Bloc 2
Architecture des données

David **RAMBEAU**

Full **S**tack in **D**ata **S**cience

Promotion 2024 -2025

- Sommaire -

- Architecture des données -

1. Objectif d'une architecture de données.

Ce que l'on appelle « architecture de données » constitue le socle technologique permettant de transformer des données brutes en informations stratégiques exploitables par les comités de direction.

Le principal objectif réside en la conception d'un système cohérent basé sur la confiance des informations mise à disposition. Pour cela, trois aspects de l'architecture data sont à prendre en compte :

- *Performance opérationnelle pour le traitement temps réel des transactions,*
- *Fiabilité analytique pour les prises de décision, et ce, de façon transverse*
- *Conformité réglementaire .*

Une architecture bien conçue doit gérer efficacement le cycle de vie complet des données : ingestion, stockage, transformation via pipelines ETL/ELT, et partage des données/informations.

Une architecture de données robuste est enrichie d'une structure de contrôle automatisé pour garantir la qualité des données.

Devant le nombre important de datas à traiter, il est important qu'une architecture de donnée fiable puisse absorber la croissance de ces volumes par la mise en œuvre de moyens techniques qui permettent de répondre au besoin de scalabilté.

Permettre à l'organisation de prendre des décisions éclairées en temps réel, anticiper les tendances grâce à l'analytique avancée et de maintenir la confiance des utilisateurs par le respect des réglementations et la sécurité de leurs données est clairement l'objectif de la mise en route d'une data architecture.

2. Contexte chez Stripe.

Stripe est un système de paiement en ligne qui fonctionne sous fortes contraintes. L'entreprise doit gérer simultanément des flux transactionnels temps-réels, des volumes massifs de données, et des exigences réglementaires strictes.

Il est évident que les enjeux des paiements en ligne nécessitent de maitriser les chaines d'échanges, notamment par la détection de fraude automatisée.

Ce cas d'usage est critique et nécessite une analyse en temps réel de multiples signaux via des modèles de machine learning, tout en garantissant la transparence et l'explicabilité exigées par la réglementation européenne.

Enfin, pour garantir une gouvernance de données moderne et efficace, Stripe doit démocratiser l'accès aux données tout en appliquant un strict contrôle d'accès de façon granulaire.

L'ensemble de ces contraintes illustre la complexité du périmètre dans lequel opère Stripe, nécessitant une rigueur d'exécution et un suivi à la hauteur des enjeux.

3. Inventaire des données

Maîtriser son patrimoine data commence par un inventaire précis des données

Cette cartographie est essentielle pour connaitre les données à manipuler et analyser. Une fois cet inventaire réalisé, nous serons à même de sécuriser les transactions et garantir la conformité.

	Objectifs
Acheteur	Gestion des comptes clients et conformité RGPD.
Compte client	Suivi des soldes, multi-devises, statuts de compte.
Marchand	Onboarding, vérification, configuration des paiements.
Moyens de paiements	Tokenisation des cartes, gestion des moyens de paiement
Transactions	Traçabilité des paiements, réconciliation.
Tentatives de paiement	Analyse des échecs, optimisation des taux de conversion.
Factures	Facturation récurrente, relances.
Frais	Calcul des frais Stripe, reporting financier.
Remboursements	Gestion des litiges et remboursements.
Événements Temps-réel	Détection de fraude, notifications, déclenchement d'actions
Logs de conformité	Audit PCI-DSS, traçabilité des accès.
Métadonnées Transactions	Données non structurées (ex : champs personnalisés pour les marchands)
Séries temporelles	Analyse des tendances, détection d'anomalies.
Graphes de relations	Détection de fraude (ex : réseaux de comptes suspects).
Données de fraude	Décision en temps réel (blocage/autorisation).
Données analytiques	Reporting business, optimisation des coûts.
Features ML	Entraînement de modèles de détection de fraudes.

4. Répartition et modèle des données

La sélection d'une architecture de données ne relève pas du hasard.

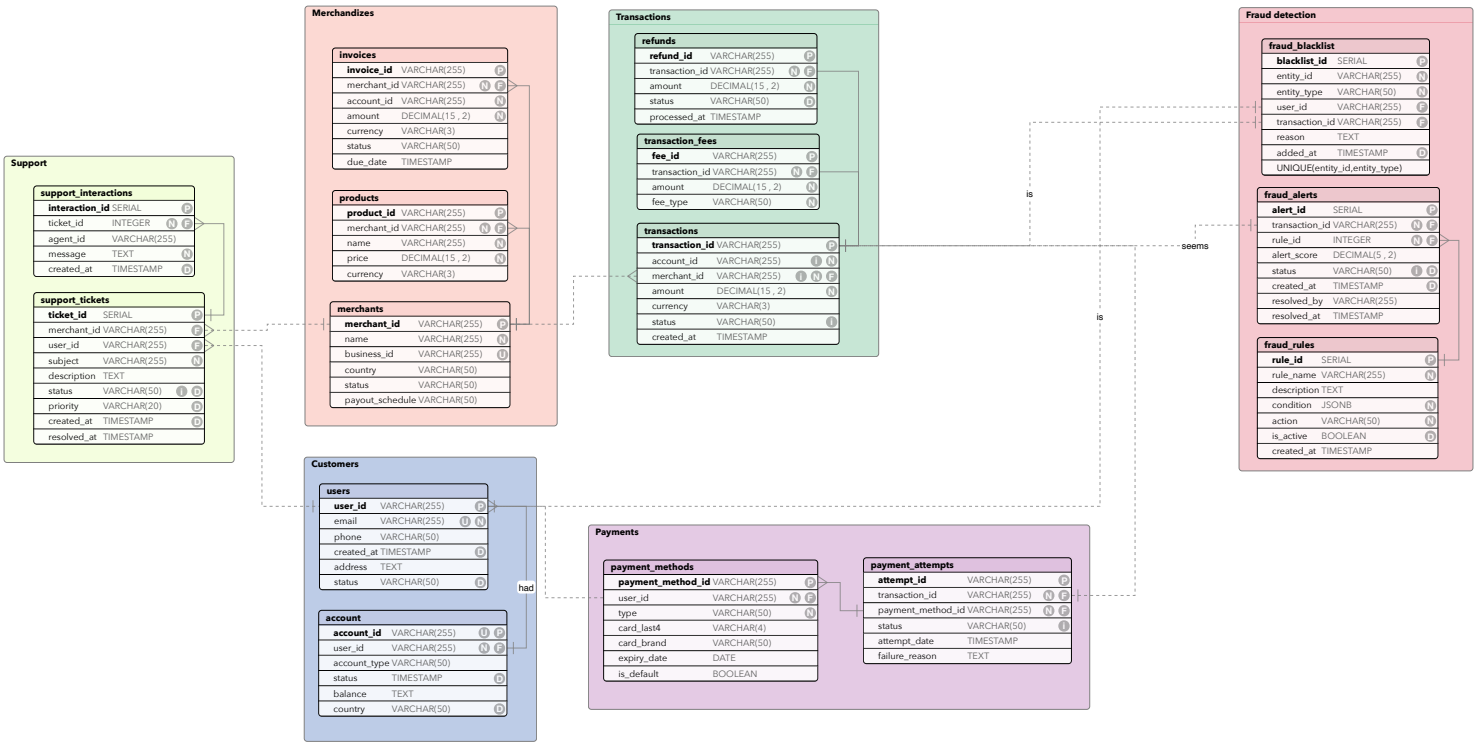
Cette décision architecturale exerce une influence directe sur la performance, le coût et la maintenabilité du système.

