

Projet SOA Selene

RAPPORT INTERMEDIAIRE - TEAM E

William D'ANDREA – Laurie FERNANDEZ – Emma GLESSER – Arthur SOENS

02-10-2022

Rappel du contexte

Sur cette page vous retrouverez un rapide rappel du contexte du projet Selene, les explications sur notre travail personnel commencent à la page suivante.

PERSONA

4 persona ont été définies dans le projet Selene pour le moment :

- **Deke**, le chef des opérations, responsable de la division de soutien à la vie sur Terre, s'assurant de la **sécurité des équipages**.
- **Buzz**, le commandant du village lunaire, responsable de la **sécurité de l'équipage du village et des missions lunaires**.
- **Dorothy**, la responsable de la logistique, responsable des missions de **transit Terre-Lune**, s'assurant de la **durabilité à long terme de l'initiative du Moon Village**.
- **Jim**, un astronaute du Moon Village, responsable des provisions et de la logistique, s'assurant du **bon déroulement des opérations** sur la base.

USER STORIES A IMPLEMENTER

Lors de ces deux premières semaines de développement, 5 user stories nous ont été demandées à développer :

1. En tant que **Deke**, je veux **m'assurer à distance que les systèmes de support de vie fournissent des conditions de vie dans les modules lunaires habités**, afin que les occupants puissent survivre.
2. En tant que **Dorothy**, je veux **organiser et suivre les missions de réapprovisionnement**, afin de fournir tous les matériaux et ressources nécessaires à la réussite d'une mission.
3. En tant que **Buzz**, je veux **observer et contrôler l'état des modules de la base** afin d'**effectuer les changements nécessaires** pour soutenir l'équipage **et les activités quotidiennes de l'équipage**.
4. En tant que **Deke**, je veux être capable d'**isoler physiquement tout module de la base** afin de préserver l'intégrité de la base en cas de défaillance d'un des systèmes de support de vie.
5. En tant que **Jim**, je veux **garder une trace de l'inventaire des ressources disponibles sur la base** afin de ne jamais tomber à cours de ressources vitales.

