

Projet de groupe JVM2 / Ingénierie des données

Flux d'événements de jeux vidéos

(v0 du 08.12.2025)

Scénario du projet



Gestion centralisée de jeux vidéos

- une **plateforme** permet à ses **utilisateurs** inscrits de posséder des **jeux**, de les installer, d'y jouer, de les mettre à jour, de les évaluer, de les recommander, etc.
- des **éditeurs de jeux vidéos** publient des jeux, les améliorent (*via* des patches), y ajoutent des contenus additionnels (DLCs)
- le jeu sur la plateforme produit des **données d'utilisation**, qui permettent notamment de produire des **rapports d'incidents** (*aka* plantages / crashes)
- ces rapports sont utilisés par les éditeurs pour proposer des **correctifs** (*patches*)

Intérêts des parties prenantes

les éditeurs veulent

- publier des jeux de qualité pour maximiser leurs ventes

- obtenir en temps réels des retours sur le fonctionnement de leur application et les exploiter pour fournir de meilleures versions de leurs jeux *via* des correctifs

les plateformes de jeux veulent

- proposer la meilleure expérience (catalogue, fiabilité) à leurs utilisateurs et maximiser leurs ventes
- optimiser la mise en avant et les prix de vente de leurs jeux en fonction de l'expérience de jeu ressentie et des évaluations des joueurs

les joueurs veulent

- avoir une expérience de jeu la plus agréable possible et l'accès à un grand nombre de jeux de qualité
- bénéficier des évaluations par d'autres joueurs
- disposer de moyens de favoriser la correction de leurs jeux et leurs extensions

Contraintes d'architecture

Important

- la communication entre les parties prenantes est **distribuée** et **asynchrone**
- des **événements** doivent être publiés, et on doit avoir la garantie qu'ils seront traités par les parties prenantes concernées (*at-least-once delivery*)
- la communication doit être régie par des **contrats sur les données échangées**, et ces contrats peuvent être amenés à évoluer dans le temps pour permettre l'évolution des applications

- les applications doivent être indépendantes et communiquer au travers d'un flux d'événements

Éléments du cahier des charges

Des applications indépendantes qui communiquent entre elles au travers d'un système de flux d'événements doivent permettre un certain nombre d'actions pour les parties prenantes :

- éditeurs de jeux vidéo
- plateformes de jeux vidéo
- utilisateurs des plateformes (*aka* joueurs)

éditeurs de jeux vidéo

- peuvent être des entreprises ou des indépendants
- publient des jeux vidéos
 - sont (au moins) caractérisés par : un nom, une plateforme d'exécution (ex. PC, PS5), une liste de genres (ex. simulation, stratégie, etc.)
 - ont un numéro de dernière version, et peuvent être en version anticipée (versions < 1.0)
- suivent les commentaires et rapports d'incidents concernant leurs jeux
 - collection de l'ensemble des commentaires dans une base de données privée
 - routage des commentaires problématiques (portant par ex. sur la stabilité) et des rapports d'incidents pour traitements spécifiques
- publient des correctifs (*patches*)
 - s'appliquent à un jeu vidéo pour un support donné et aux versions antérieures à une version ciblée (ex. 1.5.2)

- sont (au moins) caractérisés par : la nouvelle version, un commentaire de l'éditeur, une liste de modifications (au moins de types : *correction, ajout, optimisation*)

