

## **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 les infractions concernant des véhicules suites à des dépassement de vitesses détectés par des radars automatiques. Le système se compose de trois micro-services :

- Le micro-service qui permet de gérer les radars. Chaque radar est défini par son id, sa vitesse maximale, des coordonnées : Longitude et Latitude.
- Le micro-service d'immatriculation qui permet de gérer des véhicules appartenant des propriétaires. Chaque véhicule appartient à un seul propriétaire. Un propriétaire est défini par son id, son nom, sa date de naissance, son email et son email. Un véhicule est défini par son id, son numéro de matricule, sa marque, sa puissance fiscale et son modèle.
- Le micro-service qui permet de gérer les infractions. Chaque infraction est définie par son id, sa date, le numéro du radar qui a détecté le dépassement, le matricule du véhicule, la vitesse du véhicule, la vitesse maximale du radar et le montant de l'infraction.

En plus des opérations classiques de consultation et de modifications de données, le système doit permettre de poster un dépassement de vitesse qui va se traduire par une infraction. En plus, il doit permettre à un propriétaire de consulter ses infractions.

L'application est basée sur les Framework Spring Cloud et AXON. Chaque micro-service est découpé en deux parties indépendantes : la partie commande et la partie Query du micro-service.

En plus des modules représentant les différents micro-services, le projet utilise un module « common-api » sous forme d'un projet maven qui déclare les composants communs aux différents projets comme les Commandes, les Événements, les Queries, les DTOs, etc. Dans ce module, vous pouvez travailler avec Java ou Kotlin

#### **Travail à faire**

Rendre un rapport et le code source de l'application répondant aux questions suivantes :

1. Etablir une architecture technique du projet
2. Etablir un diagramme de classe global du projet
3. Développer le micro-service Radar
4. Développer le micro-service Immatriculation
5. Développer le micro-service Infractions
6. Mettre en place les services techniques de l'architecture micro-service (Gateway, Eureka Discovery service)
7. Développer votre application Frontend avec Angular ou React
8. Sécuriser votre système avec un système de d'authentification OAuth2 comme Keycloak
9. Ecrire un script docker-compose.yml pour le déploiement de ce système distribué dans des conteneurs docker.