

# Intégration des services

Rami, Antoine, Danial, Günther - Équipe #1

Ce document a pour but de montrer et d'expliquer le travail d'intégration produit pour le projet TELEGRAM. Celui-ci consistait en l'intégration de nos services de recherche de locations de voiture, de nuits d'hôtels et de billets d'avion avec les services d'autres équipes afin de constituer un système qui organise les voyages d'affaires les moins chers ; s'ajoute à cela une dimension de suivi des dépenses des voyageurs pour organiser leur remboursement ou non par leur entreprise.

## Flots d'intégration

### Recherche du moins cher

Il s'agit d'utiliser les six services produits en première partie de projet (les trois nôtres et trois venant d'autres équipes) afin de déterminer le moins cher des billets d'avion/locations de voiture/nuits d'hôtel proposés par l'ensemble des services.

### Divers paradigmes

Pour chaque catégorie (avion, voiture, hôtel), les deux services à intégrer étaient réalisés dans des paradigmes différents :

- Document (nous) et RPC (équipe #3) pour les billets d'avion.
- Ressource (nous) et RPC (équipe #5) pour les nuits d'hôtels.
- RPC (nous) et Document (équipe #8) pour les locations de voiture.

Chaque service a des attentes différentes (format de la date, granularité des localisations données, ...) et ont des réponses plus ou moins riches (descriptions, plaque d'immatriculation, options, ...).

La problématique pour cette partie est donc de faire cohabiter ces services en minimisant l'impact de leurs spécificités (dans l'idée qu'éventuellement nous pourrions bientôt ajouter les services des autres groupes et que nous voulons que cela se fasse sans trop de difficultés).

### Flux d'une catégorie

Pour cette partie, nous nous sommes inspiré de l'exemple<sup>1</sup> donné sur le site EIP sur lequel il est question de trouver le prêt le plus avantageux dans un panel de banques.

Les trois catégories ont été traitées en suivant le même schéma :

- Un flux "principal" accessible via une requête HTTP et qui retourne l'élément le moins cher provenant des services connus.
- Un flux spécifique à chaque service contacté.

---

<sup>1</sup> <http://www.enterpriseintegrationpatterns.com/patterns/messaging/ComposedMessagingExample.html>

Pour chaque catégorie, nous avons défini un objet "requête" générique qui est transmis aux flux spécifiques de cette catégorie, et un objet "réponse" générique qui est produit par les flux spécifiques et retourné dans une liste au flux principal.

Ces objets ont été constitués en suivant la spécification donnée dans la première partie du projet (par rapport à ce que les services doivent pouvoir faire).

Nous avons procédé ainsi pour fournir une interface commune entre tous les services, avec potentiellement certains champs qui ne seront pas remplis ou des informations reçues ignorées si les services ne respectent pas la spécification.

### Flux principal

- La requête arrive dans une file de messages, en attente d'être traitée.
- Nous démarshallons l'objet du corps en objet "requête" du type attendu.
- La requête est dispatchée aux différents services connus (le nôtre et celui d'un autre groupe) et nous attendons les réponses (qui sont des listes d'éléments "réponse").
- Les réponses sont attendues avec un délai ; si un service ne répond pas avant la fin du délai, on communique une erreur à une file et on considère que la réponse pour ce service est une liste vide. De cette façon, le flux principal n'est pas mis en péril à cause d'un partenaire inaccessible.
- Quand les réponses sont reçues, nous les agrégeons en concaténant les listes.
- Une fois toutes les réponses reçues, nous itérons sur les éléments de la grande liste pour récupérer le moins cher et le sauvegarder dans le corps de l'échange. Si la liste est vide, on sauvegarde un objet vide dans le corps de l'échange.
- On marshale le corps de l'échange en un document JSON. Fin du flux.

