

CarGoverload

V4 : Cargo

Encadrant : Guilhem Molines

Étudiants :

Masia Sylvain

Montoya Damien

Peres Richard

Rigaut François

Week 04

Nouveaux scénarios

Scénario principal

Deux gestionnaires cherchent à faire en simultané des réservations groupées partant du port de Marseille.

- Le gestionnaire 1 entre les différentes réservations de wagon qu'il veut en fonction de sa marchandise :
 - marchandise liquide, 3 conteneurs, arrivée Paris
 - marchandise solide, 2 conteneurs, arrivée Nice

Le système calcule la disponibilité et lui propose en une seule réservation :

- 3 wagons liquide de départ à Marseille et d'arrivée à Paris
- 2 wagons solide de départ à Marseille et d'arrivée à Nice

Le gestionnaire paye la réservation groupée.

- Le gestionnaire 2 entre les différentes réservations de wagon qu'il veut en fonction de sa marchandise :
 - marchandise liquide, 5 conteneurs, arrivée Antibes
 - marchandise solide, 3 conteneurs, arrivée Monaco

Le système calcule la disponibilité et lui propose en une seule réservation :

- 5 wagons liquide de départ à Marseille et d'arrivée à Antibes
- 3 wagons solide de départ à Marseille et d'arrivée à Monaco

Le gestionnaire ne peut pas payer la réservation car le système lui indique que l'offre n'est plus valide

Le gestionnaire 1 ayant payé la réservation, le gestionnaire 2 ne peut plus payer sa réservation car des wagons ont déjà été réservés par le gestionnaire 1.

Scénarios complémentaires

- 1) Le gestionnaire cherche à faire une réservation groupée partant du port de Marseille. Il entre les différentes réservations de wagon qu'il veut en fonction de sa marchandise :

- marchandise liquide, 3 conteneurs, arrivée Mandelieu
- marchandise solide, 2 conteneurs, arrivée Nice

Le système calcule la disponibilité et lui propose en une seule réservation :

- 3 wagons liquide de départ à Marseille et d'arrivée à Antibes
- 2 wagons solide de départ à Marseille et d'arrivée à Nice

Le fournisseur paie la réservation groupée.

Nice est un nœud qui permet le chargement et déchargement de solide et liquide. Cependant, Mandelieu ne permet le déchargement que de solide, le nœud liquide le plus proche de Mandelieu étant Antibes c'est lui qui est retourné.

Le gestionnaire paye la réservation groupée.

2) Le gestionnaire cherche à faire une réservation groupée partant du port de Marseille. Il entre les différentes réservations de wagon qu'il veut en fonction de sa marchandise :

- marchandise liquide, 3 conteneurs, arrivée Mandelieu
- marchandise liquide, 2 conteneurs, arrivée Antibes

Le système calcule la disponibilité et lui propose en une seule réservation :

- 5 wagons liquide de départ à Marseille et d'arrivée à Antibes

Le fournisseur paie la réservation groupée.

Antibes est un nœud qui permet le chargement et déchargement de solide et liquide. Cependant, Mandelieu ne permet le déchargement que de solide, le nœud liquide le plus proche de Mandelieu étant Antibes c'est lui qui est retourné. De plus, la réservation est agrégée en un seul itinéraire.

Le gestionnaire paye la réservation groupée.

User stories

En tant que Gestionnaire, je veux pouvoir spécifier plusieurs itinéraires dans ma recherche afin d'avoir une offre agrégée de plusieurs destinations

En tant que Gestionnaire, je veux pouvoir lancer une recherche composée de plusieurs itinéraires afin d'avoir une offre agrégée de plusieurs possibilités

En tant que Gestionnaire, je veux pouvoir obtenir un résultat optimisant le coût de mes différents itinéraires afin d'en limiter le prix global

En tant que Gestionnaire, je veux pouvoir obtenir un résultat optimisant le temps de trajet de mes différents itinéraires afin d'en limiter la durée globale

