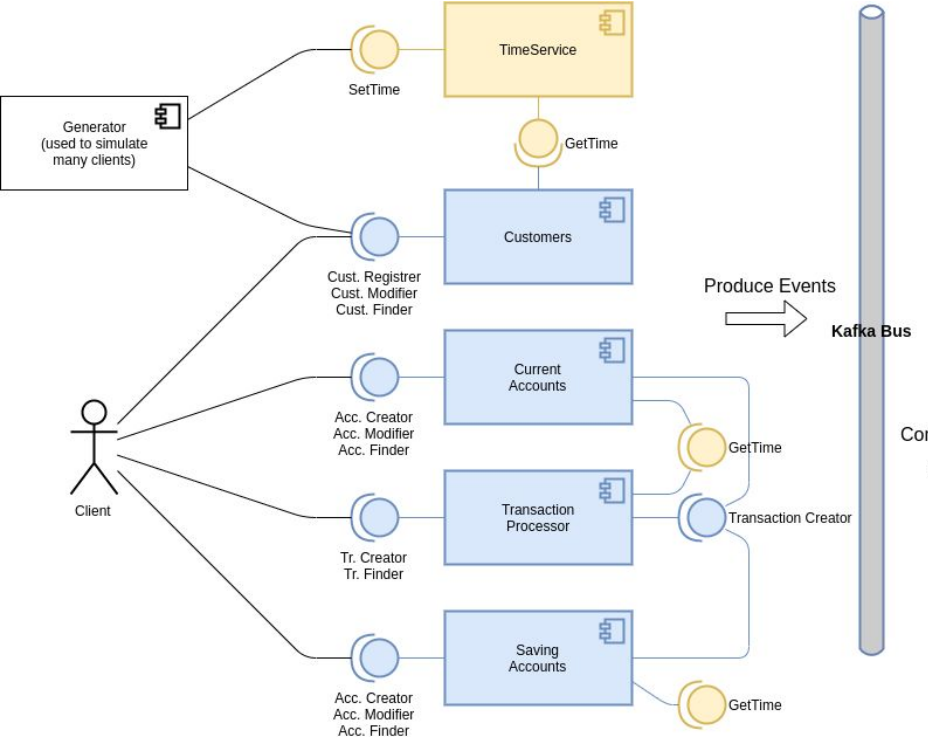
CreditRama : variante Analyste

Période 2 - Equipe A

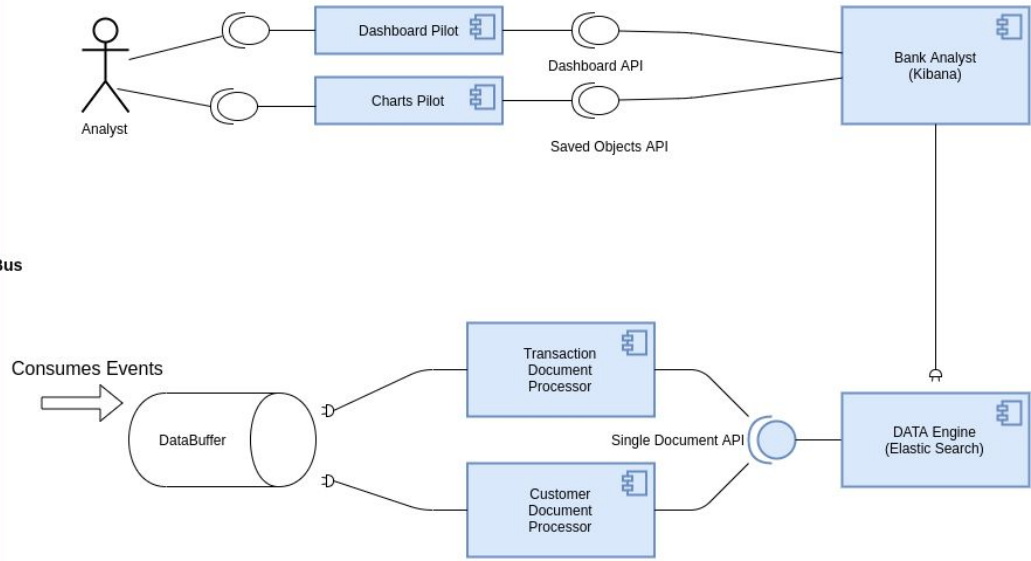
Bolot Alexandre - Artaud Corentin - Larabi Walid - Lazrak Sami

Architecture - 1ére période

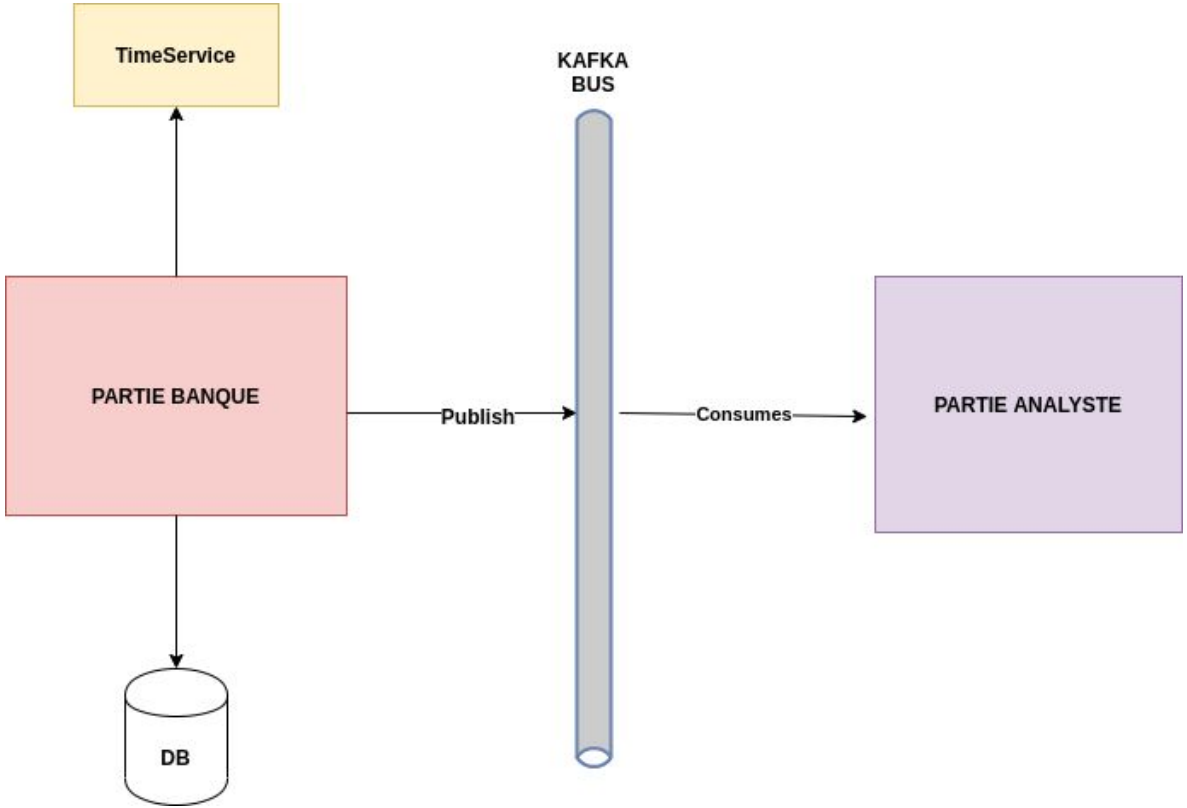
Partie banque



Partie Analyste



Architecture - 1ére période



Rappel des défis de la seconde période

- Composants dans plusieurs timezones
- Time service crash (et repart en arrière)
- La BD crash, on perd des données, on veut reconstituer d'après ce qu'on a dans le bus Kafka

Panne de base de données

La BD crash, on perd des données, on veut reconstituer d’après ce qu’on a dans le bus Kafka

Challenges pour la DB failover

Pas de perte de données

Pas d’interruption de service

(+ Avoir une seule source de vérité)

(+ Lecture des données de Kafka)

Gérer le failover

Imaginons une database qui tombe en panne

```
rt-workspace:~$ docker-compose stop mysql1
```

peut faire l'affaire

On perd la connectivité vers la base de donnée..