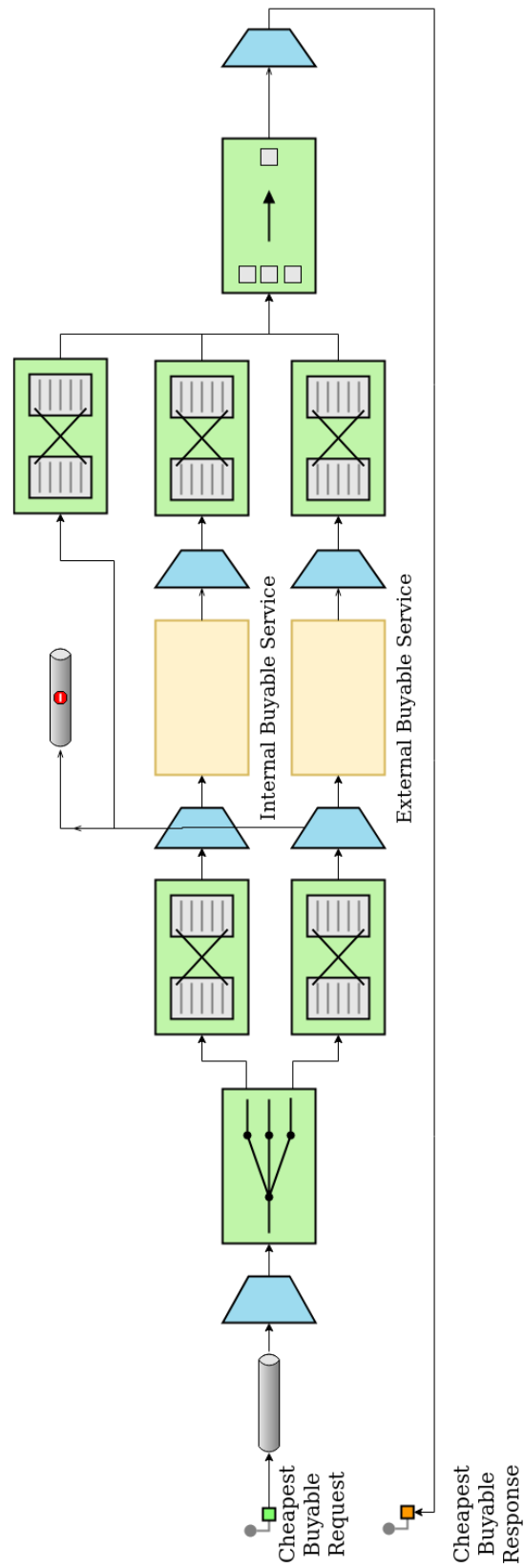
Plutôt que de concaténer des listes, nous aurions pu directement faire le choix du moins cher dans l'agrégateur, mais cela n'aurait pas respecté la sémantique de l'agrégateur.

### Flux spécifique

La procédure dépend du paradigme de service et du protocole choisis par l'équipe qui a développé le service. Nous n'avons pas pu abstraire le comportement de ces flux car les protocoles sont variés mais on peut considérer qu'ils ont une structure commune :

- Nous transformons l'objet "requête" générique transmis par le flux principal en un corps spécifique au service cible (un document JSON pour les deux services Document, un document XML-SOAP pour les services RPC, récupération des paramètres pour le service Ressource).
- Nous paramétrons puis envoyons la requête au service.
- Au retour, nous démarshallons la réponse spécifique au service en un objet "réponse" que nous avons défini. Fin du flux.

## Schéma générique des flux



*Flux principal + flux spécifiques pour une catégorie*

## Business Travels

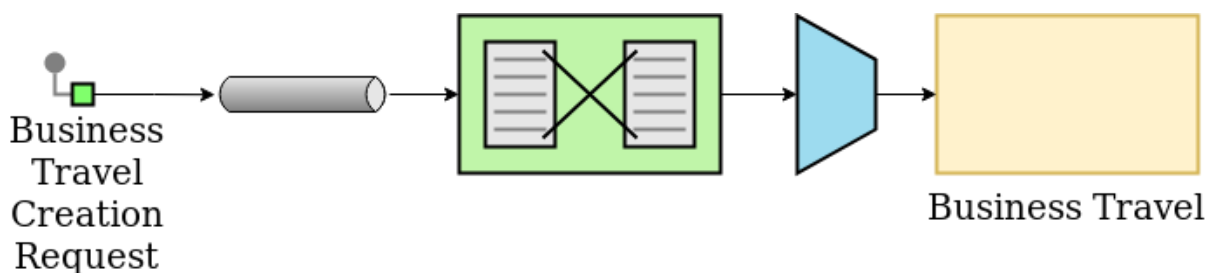
Cette partie concerne la création initiale du voyage d'affaire avec les dépenses prévues en billets d'avion, locations de voiture et nuits d'hôtel. Une fois prévues ces dépenses, l'employé soumet sa demande à un supérieur qui valide ou non la demande. Si la demande est validée, un *Travel Report* est créé.

