



Huyen's Industry

FLIGHT RESERVATIONS

github.com/juliehuyen/M2-AWOS-FlightReservations



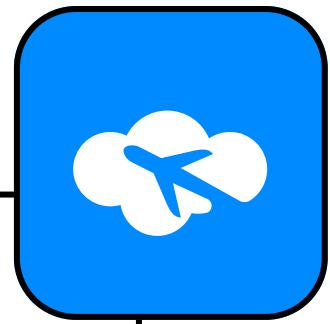
CHOIX DES MICROSERVICES



✗ **Couplage fort** : une modification peut impacter tout le système

✗ **Scalabilité limitée** : impossible de faire évoluer indépendamment certaines parties

MONOLITHIQUE



✓ **Modularité** : chaque fonction métier est isolée, testable, et évolutive indépendamment

✓ **Scalabilité** ciblée : possibilité de scaler seulement les services fortement sollicités

✓ **Résilience** : en cas de panne d'un service, le reste de l'écosystème continue à fonctionner

MICROSERVICES

VUE D'ENSEMBLE DES SERVICES CRÉÉS

SEARCH

FLIGHT

PRICING

INVENTORY

RESERVATION

PAYMENT

CHECK-IN

BOARDING

SEATING

BAGGAGE

CLIENT

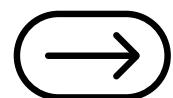
REFERENCE DATA

ACCOUNTING

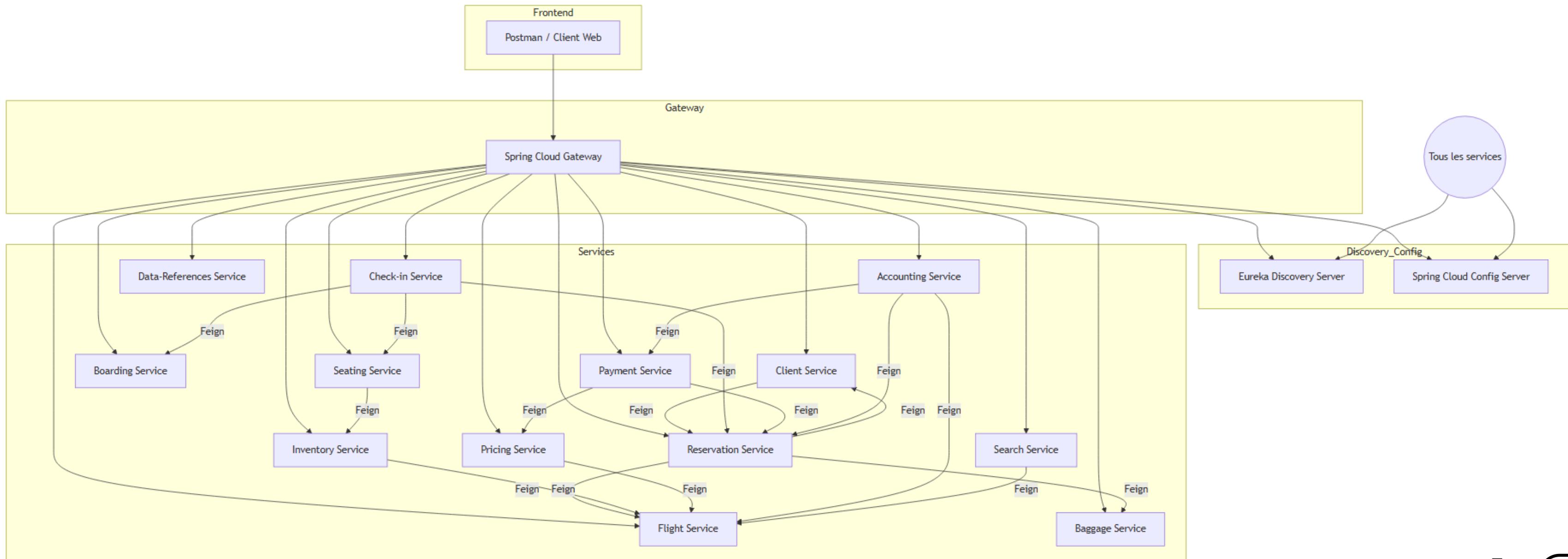
CONFIG

DISCOVERY

