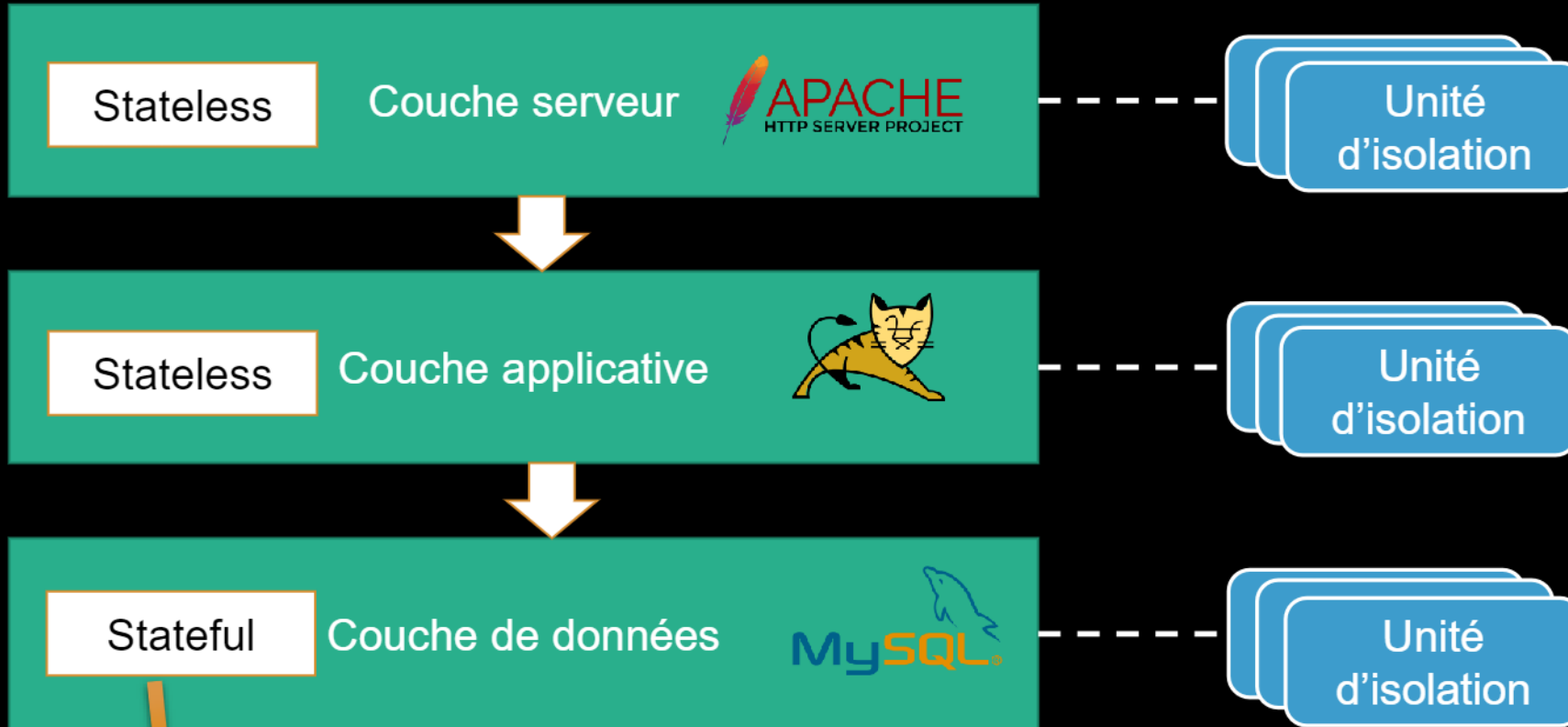


Cloud - 5

Communication inter-services

Djob Mvondo

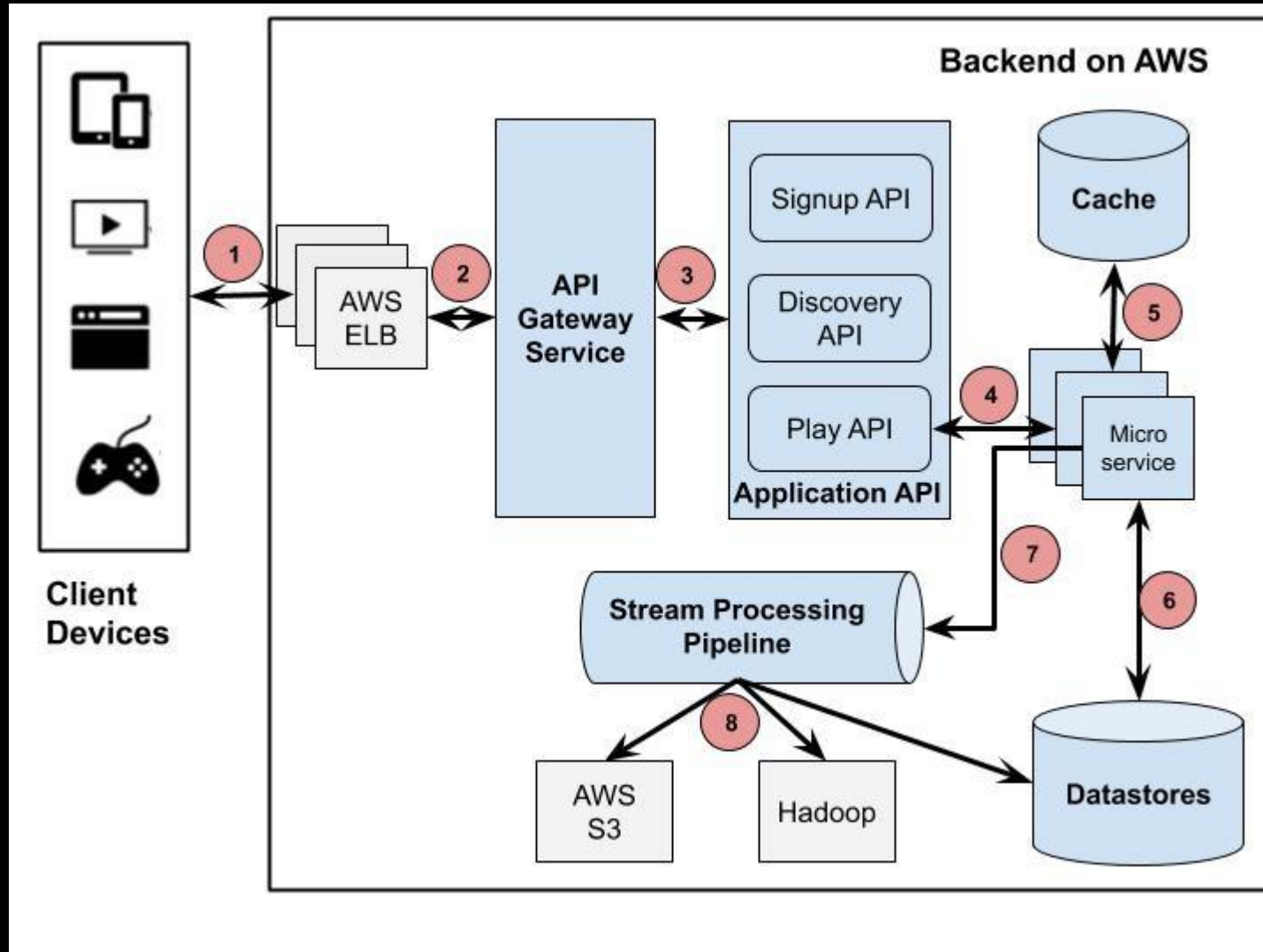
Rappel



Décomposer les couches en services pour optimiser chaque service en fonction de ses caractéristiques.

