

# CarGoverload

V4: Cargo

Encadrant : Guilhem Molines

Étudiants :

Masia Sylvain

Montoya Damien

Peres Richard

Rigaut François

# Week 40

## Scénarios **OLD**

### Scénario de la PoC

Le fournisseur cherche un wagon en fonction de son type de marchandise (liquide ou solide), son emplacement, son mode de transport jusqu'à la zone de départ (bateau ou camion) et sa date.

Le système lui propose des wagons, avec départ et arrivée dans des noeuds qui correspondent à son mode de transport (port pour bateau, autre pour camion), à son emplacement (si pas d'emplacement exact, prendre noeud à proximité) et son type de marchandise (citerne pour liquide, classique pour solide) et lui indique une date et heure de chargement.

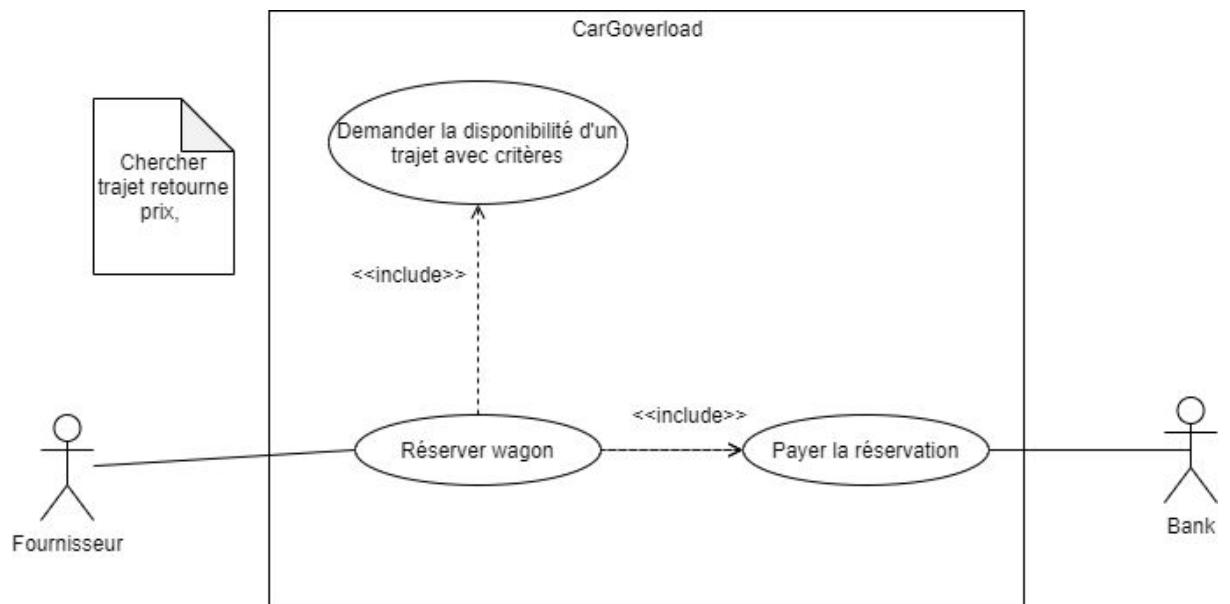
Le fournisseur réserve un wagon qui lui convient puis paie la réservation.

Le wagon appartient au fournisseur, il le charge à l'heure indiquée et la compagnie ferroviaire assure le transport jusqu'à la destination indiquée.

### Scénarios Optionnels

- 1) Obtenir des stats sur les wagons les plus utilisés (type, modèle, etc.) et sur les marchandises les plus transportées (type) pour pouvoir adapter mon parc de wagons à la demande actuelle
- 2) Le fournisseur cherche un wagon en fonction de son type de marchandise (liquide ou solide), son emplacement, son mode de transport (bateau ou camion) et sa date mais n'en trouve pas, on le redirige vers un partenaire

## Use case



Notre système sera à destination d'un seul acteur qui est le fournisseur. Ce dernier pourra rechercher la disponibilité de trajets et en réserver, en fonction de ses critères définis dans les scénarios.

