

AL –SI5-team-h

JRAD Yassine
DELABY Maël
SIMON Fabrice
VASSEUR Adrien

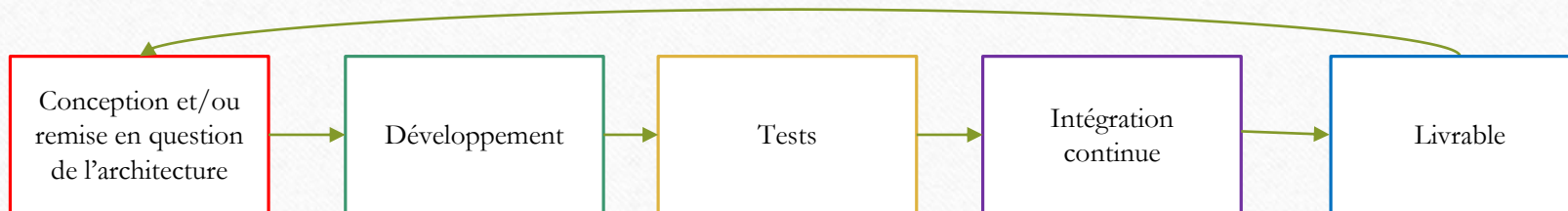


Travel Agency API

- Pouvoir réserver des trajets de trains pour un groupe de client

Organisation de l'équipe

- Tous les membres de l'équipe ont travaillé sur toutes les parties du projet en développement
- Méthode de travail :



Architecture choisie

- Architecture en micro-services
 - Pourquoi ?
 - Plus flexible (chaque service est indépendant) -> facilite la maintenance
 - Libre choix technologique sur chaque service
 - Monolithe
 - Non choisi mais envisageable car le projet n'est pas encore assez complexe
 - Aurait été un meilleur choix pour un début

Stack technologique et outils

- Langage : NodeJS
 - Plus léger -> pour tester les micro-services et pour le déploiement
 - Équipe a une préférence pour NodeJS -> permet de développer + rapidement
- TravisCI : Intégration continue
 - Simple à configurer
- Containerisation : Docker
 - Machine virtuelle léger

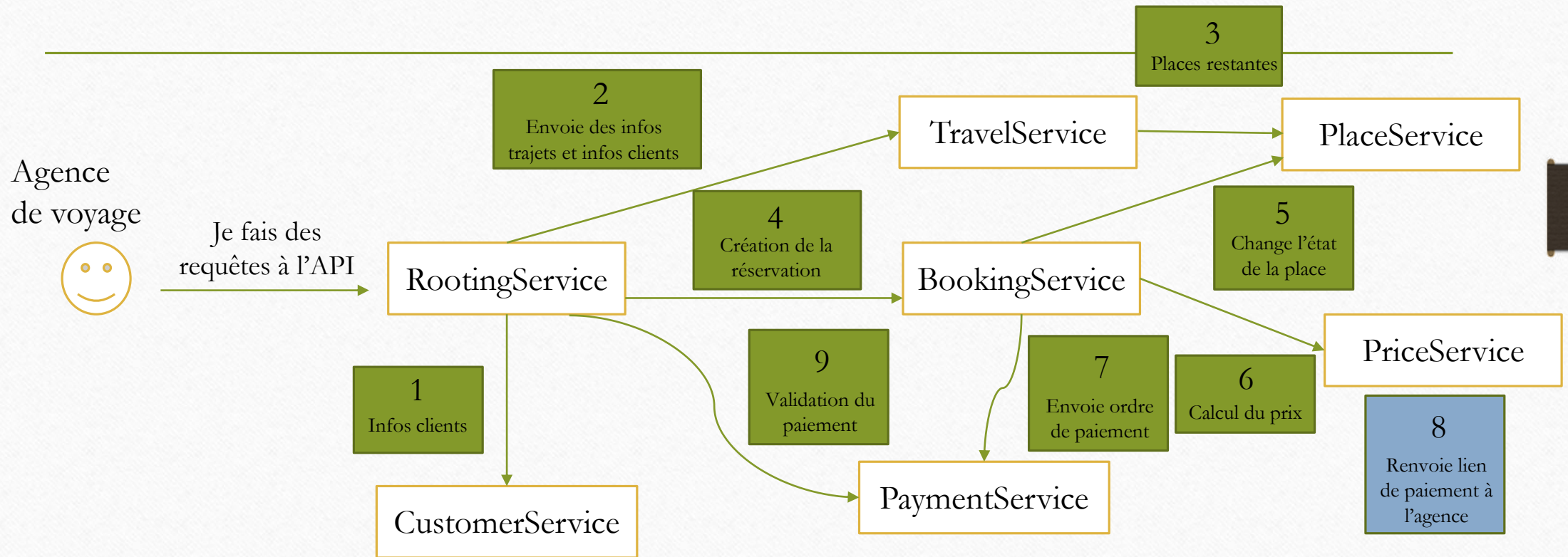
Stockage de données

- Local JSON -> toutes les données sont stockées en local pour l'instant

Scénario

- Scénario choisi :
 - Voyage d'un groupe de 3 personnes d'une même famille voyagent entre Nice et Brest avec un vélo et une PMR
 - L'agence commence par envoyé une requête afin de trouver les combinaisons de trains pouvant convenir
 - Elle choisit parmi les trajets proposés et envoi trois demandes de réservations qui lui retourne trois liens de paiements
 - Elle exécute le paiement des réservations en exécutant les liens et reçoie la confirmation que le paiement à bien été pris en compte

Scénario avec notre archi actuelle



Demos

Perspectives

- Ajout d'un nouveau service d'envoi de mail
- Continuer les autres scénarios
- Envisager un bus évènementiel pour réduire les liens entre micro-services (éviter le spaghetti)
- Stockage de données :
 - SQL relationnel pour stocker les données (clients, trajets, ...)
 - NoSQL envisagés si messages évènementiels

Merci ! Avez-vous des questions ?