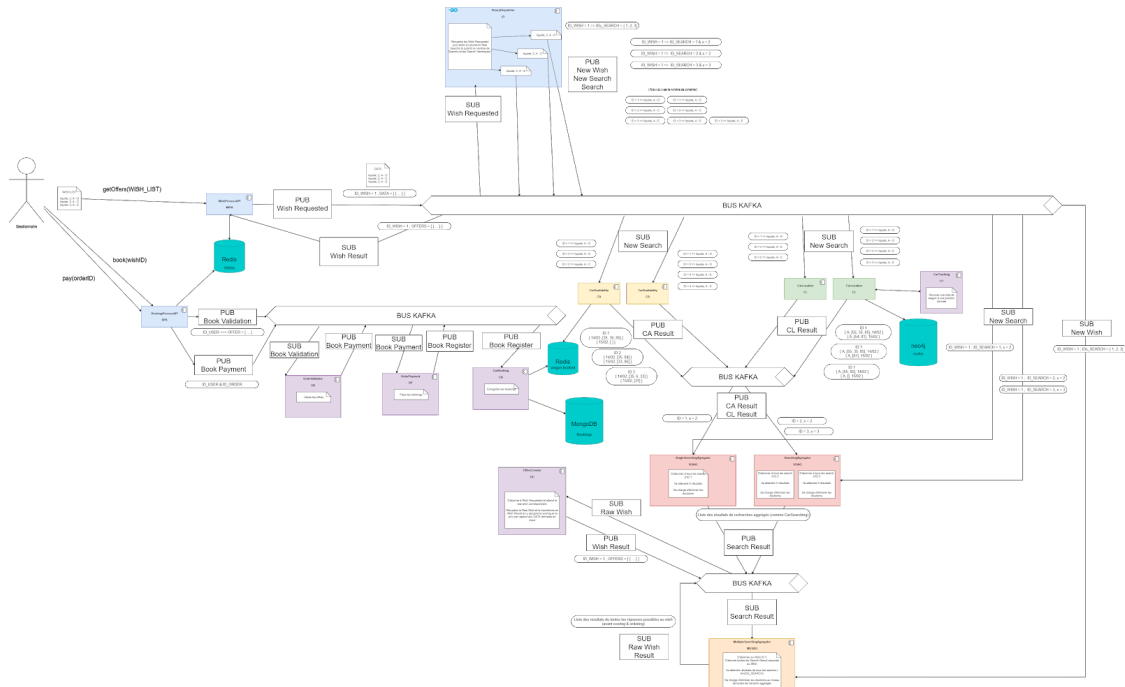
En tant que Gestionnaire, je veux pouvoir obtenir plusieurs alternatives à la même date pour un itinéraire souhaité afin d'avoir des options de secours

En tant que Gestionnaire, je veux pouvoir obtenir plusieurs alternatives à une autre date pour un itinéraire souhaité afin d'avoir des options de secours

En tant que Gestionnaire, je veux savoir si une offre agrégée que l'on m'a proposé est toujours valide

En tant que Gestionnaire, je veux pouvoir payer une offre agrégée valide que l'on m'a proposé précédemment