Et notre banque ne pourra plus fonctionner

Gérer le faile

Gérer le failover

la **logique métier** doit intervenir
et une **décision** doit être prise
de préférence ***automatiquement***

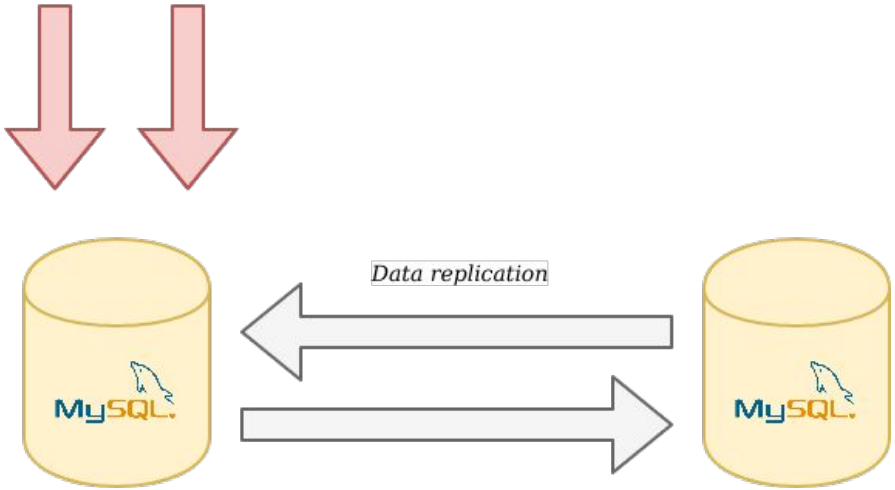
Gérer le failover

Avoir une réplication de données

- Pour ne pas perdre les données
- par une synchro rapide pour une banque
- Mais ça génère beaucoup de traffic.

Gérer le failover

Avoir une réplication de données



Gérer le failover

Rediriger le trafic sur un nouveau noeud

Plusieurs possibilités :

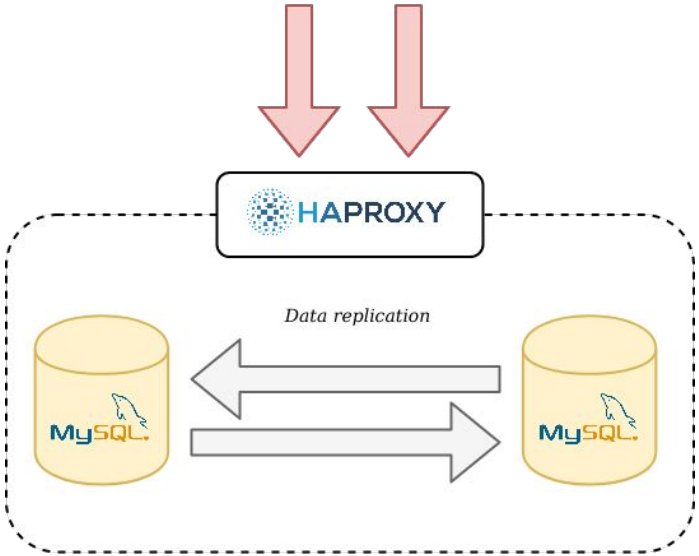
- DNS, Service Discovery, Proxies

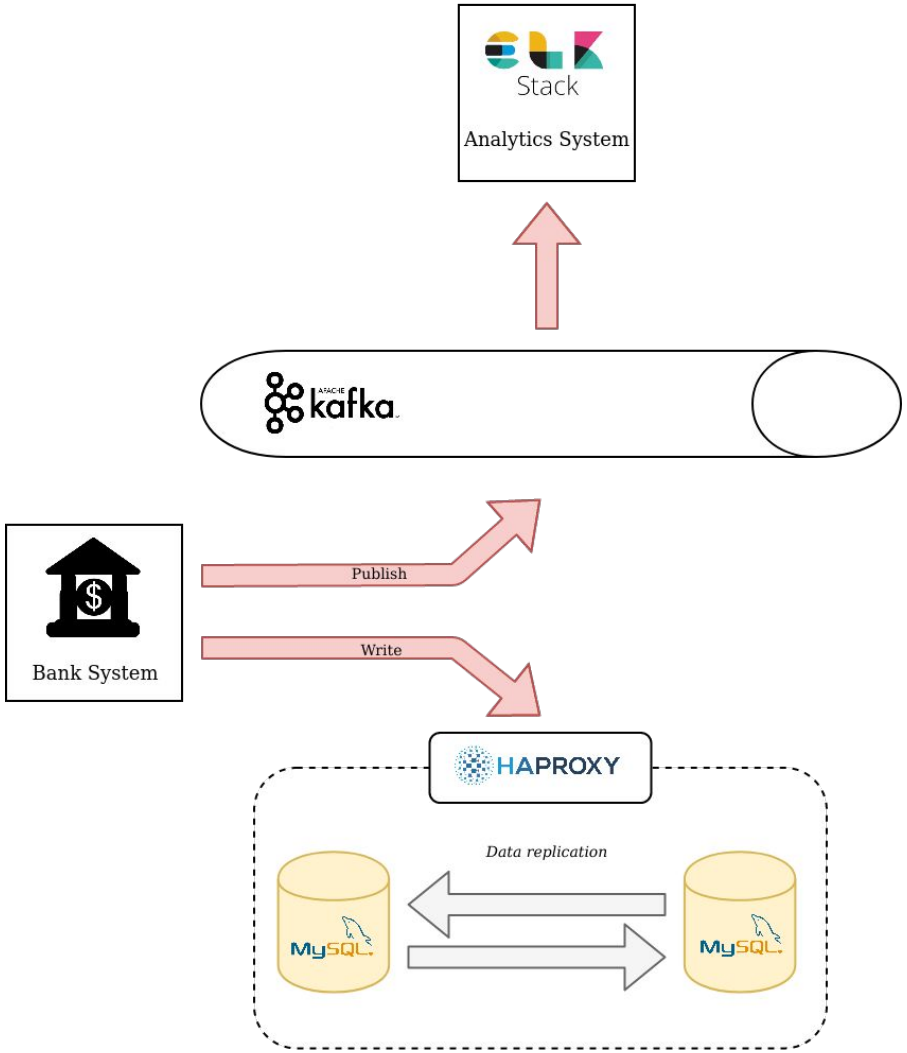
Notre choix :

- Proxy ()

