

Document d'architecture

CréditRama

Nom de l'équipe

Team A

Membres

Walid LARABI

Sami LAZRAK

Corentin ARTAUD

Alexandre BOLOT

Liens vers le wiki contenant toutes ces informations github.com/Corentin-Luc-Artaud/credirama/wiki

Table de matières

Table de matières	2
1. Scope du projet	3
2. Personas	3
3. Fonctionnalités proposés	3
3.1 Fonctionnalités pour le client	3
3.2 Fonctionnalités pour le banquier	4
Fonctionnalités pour l'analyste	4
4. User Stories	5
5. Scénarios	9
6. Deuxième période : Changements apportés au projet	10
6.1 L'heure du Time service va se dérégler aléatoirement	10
6.2 Différentes TimeZones à travers le système	10
6.3 DB Crashes sometimes with data loss	10
7. Diagramme de composants :	11
8. Plan	12
8.1 MVP	12
8.2 Rendu final	12
8.3 Backlog	12

1. Scope du projet

CréditRama est une banque en ligne où des clients créent un profil avec un compte courant et un/des comptes épargnes.

2. Personas

Les 3 persona de notre projet sont :

- **Le client** qui s'inscrit à la banque en ligne et l'utilise dans son quotidien
- **Le banquier** qui gère ses clients
- **L'analyste** qui analyse les données et propose des solutions ou optimisations.

3. Fonctionnalités proposés

3.1 Fonctionnalités pour le client

o Créer son profil (nom, prénom, mail, adresse)

- ça crée automatiquement un compte courant

o Consulter la liste de mes comptes

o Créer un nouveau compte (épargne)

- on peut avoir un compte courant et un ou plusieurs compte d'épargne

o Consulter le détail de mon compte

- historique de transactions (labels et montant et si c'est une entrée ou sortie)

o Faire un virement

- vers un compte avec son identifiant unique (rib) ça peut être mon compte ou le compte d'une autre personne, inter et intra banque

o Clôturer mon compte

- En donnant un rib ou on souhaite verser les fonds restants.
- Un compte en négatif ne peut être cloturer.

3.2 Fonctionnalités pour le banquier

o Consulter la liste de tous les profils clients de la banque

o Consulter les détails d'un profil client

- nom prénom adresse,
- comptes liés au profil

o Consulter les détails d'un compte client

- voir l'historique de ses transactions
- voir le solde d'un compte

o Avoir un contrôle sur sa banque

- Clôturer un compte
- Clôturer un profil

Fonctionnalités pour l'analyste

o Visualiser des données sur l'état globale de la banque

- Évolution de la somme d'argent détenu dans la banque pour une période donnée
- Evolution des entrées / sorties d'argent en fonction du temps
- Somme totale d'argent dans la banque
- Pourcentage des comptes en découvert (dans le rouge)

o Visualiser des données sur les transactions

- Graphe sur la nature des transactions effectués (retraits ATM, virements, paiement en ligne..)
- Le nom de la banque avec laquelle la transaction est faite
- Nombre de retraits / montants
- Nombre de retrait par localisation (position ATM/Banque propriétaire)
- Visualiser le nombre de transactions réussies/échoués
- Visualiser les raisons d'échec de transactions

o Visualiser des données sur nos clients

- Répartition par l'âge des utilisateurs
- Répartition des clients par ville (en france)

Modification le 08/10/2019 après discussion avec M. Molines et M. Collet : Ajout d'user stories pour piloter l'interface de visualisation

o Piloter l'interface de visualisation

- En ajoutant des graphiques en utilisant un langage dédié
- En créant des dashboards en combinant les graphiques à sa disposition
- En sauvegardant les configurations de dashboards et de graphiques pour les recharger à la demande.