Augmenter le nombre d'isolation augmente le temps de synchronisation pour chaque requête

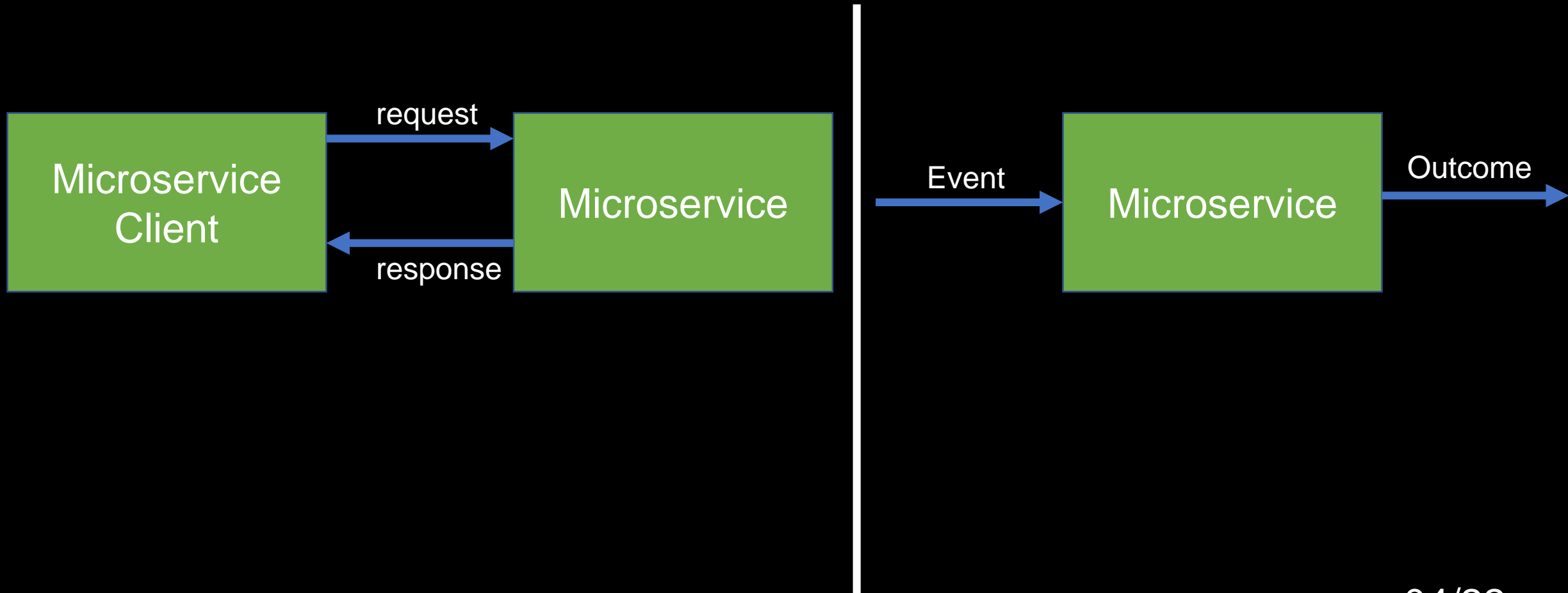
Comment les faire communiquer ?



Netflix backend overview
<https://medium.com/swlh/a-design-analysis-of-cloud-based-microservices-architecture-at-netflix-98836b2da45f>

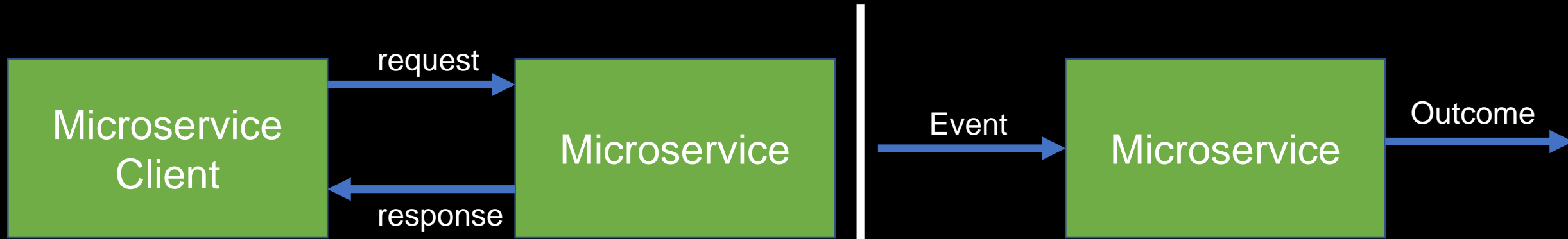
Comment les faire communiquer ?

Deux protocoles : **request/response** ou **event-driven**



Comment les faire communiquer ?

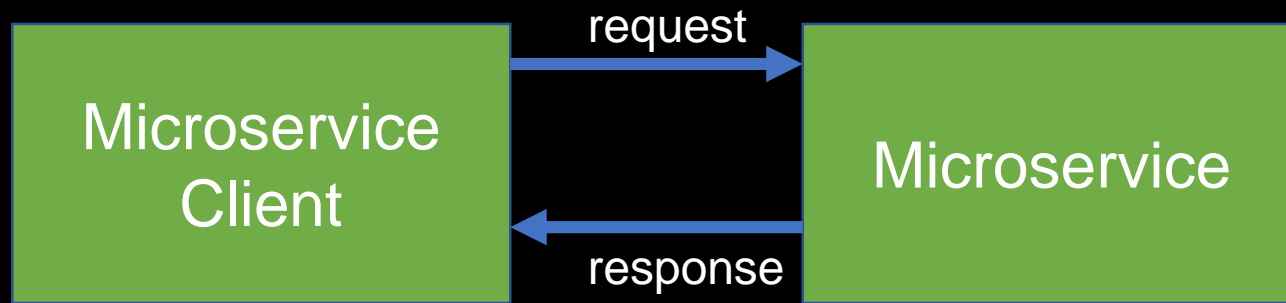
Deux protocoles : **request/response** ou **event-driven**



- + Le flow de contrôle est connu
- Modifier l'api du service entraine une modification chez tous les clients
- Faire un rollback d'une requête est compliqué
- Interdépendence entre services

Comment les faire communiquer ?

