Documentation Technique - Architecture de Données Stripe

Vue d'ensemble

Cette documentation décrit l'architecture de données complète mise en place pour Stripe, intégrant les systèmes OLTP, OLAP, NoSQL, un Data Lake centralisé et des capacités avancées de Machine Learning avec MLOps.

---

1. Sources de Données

1.1 Description

Point d'entrée de toutes les données dans le système, captant les informations provenant de multiples canaux.

1.2 Composants

- **API Stripe** : Transactions financières, paiements, remboursements
- **Applications Web** : Interactions utilisateurs, données de navigation
- **Applications Mobile** : Données mobiles, géolocalisation, type d'appareil
- **Systèmes Tiers** : Intégrations externes, webhooks, données partenaires

1.3 Types de données collectées

- Transactions financières
- Événements utilisateurs
- Logs d'application
- Données de session
- Métadonnées des requêtes

---

2. Couche OLTP - Traitement Transactionnel

2.1 Objectif

Gestion des transactions en temps réel avec garantie ACID, haute disponibilité et faible latence.

2.2 Composants

Base OLTP Transactions ACID

- **Technologies** : PostgreSQL / MySQL

- **Propriétés** : Atomicité, Cohérence, Isolation, Durabilité
- **Volumétrie** : Millions de transactions par jour
- **Latence cible** : < 100ms

Kafka Topics CDC (Change Data Capture)

- **Rôle** : Capture des modifications en temps réel
- **Pattern** : Event Sourcing
- **Format** : JSON / Avro
- **Garantie** : At-least-once delivery

Réplica Lecture

- **Objectif** : Scalabilité horizontale pour les lectures
- **Stratégie** : Réplication asynchrone
- **Usage** : Rapports, requêtes analytiques légères

Redis Cache

- **Usage** : Cache distribué pour données fréquemment accédées
- **TTL** : Configurable par type de données
- **Patterns** : Cache-aside, Write-through

2.3 Cas d'usage

- Traitement des paiements
- Gestion des abonnements
- Remboursements et rétrofacturations
- Validation des transactions

---

3. Couche d'Ingestion Temps Réel

3.1 Description

Pipeline de streaming permettant l'ingestion et le traitement des données en temps réel.

3.2 Composants

Kafka Connect CDC

- **Fonction** : Connecteur pour la capture des changements
- **Sources** : Bases OLTP, logs applicatifs
- **Destination** : Topics Kafka

Apache Kafka Streaming

- **Rôle** : Bus de messages distribué
- **Débit** : Millions d'événements/seconde
- **Rétention** : Configurable (7-30 jours)
- **Partitionnement** : Par clé métier (merchant_id, transaction_id)

Apache Flink - Stream Processing

- **Usage** : Traitement stateful en temps réel
- **Fonctionnalités** :
  - Agrégations en fenêtres temporelles
  - Enrichissement des données
  - Détection d'anomalies en temps réel
- **Garanties** : Exactly-once processing

3.3 Flux de données

1. CDC capture les changements dans OLTP
2. Kafka achemine les événements
3. Flink traite et enrichit les données
4. Distribution vers NoSQL, Data Lake et ML

---

4. Data Lake - Stockage Centralisé

4.1 Architecture Medallion

Bronze Layer (Données Brutes)

- **Contenu** : Données brutes non transformées
- **Format** : JSON, Parquet, CSV
- **Rétention** : Longue durée (plusieurs années)

- **Usage** : Archivage, retraitement, audit

Silver Layer (Données Nettoyées)

- **Contenu** : Données validées, dédupliquées, normalisées
- **Transformations** :
  - Nettoyage des valeurs nulles
  - Standardisation des formats
  - Enrichissement avec données de référence
- **Format** : Parquet optimisé
- **Usage** : Source pour ETL et ML

Gold Layer (Données Agrégées)

- **Contenu** : Données agrégées business-ready
- **Caractéristiques** :
  - Métriques pré-calculées
  - Données dénormalisées
  - Optimisées pour requêtes analytiques
- **Usage** : BI, reporting, dashboards

4.2 Object Storage (S3 / ADLS)

- **Capacité** : Pétabytes de données
- **Coût** : Optimisé avec lifecycle policies
- **Sécurité** : Chiffrement at-rest, versioning
- **Organisation** : Partitionnement par date, type, région

4.3 Gouvernance des données

- Catalogage automatique (AWS Glue, Azure Purview)
- Lineage tracking
- Data quality checks
- Politiques de rétention

