Architecture NewBank

V7: Scale

From France to Europe and beyond.
Mastering heavy loads and diverse regulations.

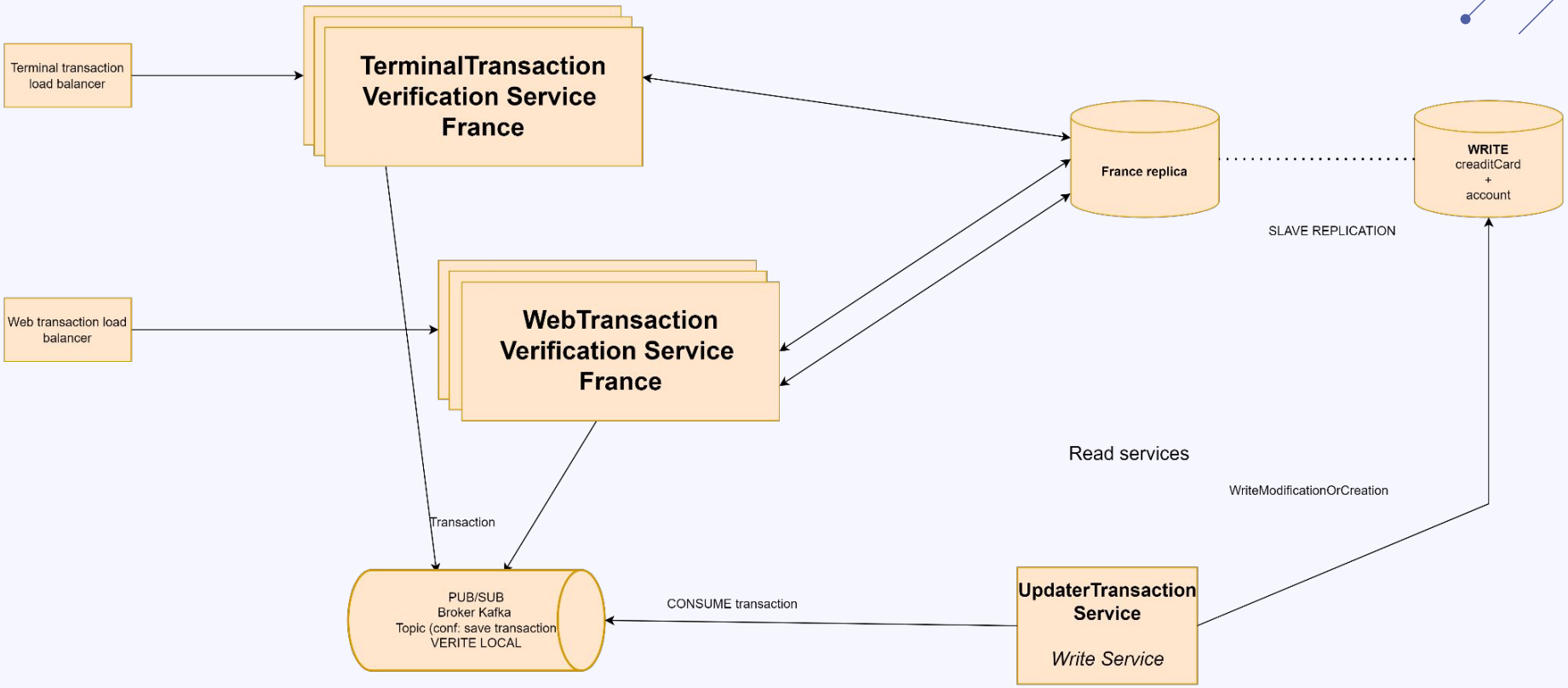
Igor Melnyk

Tobias Bonifay

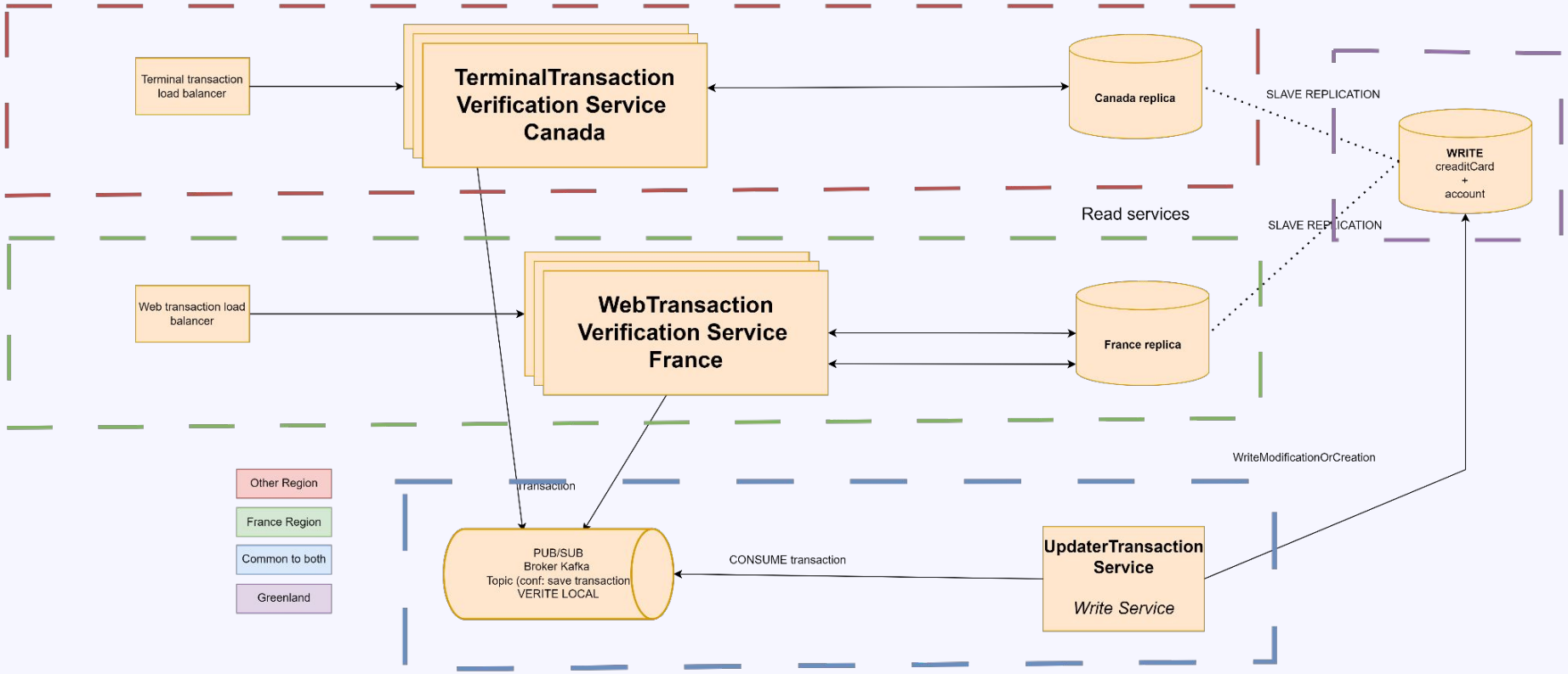
Ayman Bassy

Mathieu Schalkwijk

Architecture du PoC : une région



Architecture du PoC de Démo : Multiple regions



Services et Métiers

TerminalTransaction Verification Service

- Transaction Provenant d'un TPE.
- Validation de carte non requise.
- Validation d'information du compte client seul.

WebTransaction Verification Service

- Transaction Provenant du Web.
- Validation d'information du compte client et de sa carte.

UpdaterService

WriteThrought

- Mise à jour du compte client.

Contexte du PoC et Scénario

Hypothèse :