Deux protocoles : **request/response** ou **event-driven**



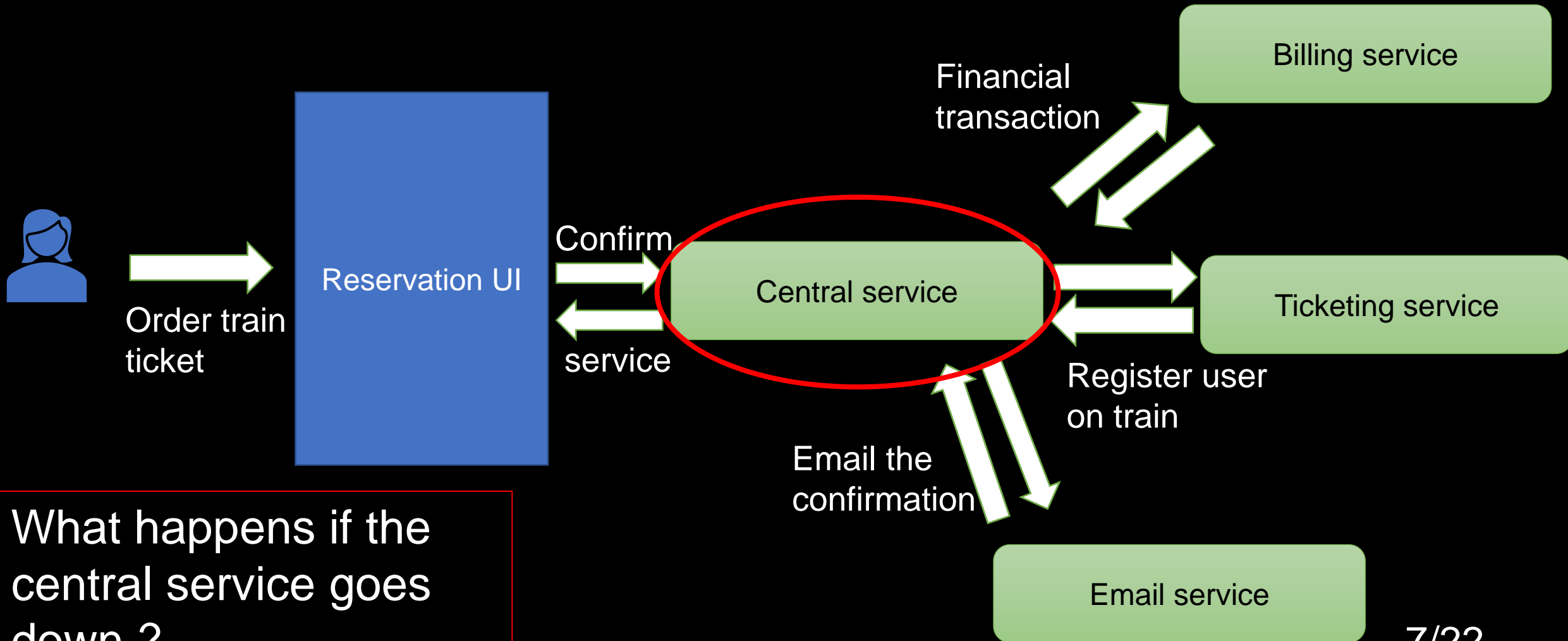
- + Le flow de contrôle est connu
- Modifier l'api du service entraine une modification chez tous les clients
- Faire un rollback d'une requête est compliqué
- Interdépendence entre services



- + Services indépendents
- + La source d'évènements ne connait pas les services
- + Rajout d'un service facile
- + Facile de rejouer les évènements
- Le flow de contrôle/données n'est pas connu
- Gestion des évènements complexe

Comment les faire communiquer ?

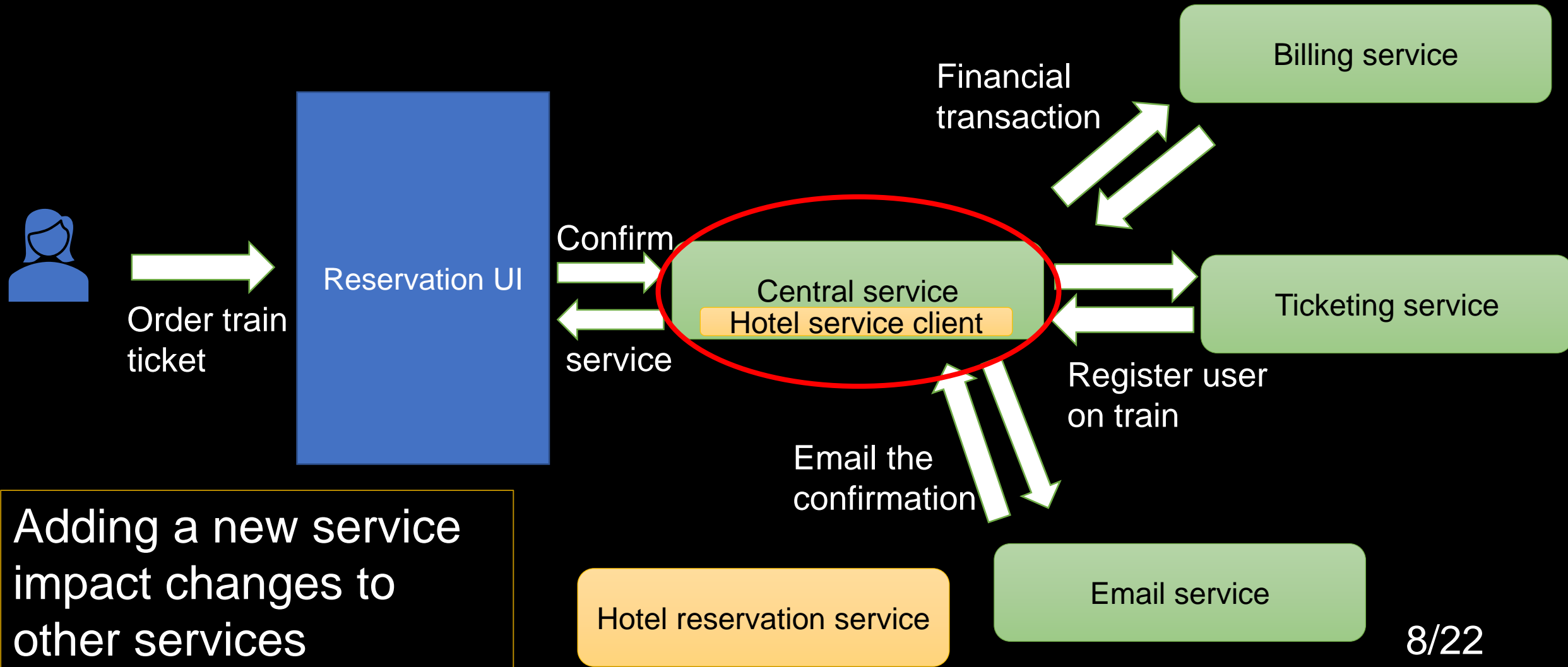
Cas pratique : Réservation d'un billet de train



What happens if the central service goes down ?

