

Document d'architecture

NewBank - V8 Cashback

Team A

(comme argent)

Antoine BUQUET

-

Benoit GAUDET

-

Ayoub IMAMI

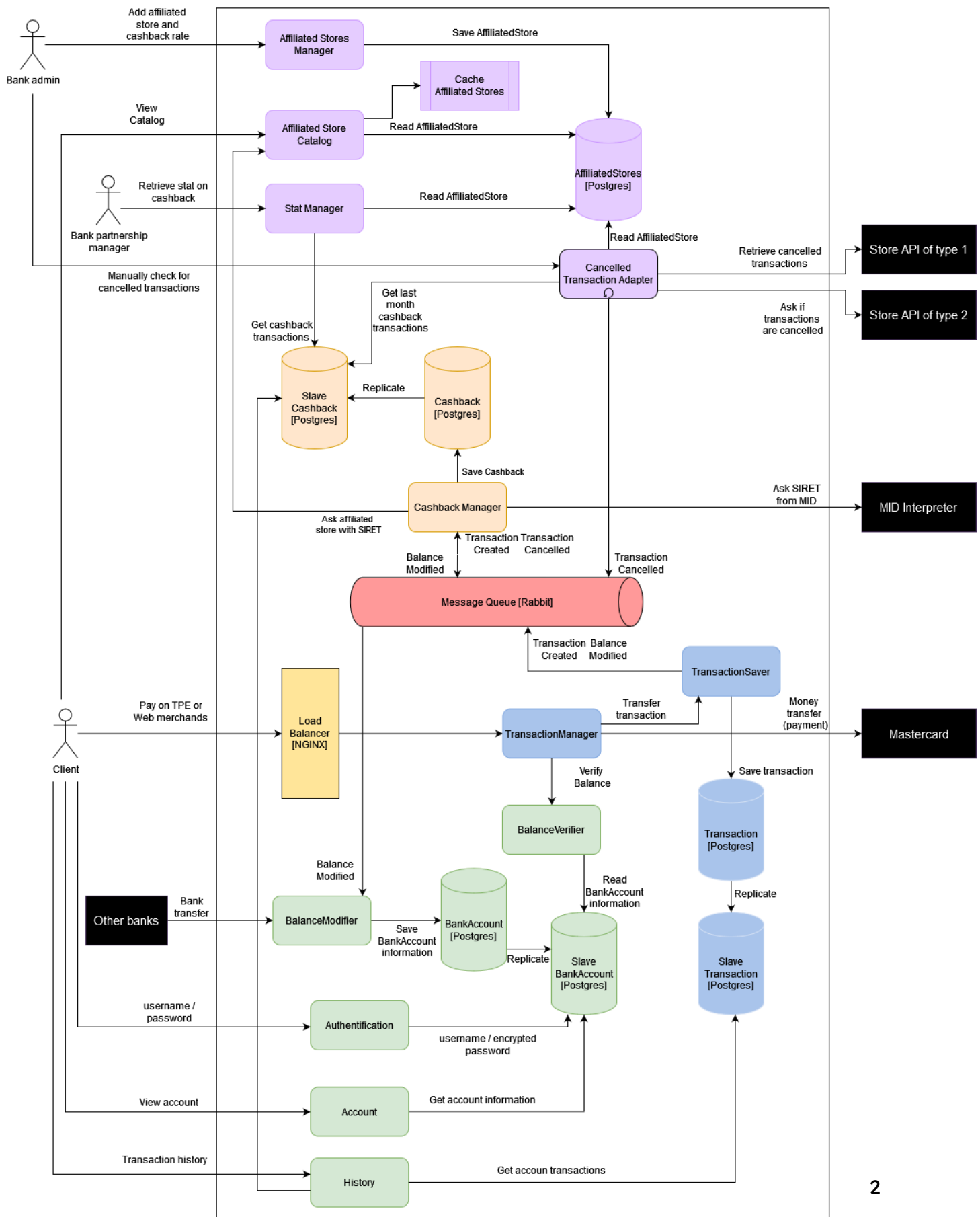
-

Mourad KARRAKCHOU

I. État de l'architecture actuel	2
1. Diagramme d'architecture	2
2. Justifications	3
3. Fonctionnalités couvertes par notre architecture	4
II. Historique de l'architecture	5
I. 13 novembre 2023	5
II. 20 octobre 2023	6
III. 1 octobre 2023	7

