

EQUIPE F

Hadil AYARI

Chahan MOVSESSIAN

Floriane PARIS

Nicolas GUIBLIN

Rapport Service-oriented Architecture

Step 3

Rendu du 1 octobre 2024

Table des matières

Diagramme de l'architecture actuelle	3
Explications des scénarios	4
Semaine 1	4
Semaine 2	5
Semaine 3	6
Les semaines suivantes	7

Diagramme de l'architecture actuelle

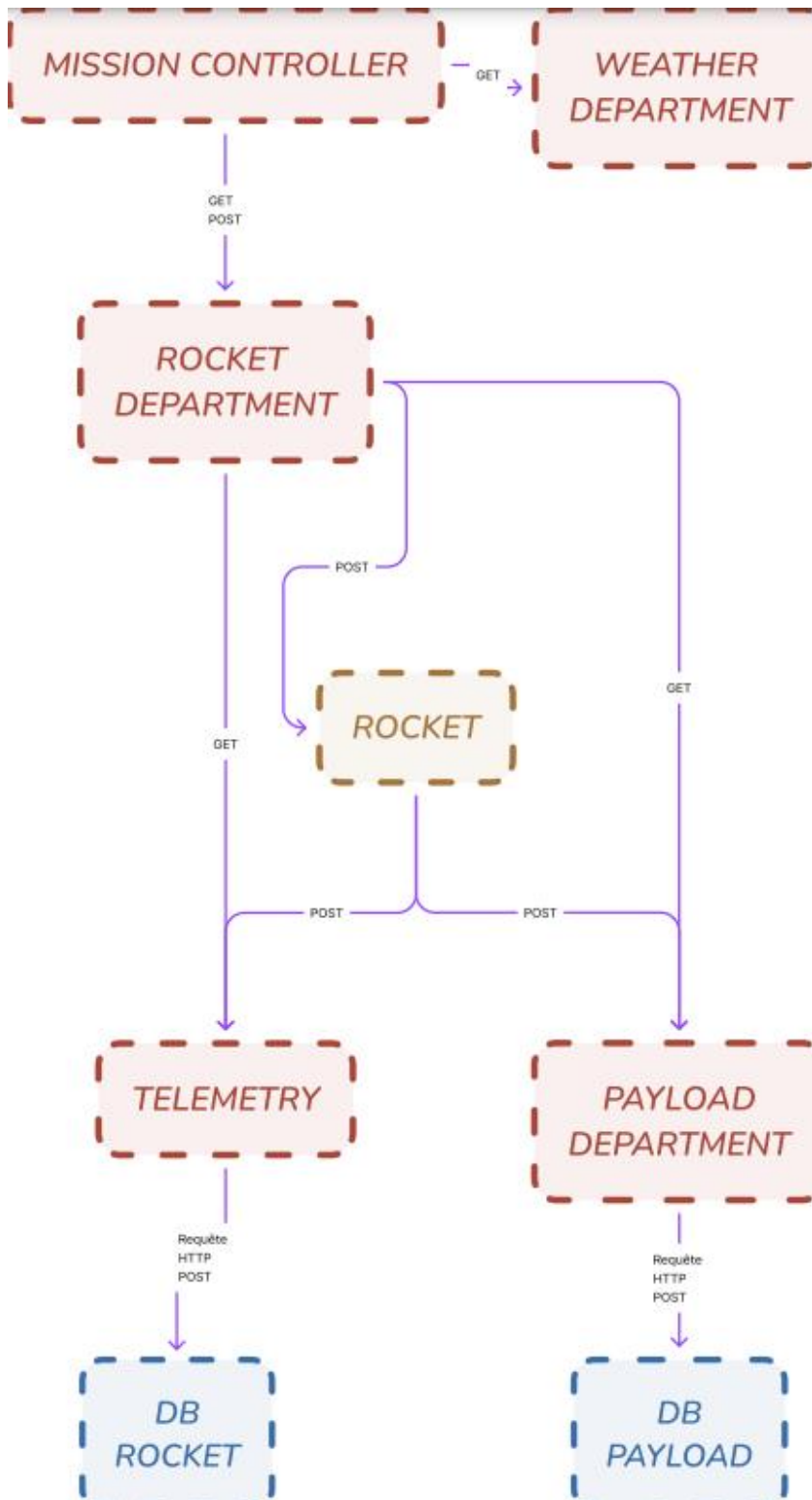


Diagramme d'architecture 1 : Diagramme actuel

Explications des scénarios

Semaine 1

En prenant en compte les scénarios de cette première semaine, nous avons commencé par une architecture à 3 modules : “weather-department”, “mission-controller” et “rocket-department”. Le script du run.sh de cette semaine lançait un main.js dans mission-controller, afin que le commandant puisse aller chercher la réponse du weather-department (Scénario 1) puis celle du rocket-department (Scénario 2). Il prend ensuite en considération ces deux avis avant d’envoyer son accord final au rocket-department (Scénario 3) qui attend cette autorisation avant de faire décoller la fusée (Scénario 4).