Gérer le failover

Rediriger le trafic sur un nouveau noeud

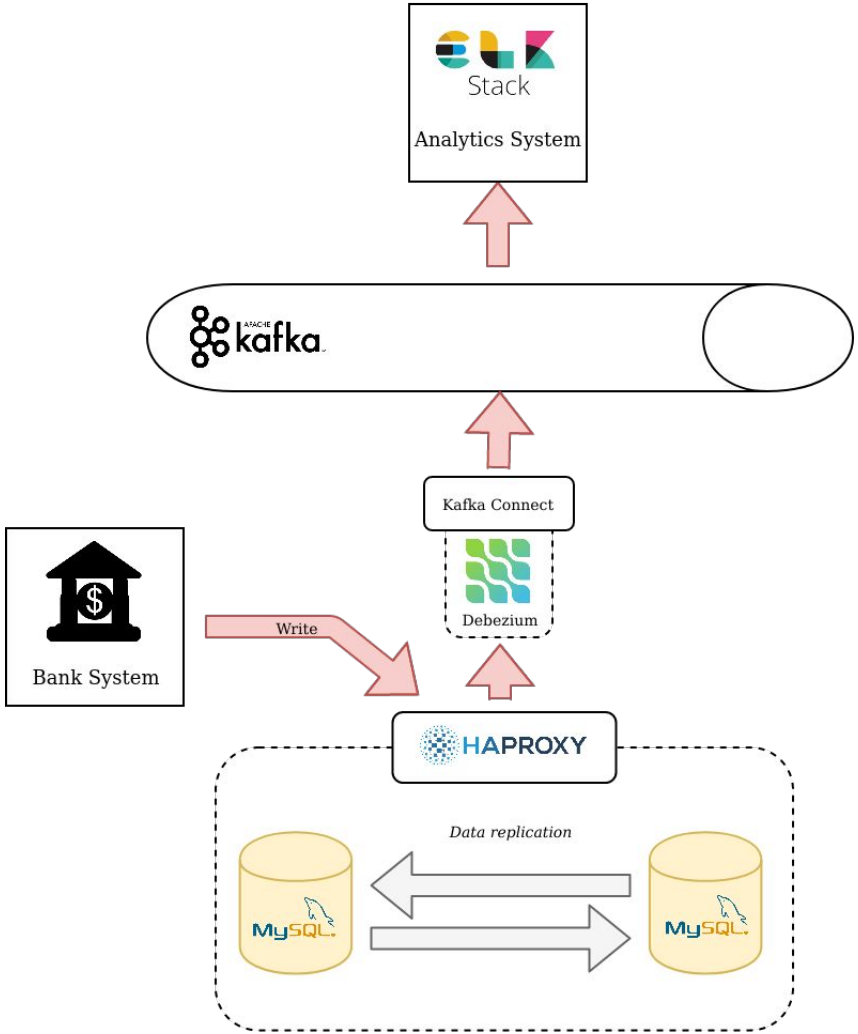




1ère solution

Réplication et load balancing

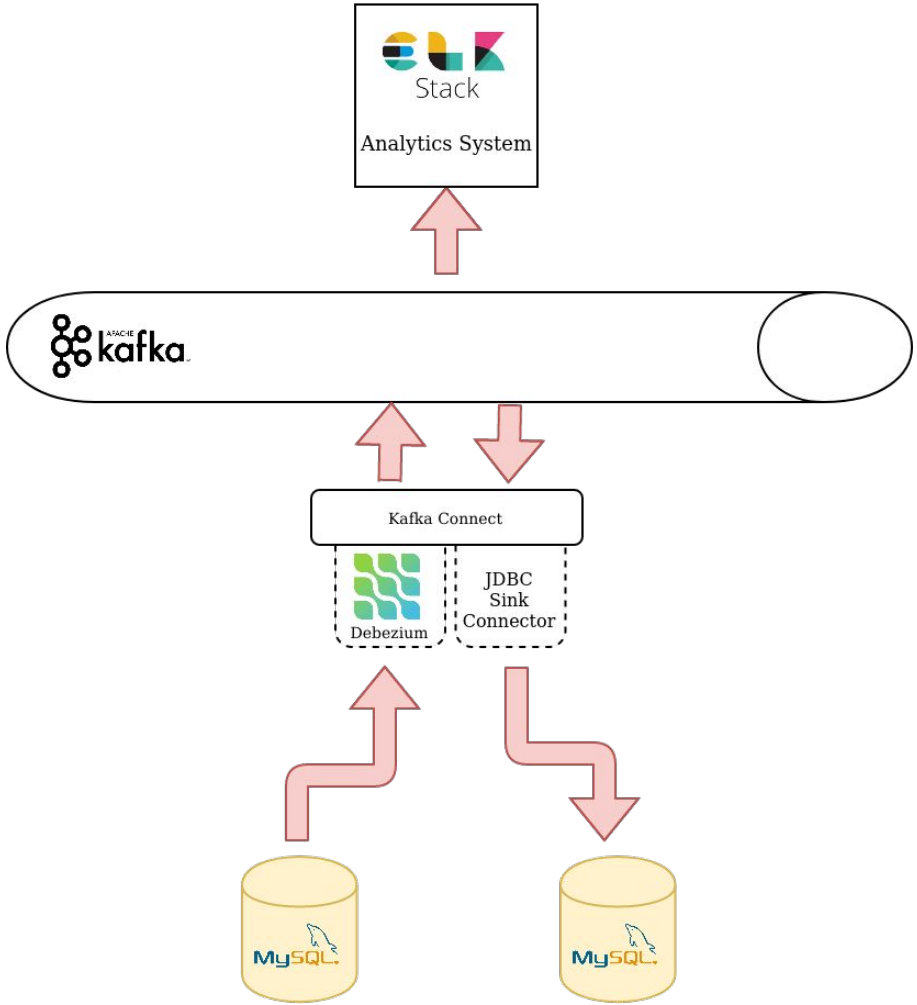
- ✓ Pas de perte de données
- ✓ Continuité de service
- ✗ Une seule source de vérité
- ✗ Lire de Kafka



2eme solution

Intégration du CDC

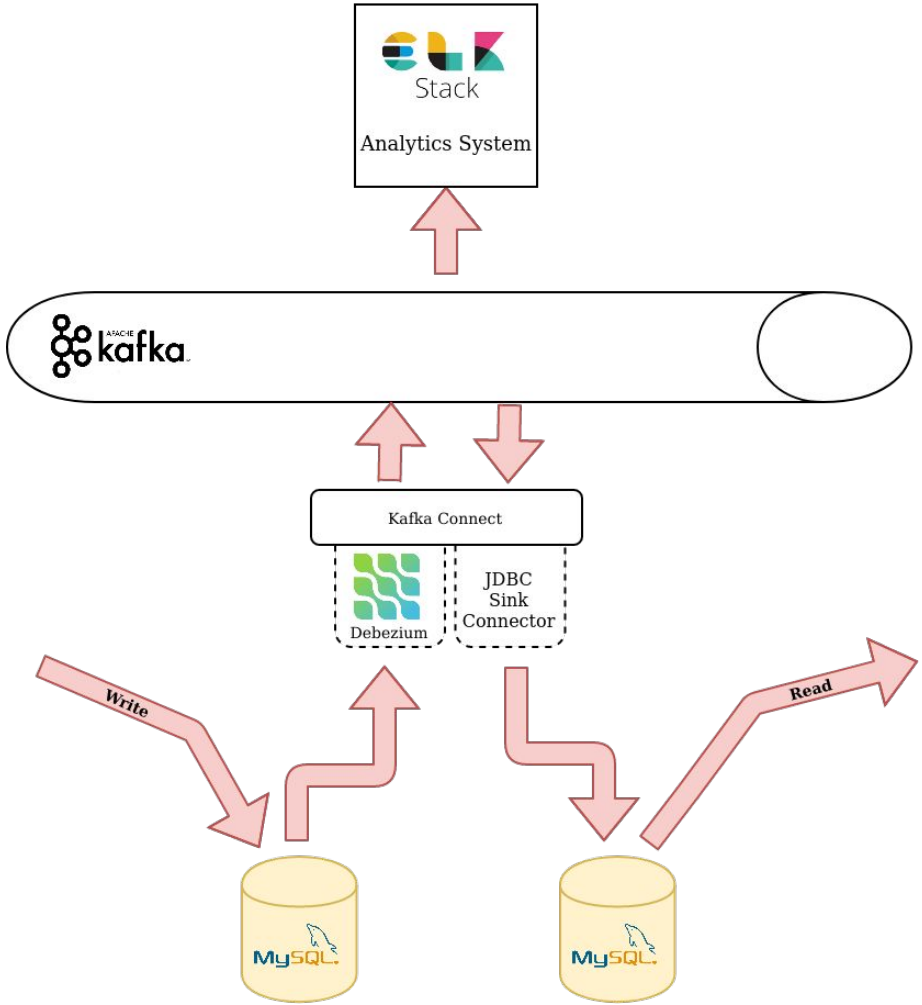
- ✓ Pas de perte de données
- ✓ Continuité de service
- ✓ Une seule source de vérité
- ✗ Lire de Kafka