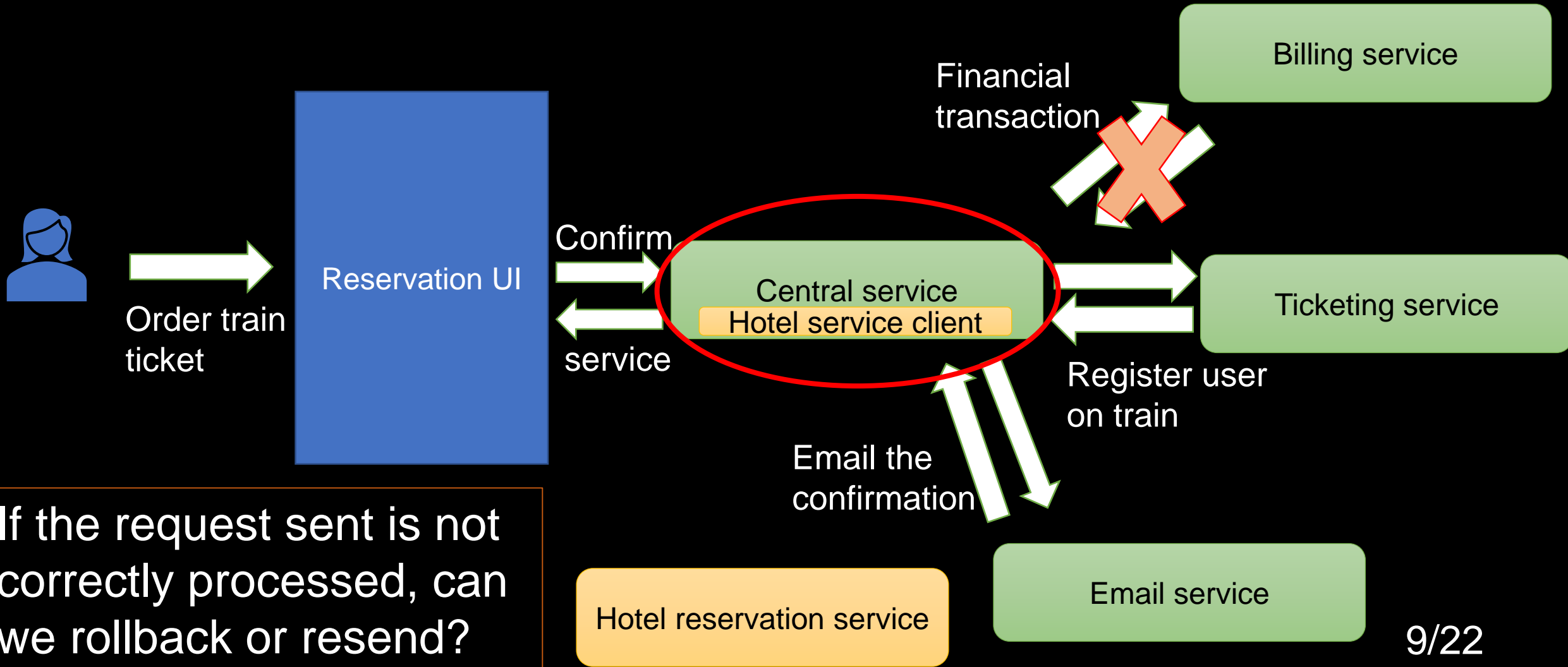
Comment les faire communiquer ?

Cas pratique : Réservation d'un billet de train



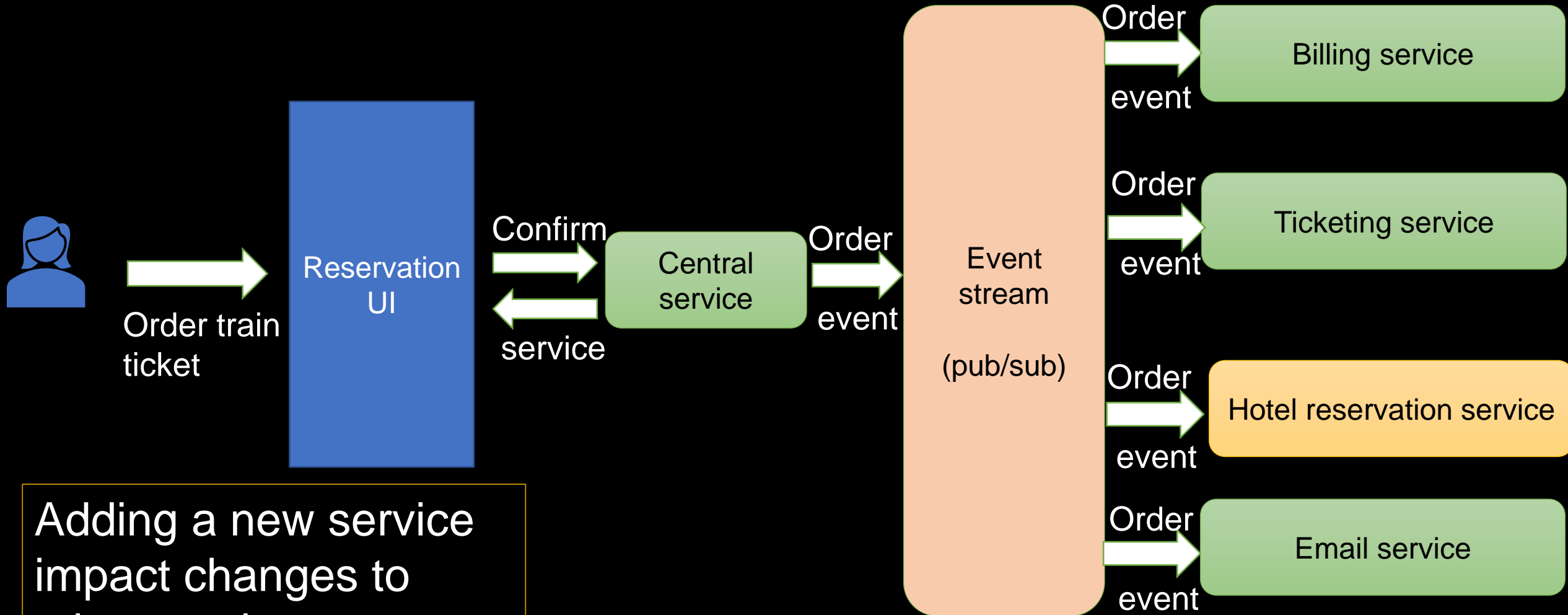
Comment les faire communiquer ?