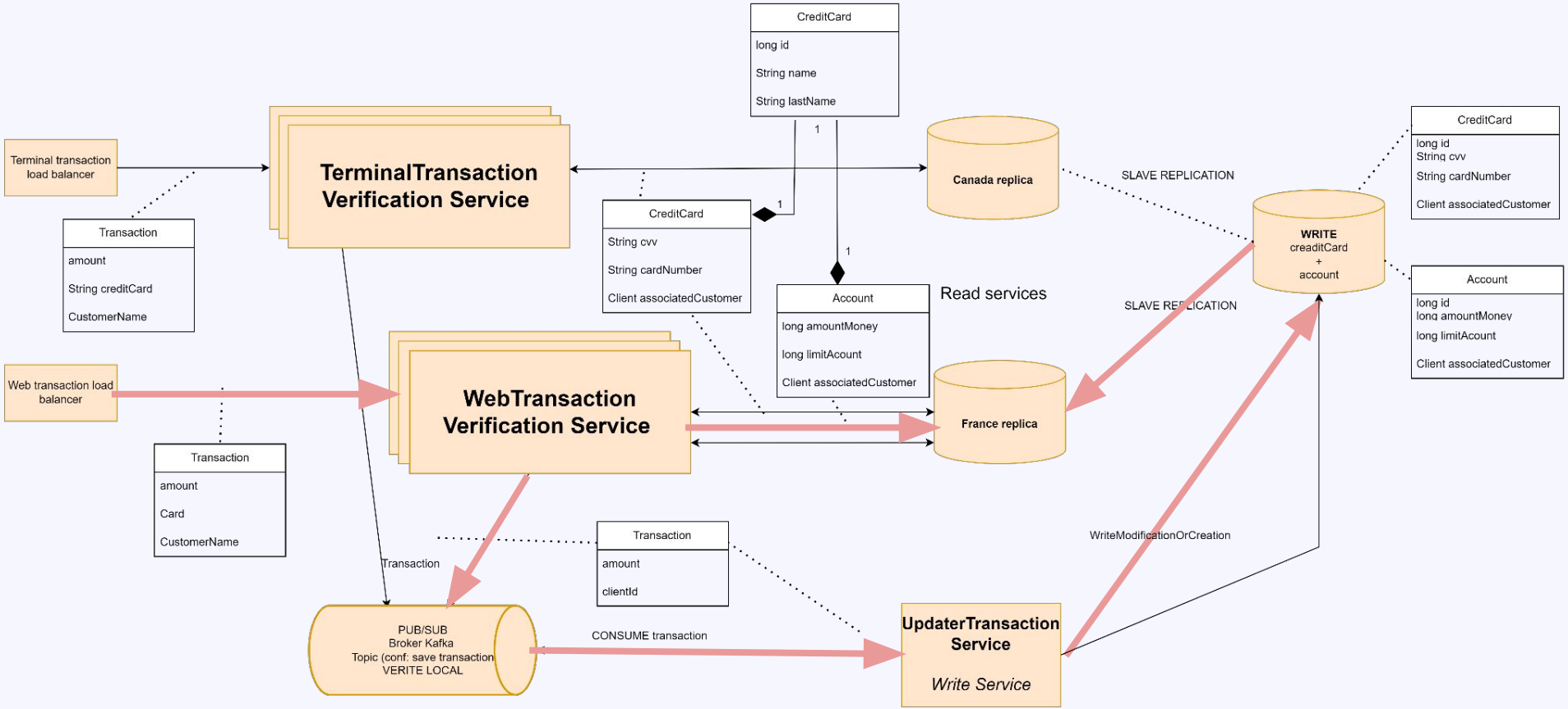
Un client **ne peut pas** enchainer deux paiements (donc transaction) en moins de 5 minutes .
=> Temporisation pour la mise à jour dans le **cadre d'une forte cohérence** de donnée.

Scénario : Test du flow de validation d'une transaction depuis le web sur un grand nombre de requêtes.

Besoin dans le contexte de la Démo :
Sur branche demo :

- **Pas de vérification sur la carte fourni** -> Besoin de generation d'un grand nombre de "client" dans la DB dans le cadre du test de haute charge.

Démo : Walking skeleton pour chaque requête (grosse charge)



