**🔍 Qu'est-ce que la découverte de services ?**

La **découverte de services** est un mécanisme essentiel dans les architectures **microservices**. Elle permet aux services de s’identifier et de se localiser dynamiquement sans avoir à coder manuellement leurs adresses IP ou ports.

**🖥️ Définition de Eureka Server**

**Eureka** est un **serveur de découverte de services**

Il joue le rôle de **registre central** dans lequel tous les microservices d'une application s’enregistrent au démarrage. Eureka stocke alors dynamiquement leurs adresses IP, ports, noms de services, etc.

* Dans eureka on ajoute l’annotation **@EnableEurekaServer** dans EurekaApplication pour identifier qu’il s’agit d’un serveur Eureka .
* et dans les autre service on ajoute l’annotation **@EnableDiscoveryClient** pour indiquer qu’il s’agit d’un service client

**🧩 Rôle du serveur Eureka**

1. **Enregistrement des services :**  
   Chaque microservice (appelé **fournisseur de service**) s’enregistre automatiquement auprès du serveur Eureka à son démarrage, en disant : *"Je suis disponible à telle adresse IP et tel port."*
2. **Découverte par d’autres services :**  
   Lorsqu’un autre microservice (**consommateur de service**) souhaite communiquer avec un service spécifique, il interroge Eureka pour obtenir son adresse.
3. **Réduction du couplage :**  
   Grâce à Eureka, les microservices ne dépendent plus d’adresses codées en dur. Cela améliore la **flexibilité**, la **résilience** et la **scalabilité** de l’application.
4. **Mise à jour dynamique :**  
   Si une instance tombe ou est redéployée, Eureka met à jour son registre automatiquement, garantissant que seuls les services **disponibles** soient utilisés.

**🚪 Définition d'une API Gateway**

Une **API Gateway** (passerelle d’API) est un **point d’entrée unique** pour toutes les requêtes provenant des clients (applications web, mobiles, etc.) dans une architecture **microservices**.

Elle agit comme un **intermédiaire** entre le client et les différents microservices, en **organisant**, **filtrant**, **redirigeant** et **sécurisant** les appels API.

**🎯 Pourquoi utiliser une API Gateway ?**

1. **Simplifie l’accès aux microservices** :
   * Le client ne connaît qu’une seule URL (celle du Gateway) au lieu d’en connaître plusieurs.
2. **Réduit la latence** :
   * En agrégeant plusieurs appels de microservices en un seul (utile pour les SPA, applications mobiles...).
3. **Centralise la sécurité** :
   * Authentification, autorisation, vérification des tokens, etc.
4. **Gère le routage dynamique** :
   * Redirige les requêtes vers le bon service, même si l’emplacement change (via Eureka).
5. **Masque la complexité** :
   * Présente une API simple au client tout en orchestrant plusieurs services derrière.
6. **Supporte le versioning et la transformation** :
   * Par exemple, transformer du JSON en XML, ou versionner les endpoints (/v1, /v2...).

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A white rectangular box with black text

AI-generated content may be incorrect.