Cas pratique : Réservation d'un billet de train



Comment les faire communiquer ?

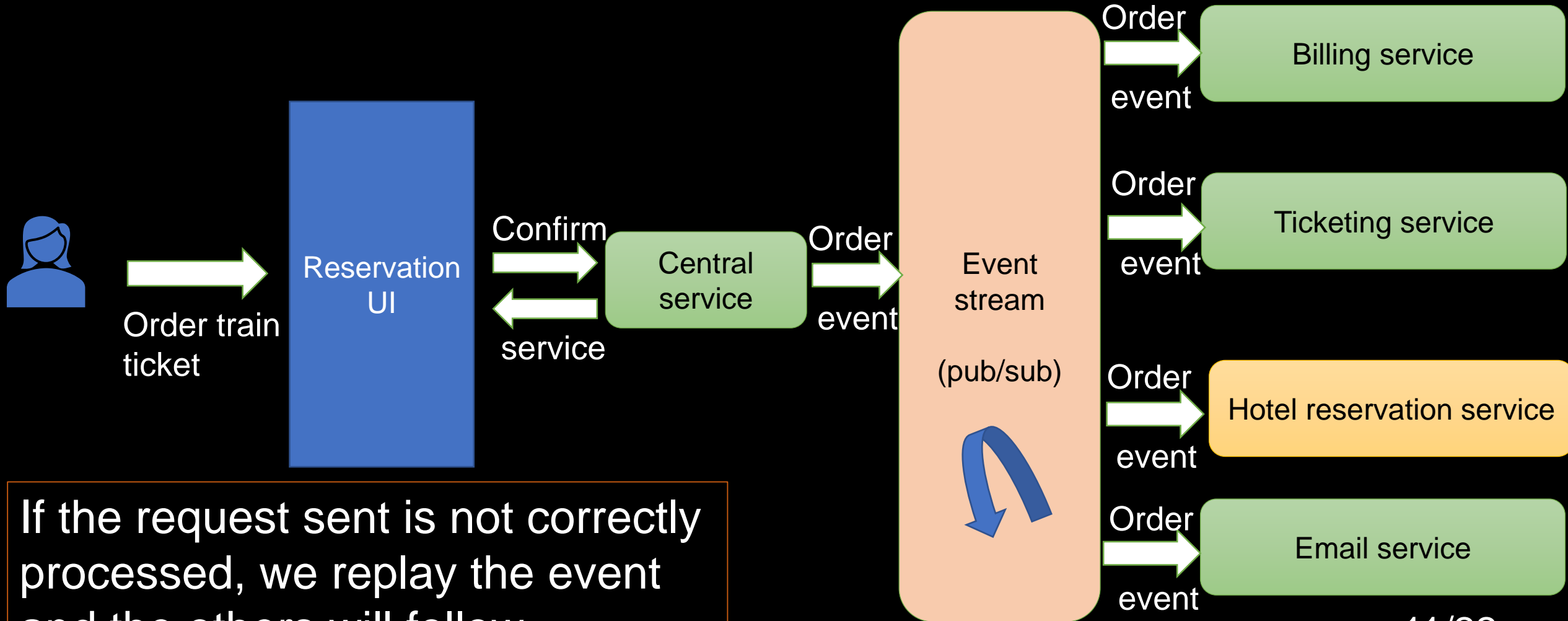
Cas pratique : Réservation d'un billet de train – event driven



Adding a new service
impact changes to
other services

Comment les faire communiquer ?

Cas pratique : Réservation d'un billet de train – event driven

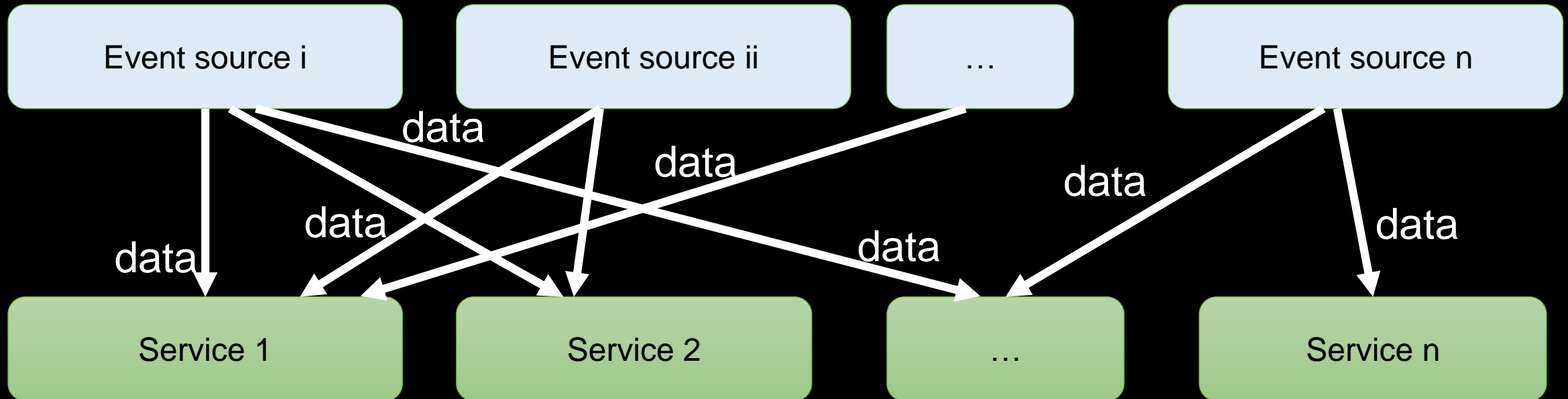


If the request sent is not correctly processed, we replay the event and the others will follow

Comment les faire communiquer ?

Il reste un problème, quel format de données utiliser ?