Par conséquent, c'est uniquement après l'analyse exhaustive de l'inventaire de données que nous sommes en mesure d'organiser les données en fonction des exigences métier et de déterminer la réponse technique la plus appropriée.

4.1. Gestion des relations transactionnelles.

Au cœur de l'infrastructure de Stripe, le modèle transactionnel repose sur PostgreSQL, une solution robuste et éprouvée pour gérer les flux financiers critiques.

Ce choix s'impose par la nécessité d'assurer une traçabilité exhaustive, une cohérence absolue et une conformité stricte aux normes, comme le PCI-DSS. PostgreSQL permet d'orchestrer avec précision les interactions entre clients, marchands, transactions et moyens de paiement, depuis l'initiation d'un paiement jusqu'à sa finalisation ou sa contestation.



Ce modèle garantit donc l'intégrité des données, tout en offrant la flexibilité nécessaire pour supporter des cas d'usage complexes, comme la détection de fraude ou la réconciliation comptable.

Le choix d'une technologie OLTP comme PostgreSQL détermine un socle idéal pour une plateforme où chaque transaction doit être auditable, infaillible et temps-réels.

4.2. Gestions des relations analytiques.

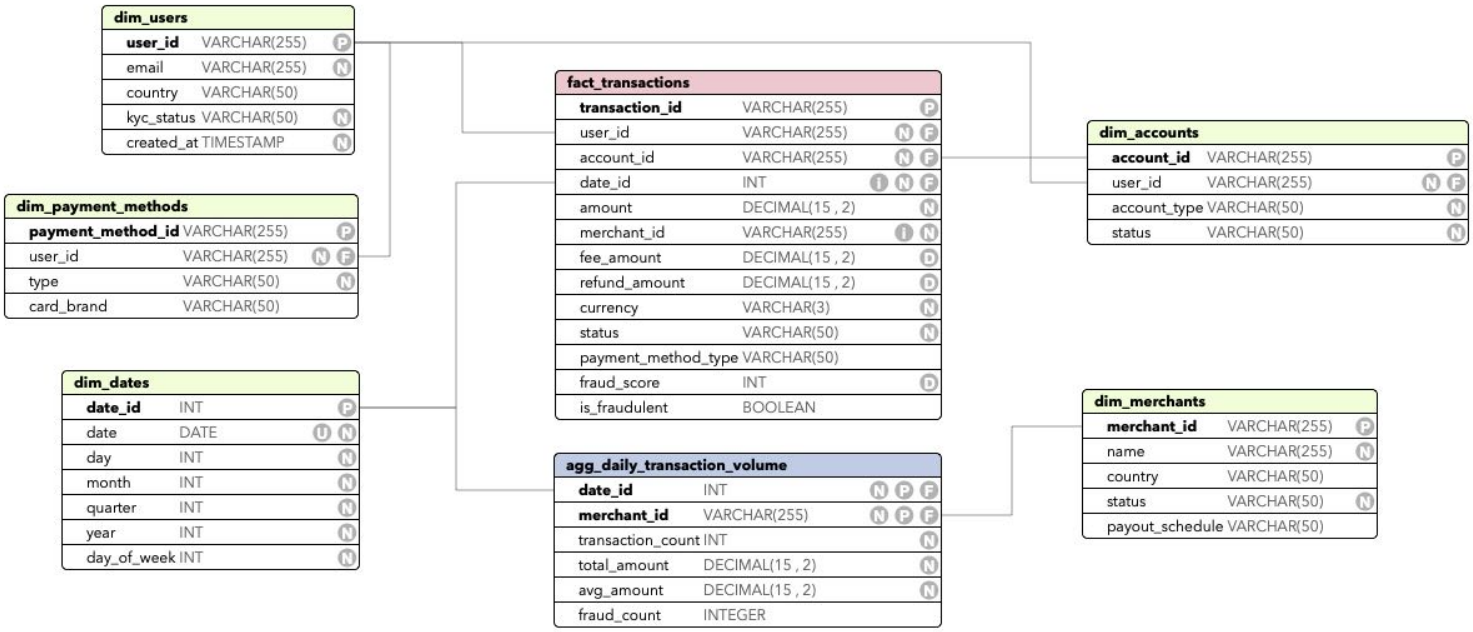
Le choix de Snowflake pour l'analyse OLAP repose avant tout sur des besoins métiers concrets : explorer, agréger et exploiter les données transactionnelles à grande échelle.