# Week 41

## Scénarios CURRENT

### Scénario de la PoC

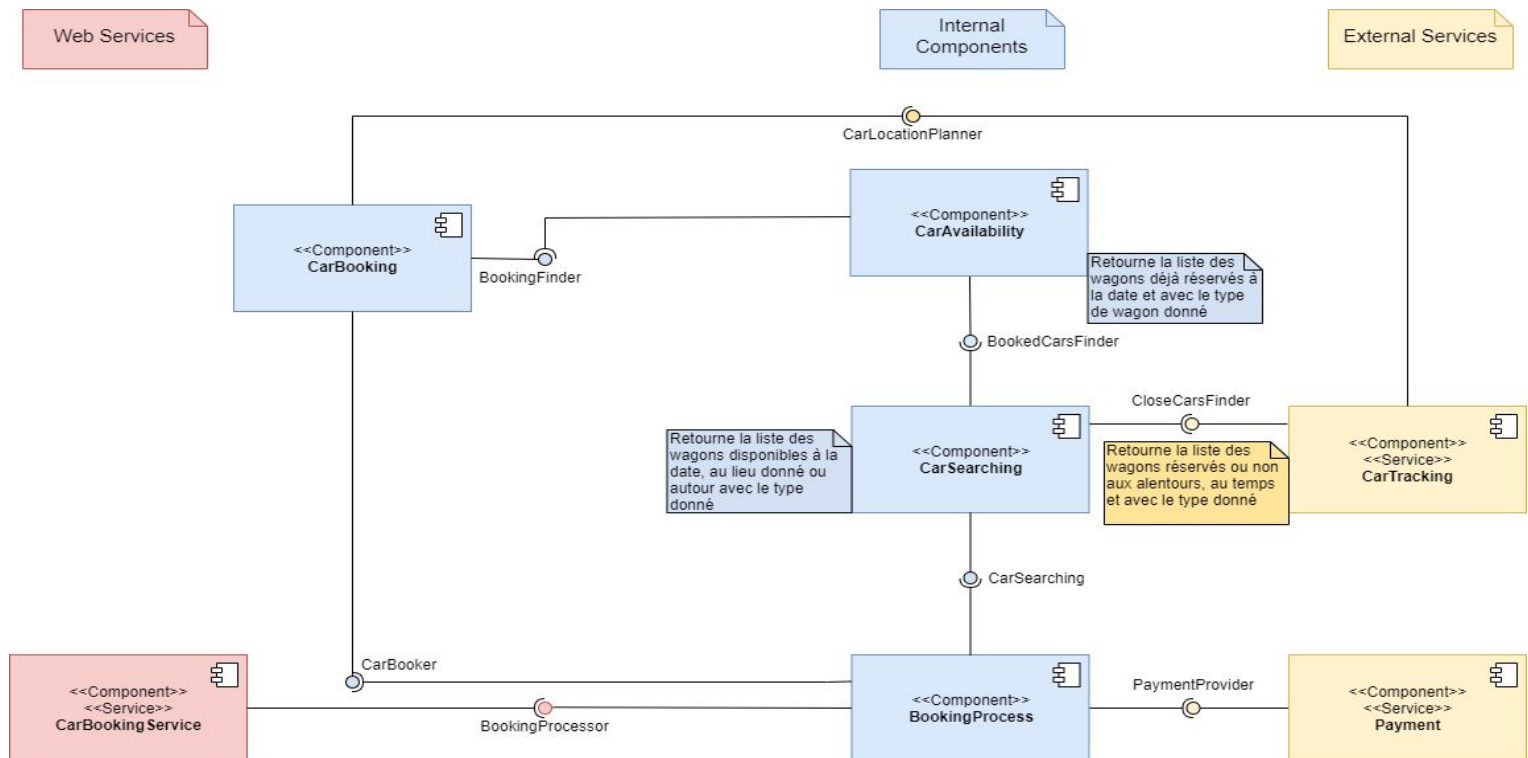
Le fournisseur recherche un wagon liquide et un wagon solide allant de Nice à Paris le 25 Decembre 2020, le système calcule la disponibilité et lui propose un wagon solide et liquide en départ de Nice et allant à Paris. Nice, Cannes et Paris sont tous munis de nœuds permettant le chargement et le déchargement de solide et liquide. Le fournisseur paie la réservation de ces deux wagons.

