Projet FinanceLake

Architecture Data Lakehouse Temps Réel

Réalisé par :

Omanou Mohamed
Boukhmira Youssef
El jaouhari Abdelmalek

Encadré par :
Pr. M. Elhajji

Janvier 2026

1 Description du Projet

Le projet **FinanceLake** est une plateforme de *Data Lakehouse* avancée conçue pour l'ingestion, le traitement et l'analyse prédictive en temps réel de données financières (actions, crypto-monnaies et matières premières). L'objectif principal est de transformer des flux de données brutes instables en signaux décisionnels exploitables (Achat/Vente) grâce à l'apprentissage automatique, tout en optimisant drastiquement les ressources de stockage.

2 Architecture du Système (Medallion Architecture)

Le cœur de **FinanceLake** repose sur une **Architecture en Médaillon** qui assure la fluidité, la fiabilité et la scalabilité du traitement des données. Le fonctionnement du pipeline suit un flux logique où chaque technologie joue un rôle précis :

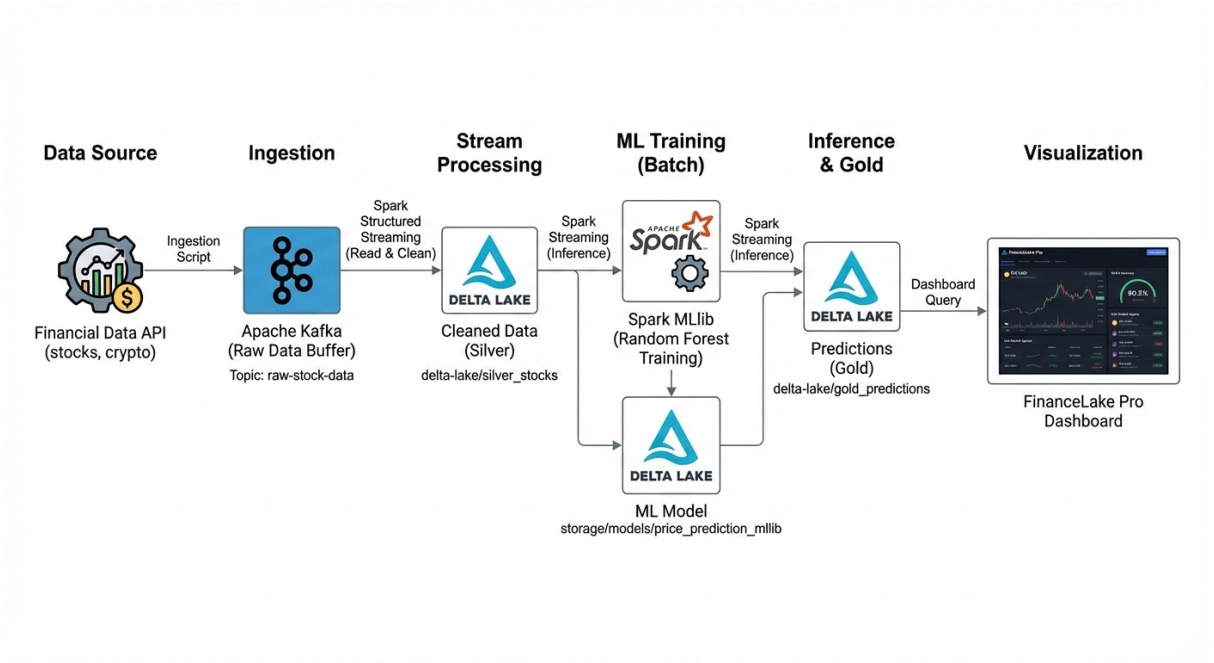

FIGURE 1 – Flux de données de FinanceLake (Kafka vers Delta Lake)

Fonctionnement du Pipeline et Rôles des Technologies

Le pipeline opère en trois étapes majeures, orchestrées par l'écosystème Spark et Kafka :

1. **Ingestion et Mémoire Tampon (API vers Kafka)** : Des scripts Python récupèrent les données brutes depuis les API financières. Au lieu d'écrire ces données sur disque, elles sont publiées directement dans **Apache Kafka**. Le rôle de Kafka

ici est crucial : il sert de tampon (*buffer*) à haut débit qui découple la source de données du moteur de traitement, permettant ainsi de gérer des flux massifs sans saturer le stockage persistant (Couche Bronze).

2. **Transformation et Fiabilité (Spark Streaming vers Silver)** : **Apache Spark Streaming** consomme les messages Kafka en temps réel. Son rôle est de transformer les flux bruts en données structurées. Il effectue le nettoyage, le typage des colonnes et la gestion des valeurs manquantes. Ces données sont ensuite stockées dans des tables **Delta Lake** (Couche Silver). **Delta Lake** apporte ici la fiabilité nécessaire grâce aux transactions ACID, garantissant l'intégrité des données lors des écritures concurrentes.
3. **Analyse Prédictive (MLlib vers Gold)** : Une fois la donnée propre dans la couche Silver, le moteur **Spark MLlib** intervient. Son rôle est d'exécuter l'algorithme *Random Forest* pour l'apprentissage automatique. Il analyse les tendances historiques pour générer des prédictions de prix et des signaux d'achat/vente. Ces résultats enrichis sont sauvegardés dans la couche **Gold**, qui sert de source de données finale pour la visualisation.