- ❑ Chaque service a ses **spécificités**

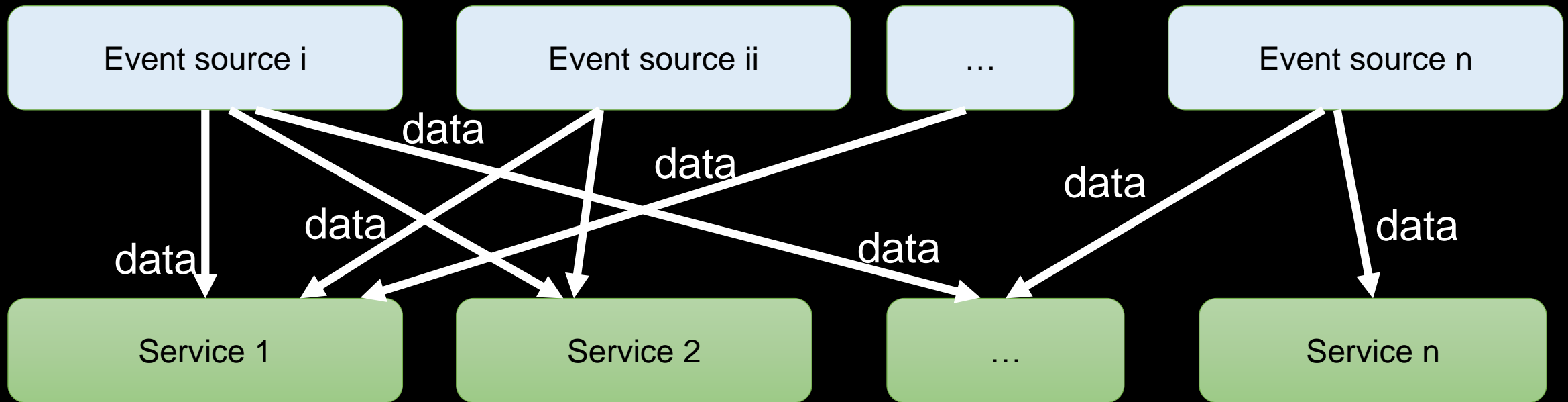


Un intégrateur spécifique pour chaque source de données, qu'il faudra maintenir.
8 services et 13 sources de données → combien d'intégrateurs ?

Comment les faire communiquer ?

Il reste un problème, quel format de données utiliser ?

□ Chaque service a ses **spécificités**



Protocole à choisir (HTTP, REST, JDBC, etc...)

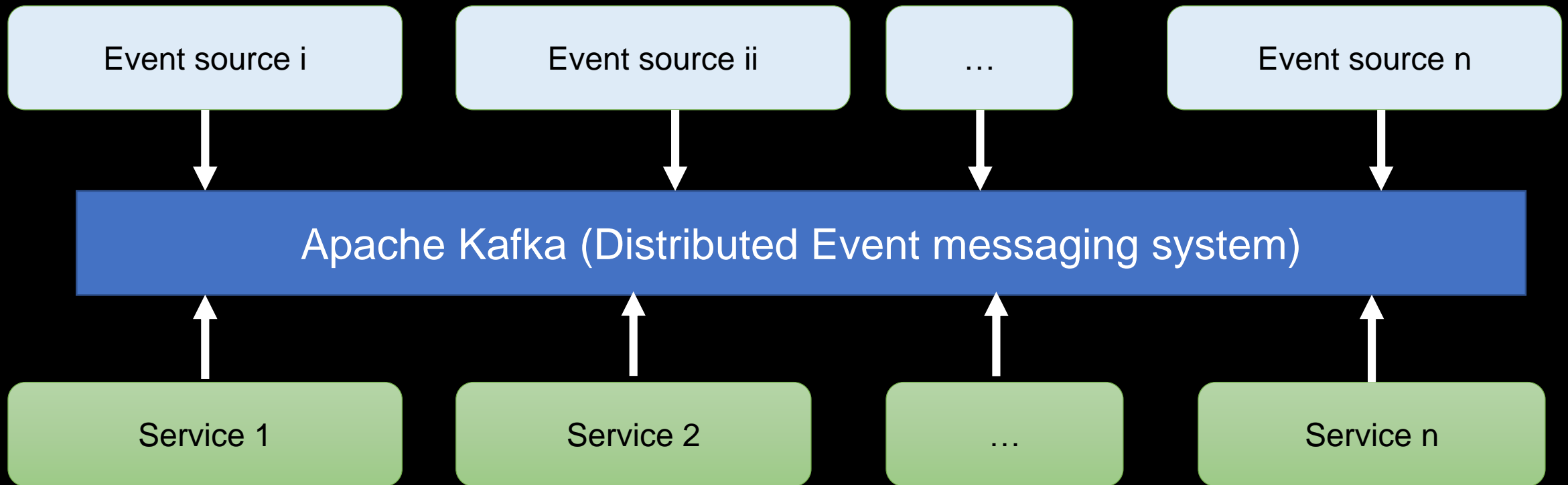
Format de données (Binaire, JSON, Yaml, etc...) => s'ils évoluent ?

Chaque service doit maintenir des connections pour chaque source ? 13/22

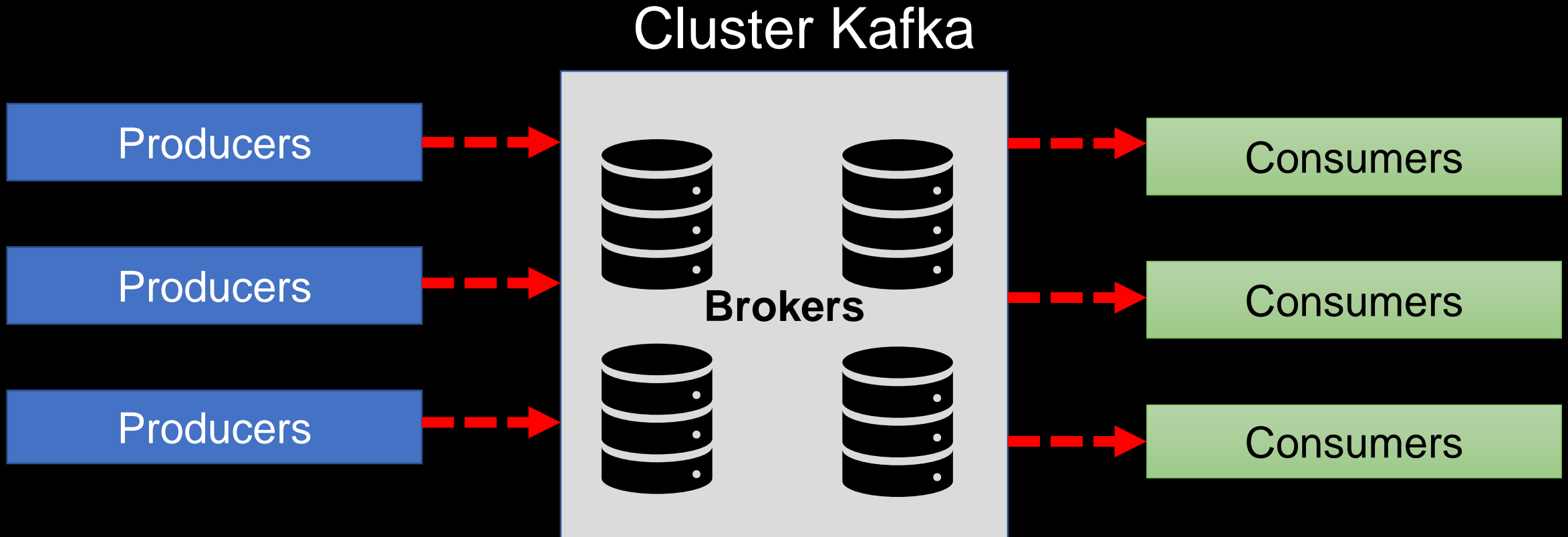
Comment les faire communiquer ?