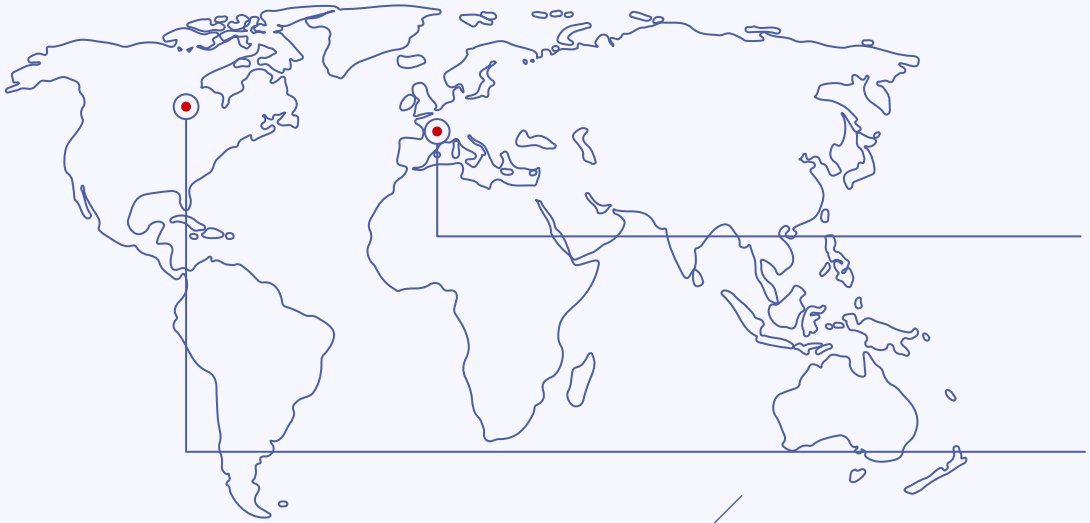
Choix Architecturaux

Gestion d'une grande charge de requêtes :

- Mise en place d'un **load balancer** vers plusieurs instance
- Mode CQRS
  - Service avec lecture exclusive vers read-only slave
  - Service de mise à jour sur master avec réplication sur les slaves
- Mise en place d'un bus de donnée **kafka** :
  - Sauvegarde du derniere état des transactions à modifier en cas de crash des services d'update.
  - **Objectif** : ne pas perdre les transactions valider pour rester cohérent.
  - **Non utilisé pour rejouer les événements**

Choix Architecturaux

Répartition géographique (LB + Services)



FRANCE Farm

Terminal Transaction LB

Web Transaction LB

CANADA Farm

Terminal Transaction LB

Web Transaction LB

Choix Architecturaux

Répartition géographique (BD)

GROENLAND
Cluster de Base de données Master.

FRANCE
Base de données slave
READ ONLY

CANADA
Base de données slave
READ ONLY

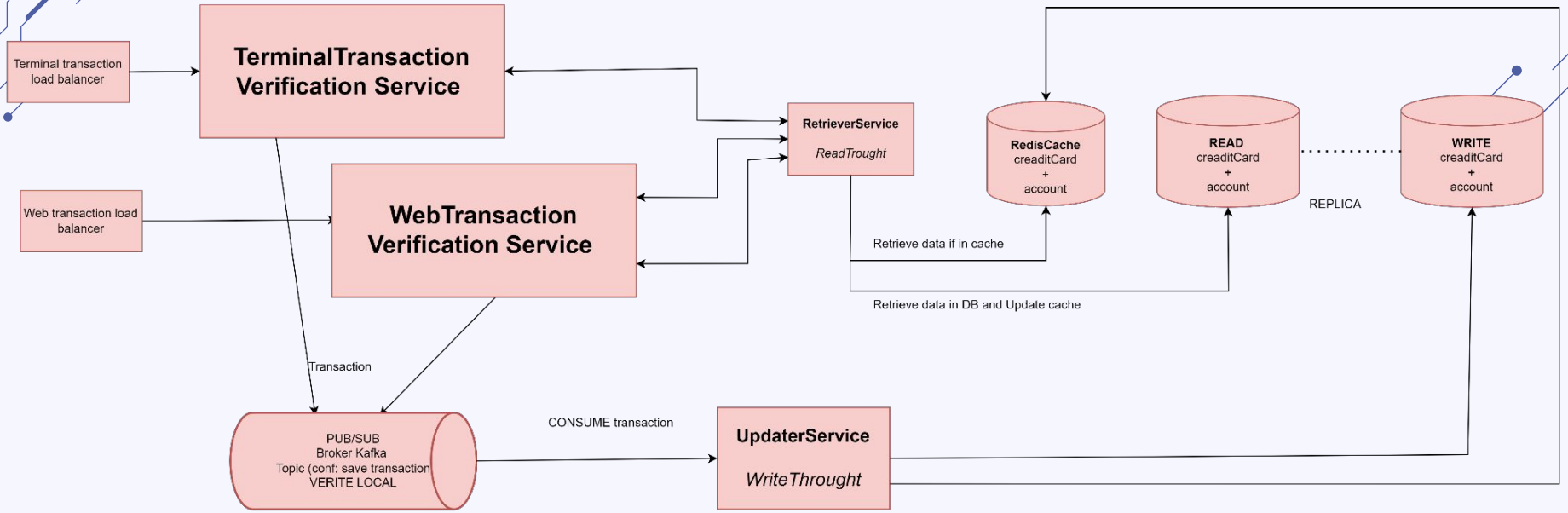
Faiblesses de l’architecture

Asynchronisation inclu dans le processus de paiement :