4. User Stories

Code couleur :

- User story livrée au dernier bimestre
- User story prévue pour prochain bimestre
- Backlog utile pour la banque mais “brique grise”

CLIENT POTENTIEL

En tant que **Client potentiel**

Je veux **m’inscrire chez CréditRama**

Afin d'**avoir un profil client**

CLIENT CREDITRAMA

En tant que **Client**

Je veux que **CréditRama crée automatiquement mon compte courant après l’inscription**

Afin d'**avoir mon premier compte disponible à l’inscription**

En tant que **Client**

Je veux **créer un compte d’épargne**

Afin d'**économiser de l’argent**

En tant que **Client**

Je veux **consulter la liste de mes comptes (identifiant du compte et solde)**

Afin de **connaître l’état de mes finances**

En tant que **Client**

Je veux **consulter l’historique de transactions d’un compte**

Afin de **vérifier le détail de mes dépenses**

En tant que **Client**

Je veux **effectuer un virement vers un autre compte bancaire grâce à son identifiant**

Afin de **déplacer de l’argent quand on a besoin**

En tant que **Client**

Je veux **utiliser l’argent de un de mes comptes**

Afin de le **retirer en liquide d’une ATM**

En tant que **Client**

Je veux **ajouter de l’argent sur un de mes comptes**

Afin de **ne pas garder de liquide chez moi**

En tant que **Client**

Je veux **utiliser l'argent de mon compte courant**
Afin d'**effectuer des paiements/transactions divers (en ligne et sur des terminaux)**

En tant que **Client**
Je veux **pouvoir clôturer mon profil**
Afin de **ne plus être Client de CréditRama**

En tant que **Client**
Je veux **pouvoir clôturer mon compte épargne**
Afin de **ne plus l'avoir dans mes comptes**

En tant que **Client**
Je veux **pouvoir imprimer le rib d'un de mes comptes**
Afin de **d'avoir les détails unique de mon compte**

BANQUIER

En tant que **banquier de CréditRama**
Je veux **pouvoir avoir la liste des clients CréditRama**
Afin d'**avoir des informations me permettant d'accéder à leur profil**

En tant que **banquier de CréditRama**
Je veux pouvoir **consulter le profil d'un client CréditRama**
Afin d'**avoir plus d'information sur le client (détails du client et la liste de ses comptes)**

En tant que **banquier de CréditRama**
Je veux pouvoir **consulter le solde de n'importe quel compte**
Afin d'**avoir plus d'information sur son état financier**

En tant que **banquier de CréditRama**
Je veux pouvoir **consulter l'historique de transactions de n'importe quel compte**
Afin de **connaître les dépenses de son client**

En tant que **banquier de CréditRama**
Je veux **pouvoir clôturer un compte**
Afin de **répondre à une requête (client ou autre) me demandant de le clôturer**

En tant que **banquier de CréditRama**
Je veux **pouvoir clôturer un profil client**
Afin de **répondre à une requête (client ou autre) me demandant de le clôturer**

ANALYST

Nous avons définis des epics pour regrouper les user stories (pour une meilleur lisibilité)

EPICS

En tant qu'**Analyste**

Je veux **visualiser des données sur l'état de la banque** (1)

Afin d'**avoir des indicateurs sur les performances de la banque**

En tant qu'**Analyste**

Je veux **visualiser des données sur les transactions**

Afin de **connaître et comprendre le déplacements d'argents** (cashflow)

En tant qu'**Analyste**

Je veux pouvoir **visualiser des données sur nos clients**

Afin d'**adapter au mieux les offres pour les satisfaire et en trouver de nouveaux**

USER STORIES

En tant qu'**Analyste**

Je veux pouvoir visualiser l'évolution de la somme d'argent dans la banque en fct du temps

Afin de voir le progrès que fait la banque

En tant qu'**Analyste**

Je veux pouvoir visualiser les entrées / sorties d'argent en fonction du temps

Afin de surveiller les finances de la banque

En tant qu'**Analyste**

Je veux pouvoir visualiser la somme totale d'argent dans la banque

Afin de communiquer les chiffres au besoin

En tant qu'**Analyste**

Je veux pouvoir visualiser le ratio des comptes en découvert

Afin de surveiller les finances de la banque

--

En tant qu'**Analyste**

Je veux pouvoir visualiser la répartition par l'âge des utilisateurs

Afin de proposer des services / offres aux clients en fonction de leur âge

En tant qu'**Analyste**

Je veux pouvoir visualiser la répartition des clients par ville (en france)

Afin de proposer des services / offres aux clients de ces villes

En tant qu'**Analyste**

Je veux pouvoir visualiser un graphe sur la nature des transactions effectués (retraits ATM, virements, paiement en ligne..)