Grâce à son architecture optimisée pour les tables de faits (transactions, événements) et les dimensions associées (clients, marchands, temps), Snowflake permet de construire des agrégations dynamiques adaptées aux besoins spécifiques, telles que le suivi des volumes de paiement, l'analyse des taux de fraude, ou encore la segmentation client.

Sa capacité à gérer des requêtes complexes sans dégrader les performances, couplée à une scalabilité facile à déployer, en fait l'outil idéal pour répondre aux exigences des équipes métiers (finance, risque, marketing).

Avec Snowflake, Stripe démocratise l'accès aux informations, en offrant une flexibilité inégalée pour créer des rapports sur mesure.

Une solution clé en main pour transformer les données brutes en décisions stratégiques.



4.3. Gestion des données non structurées.

Pour répondre au un besoin critique de stocker et d'exploiter des données non structurées sur une masse de données, notre choix se porte sur un modèle NoSQL avec MongoDB.

Contrairement aux bases relationnelles, MongoDB excelle dans la gestion de documents JSON complexes, comme les profils utilisateurs, les logs d'activités ou les métadonnées transactionnelles, où le schéma évolue fréquemment.

Son modèle sans schéma fixe permet d'adapter dynamiquement les collections aux besoins métiers (ex. : personnalisation des expériences client, historique des interactions), tandis que ses requêtes riches (agrégations, indexation avancée) facilitent l'analyse en temps réel.

C'est donc une solution adaptée aux données variées et évolutives, où la rigidité des tables relationnelles serait un frein.

4.4. Pour résumer.

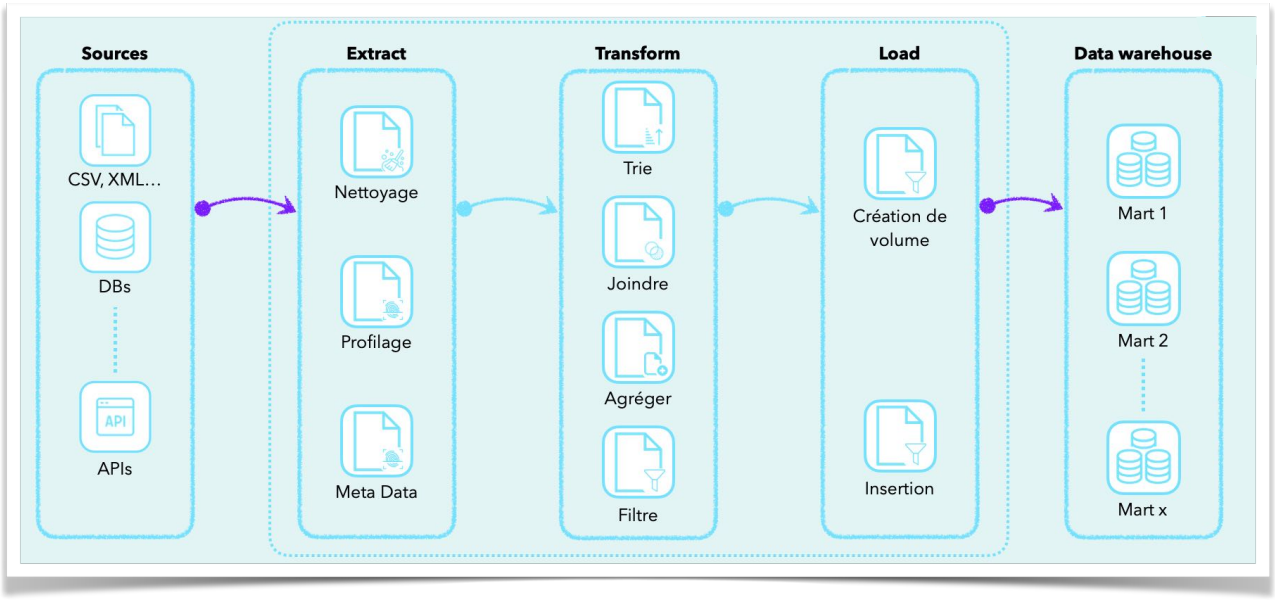
Des réflexions menées jusqu'alors, nous pouvons dégager une matrice décisionnelle nous permettant de tirer profit des notre expérience pour la mise en œuvre d'une architecture de données comme celle de Stripe.

Données	Technologie	Avantages	Critères de Sélection	Exemple
Transactionnelles (OLTP)	PostgreSQL	ACID, cohérence forte, requêtes complexes, intégrité des données	Besoins en transactions fiables, audits, conformité réglementaire, relations complexes entre entités	Transactions financières, profils marchands, historique des paiements
Analytiques (OLAP)	Snowflake	Scalabilité, performances sur requêtes complexes, gestion des tables de faits et dimensions	Analyse à grande échelle, intégration avec outils BI (Tableau), besoin de flexibilité pour les agrégations	Analyse des volumes de paiement, segmentation client, détection de fraude
Non structurées	MongoDB	Schéma flexible, scalabilité horizontale, requêtes riches (agrégations, indexation), performances en lecture/écriture	Données évolutives, absence de schéma fixe, besoin de réactivité pour les données variées	Profils clients, logs d'activité, sessions utilisateurs, données JSON complexes

5. Intégration des données

Dans un écosystème data moderne, l'intégration des données est un pilier essentiel pour transformer des sources hétérogènes en informations exploitables et fiables.

L'intégration des données dans la plateforme data repose sur un processus ETL classique (Extract, Transform, Load), une méthodologie éprouvée pour garantir la qualité, la cohérence et la disponibilité des données.



Les raisons de ce choix reposent en partie pour répondre aux contraintes d'harmonisation des sources variées (fichiers CSV/XML, bases de données, APIs), chacune nécessitant une collecte efficace et sécurisée.
Cette étape permet de capturer les données brutes tout en préservant leur intégrité.