### Scénarios complémentaires

- 1) Le fournisseur recherche un wagon liquide et un wagon solide allant de Nice à Paris le 25 Decembre 2020, le système calcule la disponibilité et lui propose un wagon solide en départ de Nice et allant à Paris et un wagon liquide disponible à Cannes sur le même parcours que celui souhaité et allant également à Paris. Nice, Cannes et Paris sont tous munis de nœuds permettant le chargement et le déchargement de solide et liquide. Le fournisseur paie la réservation de ces deux wagons.
- 2) Le fournisseur recherche un wagon liquide et un wagon solide allant de Nice à Paris le 25 Decembre 2020, le système calcule la disponibilité et lui propose un wagon solide en départ de Nice et allant à Paris et un wagon liquide disponible à Cannes sur le même parcours que celui souhaité et allant également à Paris. A Cannes et Nice, les noeuds permettent de charger/décharger le solide et liquide mais à Paris (Point d'arrivée), le noeud ne peut décharger que le solide. Le systeme propose le déchargement du wagon liquide à Auxerre, en amont du point d'arrivée. Le fournisseur accepte la proposition et paie la réservation de ces deux wagons.
- 3) Le fournisseur recherche un wagon liquide et un wagon solide allant de Nice à Paris le 25 Décembre 2020, le système calcule la disponibilité et lui propose un wagon solide en départ de Nice et allant à Paris et un wagon liquide disponible à Cannes sur le même parcours que celui souhaité et allant également à Paris. A Cannes et Nice, les nœuds permettent de charger/décharger le solide et liquide mais à Paris (Point d'arrivée), le nœud ne peut décharger que le solide. Le système propose le déchargement du wagon liquide à Auxerre, en amont du point d'arrivé. Le fournisseur réfléchit à la proposition pendant 2 jours. Étant décidé, le fournisseur décide d'accepter la proposition mais celle-ci n'est plus disponible car le wagon solide n'est plus

disponible à Nice. Le système propose un wagon solide qui se situe à Fréjus. Le fournisseur réfléchit à la proposition pendant 2 jours. Le fournisseur décide d'accepter la proposition, qui est toujours disponible et paie la réservation de ces deux wagons.

# Diagramme de composants



Notre architecture se décline comme suit:

- Un composant `BookingProcess`, qui se charge de faire tout le flow utilisateur (recherche, paiement, réservation) et maintient en cache des réservations en cours afin de pouvoir gérer la concurrence entre plusieurs utilisateurs qui voudraient réserver le même wagon
- `CarBooking`, qui gère le stockage des réservations en associant des créneaux d'indisponibilité à chaque wagon connu
- `CarAvailability`, qui fait le lien entre les réservations et qui permet de renvoyer une liste des wagons réservés selon un type et une date.
- `CarSearching`, qui implémente un algorithme de recherche poussé, en prenant en compte tous les wagons proches du lieu de recherche autour de la date demandée (via `CarTracking`), et en retirant ceux qui sont déjà réservés dans notre système (via `CarAvailability`).
- `CarTracking`, composant externe correspondant à la V9, qui permet de retrouver tous les wagons proches d'un endroit à une date donnée. Pour cela, on le met à jour avec notre composant `CarBooking` pour qu'il puisse faire sa planification.

Les avantages de cette architecture sont la possibilité de parallélisation de la recherche et le découplage entre l'algorithme d'attribution des scores des résultats et le filtrage des données des réservations, en plus d'avoir un composant `BookingProcess` purement orienté métier.

# Roadmap **OLD**

*Le fournisseur recherche un wagon liquide et un wagon solide allant de Nice à Paris le 25 Décembre 2020, le système calcule la disponibilité et lui propose un wagon solide et liquide en départ de Nice et allant à Paris. Nice, Cannes et Paris sont tous munis de nœuds permettant le chargement et le déchargement de solide et liquide. Le fournisseur paie la réservation de ces deux wagons.*

## Roadmap technique

### Semaine 41: Initialisation

- Installation de l'environnement de développement
- Initialisation du projet (hello world)
- Mise en place de la gestion de projet
- Dockerization

### Semaine 42: MVP

- Création de tous les composants
- Mock des services externes
- Mise en place de la CI
- Mise en place de la CLI / Front

### Semaine 43 : Robustesse et logique

- Persistance des données
- Implémentation primaire de la logique la réservation, et de la disponibilité (métier)
- Ajout des autres composants lié au déploiement
- Parallélisation et gestion concurrence basique

### Semaine 44 : Amélioration métier

- Affinage de l'algorithme de recherche
- Amélioration du métier
- Parallélisation et gestion concurrence avancée

### Semaine 45 : Rendu 1

- Rendu du POC
- Préparation orale

## Roadmap métier

### Semaine 41: Initialisation

- Préparation de la couche technique

### Semaine 42: MVP

- Pouvoir booker un wagon

### Semaine 43 : Robustesse et logique

- Obtenir la disponibilité des wagons
- Conserver la liste des réservations
- Rechercher un wagon selon des critères