Afin de voir la répartition des canaux utilisés et les renforcer

En tant qu'**Analyste**

Je veux pouvoir visualiser la répartition des banques avec laquelle les transactions sont faites

Afin d'améliorer nos partenariats avec les autres banques

En tant qu'**Analyste**

Je veux pouvoir visualiser le nombre de retraits par catégorie de montants

Afin de comprendre le comportement de ses clients

En tant qu'**Analyste**

Je veux pouvoir visualiser le nombre de retraits par localisation

Afin de voir où est-ce que les clients sont concentrés

En tant qu'**Analyste**

Je veux pouvoir visualiser le nombre de transactions réussies/échoués

Afin de détecter des anomalies

En tant qu'**Analyste**

Je veux pouvoir visualiser les raisons d'échec de transactions

Afin de voir les anomalies et les corriger

Modification le 08/10/2019 après discussion avec M. Molines et M. Collet : Ajout d'user stories pour piloter l'interface de visualisation

En tant qu'**Analyste**

Je veux utiliser une fonctionnalité pour afficher les dashboards que j'ai sur mon système

Afin de les réutiliser si besoin

En tant qu'**Analyste**

Je veux utiliser une fonctionnalité pour afficher les graphiques que j'ai sur mon système

Afin de les réutiliser si besoin

En tant qu'**Analyste**

Je veux utiliser une fonctionnalité pour créer un graphique avec une syntaxe spéciale

Afin de le visualiser ou l'intégrer à un ou plusieurs dashboards

En tant qu'**Analyste**

Je veux utiliser une fonctionnalité pour créer un dashboard contenant les graphiques que je souhaite
Afin de les regrouper et les visualiser sur l'outil de visualisation

En tant qu'**Analyste**

Je veux pouvoir sauvegarder la configuration d'un dashboard
afin de la re-utiliser pour créer des dashboards automatiquement

En tant qu'**Analyste**

Je veux pouvoir créer un dashboard à partir d'une configuration que je donne
afin de le retrouver sur l'interface de visualisation

En tant qu'**Analyste**

Je veux pouvoir sauvegarder la configuration d'un graphique
afin de le re-utiliser pour créer une visualisation automatiquement

En tant qu'**Analyste**

Je veux pouvoir créer un graphique à partir d'une configuration que je donne
afin de le retrouver sur l'interface de visualisation

5. Scénarios

- Le client potentiel crée un profil sur notre banque, et ça crée automatiquement son compte courant
- Le client va recevoir le numéro de son compte
- Le banquier veut visualiser les informations sur le compte pour voir s'il peut lui proposer une offre
- L'analyste
- Les données sont agrégées dans ES et accessibles par Kibana

Given an analyst

When he creates a chart named "Clients per city" and described using timeml
Then the chart is saved and displayed on the visualization interface.

6. Deuxième période : Changements apportés au projet

6.1 L'heure du Time service va se dérégler aléatoirement

Nous prévoyons d'utiliser un système de Cluster d'instances de Time Services.

Cela signifie que nous utiliserons un Time Service Gateway (pouvant être répliqué au besoin) qui sera accessible publiquement par le reste de notre système. Cette Gateway écouterait les communication broadcast entre les différentes instances du Time Service Cluster. La gateway sera donc notre nouvelle référence de temps.

Pendant ce temps, les différentes instances du Time Service vont fréquemment "voter" via broadcast pour décider quel est le temps correct. Cette décision sera basée sur la majorité des temps identiques, et les instances avec des valeurs en minorité s'adapteront à la décision majoritaire.