Les données extraites subissent ensuite un nettoyage, un enrichissement et une structuration (tri, jointure, agrégation, filtrage).
L'uniformisation des formats de formats de données passe par la correction des anomalies et la préparation des données pour les traiter ensuite à des fins d'analyse.

Les données transformées sont ensuite chargées dans un entrepôt de données organisé en « marts » métiers.
La raison de cette ségrégation des données est essentiellement pour respecter la conformité tout en assurant une accessibilité optimale pour les outils d'analyse et de reporting.

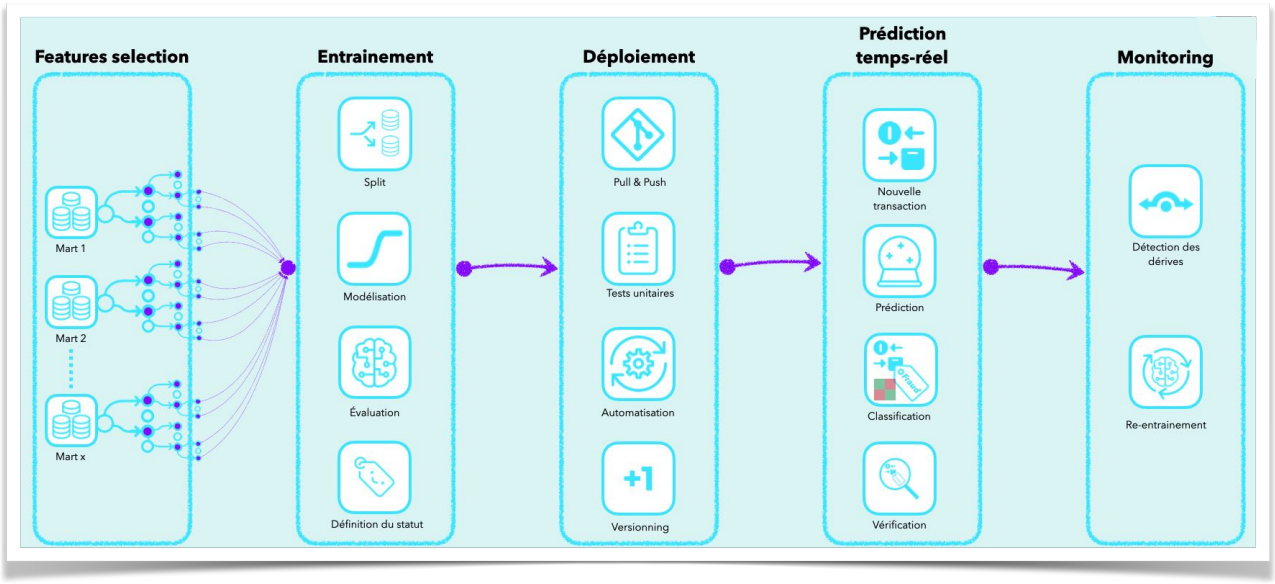
Cette architecture permet de centraliser les données, de réduire les silos et de fournir une base fiable pour la prise de décision.

6. Intégration de l'apprentissage automatique

La détection de fraude est un enjeu critique pour Stripe, où chaque transaction doit être analysée en temps réel pour identifier les comportements suspects tout en minimisant les **faux positifs** (blocage de transactions légitimes) et les **faux négatifs** (fraudes non détectées).

Afin de répondre à cet objectif, un système basé sur une régression logistique a été mis en œuvre, conjuguant performance algorithmique et supervision humaine. Ce modèle statistique, simple, mais robuste, permet d'estimer la probabilité qu'une transaction soit frauduleuse en fonction de critères tels que :

- *Le montant de la transaction,*
- *La fréquence des opérations,*
- *La répétition des transactions,*
- *La localisation géographique,*
- *Le comportement historique de l'utilisateur.*



Pour évaluer avec précision le taux de fraude et répondre efficacement au besoin initial, plusieurs points sont à prendre en compte dans la réussite de notre modèle d'apprentissage.

La sélection rigoureuse des features est un principe important au moment du choix à opérer pour l'entrainement de notre modèle. Cette phase est déterminante pour éviter d'introduire des résultats qui nous amèneraient à faire les mauvaises décisions.

D'autre part, au moment de procéder à l'isolation des données à partir des données initiales (train-test split) il est important de s'assurer de l'équilibrage du jeu de données. Il faut que le résultat soit représentatif, en tenant compte du déséquilibre naturel entre les classes, par exemple via des techniques de rééchantillonnage.

Enfin, l'évaluation du modèle se fait en conjuguant des métriques adaptées, afin de minimiser à la fois les faux négatifs (fraudes non détectées) et les faux positifs (transactions légitimes bloquées).

L'objectif final étant d'aboutir à un modèle fiable et équilibré, capable de détecter les fraudes tout en limitant les impacts sur l'expérience client.

7. Déploiement

La partie du déploiement constitue une véritable opportunité de franchir un cap vers une démarche d'excellence.

L'automatisation du déploiement est un pilier indispensable pour garantir rigueur et agilité. En éliminant les interventions manuelles, elle réduit drastiquement les erreurs humaines, assure une reproductibilité parfaite des environnements, et accélère les mises en production tout en maintenant un niveau élevé de qualité et de traçabilité.

Grâce à des pipelines CI/CD, chaque modification; qu'il s'agisse d'une mise à jour de modèle ML, d'une évolution du schéma de données ou d'un nouveau traitement ETL; est testée automatiquement (tests unitaires, d'intégration), versionnée et déployée de manière contrôlée.

Cette approche offre une souplesse opérationnelle sans compromis sur la stabilité, tout en permettant un rollback instantané en cas d'anomalie.

Résultat, une architecture véloce, fiable et auditable, où la confiance repose sur des processus reproductibles, monitorés et validés à chaque étape.

Une condition essentielle pour concilier innovation et résilience.

