Projet 3éme année ING GINF

Matière : Service Oriented Computing

Année universitaire : 2025-2026

Thème : Plateforme intelligente de services urbains interopérables

I. Contexte du projet

Aujourd'hui, les villes intelligentes se développent pour répondre aux besoins croissants des citoyens en matière de mobilité, de sécurité, de qualité d'air et de services publics en général. Dans ce contexte, il devient essentiel pour les administrations et les institutions de pouvoir interconnecter des services très divers, souvent issus de systèmes hétérogènes. Ces services peuvent être anciens, reposant sur des technologies établies comme SOAP, ou plus récents, utilisant des protocoles légers et flexibles tels que REST ou GraphQL. Certains services, en particulier ceux traitant des événements en temps réel comme les alertes d'urgence, nécessitent des communications extrêmement rapides et fiables, ce qui justifie l'usage de protocoles spécialisés tels que gRPC.

Dans ce cadre, vous devez développer une plateforme de services interopérables pour une ville intelligente. Cette plateforme devra intégrer au minimum quatre services distincts, chacun utilisant un protocole différent (SOAP, REST, GraphQL, gRPC), et les exposer via un client web unique. L'objectif est de permettre aux citoyens et aux opérateurs urbains d'accéder facilement à des informations consolidées, et d'interagir avec les services de manière simple et transparente

Exemple de services web que vous pouvez implémenter

Service 1 : Mobilité intelligente (REST)

L'objectif de ce service est de permettre aux citoyens d'accéder facilement aux informations relatives aux transports publics (bus, train, métro).
Le service peut offrir les fonctionnalités suivantes :

- La consultation des horaires en fonction d'une ligne donnée.
- Le suivi de l'état du trafic, avec les retards, annulations ou perturbations.
- Les informations sur la disponibilité et les correspondances entre différents moyens de transport.

Service 2 : Qualité de l'air (SOAP)

Ce service permet de mettre à disposition des citoyens les indices de pollution dans les différents quartiers de la ville. Il peut assurer comme exemple les fonctions suivantes :

- L’accès à l’indice AQI (Air Quality Index) pour une zone donnée.
- La consultation des principaux polluants : particules fines dioxyde d’azote (NO_2), dioxyde de carbone (CO_2), ozone (O_3).
- La comparaison de la qualité de l’air entre deux zones distinctes.

Service 3 : Urgences et santé publique (gRPC)

- Service très rapide pour la gestion d’alertes (ex. accident, incendie, demande d’ambulance).

…

II. Travail demandé

Vous êtes obligatoirement menés à : Implémenter **au moins 4 services distincts** :

- **SOAP** : données historiques ou systèmes existants.
- **REST** : service exposant des ressources CRUD.
- **GraphQL** : service permettant des requêtes personnalisées.
- **gRPC** : service pour communication rapide ou temps réel.

Partie A : En premier lieu les services sont relativement indépendants et orchestrés via un service qui modélise un processus métier que vous proposez. Par exemple vous implémentez un intégrateur qui centralisera les appels aux différents services décrivant un workflow : Exemple : Pour qu’un citoyen planifie un trajet dans une ville intelligente :

1. Vérifier la qualité de l’air (SOAP)
2. Si la zone est polluée, recommander une alternative
3. Afficher la disponibilité des transports (REST)

Partie B : Présenter l’application sous forme d’une architecture microservices, avec :

- Une **API Gateway** centralisant les accès.
- Un **client Web** consommant les services via la Gateway.

Utiliser des conteneurs : Emballez chaque microservice dans un conteneur Docker. Cela permet de garantir que chaque service s'exécute dans un environnement cohérent. Créez un Dockerfile pour chaque microservice service

Produire la documentation technique :OpenAPI pour REST, WSDL pour SOAP, SDL pour GraphQL, proto pour gRPC.).

Plan de travail suggéré

1. Choix des domaine métiers (mobilité, santé, e-commerce, énergie, agriculture, etc).
2. Définition des 4/5 …. services que vous allez implémenter et du rôle de chacun.
3. Concevoir et Développer progressivement les services/microservices

4. Mise en place de l'API Gateway.
5. Développement du client.
6. Tests & documentation.

N.B. Les étudiants peuvent ajouter des services supplémentaires, complexifier l'orchestration, ou enrichir le client Web/Mobile.