3eme solution

Intégration du CQRS

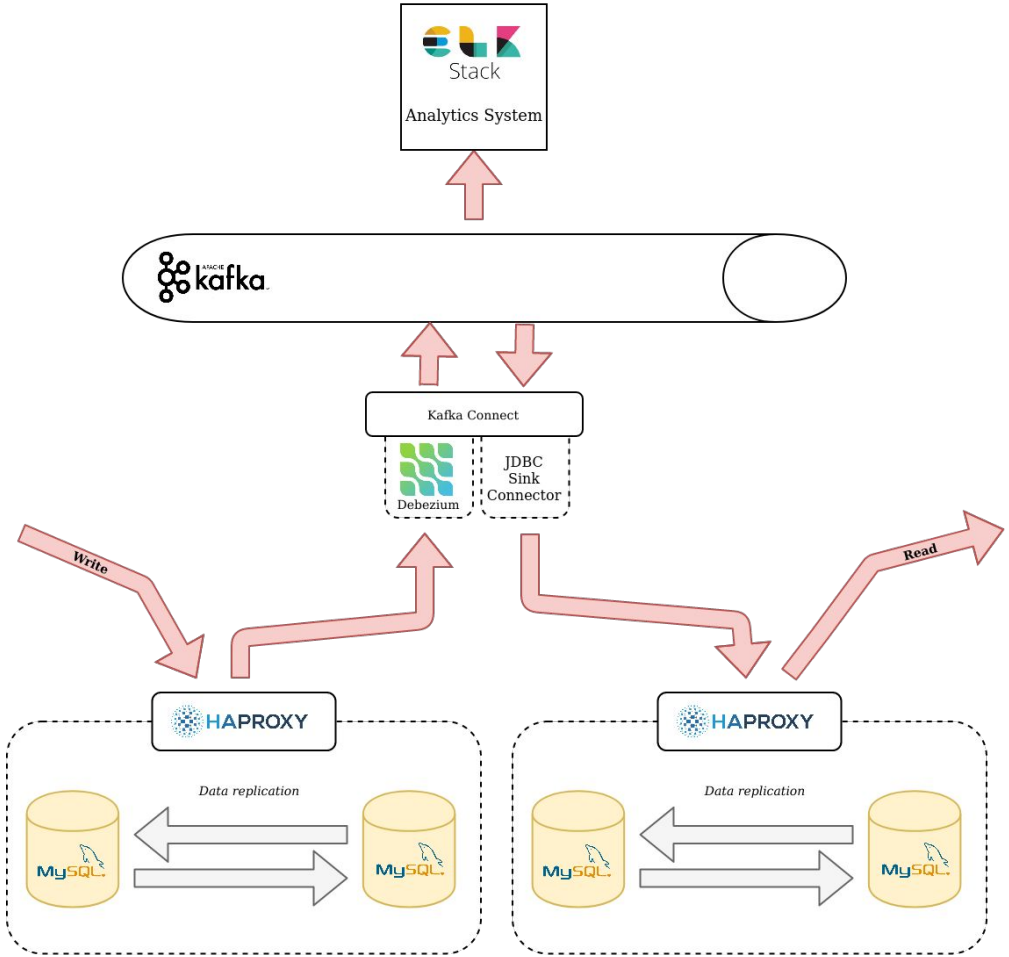
- ✓ Pas de perte de données
- ✗ Continuité de service
- ✓ Une seule source de vérité
- ✓ Lire de Kafka



3eme solution

Intégration du CQRS

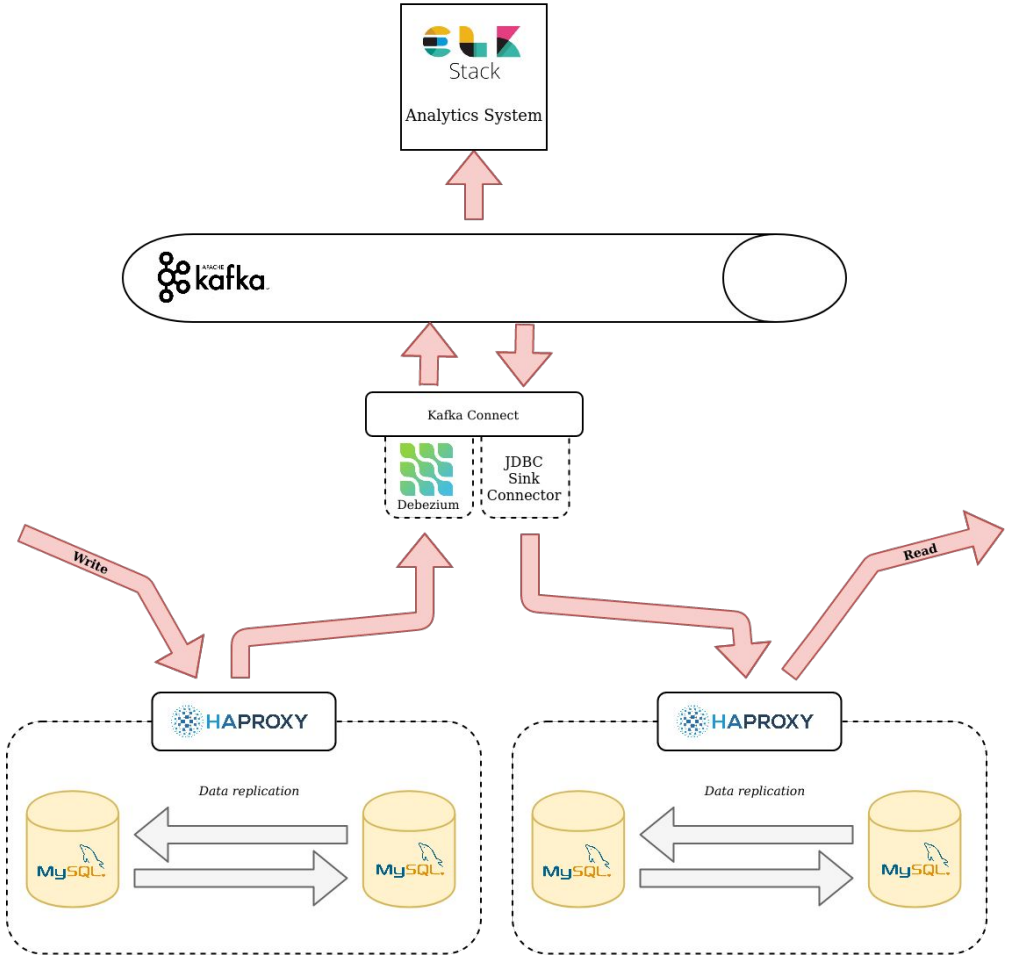
- ✓ Pas de perte de données
- ✗ Continuité de service
- ✓ Une seule source de vérité
- ✓ Lire de Kafka