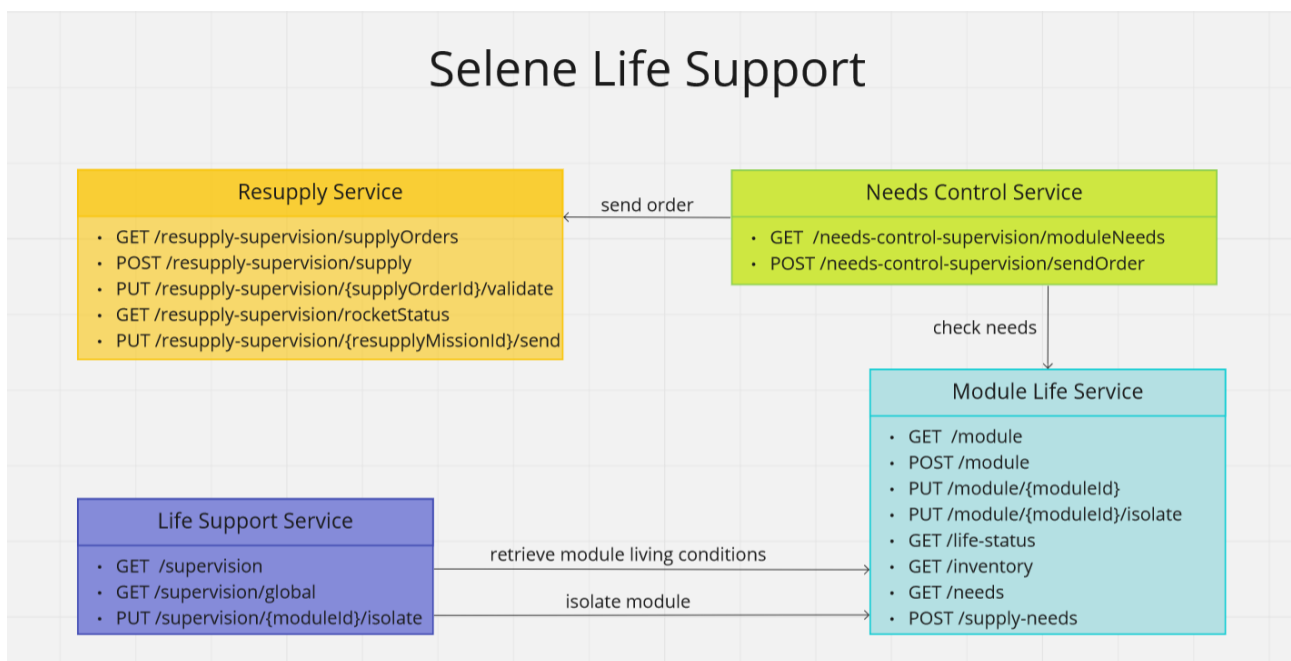
Hypothèses faites par l'équipe

Durant la lecture du sujet plusieurs hypothèses ont été faites par le groupe :

- Dans un premier temps, un module ne contient qu'un seul module de vie et pas d'habitants
- Une commande ne contient qu'un seul produit correspondant au seul support de vie des modules
- On peut rajouter des commandes à une mission de réapprovisionnement tant que celle-ci a un statut « En cours de préparation »
- Si aucune mission de réapprovisionnement n'a un statut "En cours de préparation" alors une nouvelle mission est créée
- Certaines opérations ne sont pas automatisées pour conserver l'aspect vérification par un humain comme ce serait le cas dans la vraie vie. C'est le cas de l'envoi de commande qui devra être réalisé par Buzz et non pas par Needs Control Service dès la réception de la liste des besoins. C'est également le cas du ravitaillement des modules qui devra être fait par un humain et non pas par un service dès la réception de la mission de ravitaillement pour en valider la préparation.

Architecture développée

DECOUPAGE DU SUJET EN MICRO-SERVICES



Les user stories ont montré qu'il y avait plusieurs grands domaines à traiter dans notre architecture : l'organisation des missions de ravitaillement, la gestion individuelle, la supervision des besoins et le contrôle des modules lunaires. Notre découpage en micro-services s'est basé dessus afin d'isoler chaque domaine métier dans un service.

4 micro-services différents ont ainsi été différenciés dans notre vision de l'architecture du projet Selene :

- Le Resupply Service
- Le Needs Control Service
- Le Module Life Service
- Le Life Support Service

Le Resupply Service

Le Resupply Service est un service utilisé actuellement uniquement par la persona Dorothy. Il est focalisé sur la préparation et la livraison des commandes qui nous sont passées depuis le Needs Control Service.

Le Needs Control Service

Ce service est actuellement uniquement utilisé par la persona Buzz présent sur la base lunaire. Il permet de récupérer une liste des besoins des modules qui sera ensuite utilisée pour passer des commandes à Dorothy sur Terre pour des demandes de réapprovisionnement à travers le Resupply Service.

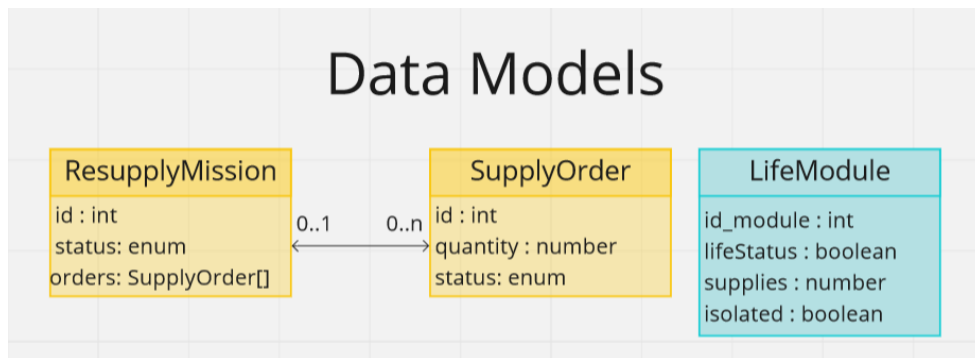
Le Module Life Service

Ce service est actuellement uniquement utilisé par la persona Jim présent sur la base lunaire. Il est focalisé sur la gestion individuelle des modules. Il sera contacté par le Needs Control Service pour récupérer la liste des besoins à remplir et par le Life Support Service pour contrôler les conditions de vie et isoler un module. Il permet également la réalisation d'un inventaire des provisions et du matériel sur la base lunaire.