Il reste un problème, quel format de données utiliser ?

❑ Chaque service a ses **spécificités**

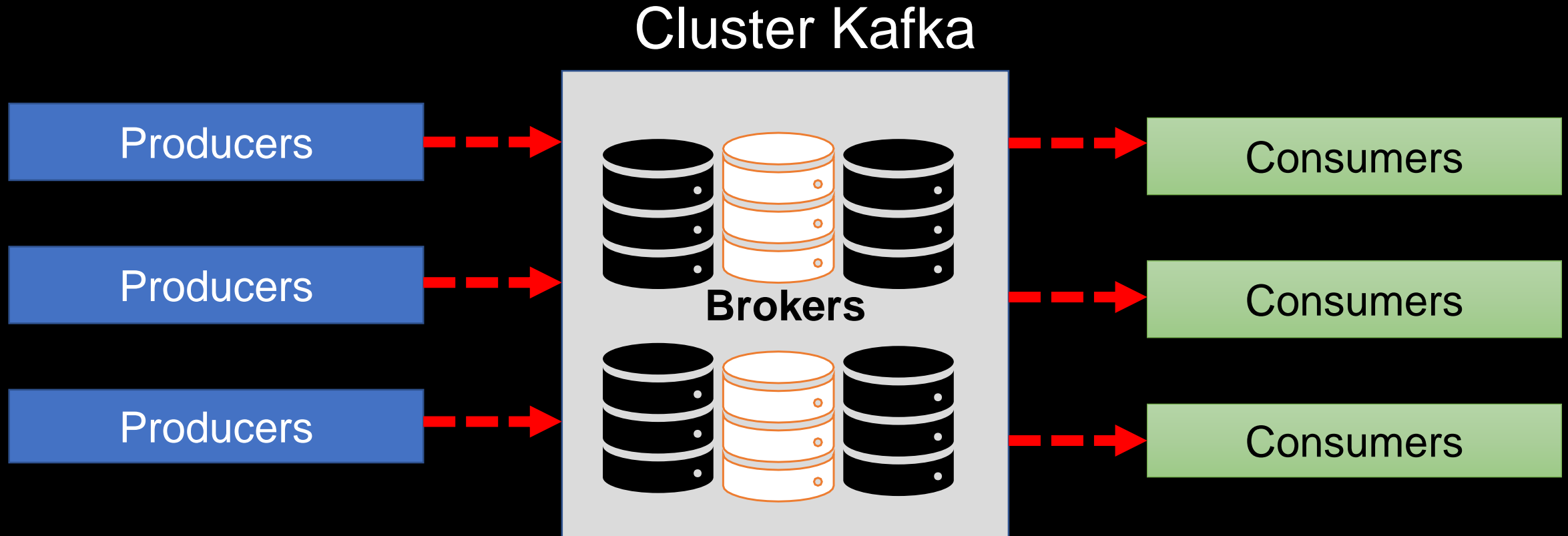


Apache Kafka



Les **brokers** représentent la base de données où sont stockés les informations liés aux **producers**.

Apache Kafka



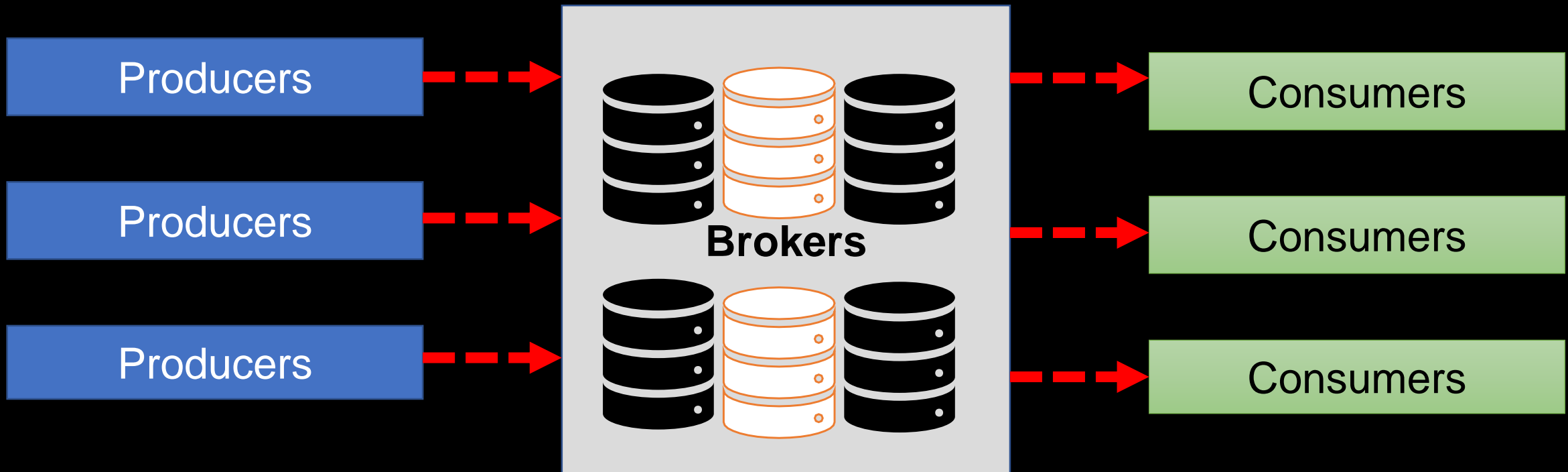
Plusieurs **brokers** peuvent être rajoutés en cours d'exécution pour l'équilibrage de charge

Apache Kafka

17/22



Cluster Kafka



Les données des brokers sont persisté de manière à ce qu'un évènement peut être rejoué. La durée est paramétrable.