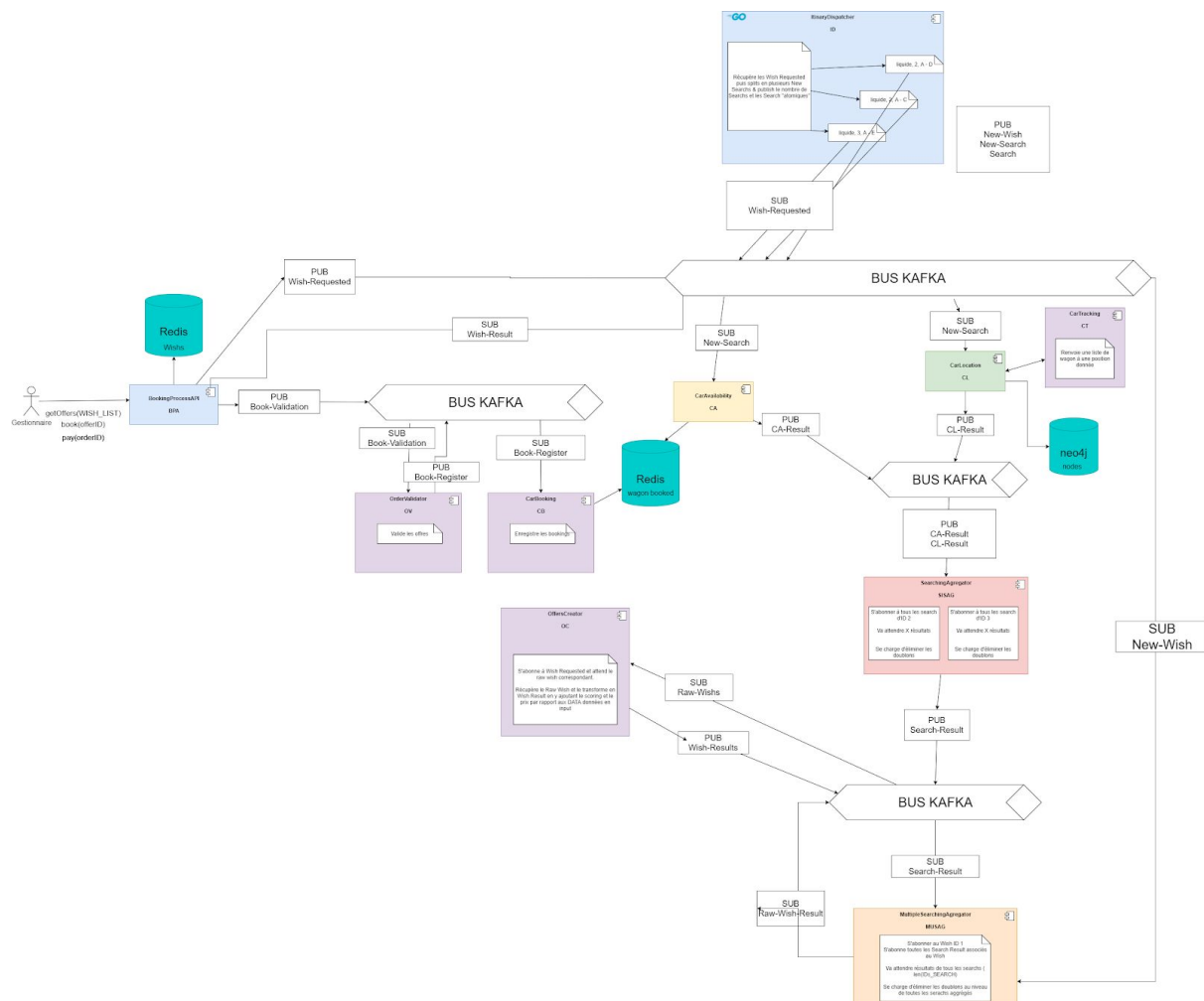
En tant que Gestionnaire, je veux savoir si l'un de mes itinéraires souhaités dans une recherche n'est pas disponible



note : nous n'avons qu'un bus kafka mais par soucis de représentation sur le schéma nous avons mis plusieurs éléments graphique représentant le bus

L'architecture ci-dessus est le fruit de notre réflexion pour convenir au nouveau scénario souhaité, cependant au vu de la charge conséquente de travail qu'il faudrait investir dans sa mise, nous avons décidé d'en diminuer la complexité

Architecture à mettre en place



Nous souhaitons mettre en place l'architecture ci-dessus. Il s'agit d'une architecture qui se rapproche d'une architecture micro-service basée sur l'event sourcing. La majorité de nos services sont donc connectés à un bus kafka (un seul et même bus, répliqué uniquement pour la représentation dans le schéma).

5 nouveaux services font leur apparition :

- ItineraryDispatcher : Chargé de splitter une recherche en plusieurs itinéraires et un itinéraire en plusieurs sous itinéraires
- OrderValidator : Permet de vérifier la disponibilité des wagons d'une offre déjà créée pour ne pas qu'entre le temps de visualisation et de paiement un wagon soit utilisé
- OffersCreator : Crée une offre et la stocke en base de données
- SearchingAgregator : Permet d'agréger les sous itinéraires en une proposition
- MultiSearchingAgregator : Permet d'agréger les itinéraires en une proposition

Sur le principe de fonctionnement, on se rapproche de l'algorithme Map Reduce.

Le but recherché de cette nouvelle architecture est de découpler nos services et de pouvoir profiter pleinement de l'asynchronisme pour paralléliser nos différentes simulations.