Solution finale

Une solution hybride

- Pas de perte de données
- Continuité de service
- Une seule source de vérité
- Lire de Kafka



Solution finale

Une solution hybride

- Pas de perte de données
- Continuité de service
- Une seule source de vérité
- Lire de Kafka

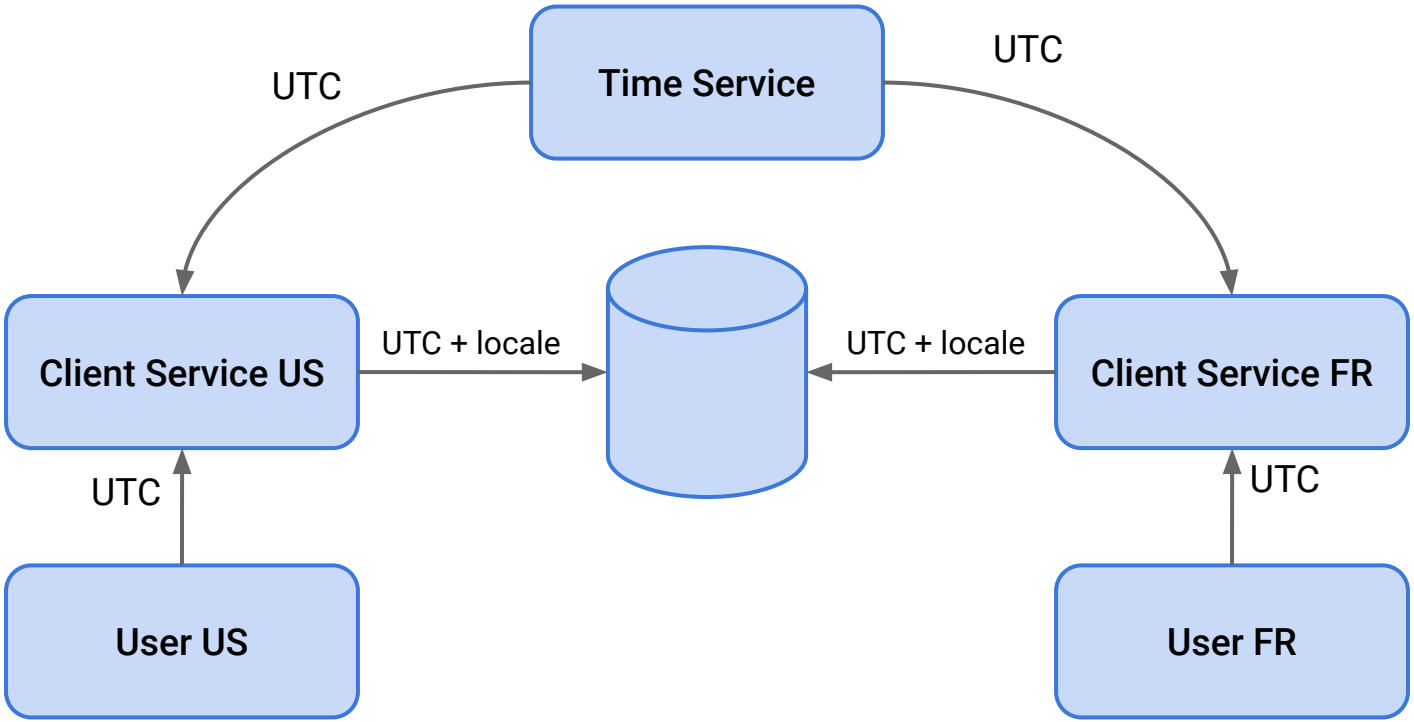
\+ ELASTICITÉ

Problématiques de gestion de temps

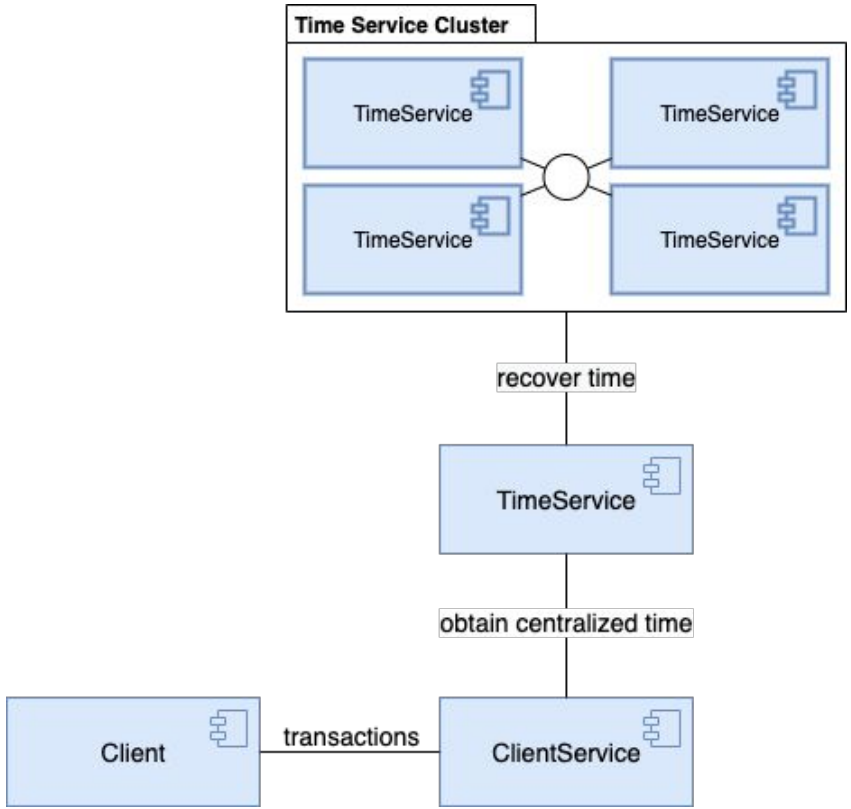
Composants dans plusieurs timezones

Time service crash (et repart en arrière)