8. Conteneurisation

Pour faire fonctionner l'ensemble de la plateforme, nous avons opté sur une technologie de conteneurisation, à savoir « Docker ». Cette méthode de travail est un levier stratégique pour assurer la reproductibilité, la portabilité et un déploiement fluide des solutions data, comme celle mise en œuvre chez Stripe.

En encapsulant chaque composant (qu'il s'agisse de PostgreSQL, d'un modèle de machine learning ou d'une API FastAPI) dans des conteneurs isolés, Docker élimine les dépendances aux environnements sous-jacents.

L'avantage premier est qu'un conteneur construit localement s'exécutera de manière identique sur un serveur on-premise, dans le cloud ou sur une machine de développement, quel que soit le système d'exploitation (Windows, Linux, macOS).

L'utilisation de Docker simplifie également la gestion des dépendances et des configurations.

Chaque conteneur intègre tout ce dont l'application a besoin pour fonctionner : code, librairies, variables d'environnement et outils système.

Cela permet de déployer des solutions complexes en quelques commandes, sans conflit de versions ni configuration manuelle fastidieuse. Par exemple, le modèle de détection de fraude, l'API d'exposition des résultats et la base de données peuvent être déployés ensemble ou séparément, avec une scalabilité et une isolation des applications renforcée pour la sécurité.

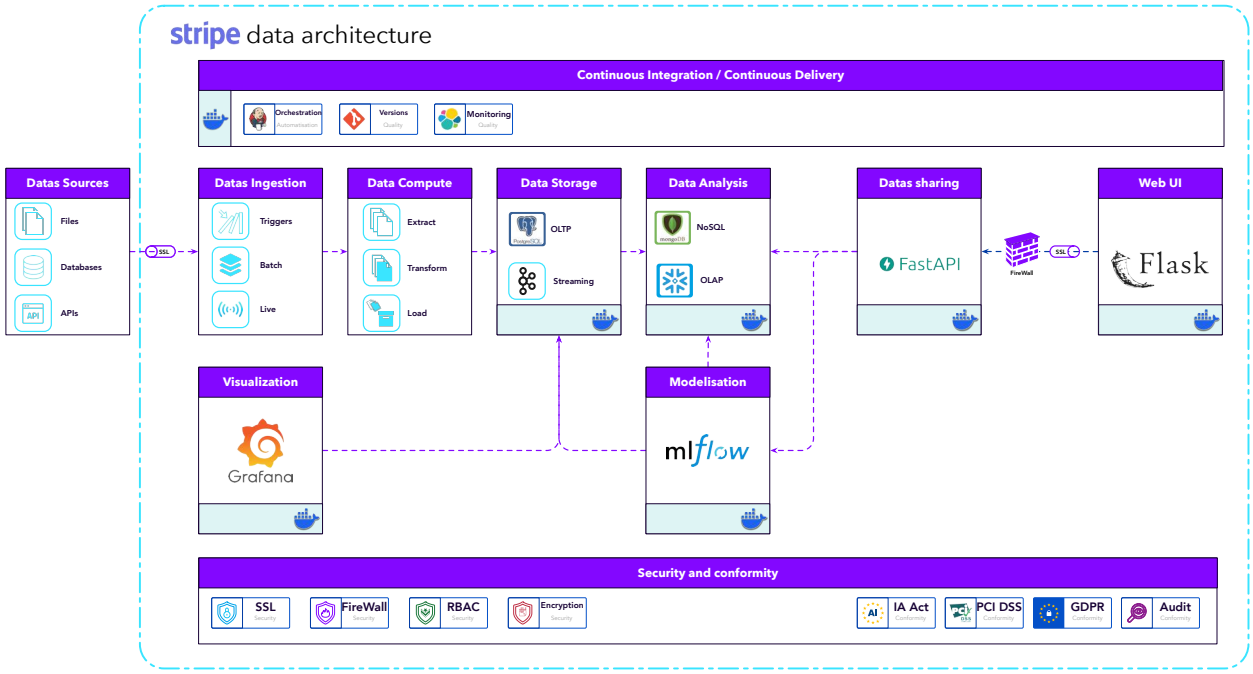
Docker s'intègre naturellement dans les pipelines CI/CD, automatisant les tests et les déploiements pour une livraison continue, versionnée et fiable.

9. Schéma d'architecture

Concevoir une architecture data performante ne relève pas d'une recette universelle, mais bien d'un équilibre subtil entre choix technologiques, contraintes métiers, ressources disponibles et évolutions futures.

Une architecture robuste n'est pas figée, elle émerge d'une compréhension fine des besoins et d'une veille constante sur les technologies et les réglementations.
La flexibilité est au cœur de cette démarche, car ce type d'architecture doit pouvoir évoluer à l'avenir en intégrant de nouvelles sources de données, supporter des charges accrues, ou répondre à des exigences réglementaires renforcées.

Par ailleurs, les choix technologiques sont toujours le fruit d'un arbitrage réfléchi entre les impératifs techniques et les réalités terrain. Ainsi, une architecture data n'est jamais « parfaite »; elle est optimisée pour un contexte donné, et sa pérennité dépend de sa capacité à se réinventer sans remettre en cause sa fiabilité. C'est ce que propose le schéma d'architecture suivant :



Une structure modulaire et évolutive, où chaque composant (ingestion, stockage, analyse, sécurité) est pensé pour s'adapter, s'étendre et se perfectionner au rythme des enjeux de l'entreprise.

Dans le domaine des données, la constante demeure l'adaptation. C'est par cette agilité qu'une entreprise visionnaire permet de transformer une infrastructure technique en un levier stratégique.

10.Plan de sécurité et de conformité.

10.1.Sur la sécurité

Ce document présente le plan de sécurité et de conformité de l'architecture de données Stripe, couvrant les aspects techniques, organisationnels et réglementaires nécessaires pour garantir la protection des données de paiement et la conformité aux réglementations en vigueur (RGPD[1], PCI-DSS[2], AI Act[3]).