- **Le modèle CQRS inclus un mécanisme asynchrone dans le processus qui est l’update d’un compte après validation d’une transaction.**

  Cette Consistance éventuel peut inclure une marge d’erreur dans le cas où le client ferai un autre paiement durant le laps de temps requis pour mettre à jour son compte

Ancienne architecture exploré

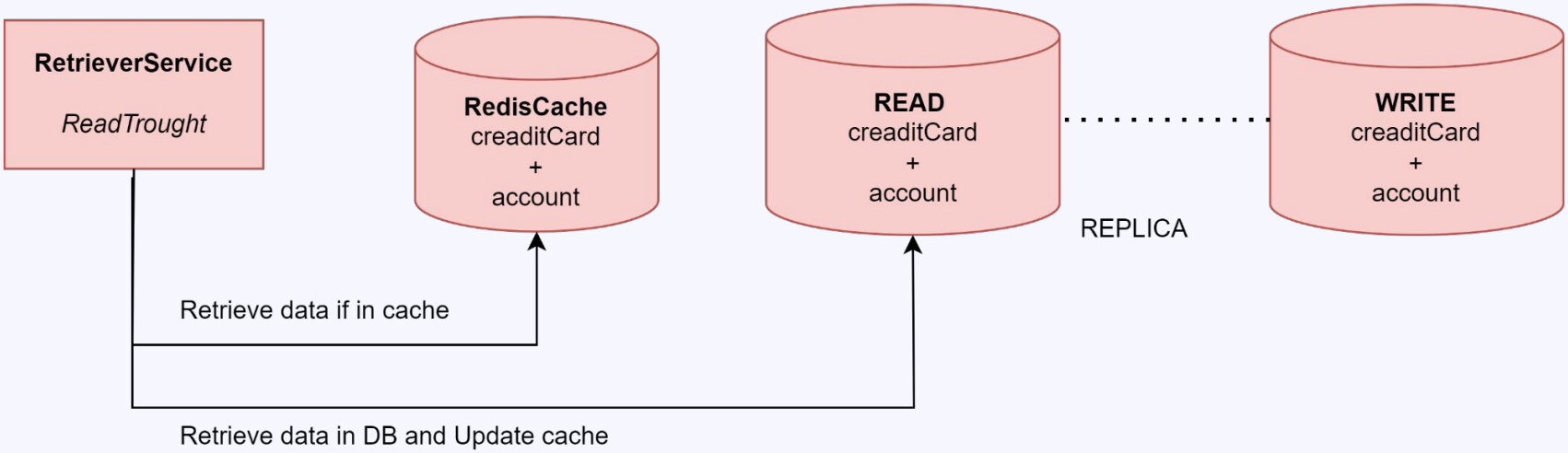


- Mise en place d'une **stratégie de cache**, augmente la rapidité de traitement :
  - **Read-through**
  - **Write-through**
- **Séparation** de la logique de récupération de donnée de celle de validation.

Ancienne architecture exploré

Problème Multiple :

- **Faible Cohérence** des données et ne **répond pas** trop au cas **d'usage**.
- Vitesse de traitement **lente** dans le cas où la donnée n'est pas présente dans le cache
- **Chronophage** dans le contexte d'une **nouvelle configuration cache redis**



Organisation de l’équipe

Igor

Updater service Kafka

Tobias

Logique des services de transaction

Mathieu

Load balancers
Tests de charge

Ayman

Base de données master et slaves

Difficulté rencontrée

- Architecture demandant beaucoup de **configuration**
- **Énormément** de temps passé sur la **conception** plutôt que la **réalisation**
- **Difficulté de réalisation de la Démo sur machine** (test de charge)

Perspectives futures

- Implementation feature restante **“diverse regulations”,** fichier de **permission** YAML ou JSON pour **configuré** par un admin de la banque les permissions de chaque **régions**.
- Finition de l'implémentation du **flow de payement complet** (notification réseau de carte VISA-MasterCard, prise en charge des fichiers de transaction interbancaire)
- Ajout d’un contexte **d’enregistrement** des clients et **d'approvisionnement** du compte
- Implémentation des services avec une autre techno : Spring Web Flux.
- Gestion des données régionales uniquement dans les slaves