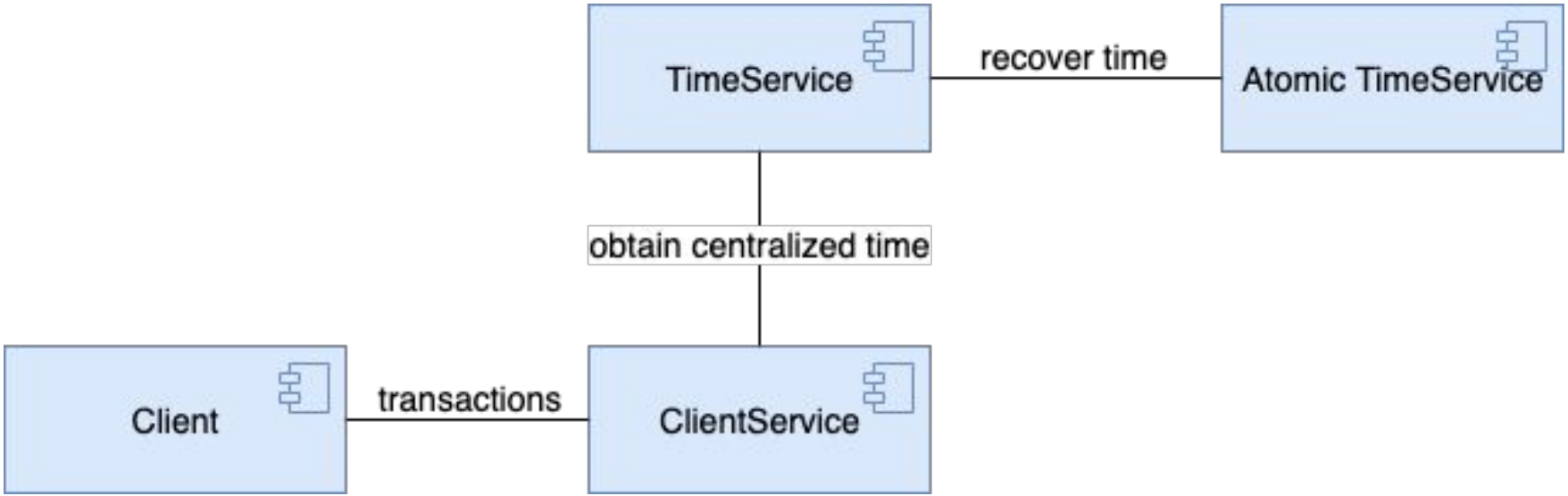
Solution pour Time Zone



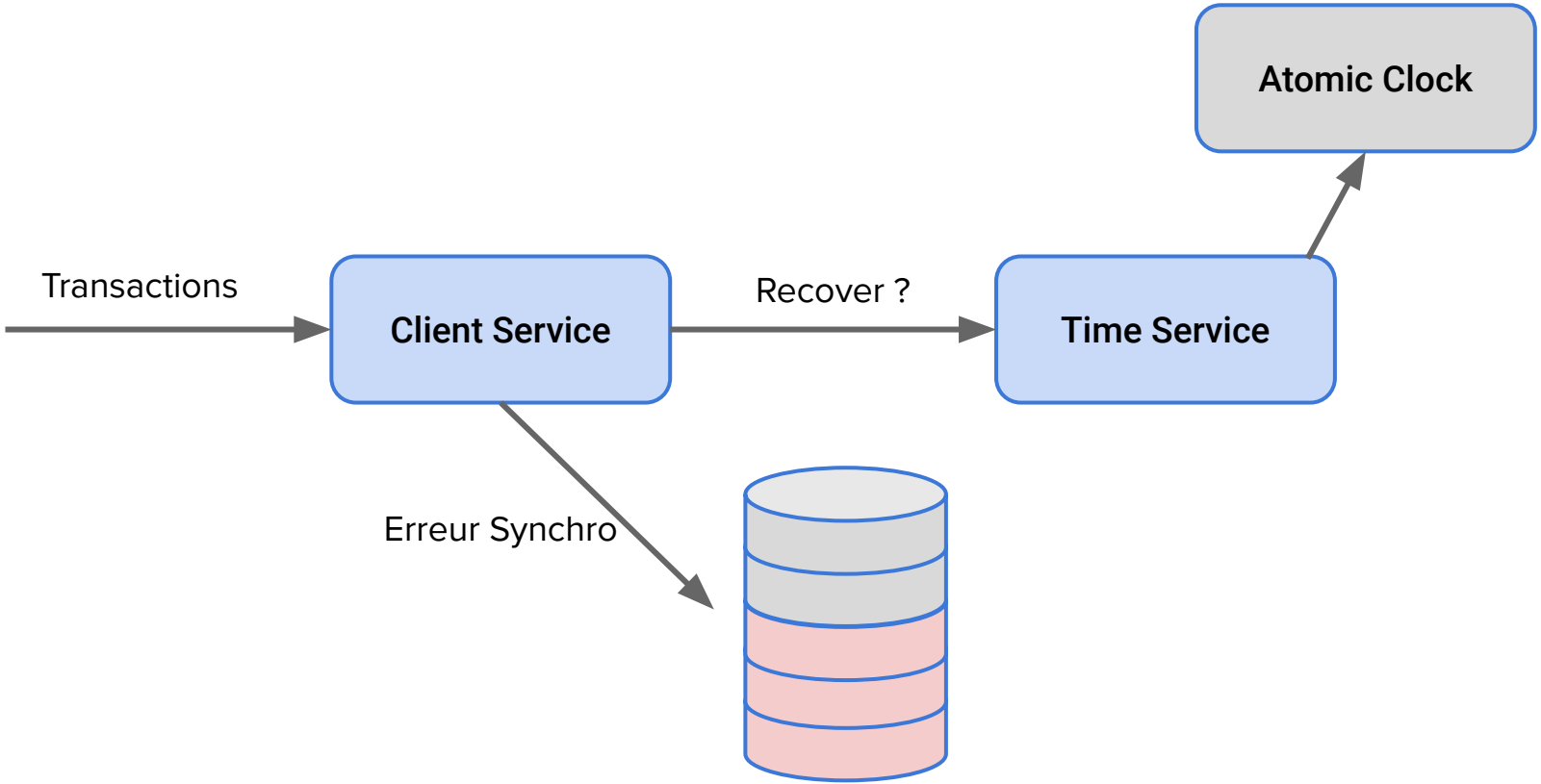
Solution envisagée



Solution retenue



Demo


Atomic Clock
Transactions
Client Service
Recover ?
Time Service
Erreur Synchro

Conclusion

Solution trouvée de manière itérative

Problématiques réelles dans les grandes entreprises

Toujours penser à mettre en place un système de failover (BD ou temps..)

Aller plus loin avec un système de diagnostic de corruption de BD (...)

Système de temps difficile

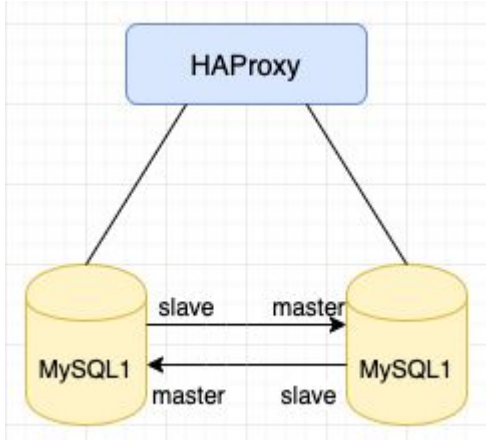
.....