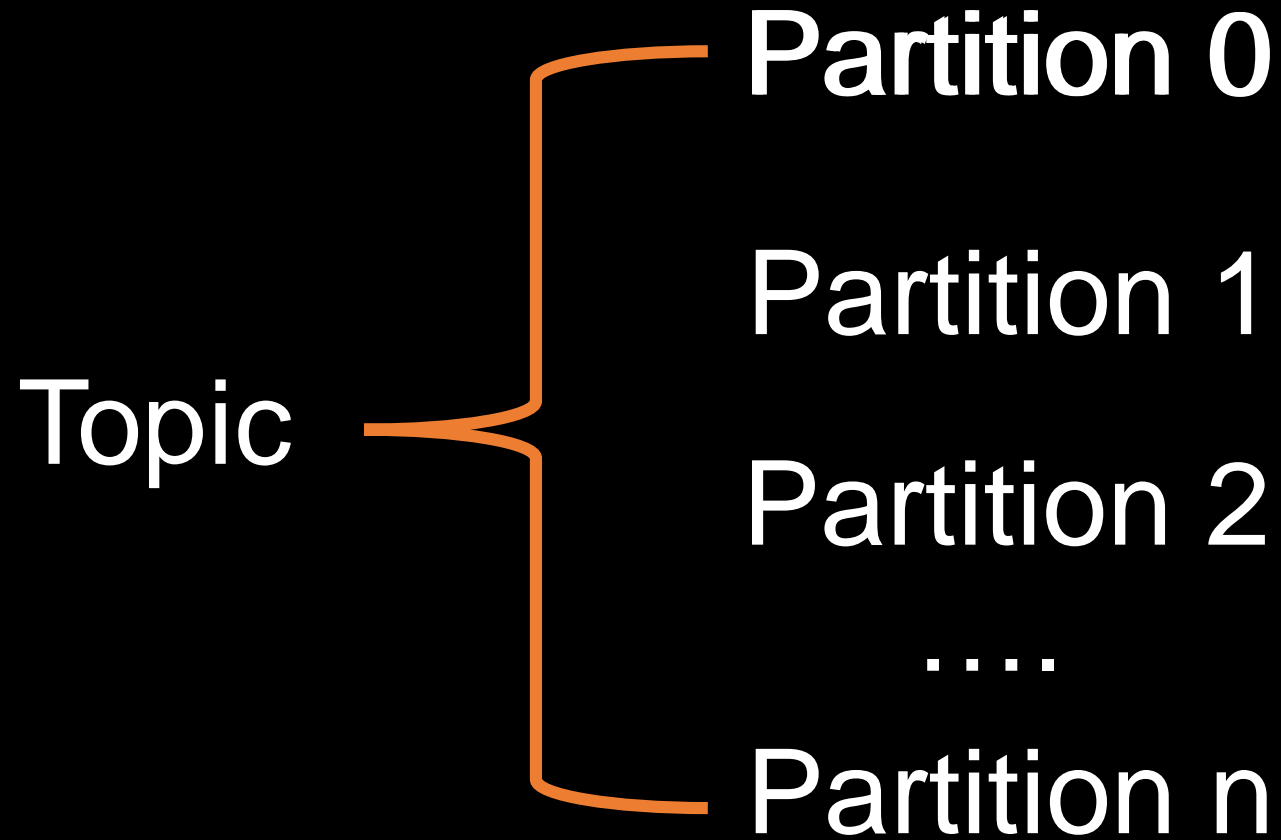
Apache Kafka

Topic



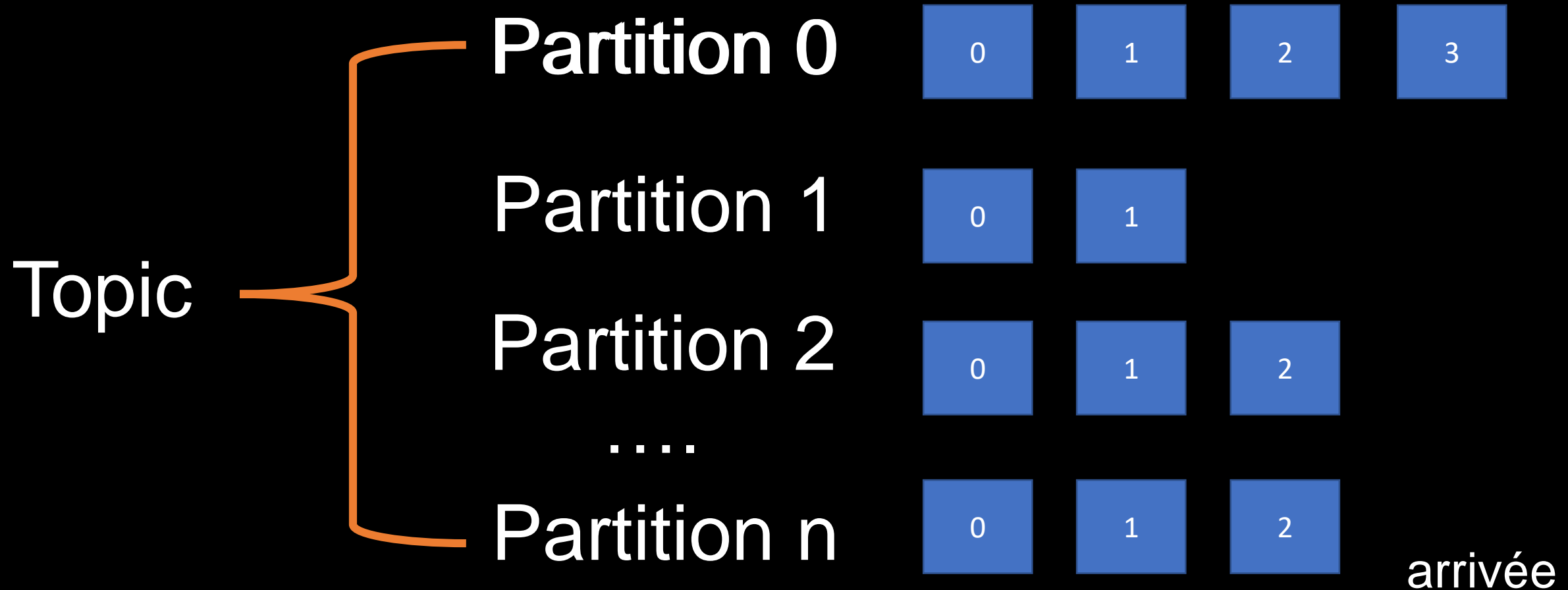
Les évènements sont ordonnés et stockés sous des catégories précises appelés **Topics**.

Apache Kafka



Chaque **Topic** est divisé en **Partition** pour faciliter la recherche d'évènements

Apache Kafka



Chaque évènement est stocké dans des messages ordonnés par ordre d'arrivée.

Apache Kafka – Cas d'utilisation

NETFLIX

Recommandations temps
réel aux utilisateurs

Uber

Collecte des données des taxis
et utilisateurs pour prédire les
prix et ajuster la tarification

Linked in®

Détection de spam et collecte des
interactions utilisateurs pour analyse

Apache Kafka – Construire les cas d'utilisations

