

## TP2 : MICROSERVICES FOR PROPELIZE.

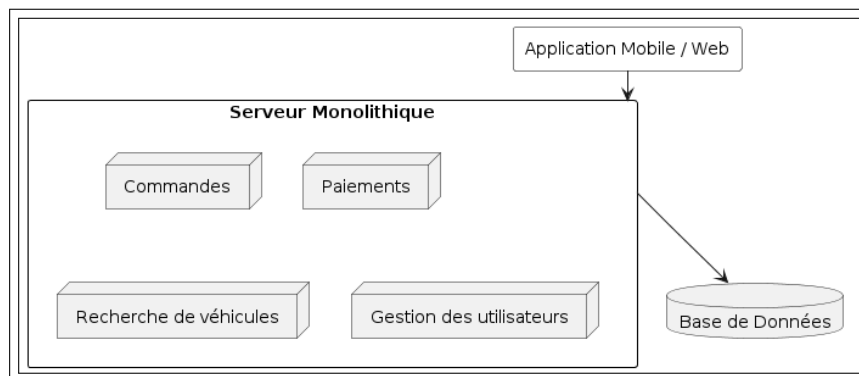
20R2071 MVONDO BILOUNGA JEAN FRANCIS  
20R2053 AMBASSIRA AMBASSIRA RYAN CRAIG  
20S2447 BIRAMAH IRAM NOUBA ADOUM  
20V2436 NJIEMOUN JUDES FRANCK INNOCENTS

May 17, 2024

# Contents

<b>1</b>	<b>Représenter l'ancienne architecture de l'application de véhicule de Propelize</b>	<b>2</b>
<b>2</b>	<b>Inconvénients de l'architecture monolithique de Propelize</b>	<b>2</b>
<b>3</b>	<b>Proposer une architecture microservices pour l'application de location de véhicules</b>	<b>2</b>
<b>4</b>	<b>Choix du stack technologique pour l'implémentation du service véhicule</b>	<b>2</b>
4.1	Node.js . . . . .	3
4.2	Express.js . . . . .	3
4.3	PostgreSQL . . . . .	3
4.4	Docker . . . . .	3

## 1 Représenter l'ancienne architecture de l'application de véhicule de Propelize

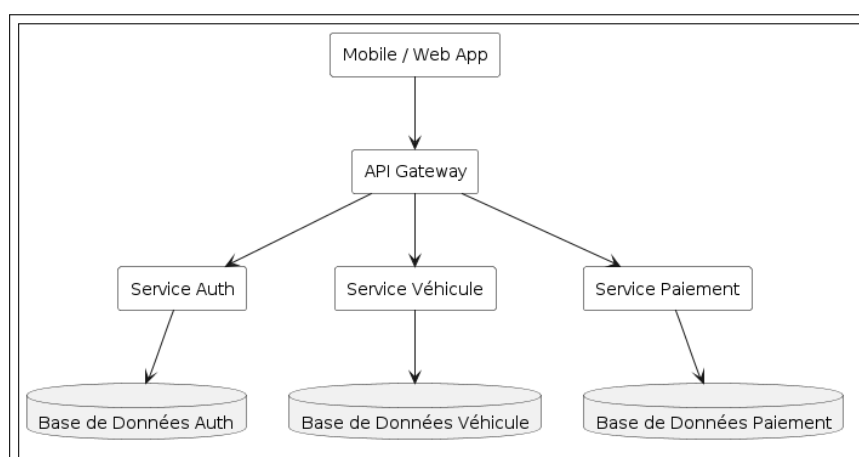


## 2 Inconvénients de l'architecture monolithique de Propelize

Les inconvénients de l'architecture monolithique incluent :

- **Complexité accrue** : À mesure que l'application grandit, la base de code devient difficile à gérer.
- **Cycles de déploiement plus longs** : Toute modification nécessite la reconstruction et le déploiement de l'ensemble de l'application.
- **Dépendances technologiques** : Difficile d'intégrer ou de mettre à jour de nouvelles technologies sans affecter l'ensemble de l'application.
- **Fiabilité réduite** : Un bogue dans une partie du système peut faire tomber toute l'application.
- **Maintenance difficile** : Code difficile à comprendre et à maintenir, surtout pour les nouveaux développeurs.

## 3 Proposer une architecture microservices pour l'application de location de véhicules



## 4 Choix du stack technologique pour l'implémentation du service véhicule

Pour implémenter le service de gestion des véhicules, nous avons choisi d'utiliser les technologies suivantes : Node.js avec Express.js, PostgreSQL, et Docker. Voici les raisons de ce choix :

## 4.1 Node.js

Node.js est une plateforme JavaScript côté serveur qui permet de construire des applications réseau évolutives. Ses avantages incluent :

- **Performances élevées** : Node.js est capable de gérer un grand nombre de requêtes simultanées grâce à son modèle d'architecture orienté événement.
- **Communauté active** : Une large communauté de développeurs qui contribue à un écosystème riche en bibliothèques et outils.
- **JavaScript** : Utilisation du même langage (JavaScript) côté client et côté serveur, ce qui simplifie le développement.

## 4.2 Express.js

Express.js est un framework web minimaliste pour Node.js. Ses avantages incluent :

- **Simplicité** : Express offre une structure légère et flexible pour construire des applications web et des API RESTful.
- **Modularité** : Permet d'ajouter uniquement les modules nécessaires, optimisant ainsi les performances.
- **Support middleware** : Gestion aisée des requêtes HTTP, des routes et des middlewares.

## 4.3 PostgreSQL

PostgreSQL est une base de données relationnelle open-source. Ses avantages incluent :

- **Fiabilité** : Connue pour sa robustesse et sa capacité à gérer des transactions complexes.
- **Fonctionnalités avancées** : Supporte les transactions ACID, les vues, les déclencheurs et les procédures stockées.
- **Extensibilité** : Capacité d'ajouter des extensions et de définir des types de données personnalisés.

## 4.4 Docker

Docker est une plateforme de conteneurisation. Ses avantages incluent :

- **Portabilité** : Les conteneurs Docker peuvent être déployés sur n'importe quel environnement compatible Docker.
- **Isolation** : Chaque conteneur fonctionne de manière isolée, ce qui réduit les conflits entre dépendances.
- **Facilité de déploiement** : Simplifie le processus de déploiement et de gestion des applications en production.

En combinant ces technologies, nous pouvons développer un service de gestion des véhicules qui est performant, évolutif et facile à maintenir.