I. État de l'architecture actuel

1. Diagramme d'architecture



Architecture composée de quatre services :

- Transaction Service
- Cashback Service
- Account Service
- Affiliated Store Service

2. Justifications

L'architecture de notre banque est basée sur une architecture en services. Le premier service, représenté en bleu dans le diagramme, aura pour objectif de gérer la partie des transactions. Le deuxième service, représenté en orange dans le diagramme, va quant à lui gérer la partie sur le cashback. Le service violet s'occupe des magasins partenaire ainsi que la connexion avec leurs services. Enfin, en vert est représenté le service des comptes pour prendre en charge les comptes utilisateurs et bancaire. Ce découpage en plusieurs services permet à la banque d'être évolutive, résiliente et mise à l'échelle.

Nous avons choisi d'implémenter notre application de cette manière en séparant le côté "transaction" de la partie cashback, car nous estimons qu'il s'agit simplement d'une extension supplémentaire venant se greffer au fonctionnement d'une banque normale.

De plus, il y a quatre bases de données dans l'application, une par service, l'une pour permettre le stockage des données sur les magasins affiliés, avec leur offre de cashback, une autre pour le stockage des utilisateurs avec leur solde. Une troisième pour sauvegarder toutes les transactions effectuées et enfin une dernière pour les cashback donné par la banque. Cette séparation permet d'isoler les données métier relatives à leurs services et évite par ailleurs une centralisation du stockage qui pourrait être problématique si l'unique base de données venait à cesser de fonctionner. Dans le cas présent, si la base de données du service de cashback n'était plus accessible, alors l'utilisateur pourrait tout de même effectuer une transaction. Nous avons également choisi de répliquer les base de données les plus sensibles afin d'assurer une meilleure résilience.