Le Life Support Service

Ce service est actuellement uniquement utilisé par la persona de Deke présent sur Terre. Il permet la supervision des modules lunaires en contrôlant l'état des conditions de vie et d'éventuellement isoler un module. Il fera pour cela appel à Module Life Service afin d'obtenir les informations des modules et de les isoler.

MODELES DE DONNEES DEFINIS



3 modèles de données ont été définis dans le projet :

- Le ResupplyMission
- Le SupplyOrder
- Le LifeModule

Le Data Model ResupplyMission

Ce premier modèle de données accédé par le Resupply Service liste les différentes Resupply Mission existantes. Il est présent uniquement dans la base de données liée à Resupply Service. Ces missions possèdent 3 informations :

- Un identifiant **id** de la mission ou fusée
- Un statut **status** qui correspond à un des 3 états de livraison ci-contre : En cours de préparation (état par défaut à la création d'un ResupplyMission) - En cours d'acheminement (livraison en cours) – Terminée (livraison arrivée sur la Lune)
- Une liste de SupplyOrder **orders** correspondant à l'ensemble des commandes associées à la mission de réapprovisionnement

Le Data Model SupplyOrder

Ce second modèle de données accédé par le Needs Control Service correspond au contenu d'une commande. Il est présent uniquement dans la base de données liée à Resupply Service. Ses 3 attributs sont les suivants :

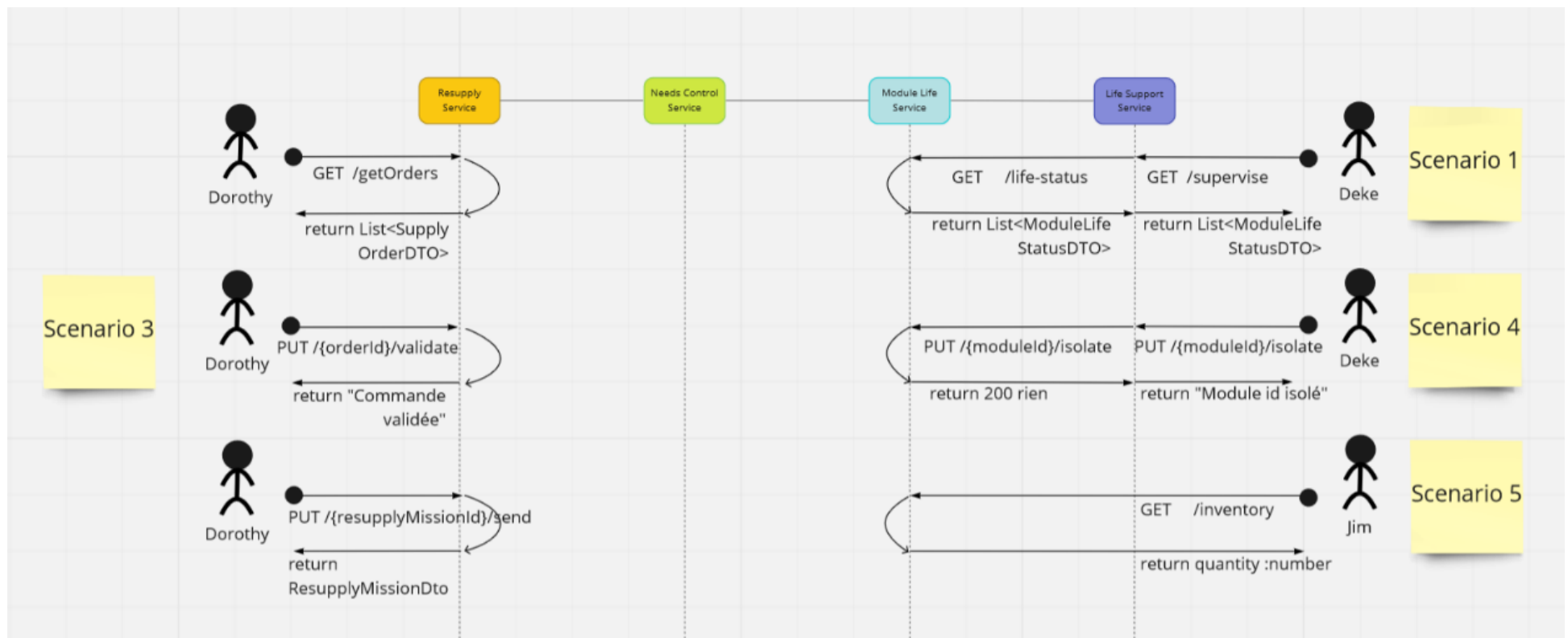
- Un identifiant de commande **id**
- Une quantité **quantity** correspondant à la quantité demandée en commande
- Un statut **status** pour savoir si la demande a été approuvée par Dorothy sur Terre

