

## **Examen Systèmes Distribués**

### **Durée : 4H | Documents autorisés**

On souhaite créer un système distribué basé sur les micro-services en utilisant une architecture pilotée par les événements respectant les deux patterns Event Sourcing et CQRS. Cette application devrait permettre de gérer des commandes de produits appartenant à des clients. L'application se compose des micro-services fonctionnels suivants :

- Le micro-service « Customer-Service » qui permet de gérer les clients. Chaque client est défini par son id, son nom, son adresse, son adresse mail et son téléphone
- Le micro-service Inventory-Service qui permet de gérer les produits. Chaque produit appartient à une catégorie. Une catégorie est définie par son id, son nom, et sa description. Un produit est défini par son id, son nom, son prix, sa quantité en stocke et son état (Disponible, Rupture, Production, Abandon)
- Le micro-service « Order-Service » qui permet de gérer des commandes. Chaque commande concerne un client et contient plusieurs lignes de commandes. Chaque ligne de commande concerne un produit. Une ligne de commande est identifiée par son id, la quantité de produit, le prix unitaire du produit, le remise. Une commande est définie par son id, la date de commande, la date de livraison, adresse livraison, son statut (Créé, Activée, Abandonnée, en Préparation, expédiée, livrée)

L'application est basée sur les Framework Spring Cloud, MySQL, AXON Framework , AXON Server, et KAFKA Broker. Chaque micro-service est découplé en deux parties indépendantes : la partie commande et la partie Query du micro-service.

#### **Travail à faire**

Développer les fonctionnalités qui vous semblent les plus importantes pour ce projet et rendre un rapport et le code source de l'application répondant aux questions suivantes :

1. Établir une architecture technique du projet
2. Établir un diagramme de classe global du projet
3. Déployer le serveur AXON Server ou KAFKA Broker
4. Développer le micro-service Customer-Service
5. Développer le micro-service Inventory-Service
6. Développer le micro-service Order-Service
7. Mettre en place les services techniques de l'architecture micro-service (Gateway, Eureka ou Consul Discovery service, Config Service)
8. Développer un micro-service qui permet faire du Real time Data Analytics en utilisant Kafka Streams (Nombre et total des commandes sur une fenêtre temporelle de 5 secondes)
9. Développer votre application Frontend avec Angular ou React
10. Sécuriser votre système avec un système de d'authentification OAuth2, OIDC avec Keycloak ou un service d'authentification basé sur Spring Security et JWT
11. Écrire un script docker-compose.yml pour le déploiement de ce système distribué dans des conteneurs docker.