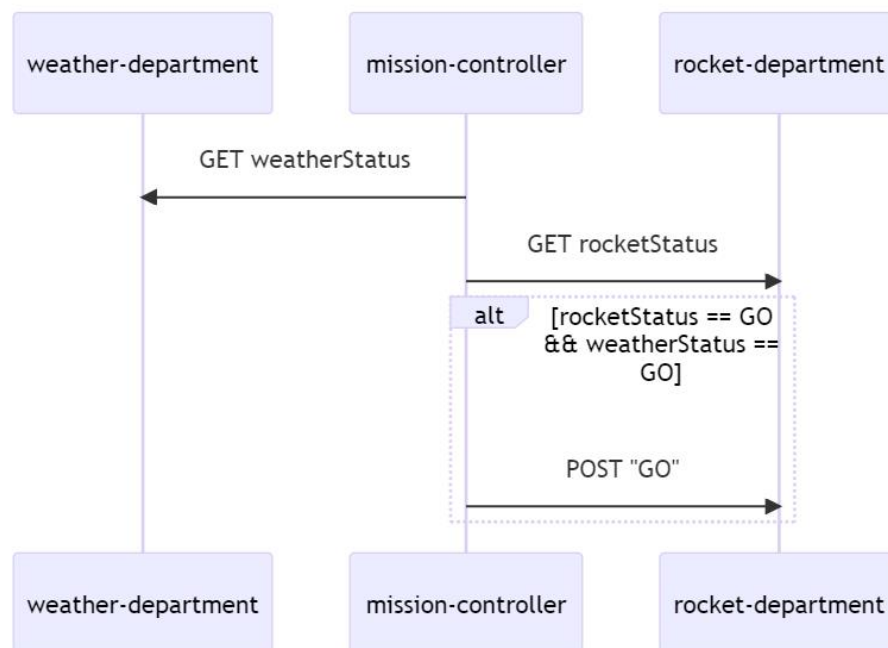


Diagramme de séquence 1

Semaine 2

La semaine d'après, nous avons donc ajouté les modules "Telemetry-department", "Payload-department" et "Rocket".

Séparer l'objet Rocket du département Rocket nous a permis d'être plus organisés. Cet objet Rocket ne reçoit d'information que du département Rocket, et envoie directement les données télémétriques au département télémétrique (Scénario 5), car on suppose que la fusée possède des appareils permettant de mesurer ses informations.

Le département Rocket peut récupérer ses données par des appels au département Télémétrie, et ainsi vérifier que la fusée s'est divisée sans problème (Scénario 6) car nous avons décidé qu'il était plus logique que le détachement d'un étage se fasse automatique lorsqu'il n'a plus de carburant.

Le département Payload peut également récupérer des données au département télémétrique afin de s'assurer que le décollage se passe bien (Scénario 7)

Une page web est également disponible pour visualiser les infos de telemetrie de la rocket et du payload à l'adresse suivante : <http://localhost:3003/> depuis la machine hébergeant les services