### Semaine 44 : Amélioration métier

- Pouvoir réserver un train afin qu'un autre utilisateur ne le réserve pas à ma place, et ce avant d'avoir payé
- Suggestions de routes alternatives

### Semaine 45 : Rendu 1

- Algorithme de recherche complet, routes alternatives selon wagons proches (en lien avec service externe correspondant au V9, donc externe) et noeuds



# Week 42

## Choix technologiques

Back-end :

- Golang : En dehors du fait que nous voulions profiter de ce projet pour pratiquer un nouveau langage de programmation, le Go est parfait pour notre projet, puisqu'il est conçu pour gérer les problèmes d'algorithmique et de concurrence et est facilement déployable et adapté à la scalabilité.
- MongoDB : MongoDB est une BDD scalable et adaptée à notre problème, puisque nous stockons des données dont la structure peut varier (Les attributs des réservations ou les noeuds de ferroutage avec des spécificités propres), c'est pourquoi l'utilisation du NoSQL semble plus pertinente.

Front-end :

- html/css + JavaScript : Pour le PoC, nous resterons en lignes de commandes dans un premier temps, puis pourrons créer une interface graphique simple en html/css par la suite (nous pourrons la faire pour le PoC si nous avons du temps, mais cela n'est pas une priorité). Étant donné la simplicité de l'interface prévue, nous ne prévoyons pas d'utiliser de framework pour l'interface.

Déploiement :

- Docker : Docker nous permet de déployer facilement notre application pour qu'elle soit testable par n'importe qui. Cela permet également de lancer nos différents modules via une seule commande docker-compose.

# Week 43

Rien à ajouter sur cette semaine, le développement suit la roadmap de la semaine 41.

# Week 44

## Modification de la roadmap

Le développement de la CLI originellement prévu pour la semaine 42 mais non réalisé (oubli) est déplacé à cette semaine car nécessaire pour la démonstration à venir.

Roadmap **CURRENT**

*Objectif POC : Le fournisseur recherche un wagon liquide et un wagon solide allant de Nice à Paris le 25 Décembre 2020, le système calcule la disponibilité et lui propose un wagon solide et liquide en départ de Nice et allant à Paris. Nice, Cannes et Paris sont tous munis de nœuds permettant le chargement et le déchargement de solide et liquide. Le fournisseur paie la réservation de ces deux wagons.*

## Roadmap technique

### Semaine 41: Initialisation

- Installation de l'environnement de développement
- Initialisation du projet (hello world)
- Mise en place de la gestion de projet
- Dockerization

### Semaine 42: MVP

- Création de tous les composants
- Mock des services externes
- Mise en place de la CI

### Semaine 43 : Robustesse et logique

- Persistance des données
- Implémentation primaire de la logique la réservation, et de la disponibilité (métier)
- Ajout des autres composants lié au déploiement
- Parallélisation et gestion concurrence basique

### Semaine 44 : Amélioration métier

- Affinage de l'algorithme de recherche
- Amélioration du métier
- Parallélisation et gestion concurrence avancée
- Mise en place de la CLI / Front => *Déplacé depuis la semaine 42*

### Semaine 45 : Rendu 1

- Rendu du POC
- Préparation orale

## Roadmap métier

### Semaine 41: Initialisation

- Préparation de la couche technique

### Semaine 42: MVP

- Pouvoir booker un wagon

### Semaine 43 : Robustesse et logique

- Obtenir la disponibilité des wagons
- Conserver la liste des réservations
- Rechercher un wagon selon des critères

### Semaine 44 : Amélioration métier

- Pouvoir réserver un train afin qu'un autre utilisateur ne le réserve pas à ma place
- Suggestions de routes alternatives

### Semaine 45 : Rendu 1

- Algorithme de recherche complet, routes alternatives selon wagons proches (en lien avec service externe correspondant au V9, donc externe) et noeuds

## Modification technologique

### Modification de la BDD

Nous sommes revenus sur notre décision quant à l'utilisation de MongoDB : En effet, nous avons décidé de repartir vers une base de données relationnelle, PostgreSQL. Après une seconde réflexion, nous en avons convenu que notre modèle objet n'allait pas changer : les nœuds de feroutages proposeront la même structure peu importe le type de marchandise qu'ils transportent et il en va de même pour les réservations qui auront une structure de donnée fixe. De plus, nous souhaitons pouvoir garantir l'intégrité et la cohérence de notre BDD avec le support des propriétés ACID. Nous avons choisi PostgreSQL car celle-ci est bien documentée, open-source, et semble plus performante que ses concurrents gratuits.