### Flux implémentés

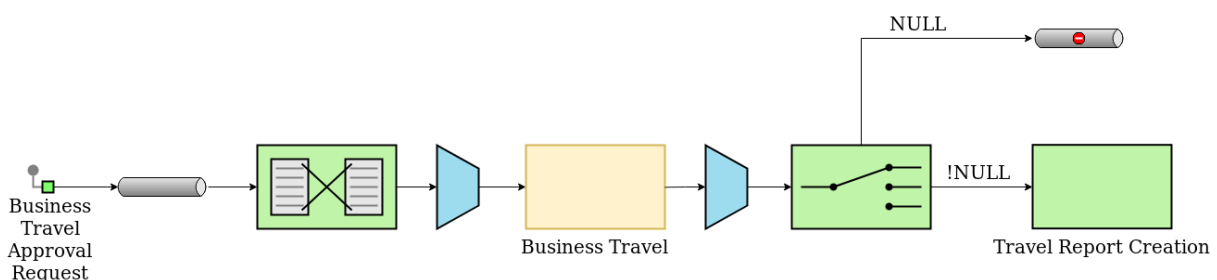
Pour cette partie, nous nous sommes reposés sur le service *Business Travels* produit lors de la première partie du projet. Nous supposons que l'employé a effectué au préalable ses recherches d'avions/voitures/hôtels au préalable grâce aux routes *Cheapest XXX*<sup>2</sup> et qu'il n'a plus qu'à les renseigner pour créer un voyage d'affaire.

Cela explique pourquoi nos routes pour cette partie sont relativement simples (accès via une requête au flux d'intégration, puis transformation si besoin, puis envoi au service) ; de tels flux existent pour la création et la demande de lister les voyages d'affaire en projet. L'unique à différer est celle de l'approbation d'un voyage d'affaire : nous validons, et si c'est bon, on initie la création d'un *Travel Report*. Sinon, le voyage d'affaire n'est pas validé et on en garde une trace dans une file d'erreur.

### Schémas des flux



Exemple de flux simple jusqu'au service Business Travel : Création d'un Business Travel



Approbation d'un Business Travel de la part d'un supérieur

## Travel Reports

Cette partie concerne le suivi des voyages d'affaire en cours. L'employé peut transmettre les pièces justificatives de ses dépenses au service. Au terme du voyage, l'employé demande à être remboursé. Si tout est en règle (dépenses inférieures ou égales à ce qui est prévu), le système effectue le remboursement ; sinon, il est demandé

<sup>2</sup> XXX étant soit *Flights*, soit *Cars* ou soit *Hotels*

une justification à l'employé puis il est décidé si oui ou non l'excédent est remboursé. Si il y a remboursement, les justificatifs sont stockés dans un dossier via FTP.

## Choix d'implémentation

Pour cette partie, nous avons créé un nouveau service, *Travel Reports*. Nous avons choisi pour lui le paradigme Document car il y a plusieurs opérations que l'on peut appliquer sur un rapport de voyage (ajout de justificatifs, clôture, approbation, refus, remboursement, archivage, ...) qui auraient surchargé le verbe POST si on avait choisi le paradigme Ressource ; et nous avons écarté le paradigme RPC car le fondement de ce service est d'appliquer des modifications aux rapports de voyage, ce qui implique éventuellement que nous en ajouterons dans le futur, et nous ne souhaitons pas devoir changer de contrat pour cela.

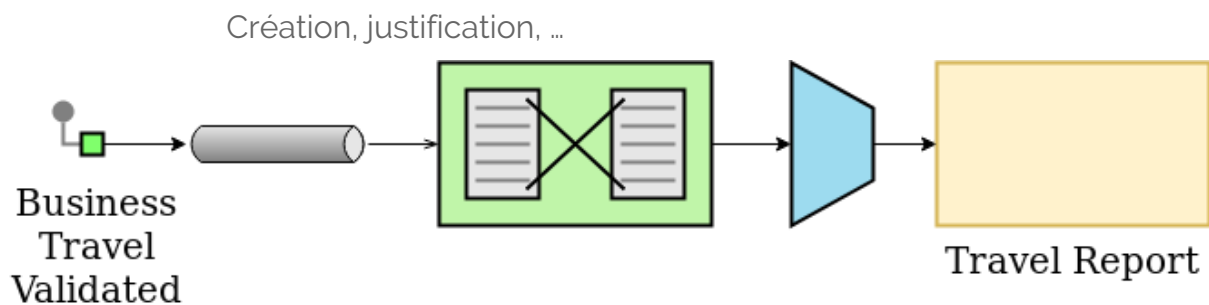
Les flux d'intégration permettent d'interagir avec ce service et implémentent l'archivage après remboursement.