plateformes de jeux vidéo

- permettent l'inscription d'utilisateurs
- s'abonnent aux événements des éditeurs
- proposent des pages de descriptions
 - d'éditeurs (au moins : listes de jeux)
 - de jeux par support (au moins : titre, éditeur, version, historique des correctifs, évaluations par les joueurs) (*en option: gestion du crossplay*)
 - de joueurs (au moins : pseudo, date d'inscription, liste de jeux, évaluations de jeux, temps de jeu par jeu), avec un affichage adapté (ex. page de l'utilisateur, page d'un utilisateur ami, page d'un utilisateur autre)
- permettent d'acquérir des jeux
 - fixent les prix en fonction de plusieurs critères (initialement : prix éditeur, puis variations en fonction de la qualité perçue du jeu et de sa demande)
 - mise en vente d'extensions (DLC) (nécessitent une version minimale d'un jeu pour le même support)
- assure un monitoring de l'exécution (distante) des jeux et de leur appréciation
 - des rapports d'incidents sont obtenus automatiquement et transférés aux éditeurs concernés
 - des évaluations par les utilisateurs sont obtenues auprès d'utilisateurs volontaires et transférées

utilisateurs de plateformes de jeux vidéo (*joueurs*)

- s'inscrivent à une plateforme de jeux vidéo
- fournissent un certain nombre d'informations protégées par le RGPD (au moins : pseudo (unique), nom, prénom, date de naissance)
- peuvent acquérir des (licences de) jeux pour un support
- peuvent consulter des pages de la plateforme (pages utilisateur (dont bibliothèque de jeux personnelle), jeux, éditeurs)
- ont un flux d'informations fondé sur leurs préférences et les jeux qu'ils possèdent ou souhaitent acquérir, incluant par exemple :
 - des mises à jour ou extensions d'un jeu possédé
 - de nouvelles évaluations ou des baisses de prix pour un jeu souhaité
- peuvent évaluer des jeux / extensions
 - qu'ils possèdent et auxquels ils ont suffisamment joué
 - peuvent évaluer l'utilité d'une évaluation par un autre joueur (ex. utile 👍, pas utile 👎)

Contraintes techniques pour le projet

Langages et outils

- Langages de réalisation : Java et Kotlin
 - utilisation de librairies tierces Java / Kotlin
 - développement d'applications en Java / Kotlin (sans GUI, mode texte possible)
- Formats d'échanges de données
 - [Avro](#)
 - (optionnellement) [Protocol Buffers](#), [JSON Schema](#)
- Outils de gestion de flux d'événements
 - [Kafka](#) (gestion des flux)
 - [Kafka Streams](#) (traitements sur les flux)

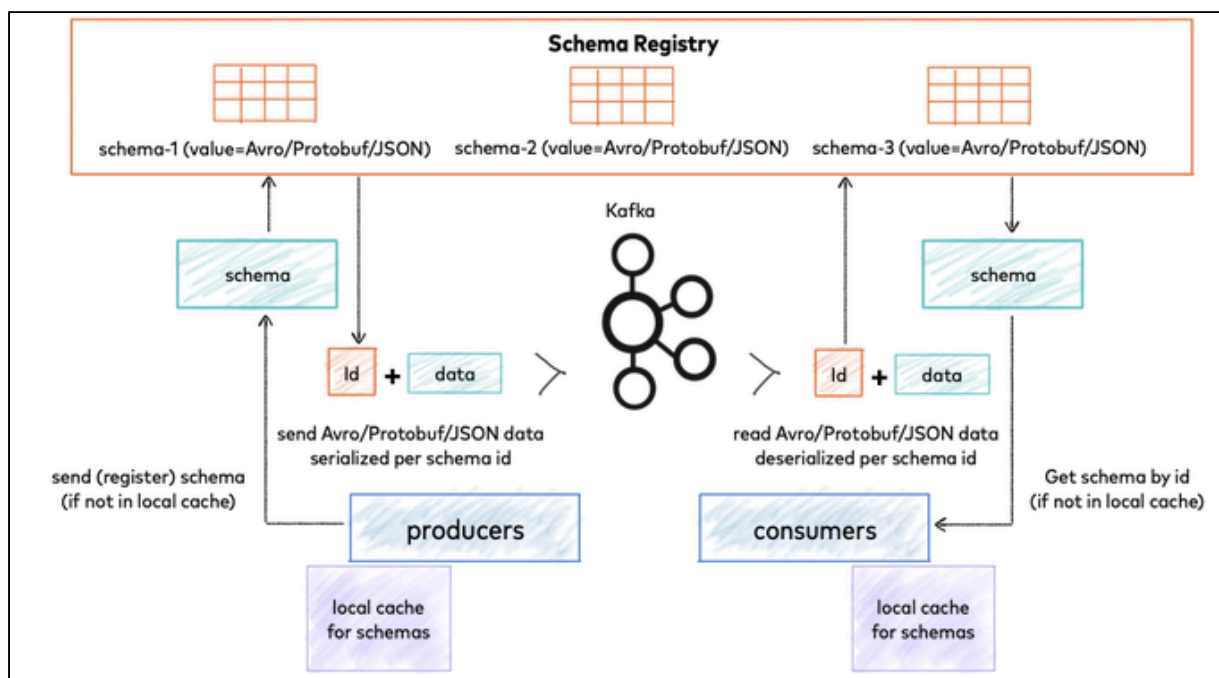
- [Schema Registry](#) (gestion centralisée de schémas utilisable avec Kafka)
- (optionnellement) [Kafka Connect](#) (connexion à des sources de données externes)
- Système de gestion de bases de données
 - choix libre (plutôt BDs relationnelles)