Synthèse des Couches de Données

— **Couche Bronze (Kafka) :** Stockage transitoire et volatil des données brutes pour optimiser les ressources de stockage.
— **Couche Silver (Delta Lake) :** Données nettoyées, normalisées et historisées de manière fiable.
— **Couche Gold (Delta Lake) :** Données agrégées et enrichies par les prédictions du modèle ML, prêtes pour l'affichage sur le tableau de bord.

3 Choix Technologiques

Technologie	Rôle	Justification
Apache Kafka	Bronze	Ingestion temps réel et réduction de l'empreinte disque.
Apache Spark	Calcul	Moteur de traitement de flux et calcul distribué pour le ML.
Delta Lake	Storage	Garantit l'intégrité des données (ACID) et le versioning.
Spark MLlib	ML	Implémentation du Random Forest pour les prédictions.
Docker	Infra	Conteneurisation pour une portabilité et scalabilité totale.

4 Résultats et Visualisation

Grâce à l'utilisation de l'algorithme *Random Forest* via Spark MLlib, le système atteint une précision de prédiction (R^2) de **90.5%**. Le tableau de bord **FinanceLake Pro** permet de visualiser ces performances et les signaux de marché en direct.

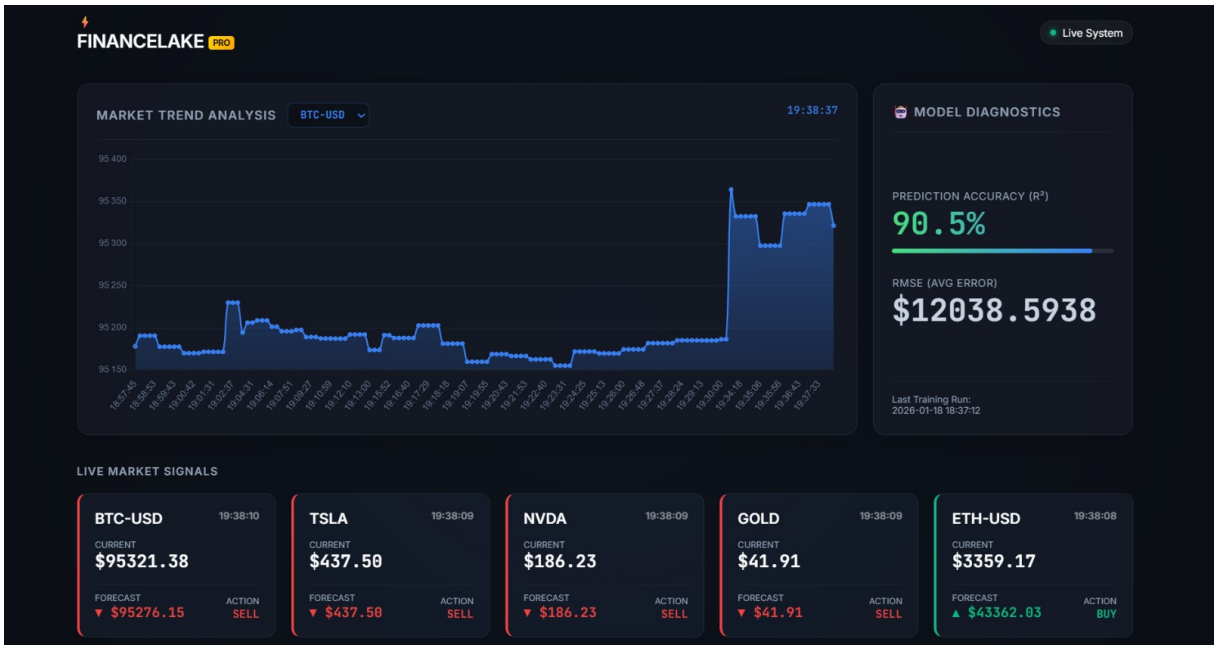

FIGURE 2 – Tableau de bord FinanceLake Pro - Analyse et Signaux en Temps Réel

5 Conclusion

Le projet **FinanceLake** met en œuvre une architecture *Data Lakehouse* temps réel efficace pour l'analyse de données financières. L'association de **Kafka**, **Spark**, **Delta Lake** et **Spark MLlib** permet une ingestion continue, un traitement fiable et une exploitation prédictive des données.

L'architecture en médaillon assure une organisation claire des données et une bonne scalabilité du système. Les résultats obtenus, avec une précision de **90.5%**, confirment la pertinence de la solution proposée.