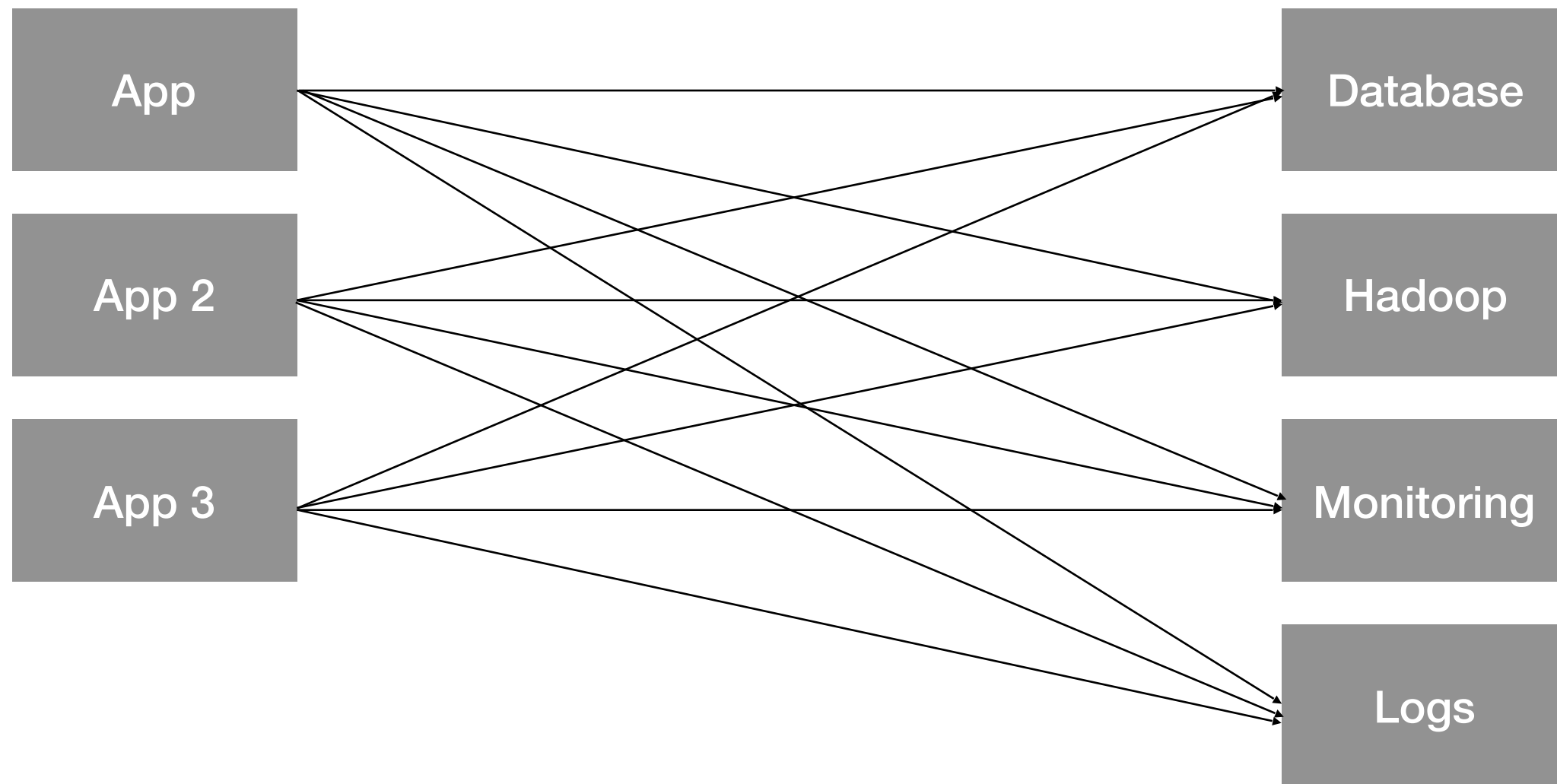


Introduction à Kafka

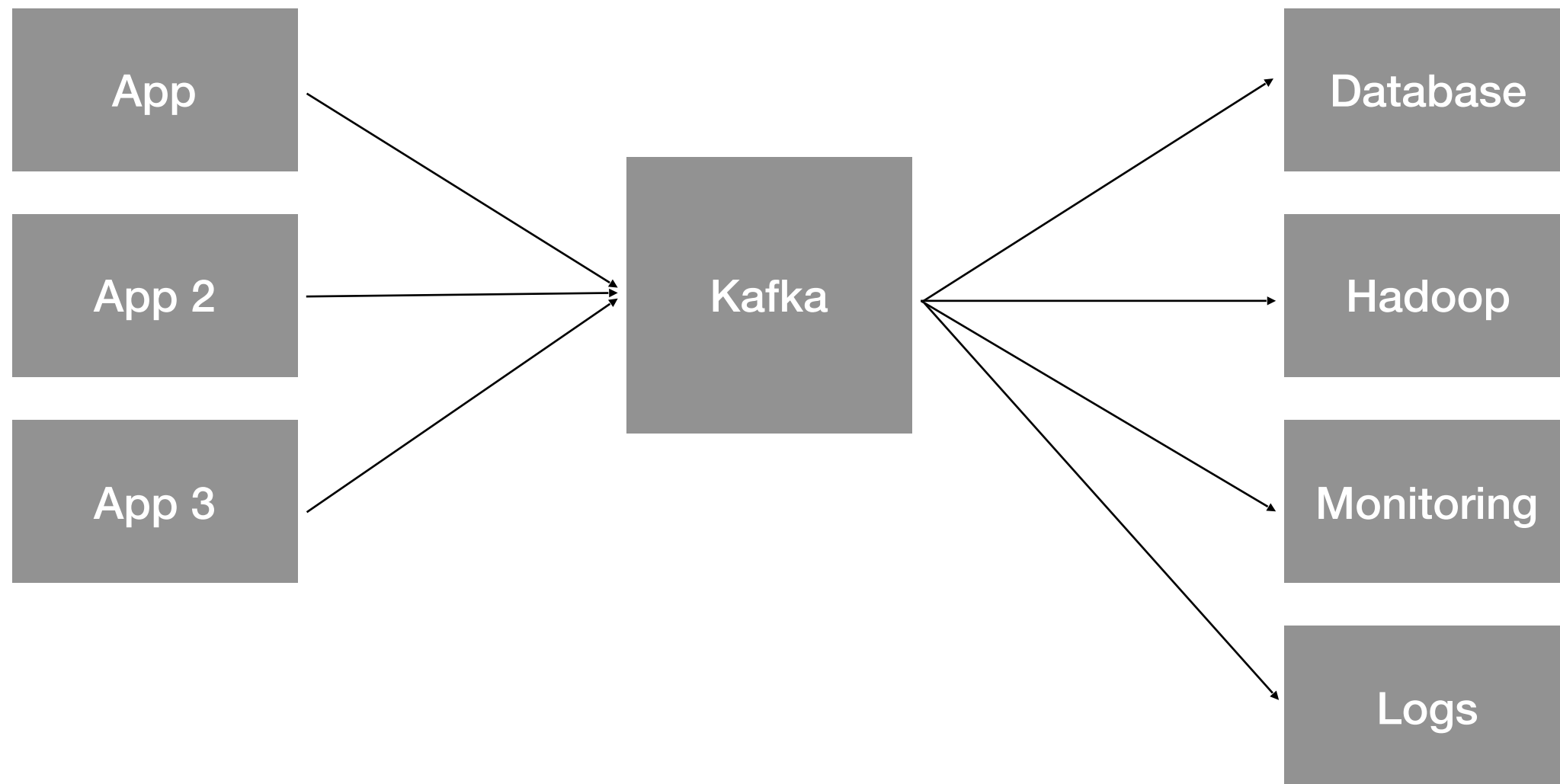
Jean-Paul LE
Geoffrey ALDEBERT

Contexte

Problématique



Problématique



Positionnement

- Complémentaire - SOA vs Microservice

Enterprise Service Bus

Message
Oriented
Middleware

Services
Web

Transformati
on

Routage

Message-oriented Middleware