Outil de gestion de flux d'événement



[source](#)

Outil de gestion de schémas



Données utilisées pour le projet

- Objectif : recourir à un dataset existant pour simuler la création de jeux par les éditeurs
- Toutes sources de données possibles
 - possibilité d'utiliser des librairies (ex. [Datafaker](#))
 - possibilité d'utiliser le [jeu de données "VGSales"](#) (Video Games Sales Data Analysis)
 - format CSV
 - "schéma" : Name, Platform, Year, Genre, Publisher, NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales

```
Captain Tsubasa: New Kick Off,DS,2010,Sports,Konami Digital
Entertainment,0,0.02,0.06,0,0.08
Puzzle Chronicles,PSP,2010,Puzzle,Konami Digital
Entertainment,0.07,0,0,0.01,0.08
Busou Shinki: Battle Masters,PSP,2010,Action,Konami Digital
Entertainment,0,0,0.07,0,0.07
Winning Eleven Playmaker 2010: Aoki Samurai no
Chousen,Wii,2010,Sports,Konami Digital
Entertainment,0,0,0.07,0,0.07
Trick DS-han: Kakushi Kami no Sumukan,DS,2010,Misc,Konami
Digital Entertainment,0,0,0.07,0,0.07
Power Pro Success Legends,PSP,2010,Sports,Konami Digital
Entertainment,0,0,0.06,0,0.06
pro evolution soccer 2011,PC,2010,Sports,Konami Digital
Entertainment,0,0.05,0,0.01,0.06
```

Rôle du projet

Objectifs

- Conception et implémentation d'une architecture pour le traitement de données en flux
- Implémentation de producteurs et de consommateurs de données
- Conception, utilisation et évolution de schémas de données
- Mise en œuvre d'outils état de l'art :
 - pour la gestion de données en flux

- pour le traitement de données en flux
 - pour la gestion de schémas
 - Réalisation à l'aide des langages de la JVM (Java, Kotlin)
-

Composition du rendu

- base de code (lien ou archive)
 - rapports de projet (format PDF)
 1. Partie développement logiciel (module *Langages de la JVM 2*)
 - trame de rapport fournie (publiée ultérieurement)
 2. Partie ingénierie des données (module *Ingénierie des données*)
 - trame de rapport fournie (publiée ultérieurement)
-

Choses à faire

- Constitution des groupes de projet (4-5 personnes) : cette séance
- Retours sur la version préliminaire du projet (v0) : séances des 9/11 décembre
- Suivi de projet : séances des 7/9 janvier
- Rendu du projet : date de rendu globale pour les modules