Le Data Model LifeModule

Ce troisième modèle de données accédé par le Module Life Service représente un module lunaire. Il est présent uniquement dans la base de données liée à Module Life Service. Les modules lunaires possèdent 4 attributs :

- Un identifiant **id_module** du module
- Un état des conditions de vie à l'intérieur du module **lifeStatus**, true si les conditions sont remplies, false sinon
- La quantité de provisions disponibles **supplies**
- Un statut **isolated** indiquant si le module est en isolement ou non

Compréhension et implémentation des user stories

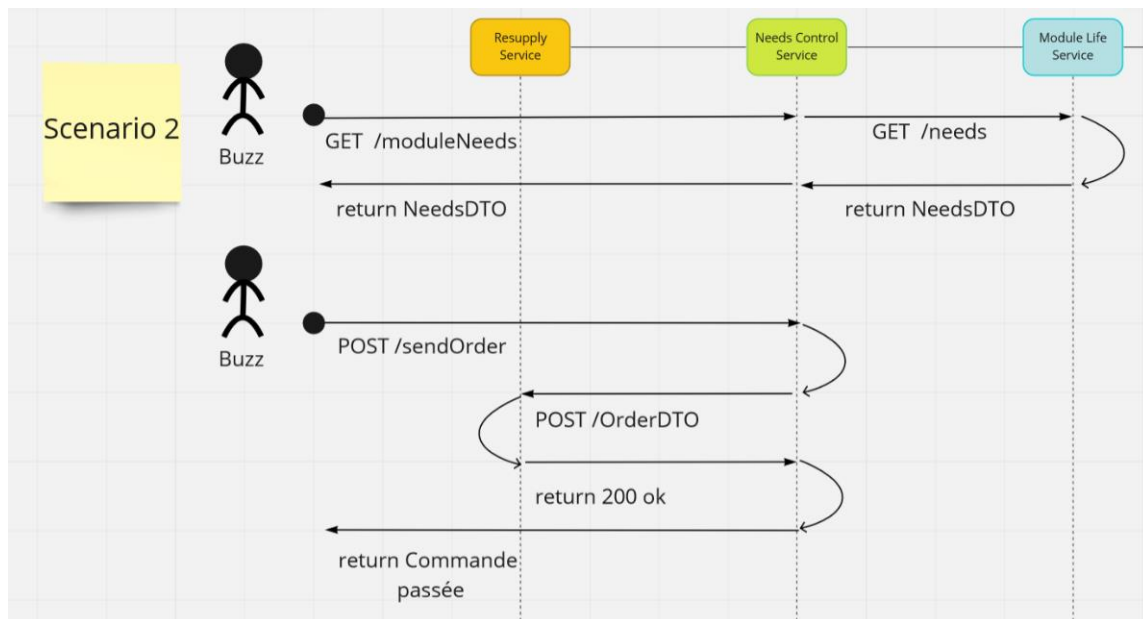


SCENARIO 1. VERIFICATION DE L'ETAT DES MODULES PAR DEKE

Cette vérification est faite à travers une unique requête de Deke au Life Support Service demandant l'état des modules, le service contacte Module Life Service qui lui renvoie l'état de tous les modules

- Le chef des opérations demande au Life Support Service à voir l'état des conditions de vie des modules
- Le Module Life Service reçoit une requête du Life Support Service demandant à voir l'état des conditions de vie des modules
- Le Module Life Service traite cette demande et renvoie la liste des états des conditions des modules au Life Support Service
- Le Life Support Service renvoie cette liste au chef des opération

SCENARIO 2. VERIFICATION DES BESOINS ET COMMANDE PAR BUZZ



Ce scénario est plus complexe que le précédent et est complété à travers 3 requêtes distinctes de Buzz :