---

5. Couche NoSQL - Données Non Structurées

5.1 MongoDB - Documents JSON

Usage :

- Profils clients enrichis
- Historique des interactions
- Métadonnées des transactions
- Configuration dynamique

Caractéristiques :

- Schéma flexible
- Indexation avancée
- Agrégation pipeline
- Sharding horizontal

5.2 Elasticsearch - Logs & Recherche

Usage :

- Logs applicatifs centralisés
- Recherche full-text
- Monitoring et observabilité
- Analyse des erreurs

Fonctionnalités :

- Indexation en temps réel
- Recherche fuzzy
- Agrégations statistiques
- Visualisation avec Kibana

5.3 Cassandra - Séries Temporelles

Usage :

- Métriques de performance
- Historique des prix

- Événements temporels
- Données IoT (devices de paiement)

Avantages :

- Write-optimized
- Scalabilité linéaire
- Haute disponibilité
- Time-based partitioning

5.4 ML Feature Store

Rôle :

- Stockage des features ML
- Versioning des features
- Serving temps réel
- Cohérence online/offline

Technologies : Feast, Tecton, ou custom

---

6. Couche ETL / ELT

6.1 Apache Airflow - Orchestration

Responsabilités :

- Planification des jobs
- Gestion des dépendances
- Monitoring et alerting
- Retry logic

DAGs principaux :

- daily_oltp_to_olap
- hourly_fraud_detection
- weekly_customer_segmentation
- monthly_compliance_reports

6.2 dbt - Transformations

Usage :

- Transformations SQL modulaires
- Tests de qualité de données
- Documentation automatique
- Version control

Modèles :

- Staging : nettoyage initial
- Intermediate : logique métier
- Marts : tables finales business

6.3 Spark Jobs - Batch Processing

Cas d'usage :

- Traitement de volumes massifs
- Agrégations complexes
- Feature engineering
- Retraitement historique

Optimisations :

- Partitionnement intelligent
- Broadcast joins
- Caching stratégique
- Dynamic resource allocation

---

7. Couche OLAP - Analytique

7.1 Snowflake / Redshift - Data Warehouse

Architecture :

- Séparation compute/storage
- Auto-scaling

- Query optimization
- Multi-cluster warehouses

Schéma :

- Modèle en étoile (star schema)
- Tables de faits : transactions, events
- Tables de dimensions : customers, merchants, products

7.2 OLAP Cubes

Dimensions d'analyse :

- Temps (jour, semaine, mois, trimestre)
- Géographie (pays, région, ville)
- Produit (type, catégorie)
- Client (segment, industrie)
- Canal (web, mobile, API)

Mesures :

- Revenus, volume, marge
- Taux de conversion
- Taux de fraude
- Customer lifetime value

7.3 Vues Matérialisées

Exemples :

- daily_revenue_by_merchant
- monthly_fraud_statistics
- customer_segmentation_summary
- product_performance_metrics

Rafraîchissement :

- Incrémental quand possible
- Planifié selon la criticité
- Triggered par événements métier

8. Couche ML & Analytics

8.1 Détection Fraude - Real-Time ML

Approche :

- Modèles de scoring en temps réel
- Ensemble methods (XGBoost, Random Forest)
- Règles métier + ML
- Threshold dynamique

Features :

- Montant, devise, géolocalisation
- Historique du marchand
- Patterns de comportement
- Velocity checks

Performance :

- Latence < 50ms
- Précision > 95%
- Taux de faux positifs < 1%

8.2 Segmentation Client - Analytics

Méthodes :

- K-means clustering
- RFM Analysis (Recency, Frequency, Monetary)
- Behavioral segmentation
- Predictive segments

Segments :

- High-value customers
- At-risk customers
- Growth potential

- Dormant accounts

8.3 Analyses Prédictives - ML Models

Modèles :

- Churn prediction
- Lifetime value estimation
- Revenue forecasting
- Demand prediction

Pipeline :

- Feature engineering
- Model training
- Backtesting
- A/B testing
- Deployment

8.4 LLM Fine-Tuning - Support Client

Objectif :

- Automatisation du support niveau 1
- Réponses contextuelles
- Support multilingue
- Escalade intelligente

Données d'entraînement :

- Historique des tickets
- Documentation produit
- FAQs
- Feedbacks clients

Métriques :

- Satisfaction client (CSAT)
- Résolution au premier contact

- Temps de réponse moyen
- Taux d'escalade

---

9. Couche de Présentation

9.1 API Analytics

Endpoints :

