Théo CLERE, Maxime BROSSARD, Sandro SPINA, Ceif-Edine MAROUANI, Julien BUC

cpe LYON

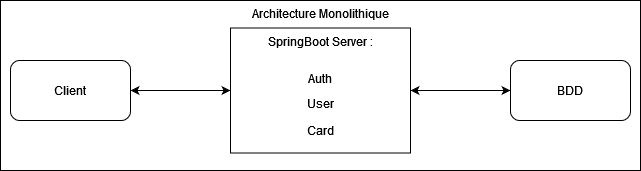
Web Statique & Dynamique

2023-2024

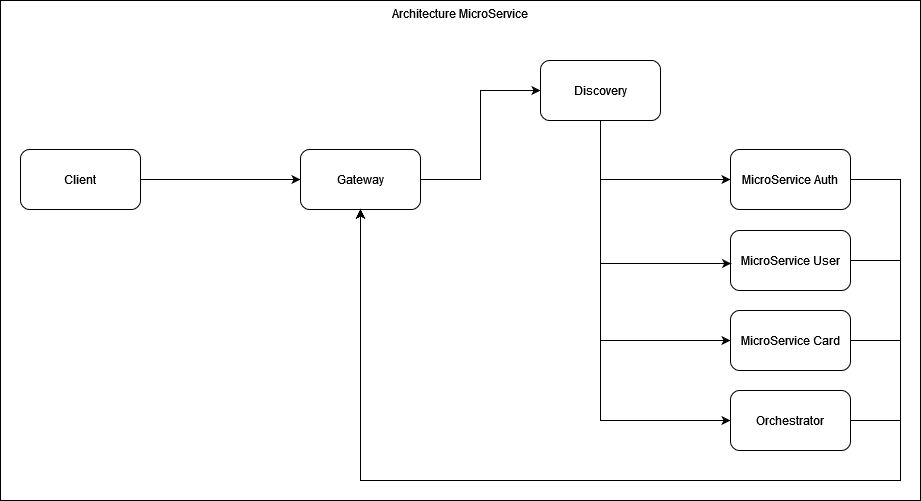
Evolution Architecturale

Au cours de cet atelier 3, nous avons pu reprendre les travaux effectués lors de l'atelier 2 : une application monolithique de gestion de cartes du jeu FIFA. Pendant cet atelier, nous avons transformé cette architecture monolithique en une architecture à micro-services.

Avant :



Après :



Ainsi, nous avons divisé les responsabilités du serveur Spring en trois nouveaux microservices :

* **MicroService Auth** :
  + **Responsabilités** : Gérer l'authentification et l'autorisation des utilisateurs.
  + **Fonctionnalités** :
    - Inscription des utilisateurs.
    - Connexion et déconnexion des utilisateurs.
    - Gestion des tokens JWT (JSON Web Tokens) pour l'authentification.
* **MicroService User** :
  + **Responsabilités** : Gérer les informations des utilisateurs et leur profil.
  + **Fonctionnalités** :
    - Création, lecture, mise à jour et suppression des profils utilisateurs.
    - Gestion du compte bancaire des utilisateurs.
    - Accès et mise à jour des informations personnelles (comme l'email, le nom, etc.).
* **MicroService Card** :
  + **Responsabilités** : Gérer les cartes du jeu FIFA.
  + **Fonctionnalités** :
    - Création, lecture, mise à jour et suppression des cartes.
    - Gestion des attributs des cartes (comme les statistiques des joueurs, les niveaux, etc.).
    - Marketplace

En plus de ces 3 micro-services, deux autres éléments nécessaires au bon fonctionnement de notre architecture. Ces éléments sont les suivants :

* **Discovery Server**:
  + **Responsabilités**: Gérer le recensement des MicroServices
  + **Fonctionnalités**:
    - Enregistrement des micro-services.
    - Gestion des informations relatives aux MicroServices
  + **Explications**: Le Discovery Server va fonctionner avec la technologie Eureka. Une simple configuration est nécessaire pour mettre en place ce serveur. Ensuite, il faudra configurer Eureka Discovery Client sur les micro-services. Il faudra lui renseigner l’adresse du Discovery Server et son port afin que les micro-services aillent s’enregistrer vers le serveur Eureka.
* **Gateway Server**:
  + **Responsabilités**: Router le trafic en fonction des requêtes utilisateurs
  + **Fonctionnalités**:
    - Définit les différentes utilisable par les requêtes utilisateurs
    - Gère le recensement des micro-services en fonction du contexte
  + **Explications**: Le serveur Gateway va gérer le « routage » de nos requêtes vers les différents micro-services. Ce serveur va fonctionner avec le composant Gateway de Spring. La Gateway aura un lien direct avec le Discovery server afin de connaitre l’adresse ou redirigé les requêtes clients.

L'orchestrateur joue un rôle crucial dans la gestion des transactions distribuées à travers différents microservices. Dans le cadre de notre application qui gère l'achat et la vente de carte, l'orchestrateur assure la coordination de plusieurs services indépendants pour garantir que à l'ensemble de l’application une cohérence. Voici comment cela fonctionne généralement :

**Fonctionnement de l'Orchestrateur dans une Transaction Distribuée**

1. **Initiation de la Transaction :**

Le processus commence lorsqu'un utilisateur initie l'achat d'une carte via l'interface utilisateur. Cette action déclenche une requête vers le service gateway.

1. **Coordination des Services :**

Service Gateway : Le service gateway reçoit la requête et la redirige vers l'orchestrateur.

Orchestrateur : L'orchestrateur reçoit la requête de transaction et commence à coordonner les différents microservices impliqués.

1. **Vérification de l'Authentification :**

MicroService Auth : L'orchestrateur envoie une requête au microservice Auth pour vérifier que l'utilisateur est authentifié et autorisé à effectuer la transaction. Cela implique la vérification des tokens JWT et des permissions de l'utilisateur.

1. **Validation des Informations Utilisateur :**

MicroService User : L'orchestrateur contacte le microservice User pour récupérer et modifier les informations de l'utilisateur, notamment le solde du compte bancaire.

1. **Réservation de la Carte :**

MicroService Card : L'orchestrateur envoie une requête au microservice Card pour vérifier la disponibilité de la carte et la réserver.

**Gestion des Échecs :**

En cas d'échec à n'importe quelle étape (comme une erreur de paiement ou une indisponibilité de la carte), l'orchestrateur doit gérer la compensation. Notre cas nous devons exécuter les étapes de compensation illustrées sur le schéma

**Schéma camunda achat de carte**

Une image contenant diagramme, ligne, texte, Parallèle

Description générée automatiquementNous avons aussi un autre exemple sur la création d’un compte utilisateur :

Description de ce qu’il a été fait sur les tests unitaires, pipeline Sonar.