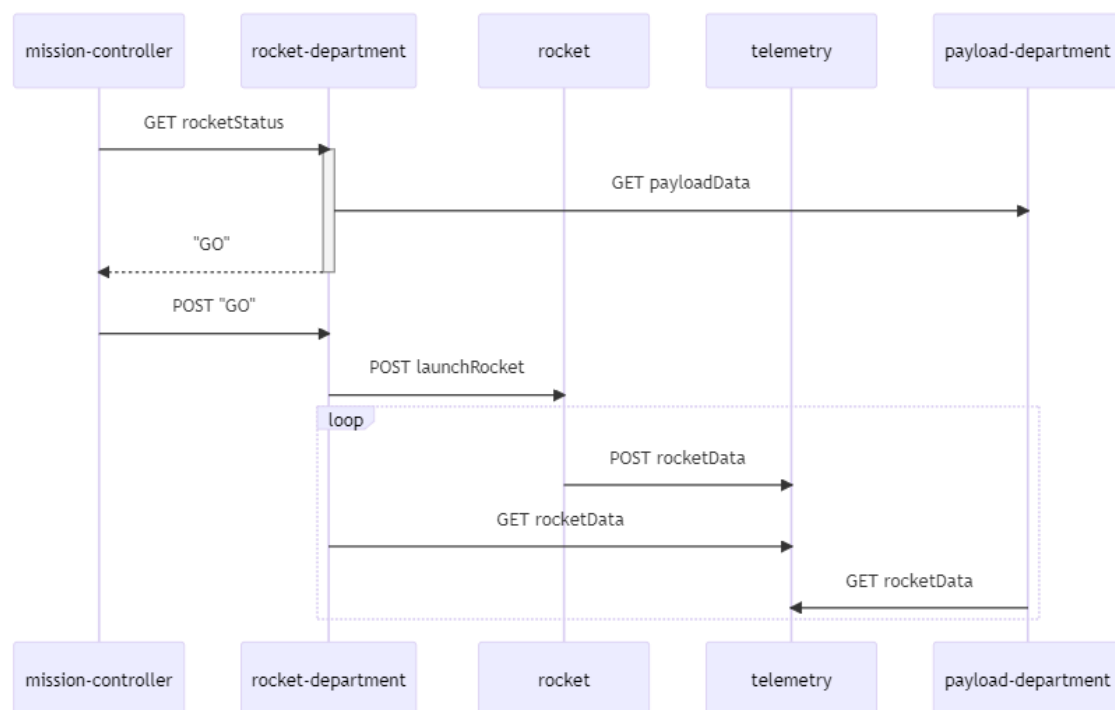


Diagramme de séquence 2 : La partie avec weather-department a été supprimée car restée inchangée depuis la semaine précédente. On suppose également sur le diagramme que le GO/NOGO pool renvoie tout le temps GO pour la simplification de lecture.

Semaine 3

Cette semaine, nous avons commencé par un refactoring des modules Payload et Rocket afin de repartir sur de meilleures bases et de mieux pouvoir justifier tous nos choix. Pendant le refactoring, nous avons rajouté une base de données au payload-department afin qu'il puisse récupérer et stocker les informations concernant le payload (Scénario 11). Pour le moment l'objet payload et encore compris dans l'objet rocket, c'est donc la rocket qui envoie les informations du payload au payload department en permanence en même temps d'envoyer les informations complète de la rocket à la telemetry.

Le rocket-department demande régulièrement des informations à la telemetry afin de pouvoir vérifier que la fusée s'est bien détachée ; si ce n'est pas le cas, elle va envoyer un message d'erreur au mission-commander. Nous avons donc rajouté le module mission-commander-department qui va recevoir ce message d'erreur et renvoyer au rocket department un ordre de détruire la fusée (Scénario 8) et le rocket-département va envoyer un ordre d'autodestruction à la rocket à la réception de cet ordre.

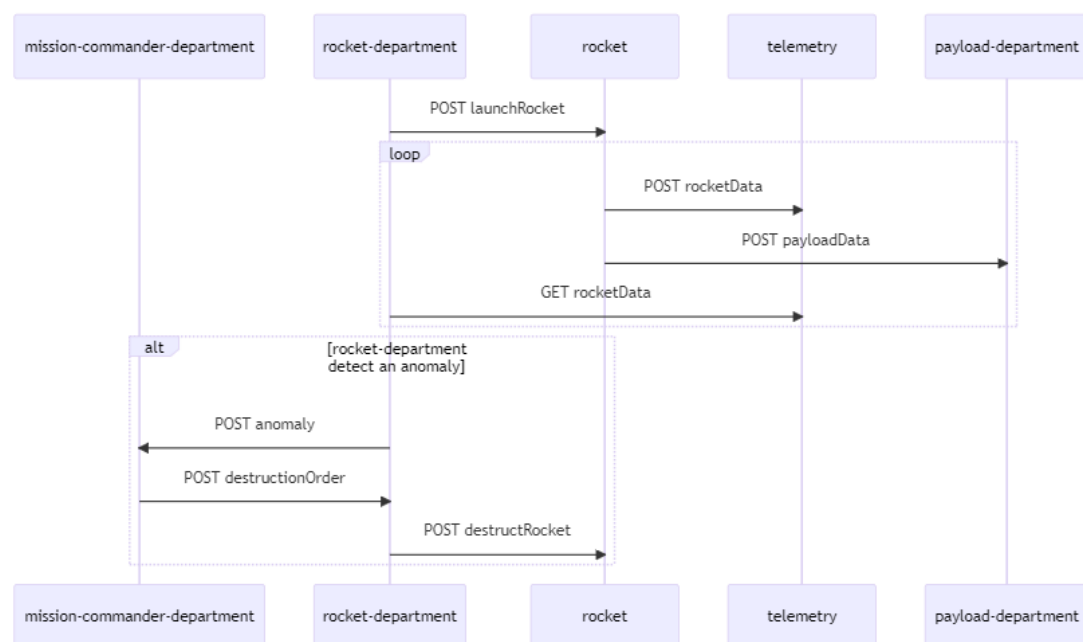


Diagramme de séquence 3 : La partie avec le mission-controller avant le POST rocket a été supprimée car inchangée depuis le dernier diagramme. On suppose également que mission-commander accepte toujours la proposition de destruction de rocket-department.

Les semaines suivantes

Par la suite, le rocket-department devrait pouvoir vérifier la vitesse de la fusée grâce aux informations reçues de la telemetrie (Scénario 12).

Nous réfléchissons également aux meilleures stratégies pour séparer la Rocket tout en conservant ses informations pour les Scénarios 9 et 10. Avoir deux objets serait une idée, mais il faudrait qu'ils soient chacun connectés afin que seul un étage de la fusée ne se vide de son carburant à la fois. Il est aussi possible que nous ayons besoin de représenter par la suite l'objet Payload indépendamment de l'objet Rocket, comme c'est le cas actuellement.