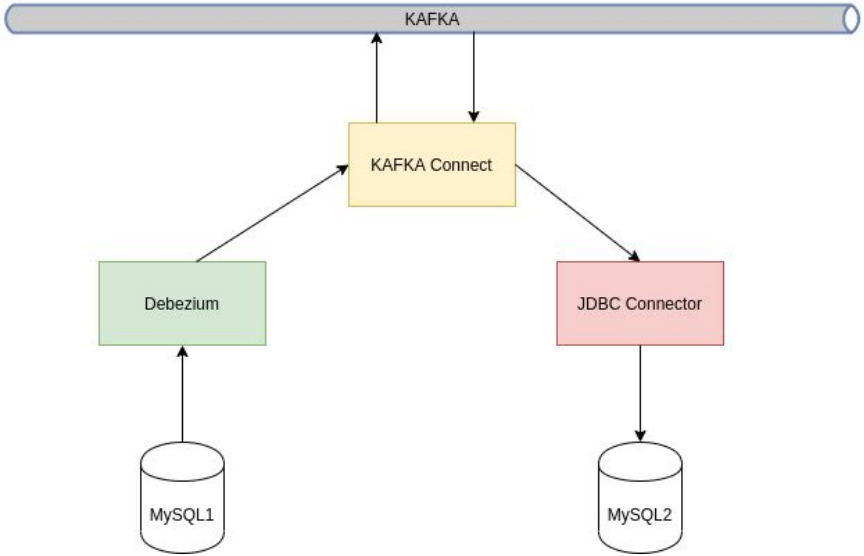
Tests : Database Crash

Protocole :

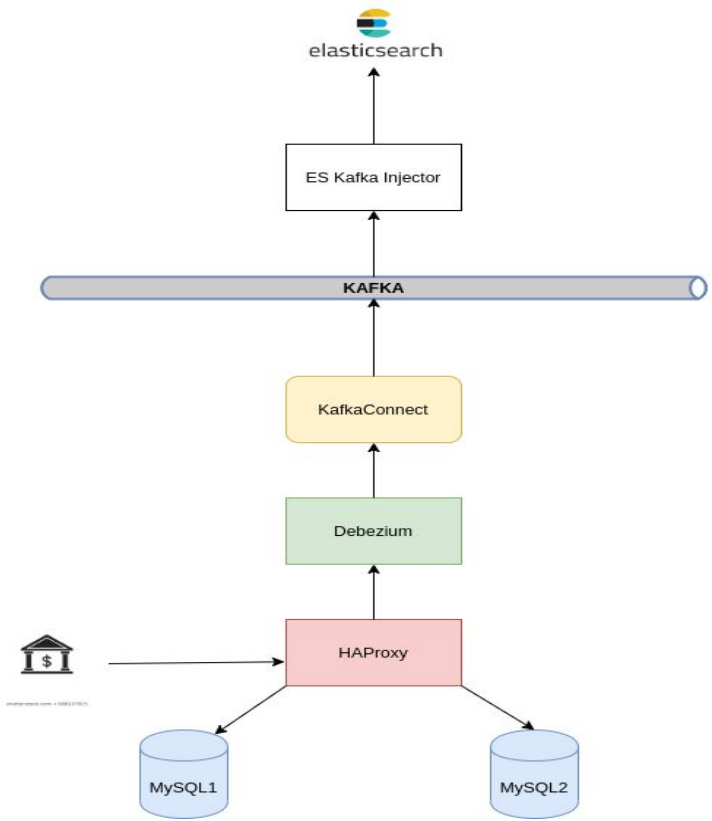
- Mettre en place toute l’architecture
- MySQL1 -> Crash
- Assurer la continuité de service en injectant des données
- Restart MySQL1
- S’assurer que les données ajoutées soient sur MySQL1

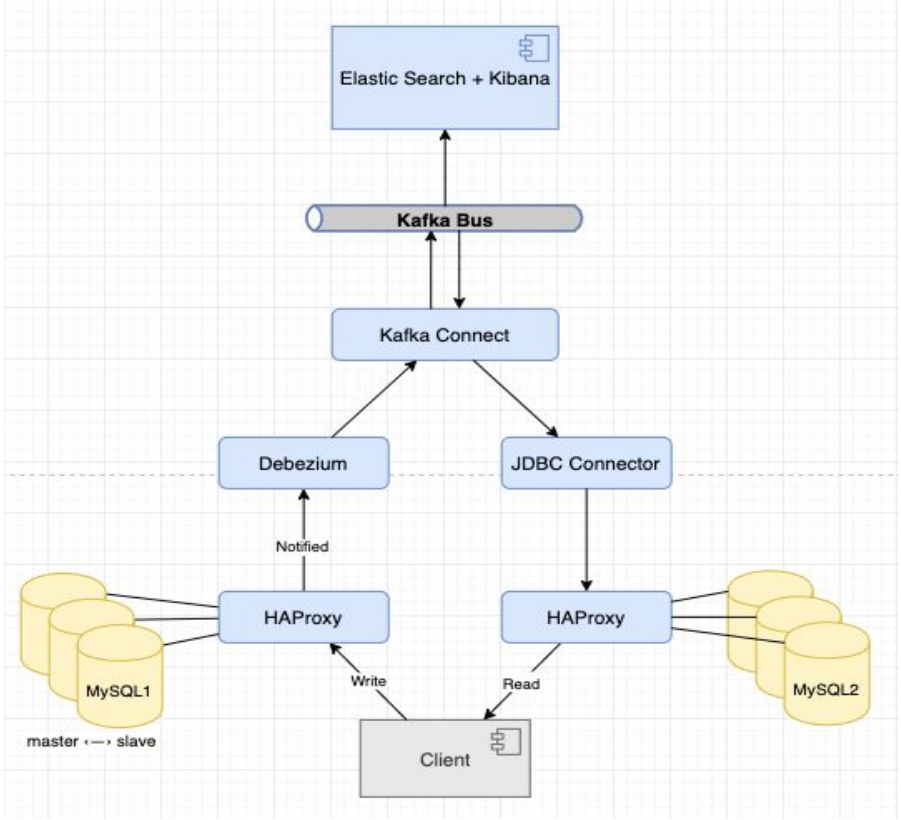
Hmmm...