À réaliser pour la Week 05

Spikes à faire :

- Init Kafka
- Refactor des services existants
- Intégration des services existants à Kafka
- Init des nouveaux services
- Intégration des nouveaux services à Kafka
- Définition des DTOs

Les tâches à réaliser cette semaine concernent principalement la mise en place de la nouvelle couche technique.

Week 05

Réalisées pendant la semaine

Spikes réalisées :

- Init Kafka => OK
- Refactor des services existants => OK sauf pour BookingProcess
- Intégration des services existants à Kafka => OK sauf pour BookingProcess
- Init des nouveaux services => OK sauf pour OfferCreator
- Intégration des nouveaux services à Kafka => OK sauf pour OfferCreator
- Définition des DTOs => OK

À réaliser pour la Week 06

- Initialisation de OfferCreator et intégration avec Kafka
- Refactor de BookingProcess et intégration avec Kafka
- US : [En tant que Gestionnaire, je veux pouvoir lancer une recherche composée de plusieurs itinéraires afin d'avoir une offre agrégée de plusieurs possibilités](#)
- US : [En tant que Gestionnaire, je veux pouvoir spécifier plusieurs itinéraires dans ma recherche afin d'avoir une offre agrégée de plusieurs destinations](#)

Week 06

Réalisées pendant la semaine

Spikes réalisées :

- Refactor du service BookingProcess
- Intégration de BookingProcess
- Init du service OfferCreator
- Intégration avec Kafka de OfferCreator

User stories réalisées :

- [En tant que Gestionnaire, je veux pouvoir lancer une recherche composée de plusieurs itinéraires afin d'avoir une offre agrégée de plusieurs possibilités](#)

User stories en cours :

- [En tant que Gestionnaire, je veux pouvoir spécifier plusieurs itinéraires dans ma recherche afin d'avoir une offre agrégée de plusieurs destinations](#)

À réaliser pour la Week 07

- US à finir : [En tant que Gestionnaire, je veux pouvoir spécifier plusieurs itinéraires dans ma recherche afin d'avoir une offre agrégée de plusieurs destinations](#)
- Adaptation du frontend pour convenir à un scénario

Week 07

Réalisées pendant la semaine

Spikes en cours :

- Initialisation de Redis avec Nestjs pour le service BookingProcessAPI

User stories réalisées :

- [En tant que Gestionnaire, je veux pouvoir spécifier plusieurs itinéraires dans ma recherche afin d'avoir une offre agrégée de plusieurs destinations](#)

À réaliser pour la Week 08

- US : [En tant que Gestionnaire, je veux pouvoir payer une offre agrégée valide que l'on m'a proposé précédemment](#)

- US : [En tant que Gestionnaire, je veux savoir si une offre agrégée que l'on m'a proposé est toujours valide](#)
- Adaptation du frontend pour convenir à un scénario

Week 08

Réalisées pendant la semaine

Spikes réalisées:

- Initialisation de Redis avec Nestjs pour le service BookingProcessAPI

User stories réalisées :

- US : [En tant que Gestionnaire, je veux pouvoir payer une offre agrégée valide que l'on m'a proposé précédemment](#)
- US : [En tant que Gestionnaire, je veux pouvoir obtenir plusieurs alternatives à la même date pour un itinéraire souhaité afin d'avoir des options de secours](#)
- Adaptation du frontend pour convenir à un scénario de plusieurs recherches

À réaliser pour la Week 09

- US : [En tant que Gestionnaire, je veux savoir si une offre agrégée que l'on m'a proposé est toujours valide](#)
- Finir complètement le scénario de paiement (lié à l'US au dessus)
- Préparer la démo du 5 mars
- Tests de charge à réaliser

Week 09

Réalisées pendant la semaine

User stories réalisées :

- US : [En tant que Gestionnaire, je veux savoir si une offre agrégée que l'on m'a proposé est toujours valide](#)
- Adaptation du frontend pour convenir à un scénario de paiement

Spikes réalisées :

- Création d'une première version de script locust pour la charge

À réaliser pour la Week 10 (Démon)

- Préparer la démon du 5 mars
- Tests de charge à réaliser à partir du script locust