- Une première requête de Buzz à Needs Control Service pour vérifier les besoins des modules, le service contacte Module Life Service qui lui renvoie la liste des besoins.
 - Le commandant du village lunaire demande au Needs Control Service à voir les besoins essentiels à la vie dans les modules
 - Le Module Life Service reçoit une requête du Needs Control Service demandant à voir les besoins essentiels à la vie dans le module
 - Le Module Life Service traite cette demande et renvoie la liste des besoins relatifs aux éléments essentiels à la vie dans le module au Needs Control Service (quantité)
 - Le Needs Control Service renvoie cette liste au commandant du village lunaire
- Requête de Buzz à Needs Control Service pour passer une commande (demande de réapprovisionnement)
 - Le commandant du village lunaire fait une demande de réapprovisionnement au Needs Control Service
 - Le Resupply Service reçoit une requête de réapprovisionnement de la part du Needs Control Service
 - Le Resupply Service traite cette demande et crée une commande en attente de validation et renvoie la commande créée.
- Requête de Buzz à Module Life Service pour réapprovisionner les modules
 - Le commandant du village lunaire fait une demande à Module Life Service pour réapprovisionner les modules
 - Le Module Life Service reçoit une requête de réapprovisionnement
 - Le Module Life Service traite cette demande et réapprovisionne autant de modules en manque que de provisions fournies

SCENARIO 3. VERIFICATION DES COMMANDES ET REAPPROVISIONNEMENT PAR DOROTHY

Ici aussi, le scénario de complétion de cette user story a été découpé en 3 requêtes de la part de Dorothy cette fois-ci :

- Requête de Dorothy demandant à voir les commandes
 - La responsable de la logistique demande au Resupply Service à voir les commandes
 - Le Resupply Service reçoit une requête demande à voir les commandes
 - Le Resupply Service traite la demande et renvoie la liste de toutes les commandes
- Requête de Dorothy pour valider une commande
 - La responsable de la logistique demande au Resupply Service pour valider une commande
 - Le Resupply Service reçoit une requête demande valider une commande
 - Le Resupply Service traite la demande, il valide la commande et l'ajoute à la liste des commandes de la prochaine fusée
- Requête de Dorothy pour envoyer la fusée/mission de réapprovisionnement
 - La responsable logistique demande au Resupply Service à passer l'état d'une fusée à "En cours d'acheminement"
 - Le Resupply Service reçoit une requête demandant à changer l'état d'une fusée
 - Le Resupply Service traite cette demande et passe l'état de la fusée à "En cours d'acheminement"

SCENARIO 4. ISOLEMENT PHYSIQUE D'UN MODULE A DISTANCE PAR DEKE

Ce scénario d'isolement d'un module par Deke se fait à travers une unique requête de sa part à Life Support Service pour isoler un module, le service contacte alors Module Life Service qui lui renvoie l'état du module

- Le chef des opérations demande au Life Support Service à isoler un module précis
- Le Module Life Service reçoit une requête du Life Support Service demandant à isoler un module
- Le Module Life Service traite cette demande, isole le module et renvoie son état
- Le Life Support Service renvoie le nouvel état du module au chef des opérations

SCENARIO 5. VERIFICATION DE L'INVENTAIRE PAR JIM

Enfin, la vérification de l'inventaire se fait grâce à une requête de Jim à Module Life Service pour vérifier l'inventaire de la base lunaire qui lui renvoie le total des provisions de la base

- L'astronaute responsable des provisions et de la logistique fait une demande de contrôle de l'inventaire à Module Life Service
- Module Life Service reçoit une requête demande à voir l'inventaire de la base
- Module Life Service traite cette demande et envoie une réponse contenant le total des provisions de chaque module

User stories développées

Pour ce premier rendu, les 5 user stories définies par le sujet ont été développées suivant l'implémentation expliquée précédemment.

Tests effectués

Pour valider notre implémentation des scénarios, notre démarche a été la suivante :

Dans un premier temps :

1. Des tests Swagger ont été effectuées pour vérifier la communication entre services
2. Un script d'intégration suivant les scénarios définis a été écrit dans un fichier `integration.sh` à l'aide d'opérations *curl*

Nos méthodes de test ont ensuite évolué vers la solution ci-dessous :

3. Un script Python permettant d'effectuer les différents appels au micro-services tels que le feraient nos persona a été réalisé selon les scénarios définis précédemment
4. Le script a été dockerisé afin de pouvoir le lancer dans un environnement contenant toutes les dépendances nécessaires
5. Le container Docker est lancé sur le même réseau que nos micro-services
6. Les logs de ce container sont récupérés une fois celui-ci éteint