DOCUMENT ARCHITECTURE GLOBAL

1 Objectif de l'architecture

L'objectif de cette architecture est d'assurer un flux de données cohérent et sécurisé entre les systèmes **OLTP**, **OLAP** et **NoSQL**.

Le pipeline permet :

- de traiter les transactions en temps réel,
- de stocker les données analytiques pour le reporting,
- de gérer les données non structurées,
- d'intégrer un modèle de machine learning pour la détection de fraude.

L'architecture est conçue pour séparer les usages transactionnels, analytiques et machine learning.

2 Sources de données

Les données proviennent de plusieurs sources :

- **Applications Stripe-like / API** : paiements, remboursements, abonnements, rétrofacturations.
- **Flux temps réel** : événements de transaction et statuts.
- **Données non structurées** : logs applicatifs, interactions utilisateurs, avis clients.
- **Données de référence** : marchands, produits, pays, devises, taux de change.

3 Système OLTP – Base transactionnelle

Les données transactionnelles sont stockées dans une base relationnelle PostgreSQL, utilisée comme système **OLTP**.

Son rôle est :

- enregistrer les transactions en temps réel,
- garantir la cohérence et l'intégrité des données,
- gérer les entités principales (transaction, client, marchand, paiement, abonnement, litige).

Outils utilisés

- API REST Stripe-like
- PostgreSQL

Ce système respecte les propriétés **ACID** et constitue la source de vérité des données transactionnelles.

4 Synchronisation des données – CDC

Les modifications effectuées dans la base OLTP sont synchronisées vers le reste du pipeline grâce à un mécanisme de **Change Data Capture (CDC)**.

Son rôle est :

- détecter les insertions et mises à jour dans PostgreSQL,
- transmettre les événements en temps réel vers un système de streaming.

Outils utilisés

- Debezium pour la capture des changements
- Apache Kafka pour le transport des événements

5 Traitement temps réel et détection de fraude

Les transactions diffusées dans Kafka peuvent être analysées en temps réel afin d'évaluer un risque de fraude.

Principe général

- chaque transaction est analysée par un modèle de machine learning,
- le modèle retourne un score de fraude et une décision simple (acceptée, à vérifier, rejetée),
- la décision est enregistrée avec la transaction.

Les résultats du scoring sont stockés afin de pouvoir être analysés et réutilisés lors des entraînements futurs.

6 Orchestration et intégration des données

Les différents traitements de données sont orchestrés à l'aide d'un outil de planification.

Son rôle est :

- gérer les flux batch et quasi temps réel,
- déclencher les transformations de données,
- orienter les données vers les bons systèmes de stockage.

Outils utilisés

- Apache Airflow pour l'orchestration
- Apache Spark pour la transformation des données
- Amazon S3 pour le stockage temporaire des données brutes

7 Système OLAP – Analyse et reporting

Les données transformées sont stockées dans un système **OLAP** sous forme de schéma analytique.

Son rôle est :

- agréger et historiser les données,
- calculer les indicateurs clés (revenus, activité, fraude),
- permettre l'analyse et le reporting.

Outils utilisés

- Snowflake (data warehouse)
- dbt pour les transformations SQL
- Tableau / Power BI pour la visualisation

Le système OLAP est utilisé uniquement pour l'analyse et ne supporte pas les opérations transactionnelles.

8 Système NoSQL – Données non structurées et ML

Les données non structurées et semi-structurées sont stockées dans une base **NoSQL**.

Son rôle est :

- stocker les logs et interactions utilisateurs,
- conserver les résultats de fraude,
- alimenter les modèles de machine learning.

Outils utilisés

- MongoDB (base documentaire)
- API de scoring pour la fraude

Les relations avec les systèmes OLTP et OLAP sont assurées par des identifiants communs (transaction_id, customer_id, etc.).

9 Sécurité, conformité et gouvernance

La sécurité est intégrée à toutes les étapes du pipeline.

Les principales mesures mises en place sont :

- chiffrement des données en transit et au repos,
- contrôle d'accès basé sur les rôles,
- journalisation des accès et des traitements,
- conformité aux réglementations RGPD, PCI-DSS et CCPA.

10 Conclusion

Cette architecture globale permet de gérer efficacement les données transactionnelles, analytiques et non structurées.
Elle offre une séparation claire des usages tout en garantissant la cohérence, la sécurité et la conformité des données.

L'architecture répond aux besoins d'analyse, de reporting et de détection de fraude dans un contexte de croissance des volumes de données.