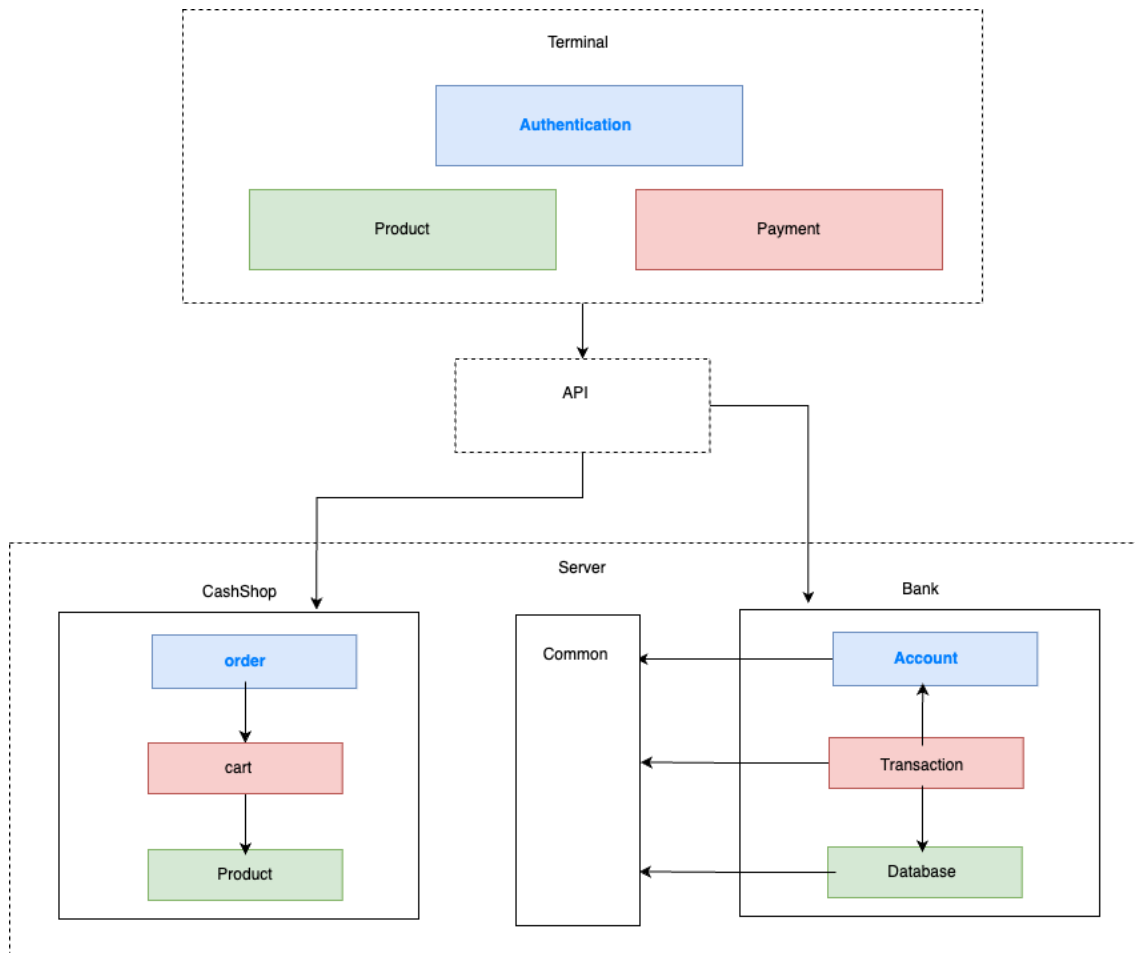


Diagramme d'architecture



Le terminal de paiement

C'est la partie représentant l'interface utilisateur du mobile.

Elle comporte le choix des produits, avec possibilité de les ajouter dans un panier et de valider ce panier pour procéder au paiement. Trois modules la composent.

- **authentication**

C'est le module responsable de l'authentification d'un utilisateur.

- **product**

Ce module sert à la visualisation des produits à vendre par la boutique. Il se charge d'afficher la liste des produits et le panier lors d'un processus pré paiement.

- **payment**

Module à la charge du paiement d'un panier, il sert à générer les QR codes, afficher un lecteur de carte et d'appeler l'API lors d'un processus de transaction bancaire.

L'api

Ce module contiendra plusieurs parties :

- **api_gateway** : router de l'application, réorientant les différentes requêtes HTTP.

- mediator : module qui va dispatcher les notifications entre les différents module bank et shop lors des étapes de commandes et de paiement. Voici son diagramme de classe :

Server

Shop

C'est la partie responsable de la gestion de la boutique. Elle assure la disponibilité des produits et la viabilité des commandes faites depuis le terminal.

Trois modules la composent.

order

Définit la commande passée et stocke l'état de la transaction bancaire.

cart

Définit le panier lors de l'achat.

product

Définit un produit en vente dans la boutique.

Bank

Cette partie simule une application bancaire. Elle permet de gérer les comptes utilisateur et de vérifier la viabilité d'une transaction bancaire lors d'un achat.

Trois modules la composent.

Les noms des modules commencent par : com.cashmanager.

Exemple : Pour le module account -> com.cashmanager.account

Transaction

C'est un module central, il permet de gérer l'état d'une transaction et d'envoyer les informations nécessaires. Il est décisionnaire lors d'un mouvement d'argent, il effectue les transactions d'argent entre les comptes.

Il sert aussi de point d'entrée et de sortie de l'application pour prévenir un service extérieur du bon fonctionnement ou non de la transaction.

Fonctionnalités :

- Créditer un compte,
- Décréditer un compte,
- Prévenir lors d'un mouvement,
- Vérifier l'état global des transactions et créer des logs en lien avec elles.

Account

Ce module sert à gérer la partie compte des clients de la banque. Il définit le profil d'un client et gère les différents comptes qu'il peut avoir.

Fonctionnalités :

- CRUD des comptes,
- Vérifier la véracité d'un compte,
- Bloquer un compte (si fraude).
- Effectuer un transfert d'argent entre deux comptes.

Database

Ce module est centré sur la base de données de module parent Bank. Il contient les entités dont a besoin l'application pour fonctionner. Il a également un dossier pour les mappers, fonctionnalités permettant de transformer les entités en DTO et inversement.

Common

Ce module permet de regrouper les parties communes de l'application, il peut être appelé mais n'a pas de dépendances à d'autres modules en son sein. Il se trouve à la même racine que le module Bank.

Fonctionnalités :

- Modèles communs,
- Énumérations,
- Fonctions services globalisées.