Rétrospectivement, nous avons pensé que nous aurions pu implémenter toute la partie *Travel Reports* via des flux. Cependant, cela nous semble correct d'avoir un service qui s'occupe de la partie métier et les flux qui viennent faire des branchements, comme c'est le cas dans notre implémentation.

## Flux implémentés

Comme annoncé et de façon analogue aux *Business Travel*, nous nous avons plusieurs flux très simples qui ne font que interroger le service *Travel Report* : lister les rapports en cours, fournir une explication d'une sur-dépense, ajouter un justificatif de dépense. Un flux, la création de rapport, est appelé par l'approbation d'un voyage d'affaire et suit le même schéma que les flux précédemment cités.

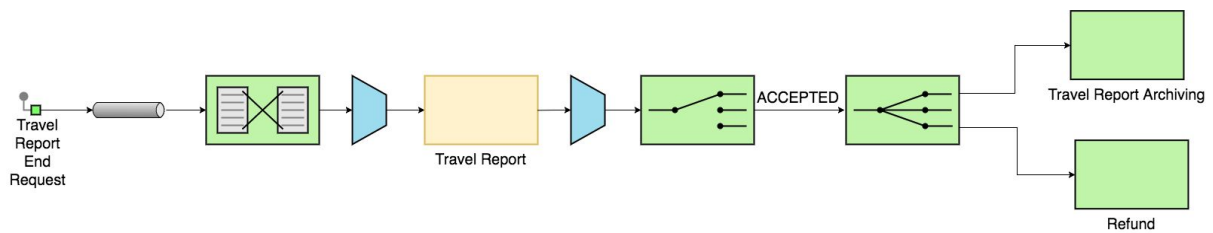
Les autres flux - la clôture du voyage, l'approbation d'une justification et le remboursement - sont décrits plus bas.



### *Exemple d'un flux simple : la création d'un rapport de voyage*

Ce schéma est semblable pour la création, l'envoi d'un justificatif de dépense, l'envoi d'une explication et pour lister les rapports en cours.

### Clotûre d'un voyage, approbation d'une explication



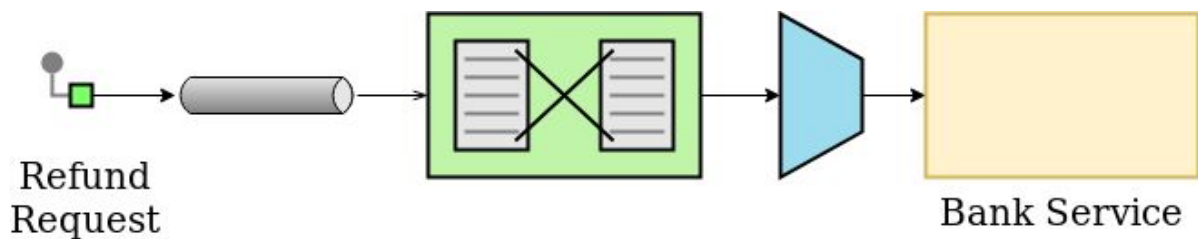
### Clôture d'un rapport de voyage

La clôture d'un voyage et l'approbation d'une justification ont le même schéma car ils ont la même logique mais pas la même condition (pas la même route de Travel Report à contacter, donc pas les mêmes requêtes/réponses).

Rétrospectivement, nous aurions pu extraire dans un flux à part ce qui suit après l'acceptation dans les deux cas (gain minime en terme de code, cela étant).

### Remboursement

Nous ne joignons pas de schéma de ce que nous avons fait pour ce flux car il se limite pour notre implémentation de l'affichage d'une trace sur la console. Dans une implémentation réelle, ce flux contacte la banque pour effectuer un virement.



### Flux de remboursement avec contact à la banque

# Organisation du travail

Le tableau ci-dessous indique le rôle de chacun dans la partie "intégration" du projet.

Rami	Implémentation de flux de Business Travel, des tests d'intégration des flux spécifiques aux avions, hôtels et voitures ; mise en place de scripts de build & run.
Antoine	Implémentation des flux principaux et spécifiques aux services pour les billets d'avion, les nuits d'hôtel et les locations de voiture.
Danial	Implémentation des flux spécifiques aux services Business Travel & Travel Report et mise en place des scénarios.
Günther	Implémentation de certains flux de Business Travel et mise en place des scripts et scénarios.

Après concertation et suivant l'appréciation de chacun de son propre travail et du travail des autres, nous sommes d'accord que nous avons chacun réalisé 25% du travail final.