Exemple : un cluster de 6 instances de Time Service (ici le temps est représenté par un entier, pour simplifier nous commencerons à 1)

$T1 = [1, 1, 1, 1, 1, 1]$ → tout le monde a le même temps, personne ne s'adapte, et on est certain que le temps exacte vaut 1.

Tout se passe bien jusqu'à T6

$T6 = [6, 6, 4, 6, 6, 6]$ → ici l'instance n° 3 a pour temps la valeur 4, lors du vote elle va se rendre compte qu'elle est minoritaire et va prendre le temps de la majorité soit 6.

Imaginons maintenant qu'il n'y ait pas 1 mais 4 instances en fautes.

$T6 = [6, 6, 4, 3, 2, 1]$ → le temps majoritaire est toujours 6.

En cas d'égalité lors du vote, on considère que le temps majoritaire le plus haut est le bon.

$T7 = [7, 7, 5, 5, 8, 3]$ → le temps élu est le plus grand majoritaire soit 7

6.2 Différentes TimeZones à travers le système

Notre Time Service Gateway fournit une source unique de temps absolu UTC. De cette façon nous évitons de devoir gérer plusieurs représentation du temps dans la partie du système nous permettant de garantir la cohérence du temps fourni (plus simple car cela évite les conversions). Il incombe donc aux différents partis utilisant ce temps de le convertir si besoin est.

6.3 Pertes de données lorsque la base de donnée crash

Nous allons créer un scénario où notre base de données crash et perd une partie ou la totalité de ses données. Il faut récupérer cette perte de donnée, et pour se faire il faut définir une architecture et une stratégie pour le faire.

Après plusieurs recherches, l'événement sourcing et le CQRS sont deux patterns utilisés souvent ensemble et qui peuvent répondre à notre problématique. En effet, il y a plusieurs avantages à utiliser ces patterns.

ça fournit un journal complet de chaque changement d'état apporté, sous forme de log; donc, il serait facile de reconstruire l'état d'un objet en suivant les logs.

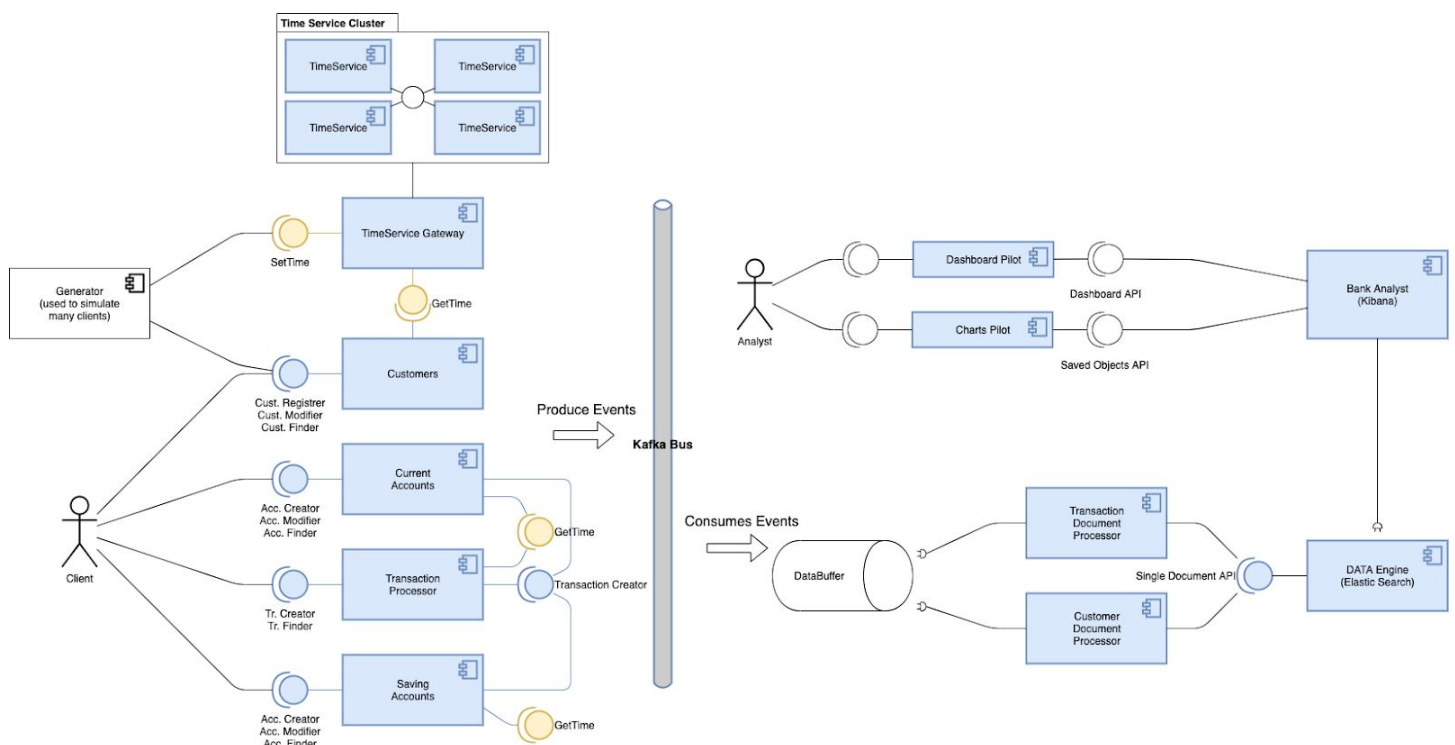
Notre première intuition pour l'Event Sourcing c'est d'utiliser le Change Data Capture (CDC) sur notre base de donnée ou l'on écrit. qui signifie qu'à chaque changement dans la base de donnée, un événement sera généré et sera envoyé sur le bus Kafka. Grâce à nos recherches, Debezium est la technologie qui est adapté à ça. Ensuite, on aura tout les composants qui peuvent écouter sur le topic ou on écrit les changements et on peut écrire (monitoring, recovery database par exemple, etc). L'avantage c'est qu'on aura une seule source de vérité.