3. Fonctionnalités couvertes par notre architecture

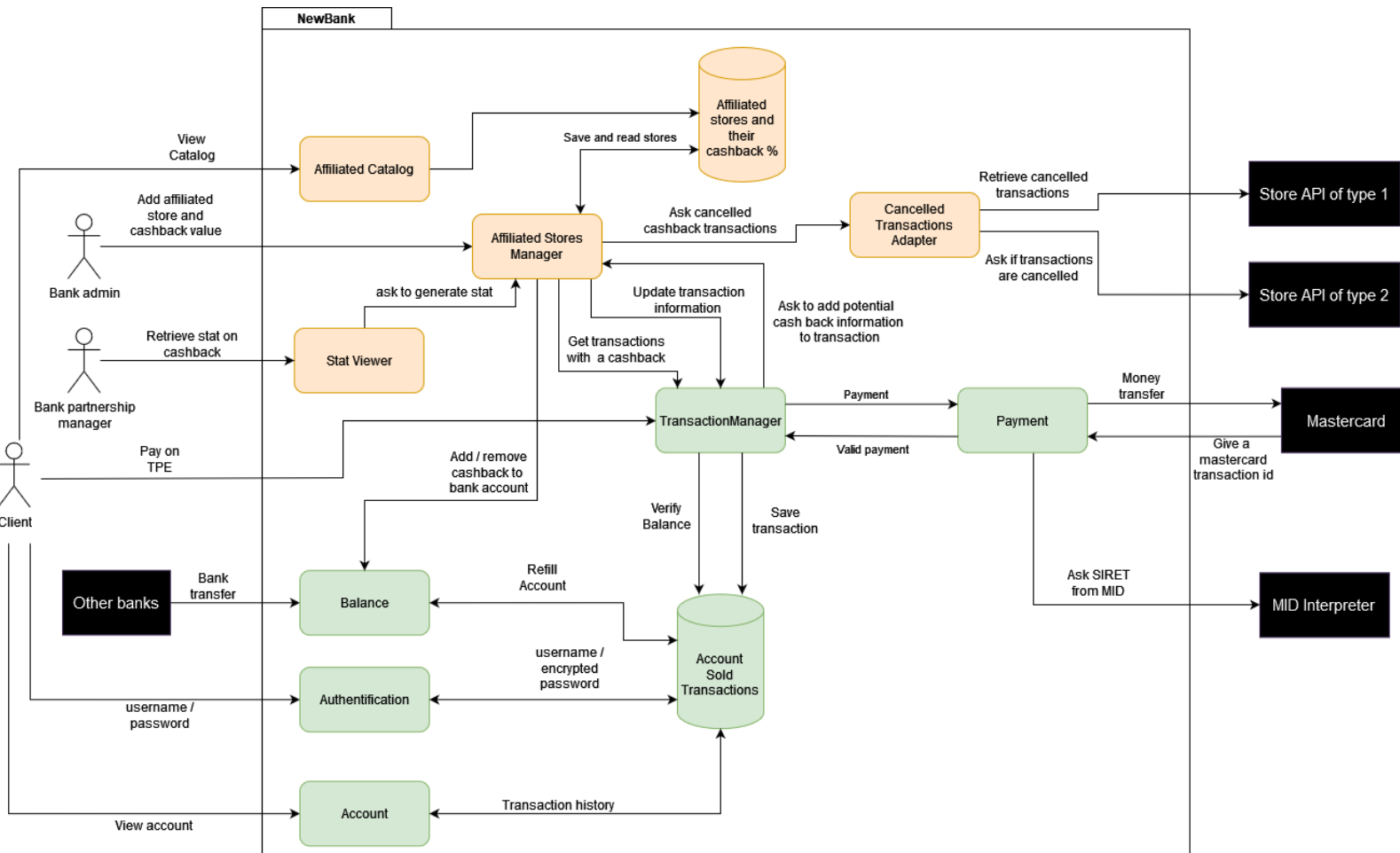
Notre architecture nous permet à l'heure d'aujourd'hui d'effectuer les différentes fonctionnalités suivantes :

- Pour l'utilisateur :
 - Créer un compte bancaire
 - Ajouter des fonds depuis une banque externe
 - Effectuer un paiement normal
 - Effectuer un paiement avec cashback
 - Visualiser des informations concernant son compte (solde, historique des transactions...)

- Pour la banque :
 - Enregistrer un magasin partenaire
 - Générer des statistiques sur les cashback pour le manager des partenariats
 - Retirer le cashback des utilisateurs ayant annulé ou remboursé une transaction que leur en avait octroyé

