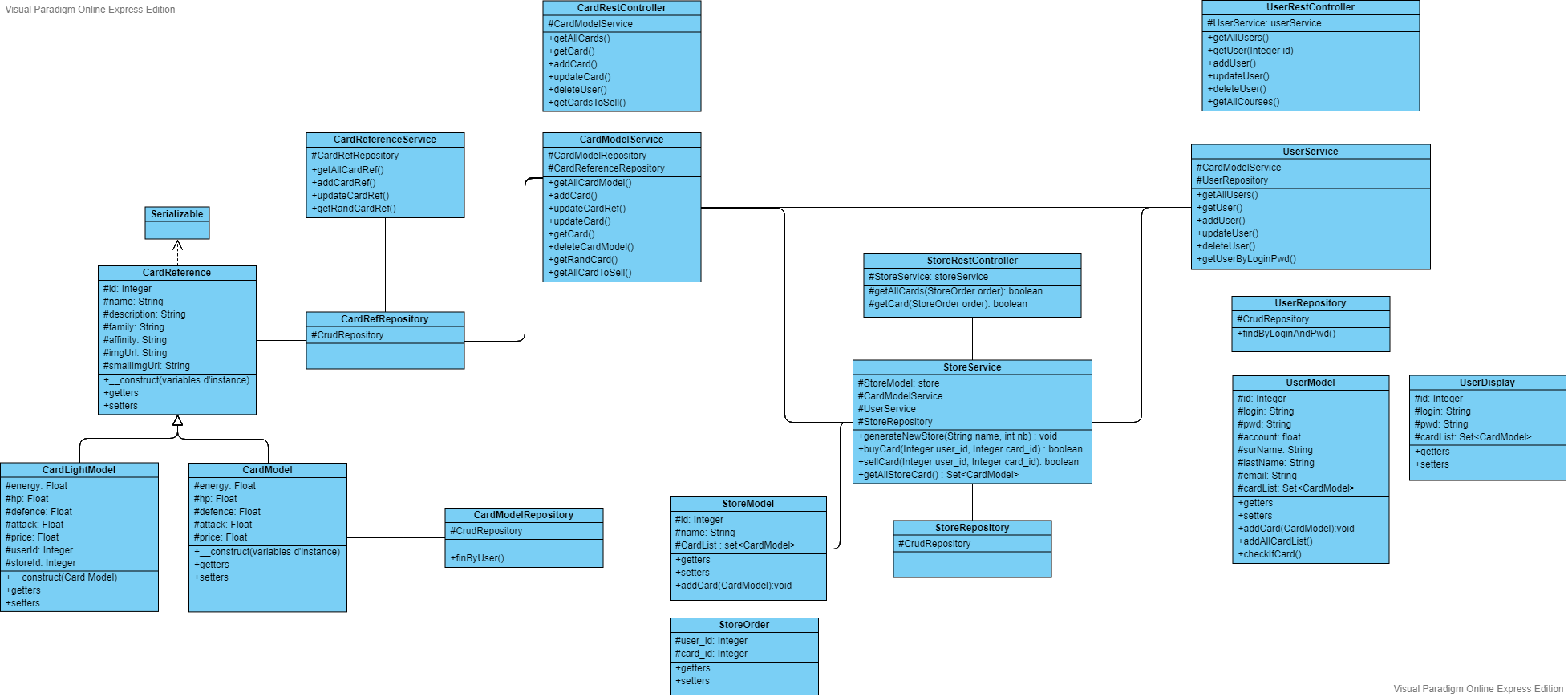
Diagramme de classe existant de l'architecture monolithique



Bus de communication les plus répandus :

RabbitMQ, Kafka et ActiveMQ sont toutes des technologies de messagerie utilisées pour fournir une communication asynchrone et découpler les processus. Kafka est un système de messagerie distribué à haut débit. RabbitMQ est un message broker fiable basé sur AMQP. ActiveMQ et Kafka sont tous deux des produits Apache, tous deux écrits en Java ; RabbitMQ est écrit en Erlang.

|  |  |  |  |
| --- | --- | --- | --- |
|  | ActiveMQ | Kafka | RabbitMQ |
| Avantages | Open-source  Facile à utiliser  Création dynamique de files d'attente.  Interface utilisateur Web pour la gestion des files d'attente.  Le broker est configurable par XML.  Le produit est activement entretenu | Distribué  Scalable  Rendement élevé  Convient au traitement en temps réel  Open source  Excellent pour les projets Big Data | Facile à configurer  Convient à de nombreux langages de programmation et protocoles de messagerie  Peut être utilisé sur différents systèmes d'exploitation et environnements de cloud computing  Interface administrateur  Système de plugins qui permet d’apporter de nouvelles fonctionnalités  Système de clustering pour la haute disponibilité et la scalabilité  Les vhost permettent de cloisonner les environnements |
| Inconvénients | Les messages doivent être envoyés à des files d'attente ou à des sujets. | Manque d'éléments prêts à l'emploi  L'absence d'un système de surveillance complet  Pas de routage  Problèmes avec un nombre croissant de messages | Non transactionnel (par défaut)  Besoin d'Erlang  Problèmes liés au traitement de grandes quantités de données |

Framework Front-end

|  |  |  |  |
| --- | --- | --- | --- |
|  | React Js | Angular Js | Vue Js |
| Avantages | Haut niveau de flexibilité et de réactivité  DOM virtuel (modèle d’objet de document)  La migration entre les versions est généralement très facile, avec « codemods », fournit par Google, pour automatiser une grande partie du processus  Facile à apprendre, en raison de sa simplicité en ce qui concerne la syntaxe  Grande communauté | Injection de dépendance des fonctionnalités liées aux composants avec modules et modularité en général  MVVM (Model-View-ViewModel) qui permet aux développeurs de travailler séparément sur la même section de l’application en utilisant le même ensemble de données  Liaison de données bidirectionnelle qui minimise les risques d’erreurs possibles  Nouvelles fonctionnalités améliorées telles que RXJS, compilation plus rapide | HTML optimisé  Documentation détaillée  Adaptabilité. Il fournit une période de transition rapide d’un autre framework à Vue.js, en raison de la similitude avec Angular et React en termes de conception et d’architecture.  Intégration impressionnante  Petite taille. Vue.js peut peser environ 20Ko en gardant sa vitesse et sa flexibilité qui permet d’atteindre de bien meilleures performances par rapport aux autres framework |
| Inconvénients | Mélanger des templates avec la logique (JSX) peut être déroutant pour certains développeurs  Les développeurs ont parfois trop de choix  Long à maîtriser, ReactJS nécessite une connaissance approfondie de la façon d’intégrer l’interface utilisateur dans le framework MVC. | La syntaxe complexe qui vient de la première version d’Angular. Néanmoins, Angular 6 utilise TypeScript 2.7 qui est moins difficile à apprendre.  Les problèmes de migration qui peuvent apparaître lors du passage de l’ancienne version aux plus récentes | Moins utilisé donc plus petite communauté  Risque de sur-flexibilité. Parfois, Vue.js peut avoir des problèmes lors de l'intégration dans de grands projets  Scalabilité |