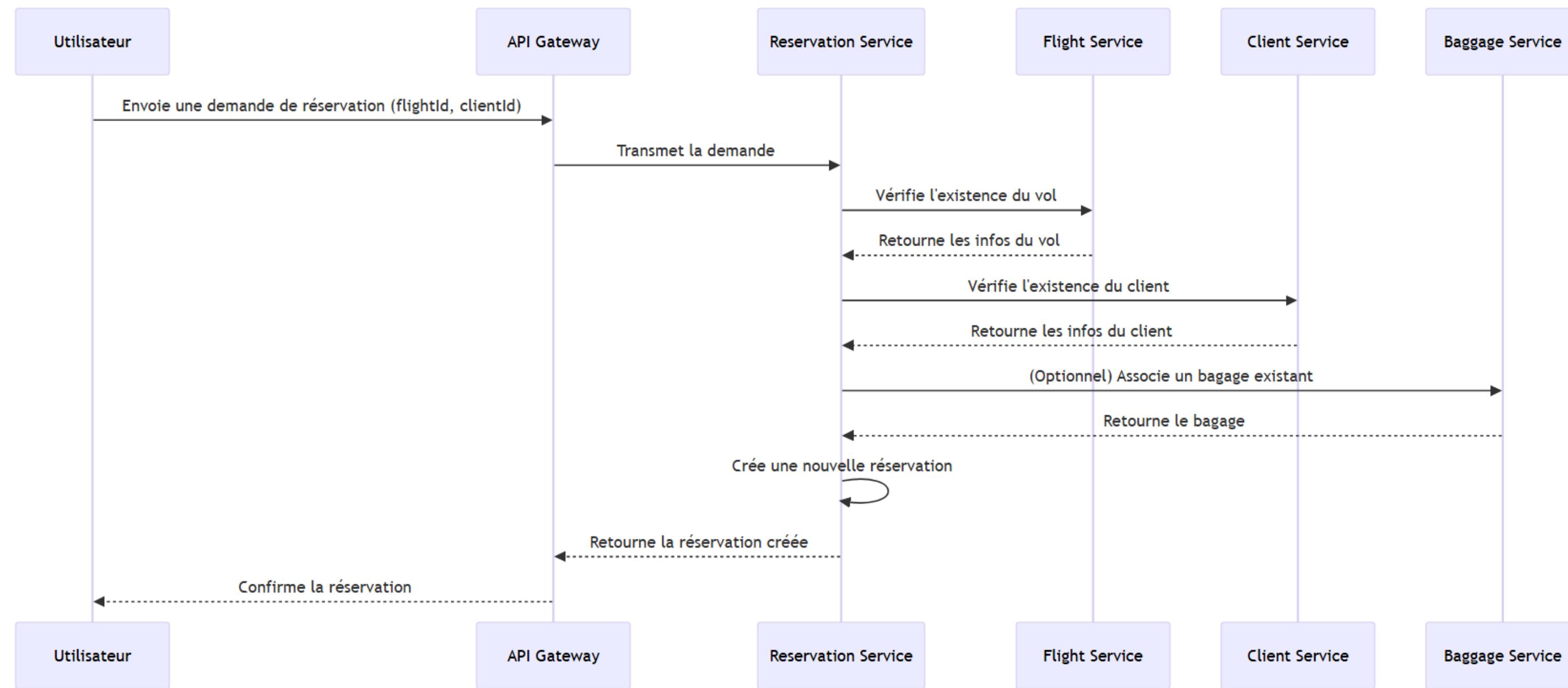
GATEWAY



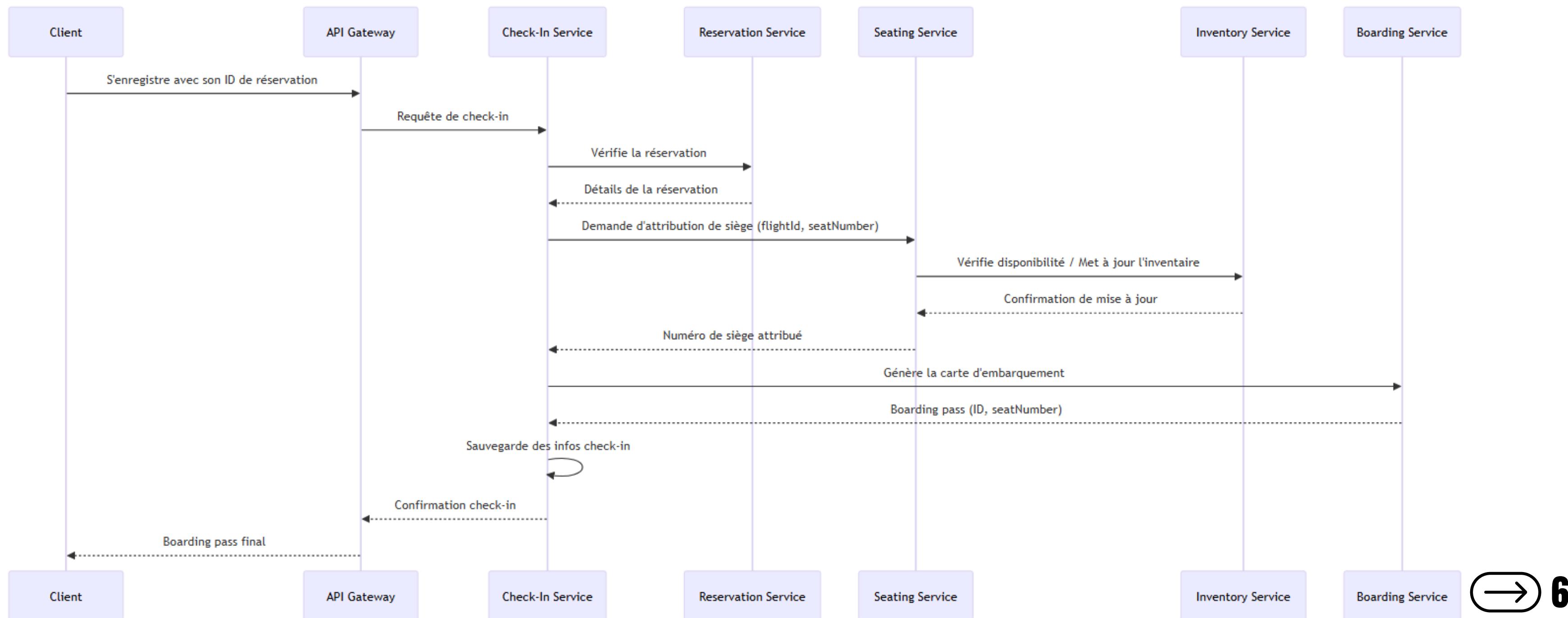
SCHEMA D'ARCHITECTURE TECHNIQUE



ZOOM SUR UN PROCESSUS METIER



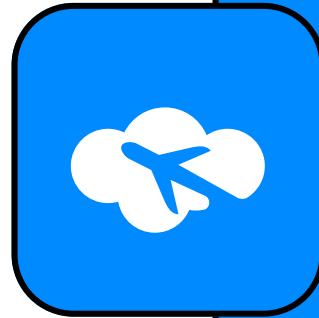
ZOOM SUR UN PROCESSUS METIER



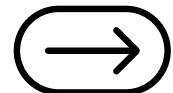


Huyen's Industry

DEMONSTRATION SUR POSTMAN



7





LES AVANTAGES DE NOTRE SOLUTION



Maintenance facilitée

Changement d'un service sans impacter tout le système