10.1.1.Infrastructure couverte par le plan sécurité

- ***OLTP*** *: PostgreSQL (données transactionnelles)*
- ***NoSQL*** *: MongoDB,*
- ***Données streaming*** *: Apache Kafka*
- ***OLAP*** *: MongoDB*
- ***Storage*** *: S3*
- ***Machine Learning*** *: MLflow (modèle fraude)*
- ***APIs*** *: FastAPI*
- ***WebApp*** *: Flask (simulation transactions)*

10.1.2.Parties prenantes couvertes par le plan sécurité.

Le plan de sécurité couvre l'ensemble des parties prenantes impliquées dans l'architecture, garantissant une protection cohérente et adaptée à leurs rôles.

- Les équipes techniques bénéficient de règles strictes d'accès, de chiffrement et d'audit pour sécuriser les données en développement et en production.
- Les utilisateurs métiers (marketing, finance, support) sont protégés par des politiques de contrôle d'accès (RBAC) et de chiffrement des données, assurant la confidentialité des informations sensibles.
- Les clients finaux voient leurs transactions et données personnelles sécuriser via des protocoles TLS 1.3 et des normes PCI-DSS/RGPD.
- Les prestataires externes (auditeurs, sous-traitants) sont encadrés par des accès temporaires et supervisés, tandis que les régulateurs (PCI-DSS, RGPD, AI Act) trouvent dans ce cadre une réponse aux exigences de conformité.
- Les partenaires technologiques (cloud providers, éditeurs SaaS) sont intégrés via des accords de sécurité (DPA) et des chiffrements end-to-end.

L'objectif visé par l'identification des parties prenantes, permet d'assurer une sécurité alignée sur les enjeux métiers, technique et réglementaire de Stripe.

---

[1] *Réglementation générale sur la protection des données.*
[2] *Payment Card Industry Data Security Standard*
[3] *Artificial Intelligence Act*

10.1.3. Périmètre de sécurité

Dans le cadre d'une architecture de données sécurisée comme celle de Stripe, la classification des données et leur niveau de sensibilité sont essentiels pour garantir la conformité et la protection des informations.

- Les numéros de carte bancaire, classés comme critiques, relèvent de la norme PCI-DSS, exigeant un chiffrement strict et un accès restreint.
- Les données personnelles clients et les transactions, de sensibilité élevée, sont soumises au RGPD, imposant anonymisation et traçabilité.
- Les modèles ML de détection de fraude, également élevés, doivent respecter l'AI Act, notamment en termes de transparence, d'explicabilité et d'auditabilité.
- Enfin, les logs d'audit, bien que tolérables, restent couverts par le RGPD pour assurer la traçabilité des accès.

Ce cadre permet de structurer les politiques RBAC[4], en alignant les rôles et permissions sur les exigences réglementaires, tout en minimisant les risques de fuites ou de non-conformité.

Une gouvernance data rigoureuse, combinée à des outils de monitoring, assure une protection adaptée à chaque type de donnée.

Type de données	Niveau de sensibilité	Réglementation
Numéros de carte bancaire	Critique	PCI-DSS
Données personnelles clients	Elevé	RGPD/CCPA
Données de transaction	Elevé	PCI-DSS + RGPD
Modèles ML de détection fraude	Elevé	AI Act
Logs d'audit	Tolérable	RGPD

[4] *Role-Based Access Control*

10.1.4. Contrôle d'accès.

Pour assurer le suivi, le contrôle et la gestion des droits attribués à chaque utilisateur, une matrice d'autorisations est élaborée en cohérence avec le dispositif de gouvernance des données et sous la supervision des responsables de la sécurité.

Rôle	Groupes d'Utilisateurs	Accès aux Données	Permissions Spécifiques
Administrateurs	DG	Accès complet (CRUD) sur toutes les couches de données.	Créer/supprimer des tables, gérer les rôles, configurer les politiques de sécurité.
Data engineers	DE	Lecture/écriture sur les couches ingestion, stockage et traitement.	Déployer des pipelines, optimiser les requêtes, gérer les schémas.
Data analysts	DA	Lecture seule sur OLAP et Data Lake (S3).	Créer des rapports, exécuter des requêtes analytiques, accéder aux dashboards.
Data scientists	DS	Lecture sur OLAP et Data Lake Ecriture sur des features et modèles ML.	Entraîner des modèles, accéder aux features, déployer des modèles en production.
Développeur Backend	Dev	Lecture/écriture sur OLTP (PostgreSQL) et APIs.	Développer des APIs, lire/écrire des données transactionnelles.
Responsable Sécurité	Secu	Lecture seule sur les logs et métadonnées ; accès aux outils de gouvernance.	Auditer les accès, configurer les politiques de sécurité, gérer les clés de chiffrement.
Support Client	Sup	Lecture seule sur les données clients (anonymisées) dans OLAP et OLTP.	Accéder aux données clients pour résoudre les tickets, sans accès aux données sensibles.
Auditeur	Audit	Lecture seule sur les logs, données transactionnelles et métadonnées.	Générer des rapports de conformité, accéder aux logs d'audit.
Utilisateur Métier	Marketing, Finance, Produit	Lecture seule sur les dashboards et rapports pré-approuvés.	Visualiser les données, exporter des rapports (avec restrictions).
Automatisation/ CI-CD	Comptes techniques	Écriture restreinte aux environnements de test/staging.	Déployer du code, exécuter des tests, accéder aux environnements non-prod.
Invité	Prestataires externes	Lecture seule sur des jeux de données spécifiques et anonymisés.	Accès temporaire et limité aux données nécessaires à leur mission.

Ce tableau se verra agrémenté également des outils et technologies associés à chaque permission.

10.1.5. Firewall

Un des piliers de notre système de sécurité dans l'architecture est la délégation du découpage par zone du réseau.

Ce découpage renforce la sécurité du réseau de l'architecture qui repose ainsi sur une isolation stricte en zones de sécurité, protégées par des pare-feu et des règles d'accès circonscrites par type d'application.