- `/metrics/revenue`
- `/analytics/fraud-detection`
- `/reports/compliance`
- `/insights/customer-behavior`

Caractéristiques :

- RESTful design
- Rate limiting
- Authentication OAuth2
- Versioning API

9.2 Dashboards BI (Tableau/Looker/Power BI)

Tableaux de bord :

- Executive Dashboard
- Operations Dashboard
- Fraud Monitoring Dashboard
- Customer Analytics Dashboard

Fonctionnalités :

- Drill-down/up
- Filtres interactifs
- Exports automatisés
- Alertes personnalisées

9.3 Rapports de Conformité

Types :

- RGPD : Registre des traitements, audits
- PCI-DSS : Rapports trimestriels de sécurité
- CCPA : Rapports de droits des consommateurs
- SOC 2 : Contrôles de sécurité

Automatisation :

- Génération planifiée
- Validation automatique
- Distribution sécurisée
- Archivage auditable

9.4 Chatbot IA - Support Client

Capacités :

- Compréhension du langage naturel
- Réponses contextuelles
- Intégration système de tickets
- Handoff vers agents humains

Technologies :

- LLM fine-tuné sur données Stripe
- Vector database pour RAG
- Intent classification
- Sentiment analysis

---

10. MLOps - Opérations Machine Learning

10.1 Composants

Model Registry (MLflow)

- Versioning des modèles

- Tracking des expérimentations
- Métriques de performance
- Artifacts management

Training Pipeline

Étapes :

1. Data validation
2. Feature engineering
3. Model training
4. Hyperparameter tuning
5. Model evaluation
6. Model registration

Automatisation :

- Réentraînement déclenché par drift
- Scheduled retraining
- Performance-based triggers

Model Monitoring

Métriques surveillées :

- Performance metrics (accuracy, AUC)
- Data drift
- Concept drift
- Feature drift
- Prediction drift

Alerting :

- Dégradation de performance
- Anomalies dans les distributions
- Latence excessive
- Erreurs de prédiction

Feedback Loop

Sources de feedback :

- Interactions chatbot
- Décisions de fraude validées
- Comportement utilisateur post-prédiction
- Labels manuels des équipes

Utilisation :

- Enrichissement des données d'entraînement
- Active learning
- Continuous improvement
- A/B testing validation

10.2 Flux de réentraînement

1. Monitoring détecte un drift
2. Notification déclenchée
3. Extraction données du Data Lake
4. Préparation dataset (Silver → Gold)
5. Entraînement avec nouveaux paramètres
6. Validation sur données de test
7. Comparaison avec modèle actuel
8. Approbation (automatique ou manuelle)
9. Déploiement canary
10. Rollout complet
11. Monitoring intensifié post-déploiement

---

11. Sécurité & Conformité

11.1 Chiffrement TDE/SSL

At-rest :

- Chiffrement transparent des bases de données
- Chiffrement des fichiers dans le Data Lake
- KMS (Key Management Service) centralisé
- Rotation automatique des clés

In-transit :

- TLS 1.3 pour toutes les communications
- Certificats SSL/TLS gérés
- mTLS pour communications inter-services

11.2 RBAC & IAM

Rôles :

- Data Engineer : accès aux pipelines ETL
- Data Analyst : lecture Data Warehouse
- Data Scientist : accès ML Platform
- Compliance Officer : accès audits et logs

Principe : Least privilege access

Implémentation :

- IAM groups et policies
- Service accounts
- Temporary credentials
- MFA obligatoire

11.3 Audit Logs

Événements tracés :

- Accès aux données sensibles
- Modifications de configuration
- Déploiements de modèles
- Requêtes SQL sur données PII
- Exports de données

Rétention : 7 ans (conformité)

Analyse :

- SIEM integration
- Détection d'anomalies
- Alertes temps réel

- Rapports périodiques

11.4 Masquage de Données

Techniques :

- Tokenization pour numéros de carte
- Hashing pour identifiants
- Masquage dynamique en fonction du rôle
- Anonymisation pour environnements non-prod

Données masquées :

- PII (Personally Identifiable Information)
- PCI (Payment Card Information)
- Données médicales
- Informations contractuelles sensibles

11.5 Conformité Réglementaire

RGPD (Règlement Général sur la Protection des Données)

- Droit à l'oubli : procédure de suppression
- Portabilité : export de données personnelles
- Consentement : tracking et gestion
- DPO : Data Protection Officer désigné

PCI-DSS (Payment Card Industry Data Security Standard)

