

ATELIER1 CHEVEREAU DOHOU SPINARD MALEK

Schéma d'architecture monolithique existantes

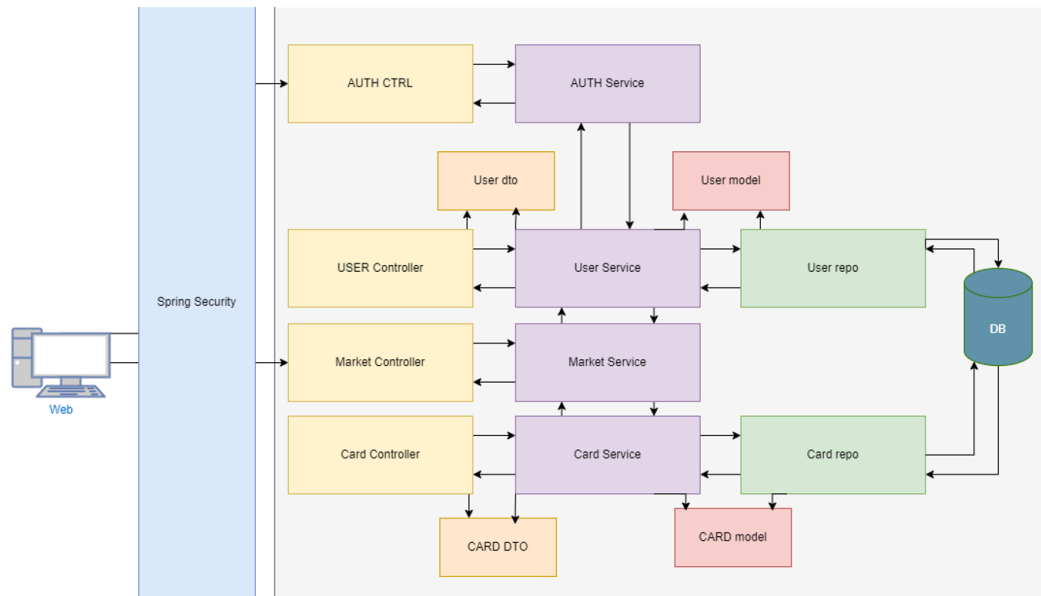


Schéma d'architecture de notre nouvelle application

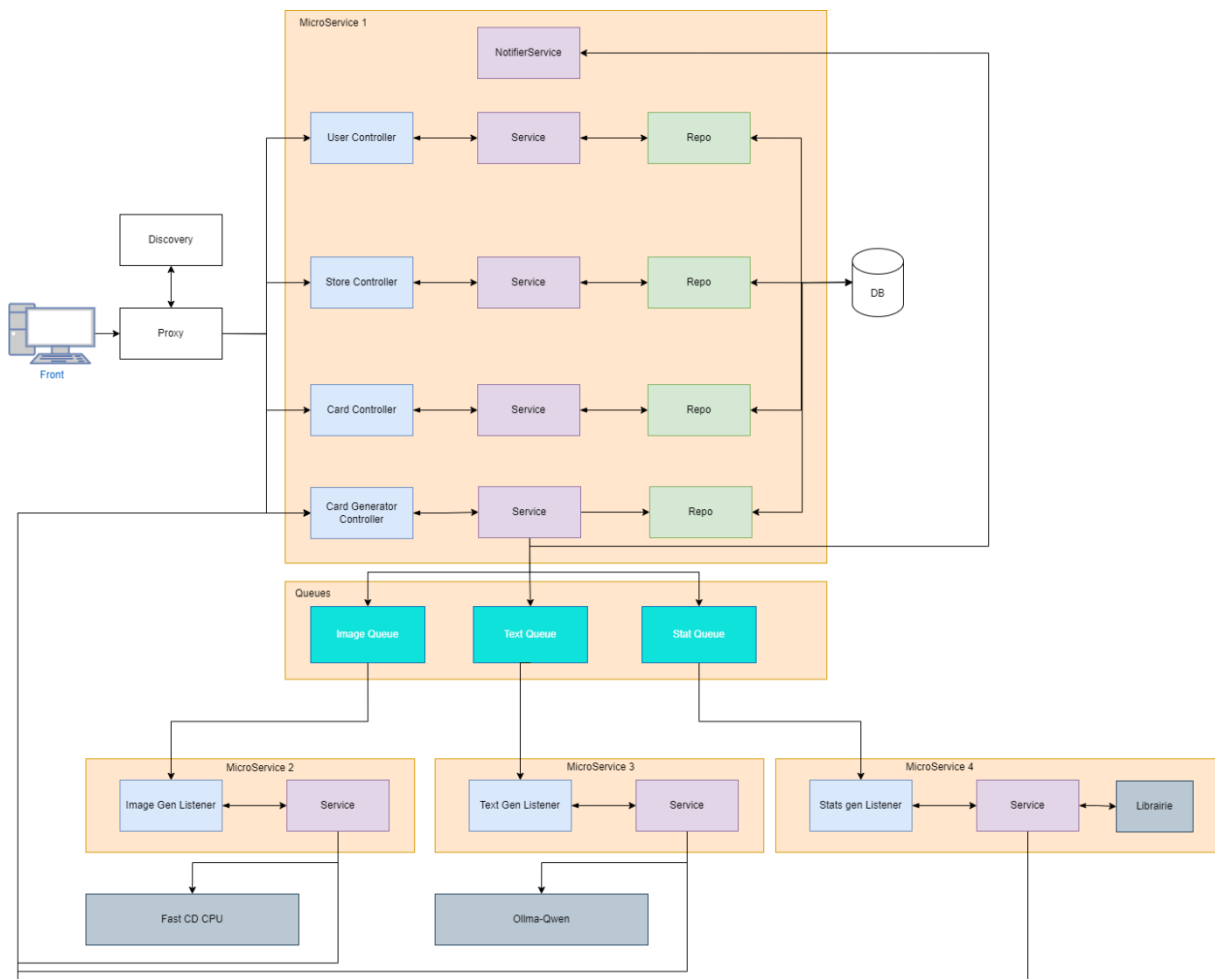


Tableau récapitulatif des bus de communication

Réaliser un tableau récapitulatif des bus de communication les plus répandus

| Bus de communication | Avantages | Inconvénients |
|----------------------|---|---|
| Kafka (Apache Kafka) | <ul style="list-style-type: none">- Haute tolérance aux pannes- Très scalable- Haute performance- Standard dans l'industrie- Installation pas trop complexe | <ul style="list-style-type: none">- Complexité d'administration |
| RabbitMQ | <ul style="list-style-type: none">- Bonne gestion des files d'attente- Supporte plusieurs protocoles (AMQP, MQTT) | <ul style="list-style-type: none">- Configuration plus complexe que Kafka |
| MQTT | <ul style="list-style-type: none">- Très léger, adapté aux projets IOT | <ul style="list-style-type: none">- Ne convient pas aux cas d'usage où une haute sécurité est nécessaire |
| SQS d'AWS | <ul style="list-style-type: none">- Géré par AWS (pas d'infrastructure à gérer)- Très scalable- Bonne intégration avec les services cloud AWS | <ul style="list-style-type: none">- Dépendance vis-à-vis d'AWS- Latence plus élevée comparée à d'autres solutions on-premise |