HAProxy
slave
master
MySQL1
MySQL1
master
slave

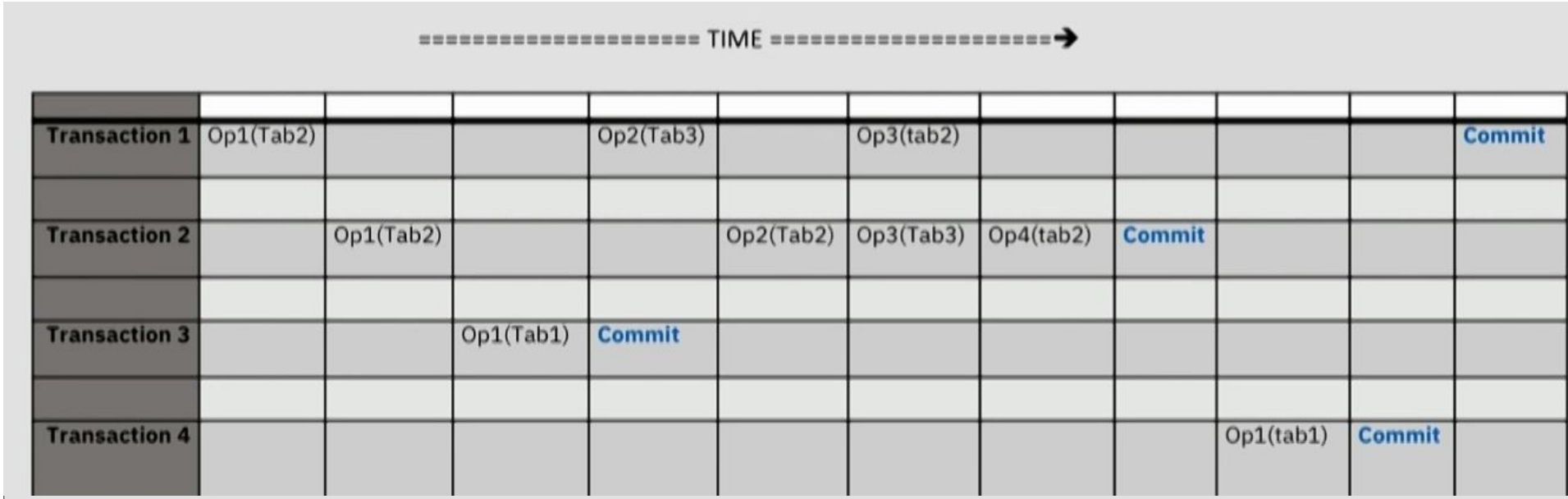

KAFKA
KAFKA Connect
Debezium
JDBC Connector
MySQL1
MySQL2

Comparaison des solutions





Transaction stream in practice

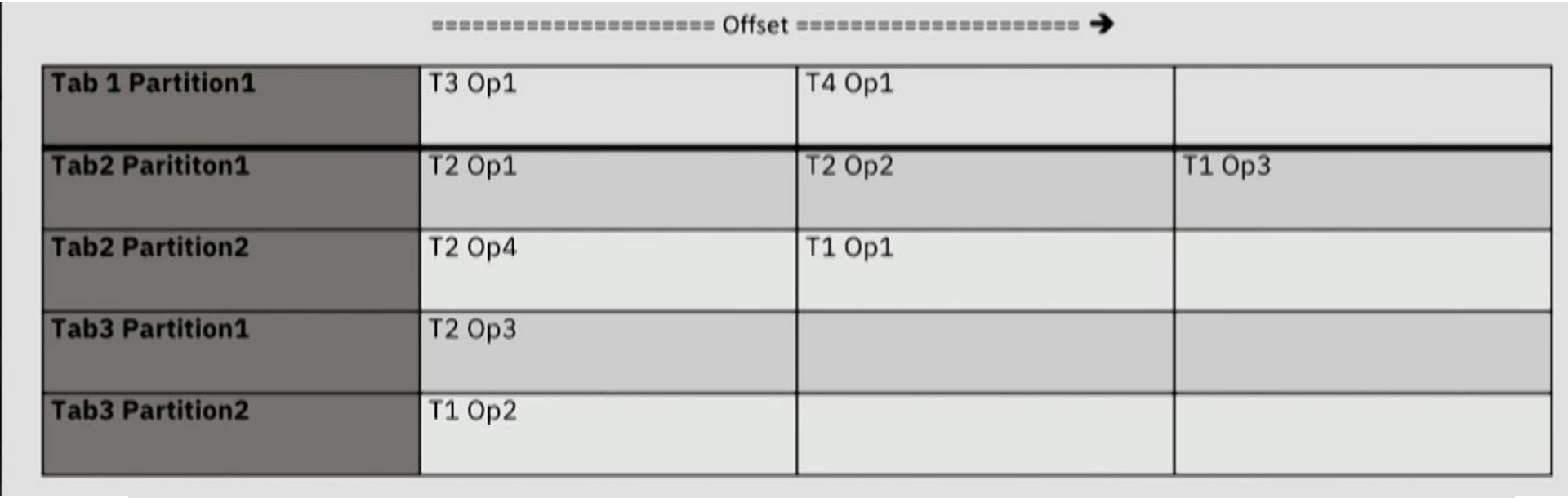


IIDR Kafka Target <========= T3 Op1, T2 Op1, T2 Op2, T2 Op3, T2 Op4, T4 Op1, T1 Op1, T1 Op2, T1 Op3

(send)

Kafka multiple partitions

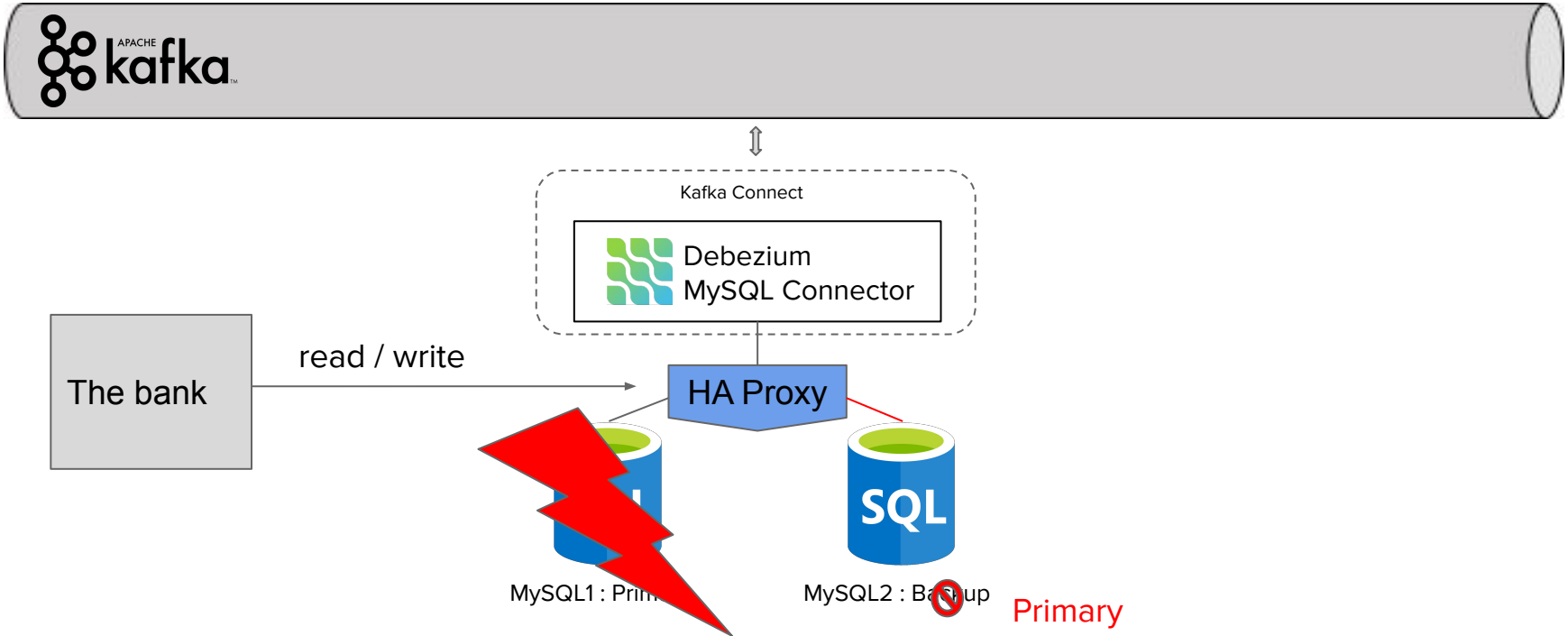
Kafka only provides a total order over messages within a partition, not between different partitions in a topic



Kafka Consumer Application <======== T3 Op1 , T2 Op1, T2 Op2 , T2 Op3, T2 Op4 , T4 Op1, T1 Op1, T1 Op2, T1 Op3

Solutions potentielles?

Solution 1 : Une solution de duplication sur une autre instance



Solutions potentielles?

Solution 1 : Une solution de duplication sur une autre instance

Pour répondre au problème, on peut avoir ce déploiement:

- Base de données
  - Une instance “MySQL 1” (configuré comme slave au “MySQL 2”) avec le GTID activé
  - Une instance “MySQL 2” (configuré comme slave au “MySQL 1”) avec le GTID activé
  - Une instance HAProxy - “MySQL 1” est configuré comme serveur primaire, “MySQL 2” comme backup
- Système de streaming
  - Apache ZooKeeper
  - Apache Kafka broker
  - Apache Kafka Connect avec Debezium MySQL Connector (le connector se connecte au HAProxy)

Pour aller plus loin ...

- Permettre de remettre l'offset de Kafka en arrière
- Performance/Vitesse de la récupération des données à grande échelle
- Kafka cluster/node failure ?
- Détection de pertes de données ? Comment détecter une perte partielle ou corruption des données ?