées.
  Une fois cette étape franchie, les données historiques s'accumulent dans la base pendant que le flux continue d'alimenter le système.

3. DATA VISUALIZATION

- **ÉTAPE 5 : TABLEAU DE BORD POWER BI**
  Construire une interface analytique avec Power BI connectée à MySQL. Ce dashboard offrira une vue d'ensemble des indicateurs clés et permettra une exploration interactive des données.

- **ÉTAPE 6 : VISUALISATION TEMPS REEL EN PYTHON**
  Développer une application Python (Streamlit, Dash ou équivalent) qui affiche des graphiques actualisés automatiquement. Cette interface combinera les données MySQL avec les messages Kafka reçus en direct pour offrir une vision instantanée de l'activité.

4. DATA CACHING

- **ÉTAPE 7 : COUCHE DE CACHE MONGODB (SCRIPT PYTHON 2)**
  Implémenter un mécanisme de cache avec MongoDB pour améliorer les temps de réponse. Un script Python extraira périodiquement les agrégations depuis MySQL et les stockera sous forme de documents. Les dashboards pourront ainsi charger ces données précalculées plutôt que d'exécuter des requêtes coûteuses.

5. DATA PROCESSING (MACHINE LEARNING)

- **ÉTAPE 8 : CONSTRUCTION DU MODÈLE PREDICTIF (SCRIPT PYTHON 3)**
  Concevoir un pipeline d'apprentissage automatique qui exploite les données accumulées pour anticiper les tendances futures. Le processus d'entraînement sera planifié pour s'exécuter quotidiennement, intégrant ainsi les dernières observations.
  Le script consommera les données depuis MongoDB (données agrégées).
  Implémenter le même modèle avec trois technologies différentes : Spark MLlib, Dask-ML et Scikit-learn (Pandas).
  Analyser les résultats pour déterminer l'approche la mieux adaptée à votre contexte.

- **ÉTAPE 9 : ANALYSE DES PERFORMANCES**
  Mesurer et comparer les trois implémentations selon plusieurs axes :
  - Durée d'entraînement
  - Utilisation mémoire
  - Capacité à monter en charge
  - Complexité de mise en œuvre.

  Documenter les conclusions et justifier le choix final.

BONUS

- **ORCHESTRATION AUTOMATISEE**
  Identifier et récupérer un jeu de données adapté à votre problématique. Cette décision orientera l'ensemble de votre démarche analytique. Veillez à sélectionner des données de qualité suffisante pour permettre une exploitation pertinente.
  Une validation préalable par l'encadrant est requise avant de poursuivre.

- **MIGRATION CLOUD**
  Transposer l'architecture sur une plateforme cloud en substituant les composants locaux par leurs équivalents hébergés (exemple : RDS pour MySQL, EMR pour Spark).

- **SOURCES ADDITIONNELLES**
  Enrichir le flux de données en intégrant des sources externes via NiFi : APIs tierces, fichiers complémentaires.

- **SYSTÈME D'ALERTES**
  Configurer des notifications automatiques déclenchées par les prédictions du modèle ou le dépassement de seuils critiques.

ROLE DES SCRIPTS PYTHON

Script	Source	Destination	Fréquence
Script 1 : Consumer	Kafka	MySQL	Continu
Script 2 : Cache	MySQL	MongoDB	Périodique
Script 3 : ML	MongoDB	Modèle sauvegardé	Quotidien

RECOMMANDATIONS POUR LA SOUTENANCE

- Prévoir une démonstration fonctionnelle du pipeline complet.
- Argumenter vos décisions techniques face aux alternatives possibles.
- Partager les obstacles rencontrés et les solutions mises en œuvre.
- Illustrer la valeur ajoutée métier de votre réalisation.
- Préparer des réponses sur l'évolutivité et les améliorations envisageables.
- Défendre le choix du framework ML avec des données chiffrées.

Bon courage !