| gRPC | <ul style="list-style-type: none">- Communication rapide grâce à Protobuf- Excellente performance dans les environnements polyglottes | <ul style="list-style-type: none">- Protobuf est moins lisible que JSON |

Tableau comparatif des frameworks FrontEnd

Framework Front End

| Frameworks | Avantages | Inconvénient |
|------------|--|---|
| React | Performances élevées avec un DOM virtuel. Optimisation via la mise à jour rapide des interfaces. Composants réutilisables et modulaires. Grande communauté avec plusieurs tutoriels ainsi qu'une documentation complète et de qualité. | Apprentissage d'un nouveau langage : le JSX (HTML + JS). Trop de choix au sein de l'écosystème notamment pour les bibliothèques. |
| Angular | Framework complet Utilisation des compocants Amélioration de la qualité du code avec TypeScript Support de Google, stabilité et améliorations continues du Framework. | Framework dense et complexe en fonctionnalités, difficultés d'apprentissages. Performances moins bonnes sur des petites application, architecture trop lourde. |
| Vue | Facile à prendre en main. Utilisation du DOM virtuel Documentation excellente. Composants réactifs | Manque de structure pour les grandes applications Ecosystème plus petit. |
| Ember | Structure complète pour développer des applications avec des conventions strictes. Mise à jours régulières du Framework. | Architecture lourde pour des petites applications. Difficultés à maitriser les conventions trop rigides |