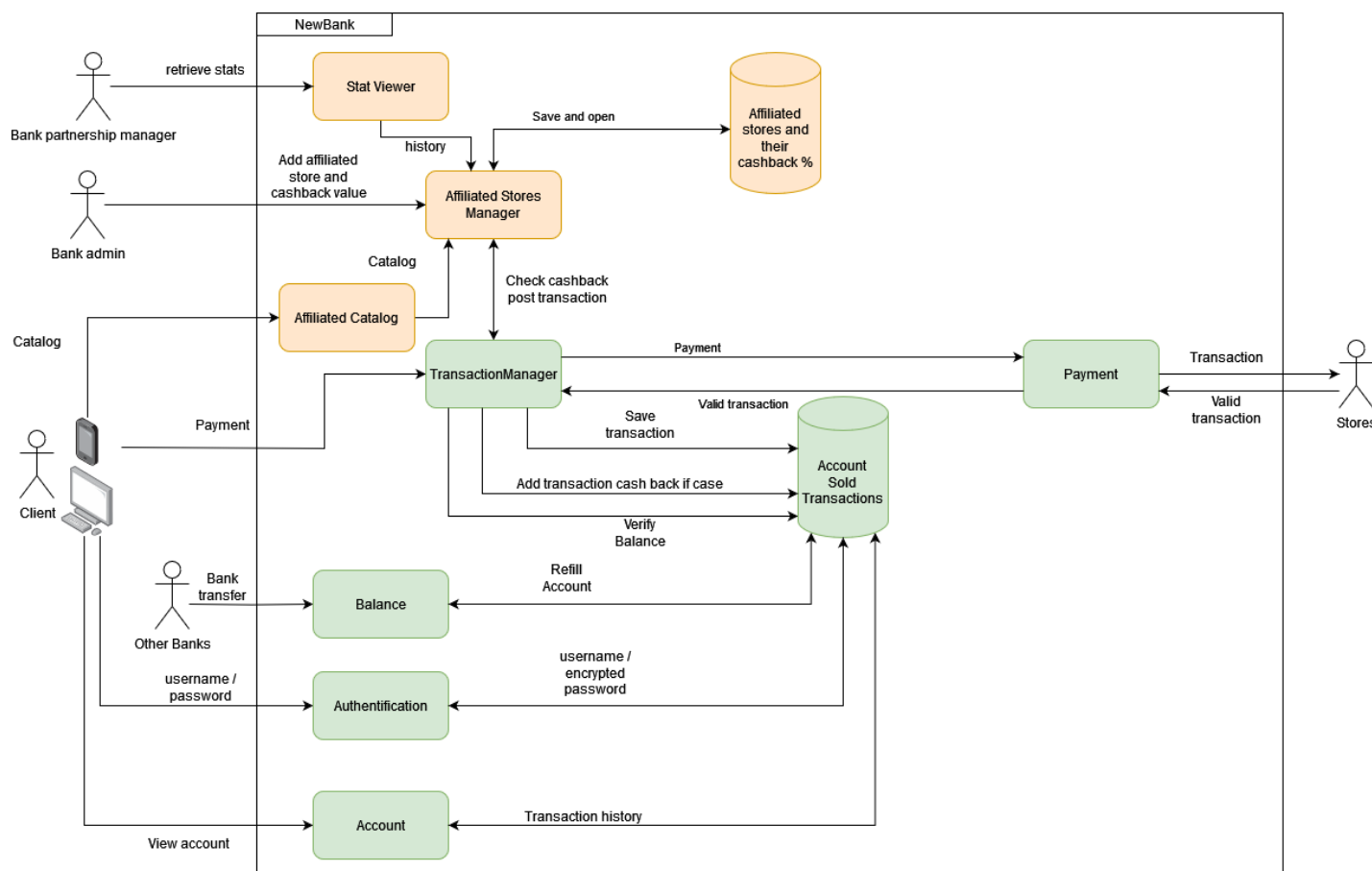
II. Historique de l'architecture

I. 13 novembre 2023



Voici notre architecture lorsqu'elle était composée seulement de deux services. Elle présentait plusieurs Single Point Of Failure, notamment le Transaction Manager et le Affiliated Stores Manager. Les responsabilités de chacun étaient trop importantes dû à un mauvais découpage. De plus, les base de données n'étaient pas répliquées non plus, impliquant alors une moins bonne résilience.