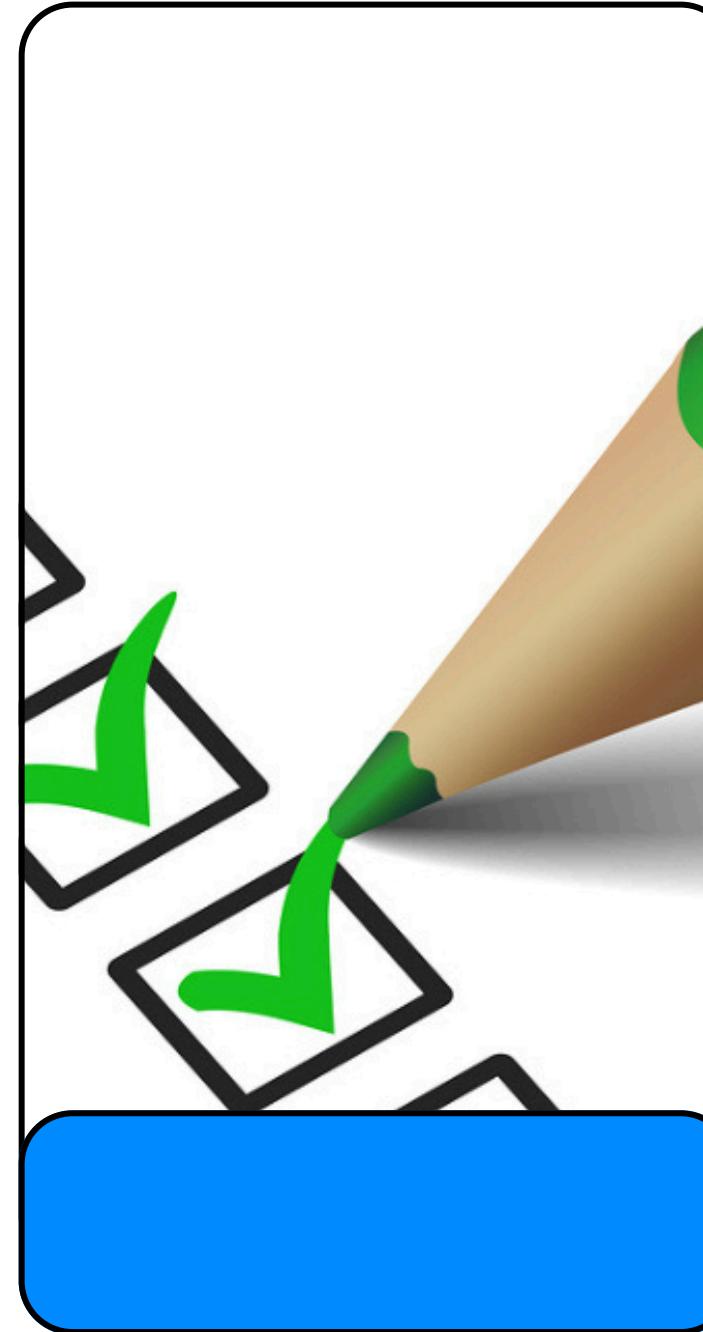
Evolutivité future

Remplacement d'un service



Tests unitaires

Tests unitaires/mocks intégrés pour assurer la robustesse métier



RÉPARTITION

Répartition et collaboration

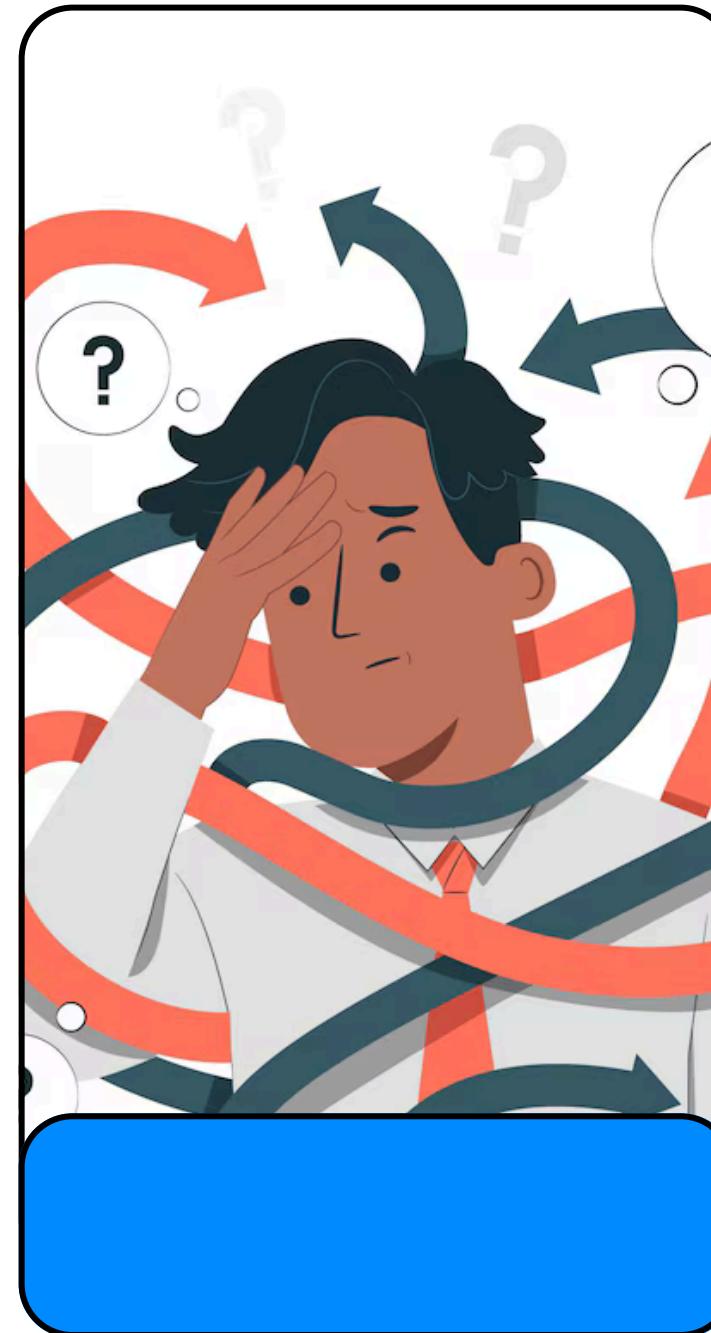
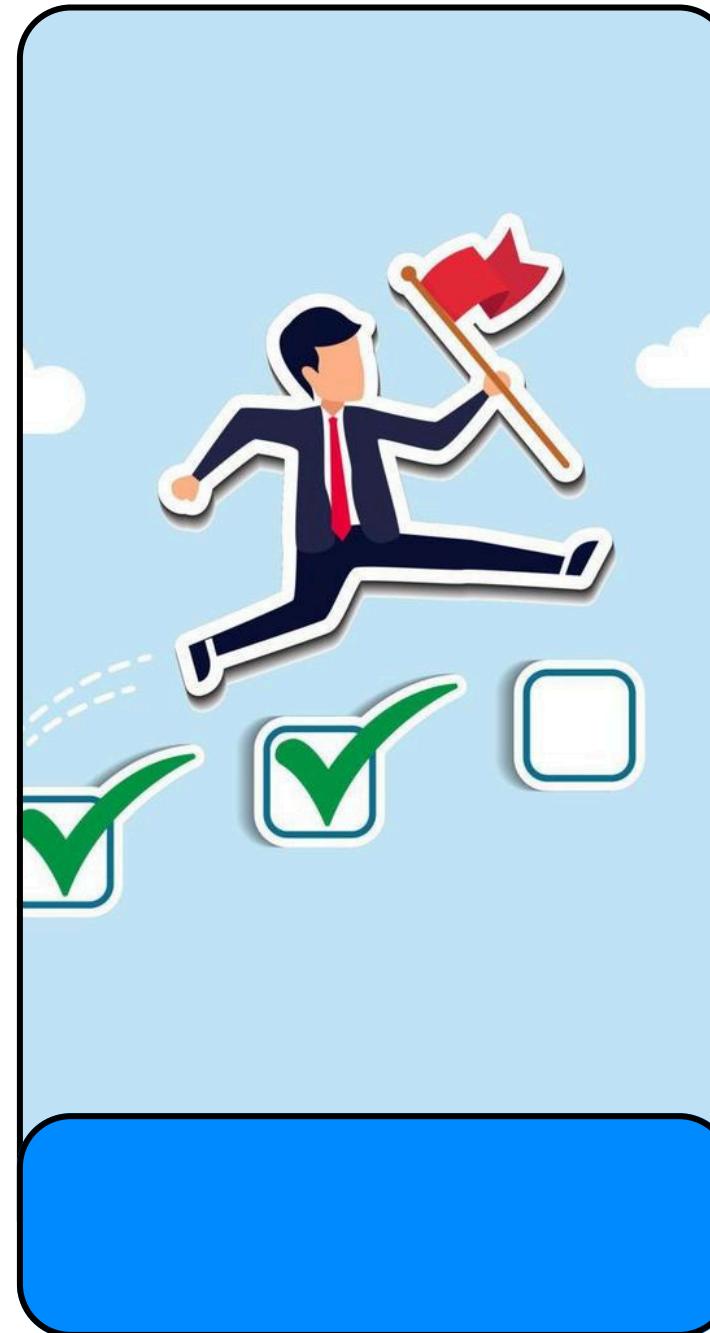


- Travail facilité par le fait d'habiter ensemble
- Conception de l'architecture discutée à deux
- Répartition de l'implémentation



Vérification continue

- Tests systématiques avec Postman
- Vérification des PR



MARGE DE PROGRESSION



Débuts compliqués

Peu à l'aise avec Spring Boot, difficulté à structurer les modules.



Appui sur les TP du cours

Meilleure compréhension grâce aux TP et à nos camarades



Répétition des patterns

Avec plus de 10 services : meilleure maîtrise progressive.



Problèmes techniques

Dépendances et gateway



THANK
you!

github.com/juliehuyen/M2-AWOS-FlightReservations