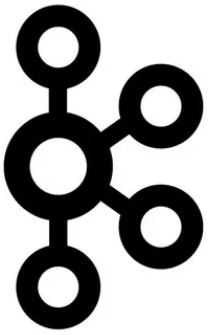
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES HAUTS-DE-FRANCE

Université Polytechnique HAUTS-DE-FRANCE

Projet Intergiciel

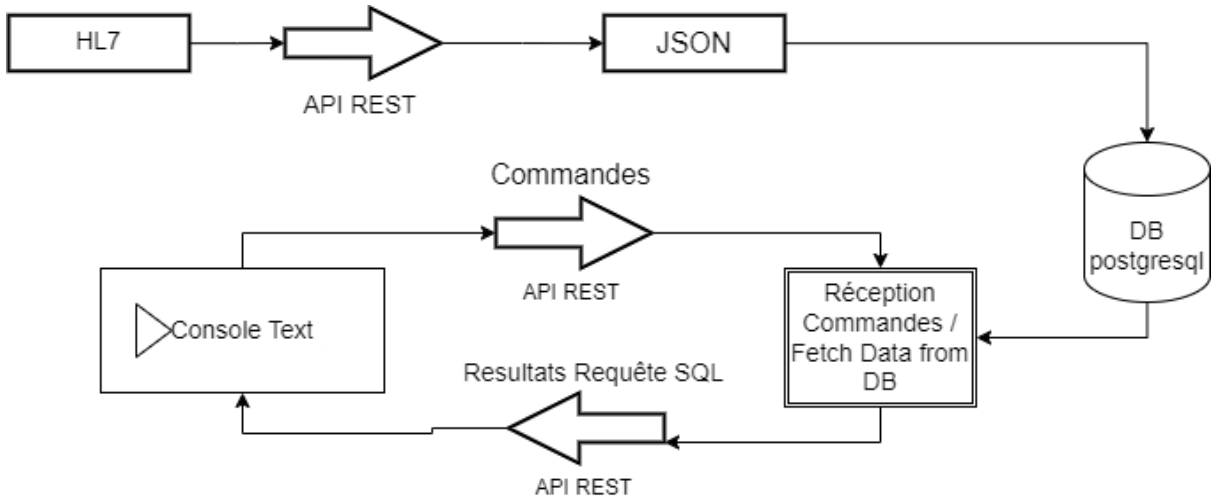


Rapport - Réponses aux questions

Alaaeddin ALMAJJO
Ayman DOULKOM
Pape THIAM

FISA 4 Informatique / 2023-2024

Question N°1 :
Oui, il aurait été possible de répondre au même besoin sans utiliser Kafka ou un autre bus intergiciel. Une alternative pourrait consister à utiliser une architecture basée sur des appels API synchrones ou asynchrones entre les différents composants. Chaque composant pourrait exposer des points de terminaison API REST ou GraphQL, permettant aux autres composants de les appeler pour échanger des données. Cependant, cette approche pourrait être moins scalable et moins résiliente que l'utilisation d'un bus de messages comme Kafka, car elle dépendrait davantage de la disponibilité et de la performance des différents services.



Synoptique des échanges

Question N°2 :
Les deux architectures ont leurs avantages et leurs inconvénients. L'architecture proposée dans le TP avec Kafka offre une scalabilité élevée, une tolérance aux pannes, une faible latence et une bonne capacité à traiter de gros volumes de données. Cependant, elle est plus complexe à mettre en place et à maintenir en raison de la nécessité de configurer et de gérer Kafka et ses composants associés. En revanche, une architecture basée sur des appels API peut être plus simple à mettre en place et à maintenir, mais elle peut être moins scalable et moins résiliente.

Question N°3 :
Pour sécuriser les échanges dans un bus Kafka, plusieurs possibilités sont envisageables :

- Utilisation de TLS/SSL pour chiffrer les communications entre les différents composants Kafka.
- Utilisation de SASL (Simple Authentication and Security Layer) pour l'authentification des clients Kafka.
- Utilisation de listes de contrôle d'accès (ACL) pour contrôler les autorisations d'accès aux topics Kafka.
- Utilisation de protocoles de sécurité externes comme Kerberos pour l'authentification.

Pour le projet actuel, une approche possible serait d'utiliser TLS/SSL pour chiffrer les communications entre les producteurs, les consommateurs et le cluster Kafka. Cela

impliquerait de configurer correctement Kafka pour utiliser TLS/SSL et de générer des certificats SSL pour chaque composant Kafka. Ensuite, les clients Kafka devraient être configurés pour utiliser ces certificats pour établir des connexions sécurisées avec le cluster Kafka. Cela garantit que les données échangées sur le bus Kafka sont protégées contre les interceptions et les altérations non autorisées.