Cette architecture de défense en profondeur minimise les surfaces d'attaque, garantit la confidentialité des données sensibles, et permet une détection rapide des intrusions.

Les flux inter-zones sont chiffrés et audités, conformément aux normes PCI-DSS et ISO27001[5].

Zone	Composants	Protocole/ Accès	Rôle	Niveau de Sécurité
WAN	-	-	Point d'entrée/sortie des requêtes externes.	Aucune confiance.
DMZ	Load Balancer	HTTPS (TLS 1.3)	Filtrage des requêtes, protection contre les attaques DDoS et injections.	Haute sécurité, exposition contrôlée.
Application	FastAPI (8000) Flask (5001)	Firewall Rules, TLS 1.3	Traitement des requêtes applicatives, exposition des APIs.	Accès restreint depuis la DMZ.
Streaming	Kafka, Redis	TLS 1.2+, ACLs	Gestion des flux temps réel et cache distribué.	Accès privé, chiffrement en transit.
Stockage	MinIO (S3), Snowflake	TLS 1.3, IAM	Stockage des données structurées et non structurées.	Accès privé, chiffrement SSE.
Monitoring	Grafana, Prometheus	Accès VPN + MFA	Surveillance, logging et alerting.	Accès ultra-restreint, audit complet.
Réseau Privé	10.0.0.0/16	VLAN, ACLs réseau	Isolation des zones privées, communication interne sécurisée.	Confiance limitée, segmentation stricte.

[5] *Norme internationale de sécurité des systèmes d'information (Systèmes de management de la sécurité de l'information)*

10.1.6. Chiffrement en transit

Le chiffrement des données en transit est systématiquement appliqué pour sécuriser les échanges entre tous les composants de l'architecture.

Ces protocoles, combinés à des certificats signés par une autorité interne, éliminent les risques d'écoute ou de falsification, conformément aux exigences PCI-DSS et RGPD. Une revue régulière des certificats et des algorithmes renforce la posture de sécurité globale.

Connexion	Protocole	Algorithme	Justification
Client → API	TLS 1.3	ECDHE-RSA-AES256-GCM-SHA384	Standard industrie
API → PostgreSQL	TLS 1.2+	AES-256-GCM	Chiffrement DB
API → MongoDB	TLS 1.2+	AES-256-GCM	Chiffrement DB
Kafka → Consumers	TLS 1.2+	AES-256-GCM	Streaming sécurisé
API → Snowflake	TLS 1.3	ChaCha20-Poly1305	Cloud DWH

10.1.7. Chiffrement des données

Le chiffrement des données est un pilier central de la sécurité de l'architecture, appliqué à chaque couche de stockage pour protéger les informations sensibles.

Cette approche assure une protection conforme aux exigences PCI-DSS et RGPD, tout en centralisant la gestion des clés pour limiter les risques d'exposition. Les clés sont systématiquement rotatives et accessibles uniquement aux rôles autorisés, renforçant la résilience face aux cybermenaces.

Base de données	Méthode	Algorithme	Clés
PostgreSQL	TDE (pgcrypto)	AES-256-CBC	Vault (HashiCorp)
MongoDB	WiredTiger Encryption	AES-256-GCM	KMIP Server
Cassandra	Transparent Data Encryption	AES-256	Local Keystore
Redis	RDB Encryption	AES-256-CBC	Vault
MinIO	SSE-C	AES-256	KMS AWS

10.2.Sur la conformité

10.2.1.Respect de la vie privée - RGPD

La conformité en lien avec le respect de la vie privée est une ligne directrice chez Stripe face aux conséquences qu'il pourrait en résulte en cas de piratage. Par conséquent, les données personnelles utilisées tout au long du flux d'informations, de la collecte à l'entraînement et l'exécution du modèle, sont minimisées et pseudonymisées afin de limiter les risques d'identification non autorisée.

Par ailleurs, les clients bénéficient de leurs exercices de droits RGPD.
Une information claire et transparente est fournie aux utilisateurs sur l'utilisation de leurs données et les finalités du traitement, conformément à l'article 13 du RGPD.

De plus, une analyse des risques pour les droits et libertés des personnes concernées a été réalisée, et des mesures techniques et organisationnelles ont été mises en place pour garantir la sécurité et la confidentialité des informations.

Principe RGPD	Implémentation	Technologie
Minimisation des données	Collecte strict nécessaire	Transverse
Limitation de la conservation	Retention 36 mois max	Airflow DAG purge
Intégrité et confidentialité	Chiffrement AES-256	pgcrypto + TLS
Responsabilité	Logs d'audit immutables	Elasticsearch
« Privacy by Design »	Pseudonymisation	Tokenisation et hachage

Enfin, un registre des activités de traitement est tenu à jour pour s'assurer que les mesures de protection sont proportionnées aux risques encourus. Les durées de conservation en fonction du traitement des données ont été définies de concert avec les équipes juridiques.
Cette approche globale permet de concilier innovation technologique, respect des droits privés et conformité réglementaire.

Traitement	Finalité	Base légale	Durée conservation	Mesures sécurité
Transactions paiement	Exécution contrat	Art. 6(1)(b)	36 mois	Chiffrement AES-256 + TLS
Détection fraude	Intérêt légitime	Art. 6(1)(f)	24 mois	Pseudonymisation + RBAC
Analytics clients	Consentement	Art. 6(1)(a)	12 mois	Agrégation anonyme
Logs d'audit	Obligation légale	Art. 6(1)(c)	60 mois	Immutabilité + chiffrement

10.2.2. Respect des transactions financières - PCI-DSS

Stripe étant un acteur bancaire, l'entreprise est en obligation de se conformer au PCI-DSS. Par conséquent, pour être en conformité avec cette norme, des mesures de sécurité renforcées doivent se faire jour, notamment :

- *Chiffrement des données*
- *Contrôle d'accès strict*
- *Segmentation réseau*
- *Surveillance continue*