- Ségrégation du réseau
- Logs d'audit détaillés
- Tests de pénétration trimestriels
- Gestion des vulnérabilités

CCPA (California Consumer Privacy Act)

- Transparence sur collecte de données
- Opt-out de la vente de données
- Suppression sur demande

- Rapports annuels de conformité

---

12. Flux de Données Principaux

12.1 Flux Transactionnel Temps Réel

```
Source → OLTP → Kafka CDC → Flink Processing →
  ├─→ NoSQL (détection fraude)
  ├─→ Data Lake (archivage)
  └─→ ML Models (scoring temps réel)
```

12.2 Flux Analytique Batch

```
OLTP Replica → Data Lake Bronze → ETL Processing →
Data Lake Silver → Transformations dbt → Data Lake Gold →
OLAP Warehouse → Dashboards BI
```

12.3 Flux ML & Feedback

```
Data Lake Gold → Feature Store → ML Training →
Model Registry → Deployment → Predictions →
User Feedback → Data Lake Silver → Retraining
```

---

13. Métriques et KPIs

13.1 KPIs Techniques

- **Latence P99 OLTP** : < 100ms
- **Débit Kafka** : > 1M messages/sec
- **Disponibilité système** : 99.99%
- **Temps de traitement ETL** : < 4h pour batch quotidien
- **Fraîcheur des données OLAP** : < 15 minutes

13.2 KPIs ML

- **Accuracy détection fraude** : > 95%
- **False positive rate** : < 1%

- **Latence prédiction** : < 50ms
- **Model drift** : < 5% par mois
- **Coverage chatbot** : > 80% des questions niveau 1

13.3 KPIs Business

- **Coût par transaction** : optimisation continue
- **ROI prévention fraude** : > 10:1
- **Satisfaction client (CSAT)** : > 4.5/5
- **Temps de résolution moyen** : < 2h
- **Adoption self-service** : > 60%

---

14. Technologies et Outils

14.1 Stack Technique Complet

Bases de données :

- OLTP : PostgreSQL 15, MySQL 8.0
- OLAP : Snowflake, Amazon Redshift
- NoSQL : MongoDB 6.0, Elasticsearch 8.x, Cassandra 4.x
- Cache : Redis 7.x, Memcached

Data Processing :

- Streaming : Apache Kafka 3.x, Apache Flink 1.17
- Batch : Apache Spark 3.4, dbt 1.6
- Orchestration : Apache Airflow 2.7

Storage :

- Object : AWS S3, Azure Data Lake Storage
- File formats : Parquet, Avro, JSON

ML Platform :

- Training : PyTorch, TensorFlow, Scikit-learn
- MLOps : MLflow, Kubeflow

- Feature Store : Feast, Tecton
- LLM : GPT-4, Claude, modèles custom

BI & Visualisation :

- Tableau, Looker, Power BI
- Superset, Metabase

Infrastructure :

- Cloud : AWS, Azure, GCP
- Containers : Docker, Kubernetes
- IaC : Terraform, CloudFormation

Sécurité :

- IAM : AWS IAM, Azure AD
- Secrets : HashiCorp Vault, AWS Secrets Manager
- Monitoring : Datadog, Prometheus, Grafana

---

15. Évolutions Futures

15.1 Court terme (3-6 mois)

- Implémentation du data mesh
- Amélioration des temps de réponse OLAP
- Extension du feature store
- Optimisation des coûts cloud

15.2 Moyen terme (6-12 mois)

- Migration vers lakehouse architecture (Delta Lake)
- Real-time OLAP avec Apache Druid
- Federated learning pour ML distribué
- Data observability avancée

15.3 Long terme (12-24 mois)

- Intégration quantum computing pour détection fraude

- Graph databases pour analyse de réseau
- Edge computing pour processing géolocalisé
- Zero-trust data architecture

---

16. Conclusion

Cette architecture de données moderne et scalable permet à Stripe de :

✅ Traiter des millions de transactions quotidiennes avec haute fiabilité
✅ Fournir des analyses en temps réel et historiques
✅ Détecter et prévenir la fraude efficacement
✅ Personnaliser l'expérience client
✅ Maintenir la conformité réglementaire
✅ Améliorer continuellement grâce au ML

L'architecture est conçue pour évoluer avec la croissance de Stripe tout en maintenant performance, sécurité et gouvernance des données.

Version : 1.0
Date : Décembre 2024
Auteur : Équipe Data Engineering Stripe
Classification : Confidentiel - Usage Interne