II. 20 octobre 2023

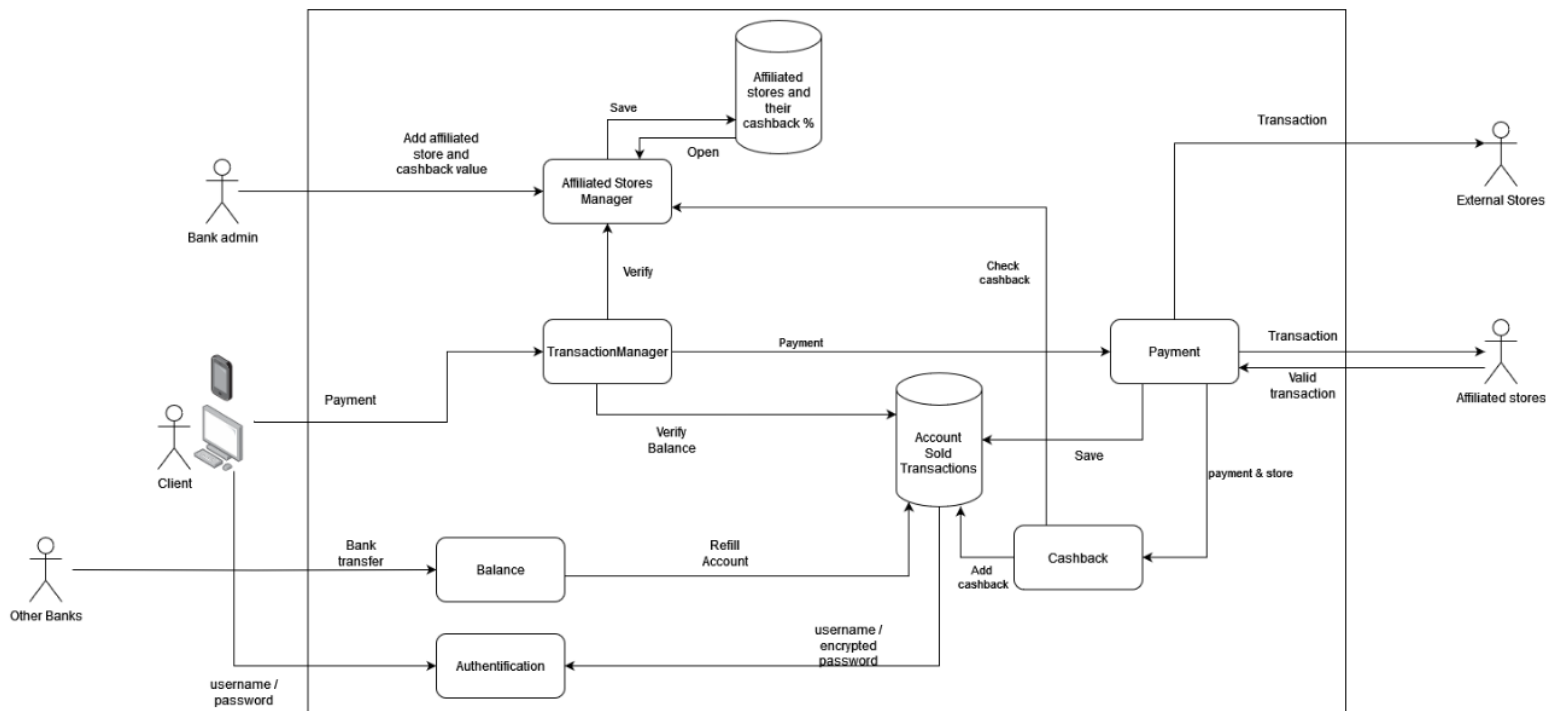


À cette date, notre application manquait cruellement de réalisme en ne faisant pratiquement aucun appel à des services externes. En effet, la transaction était directement transmise au magasin qui validait alors la transaction. Dans la réalité, le fonctionnement d'un paiement est bien plus complexe, en impliquant plusieurs acteurs externes pour effectuer le transfert des fonds.

L'application était donc fonctionnelle, mais certaines logiques n'étaient pas cohérentes. C'était le cas notamment avec le composant Transaction Manager qui avait la responsabilité d'ajouter le cashback dans le compte bancaire de l'utilisateur alors que théoriquement, il devrait s'agir d'une action effectuée depuis le service cashback.

Enfin, nous n'avions pas encore pensé à la fonctionnalité de vérification des transactions qui avaient du cashback mais qui ont été annulées ou remboursées par le magasin partenaire. Toute la partie avec les mocks simulant des API distante différentes afin de récupérer cette information n'était donc pas encore présente dans notre architecture.

III. 1 octobre 2023



Il s'agit du premier diagramme d'architecture élaboré dans le cadre de ce projet. Les briques les plus importantes de notre diagramme d'architecture actuel y sont déjà présentes telles que Affiliated Stores Manager, Transaction Manager ou bien Payment.