Pour chacun de ces points, un élément organisationnel ou technologique doit être opéré

- *Gestion des risques (Art. 9)*
- *Gouvernance des données (Art. 10)*
- *Transparence et traçabilité (Art. 12-13)*
- *Supervision humaine (Art. 14)*
- *Robustesse et cybersécurité (Art. 15)*

Exigence	Implémentation
Firewall	iptables + Security Groups
Pas de mots de passe par défaut	Rotation 90j + complexité
Protection données stockées	AES-256 + Tokenisation
Chiffrement transmission	TLS 1.3 obligatoire
Anti-virus	ClamAV + EDR
Accès sur besoin	RBAC + Principe moindre privilège
Identité unique	SSO + MFA
Accès physique	Datacenter
Logs et monitoring	Elasticsearch + Grafana
Tests sécurité	Pentests semestriels
Gestion des risques	Gouvernance data, PSSI

10.2.3. Respect de l'éthique - IA Act

Le respect de l'éthique et des exigences de l'IA Act est au cœur du système de détection de fraude de Stripe, classé comme à « haut risque » en raison de son impact direct sur l'accès aux services financiers et à la protection de ses clients.
Au-delà de la simple conformité, cette démarche reflète un engagement fort en faveur d'une intelligence artificielle transparente, gage de confiance.

Avant tout déploiement, des tests rigoureux sont menés pour éviter toute dérive et introduction de biais, tandis qu'une documentation exhaustive garantit la traçabilité et l'explicabilité des décisions algorithmiques.

Un suivi continu des performances et des disparités permet de corriger en temps réel les dérives potentielles, assurant que le modèle reste juste, impartial et aligné sur les valeurs réglementaires et sociétales.

En plaçant l'éthique au même niveau que la performance, Stripe renforce non seulement sa conformité, mais aussi la confiance de ses utilisateurs et partenaires.

Critère	Évaluation	Justification
Domaine d'application	Crédit/Finance	Haut risque
Impact sur droits fondamentaux	Blocage des transactions	Potentiel discrimination
Automatisation décision	Scoring fraude automatique	Décision sans humain
Niveau de risque	Haut risque	Conforme AI Act Art. 6

10.2.4. Audits de sécurité réguliers

Dans un environnement où les cybermenaces évoluent en permanence, les audits de sécurité réguliers et les tests d'intrusion (pentests) constituent une démarche indispensable pour Stripe.

Ces évaluations systématiques permettent d'identifier les vulnérabilités avant qu'elles ne soient exploitées, de valider l'efficacité des mesures de protection et de renforcer la confiance dans l'infrastructure.

En simulant des attaques réelles, les pentests révèlent non seulement les failles techniques, mais aussi les risques liés aux processus ou aux comportements humains.

Une approche récurrente et à large spectre permet de couvrir les systèmes, les réseaux, les applications et les pratiques internes, ainsi, Stripe assure une amélioration continue de la posture de sécurité.

Point notable de ce choix organisationnel, l'intégration de ces audits transverses à l'entreprise permettre de renforcer la culture data au travers de la cybersécurité et ainsi donner à Stripe un avantage concurrentiel.

Type d'audit	Fréquence	Périmètre	Responsable
Audit PCI-DSS	Annuel	Environnement de données des titulaires de cartes	Évaluateurs de sécurité qualifiés
Pentest externe	Semestriel	APIs publiques	Équipe « Red Team »
Pentest interne	Trimestriel	Réseau interne	RSSI
Audit RGPD	Annuel	Traitements données	DPO
Revue de code	Mensuel	Nouveau code	DevSecOps
Analyse de vulnérabilité	Hebdomadaire	Toute infrastructure	Automatisé
Audit AI Act	Annuel	Modèles ML	Agent de conformité

11.Formation utilisateur & sensibilisation data

Dans un écosystème technologique aussi complexe que celui de Stripe, la formation transverse des équipes est un levier stratégique pour garantir l'efficacité et la sécurité de l'architecture.

Les comportements humains; qu'il s'agisse de méconnaissance des outils, de résistances au changement ou d'erreurs involontaires; représentent souvent le maillon faible face aux risques opérationnels et cybersécurité.

Former l'ensemble des collaborateurs, des équipes techniques aux métiers, permet de renforcer la culture data, d'optimiser l'utilisation des outils et de minimiser les erreurs liées à une mauvaise manipulation des systèmes.

Pour ce faire, un plan annuel est donc mis en place pour s'assurer que l'ensemble des acteurs de l'entreprise connait les enjeux liés à la cybersécurité et est suffisamment à niveau.

Formation	Public	Fréquence recyclage	Durée
Sensibilisation cybersécurité	Tous	Annuel	2h
RGPD et vie privée	Tous	Annuel	1h
Simulation de phishing	Tous	Trimestriel	Test inopiné
Développement sécurisé	Développeurs	Semestriel	4h
Réactivité aux incidents	DevOps	Annuel	8h
PCI-DSS et cybersécurité	Équipe paiement	Annuel	4h

12.Conclusion

Ce dossier présente une architecture de données moderne, scalable et sécurisée, spécialement conçue pour répondre aux défis de Stripe en matière d'excellence opérationnelle et de détection de fraude par IA.

En s'appuyant sur des technologies éprouvées et des mécanismes de sécurité avancés, elle garantit disponibilité, intégrité et confidentialité des données, tout en permettant une analyse en temps réel des transactions grâce à des pipelines optimisés et des modèles d'apprentissage automatique.

Cette infrastructure réactive et résiliente permet non seulement d'identifier et de bloquer les fraudes instantanément, mais aussi d'alimenter des tableaux de bord analytiques et des APIs sécurisées pour une visibilité opérationnelle complète.

Alliant automatisation DevOps, standards industriels et innovation IA, elle offre une protection contre les fraudes, tout en assurant une expérience client fluide et de confiance. C'est tout l'enjeu de la mise en place d'une architecture data au sein d'une organisation.