Ensuite, d'un autre côté, on peut créer une application (event handler) qui post-process les événements CDC et crée une vue accessible en lecture seule pour lire les données (via une bd ou un cache) et comme ça on applique le pattern CQRS (en séparant l'écriture et la lecture)

Une base de donnée de secours ou de stockage froid peut être créé.

Grâce à des technologies comme JDBC Sink Connector, on s'abonne à un topic kafka ou on envoie les événements du CDC et on remplit la base de donnée de secours. Ensuite, si un jour nous avons un problème sur la base de donnée principale, on aura l'archive sur la deuxième base de donnée. On peut indexer la base de donnée par collections (par jour ou autre) et si un index est impacté, on ira directement chercher les données de secours sur le même index de la base de données de secours par exemple.

7. Diagramme de composants :



Le schéma d'architecture a été partiellement mis à jour.

Il comprend les modifications apportées pour la gestion des défaillances de temps et des différences de TimeZone.

En revanche, la modification concernant la perte des données dans la base de données n'est pas encore présente étant donné que nous étudions plusieurs approches et souhaitons prendre le temps de choisir celle qui nous semble la meilleure.

8. Plan

8.1 MVP

Lien vers la page du Milestone Github : [milestone 1](#)

- [M'inscrire chez CréditRama](#)
- [Création de compte à l'inscription](#)
- [Utiliser l'argent de un de mes comptes](#)
- [Ajouter de l'argent sur un de mes comptes](#)
- [Avoir la liste des clients CréditRama](#)
- [Visualiser l'évolution de la somme d'argent dans la banque en fonction du temps](#)
- [Visualiser la somme totale d'argent dans la banque](#)
- [Visualiser le nombre de retraits par catégorie de montants](#)

8.2 Rendu final

Lien vers la page du Milestone Github : [milestone 2](#)

- [Consulter la liste de mes comptes](#)
- [Consulter l'historique de transactions d'un compte](#)
- [Effectuer un virement vers un autre compte](#)

8.3 Backlog

Lien vers la page du Milestone Github : [backlog](#)