- Transport de messages
- Communication asynchrone
- Persistance des messages
- Fiabilité
- Routage basique (topics)

Présentation Kafka

Cas d'usage Kafka

- Internet des Objets (IoT)
- Métriques applicatives
- Notifications
- Complex Event Processing
- Logs
- Transaction bancaire
- Chats
- ...

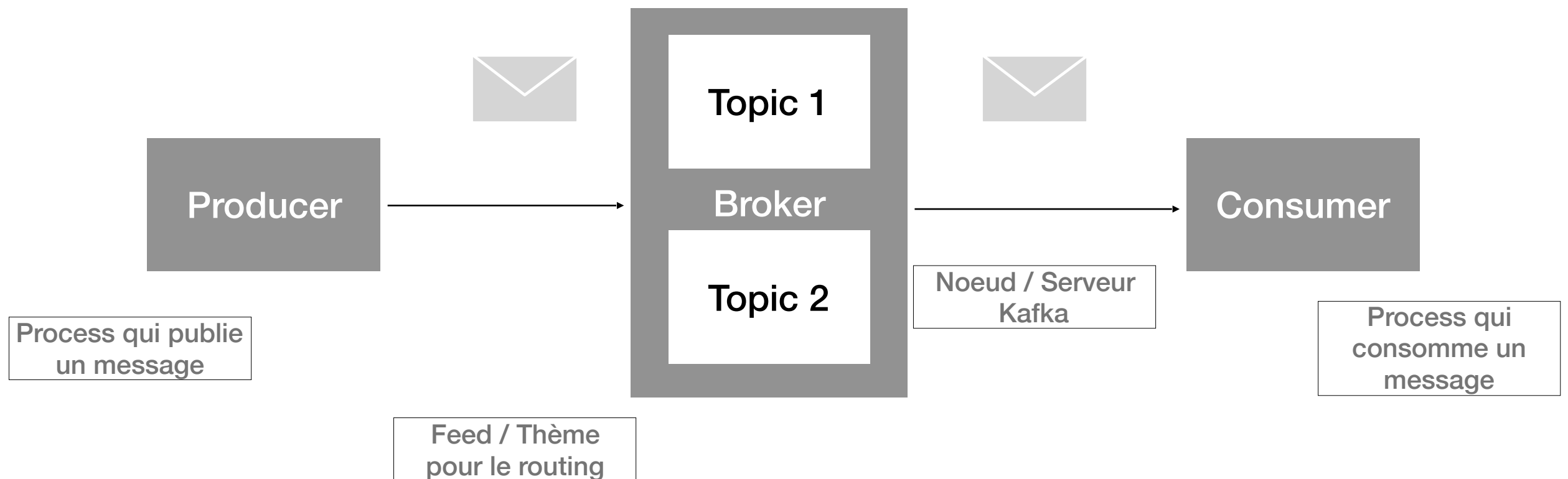
Utilisateurs



Avantages / Limites

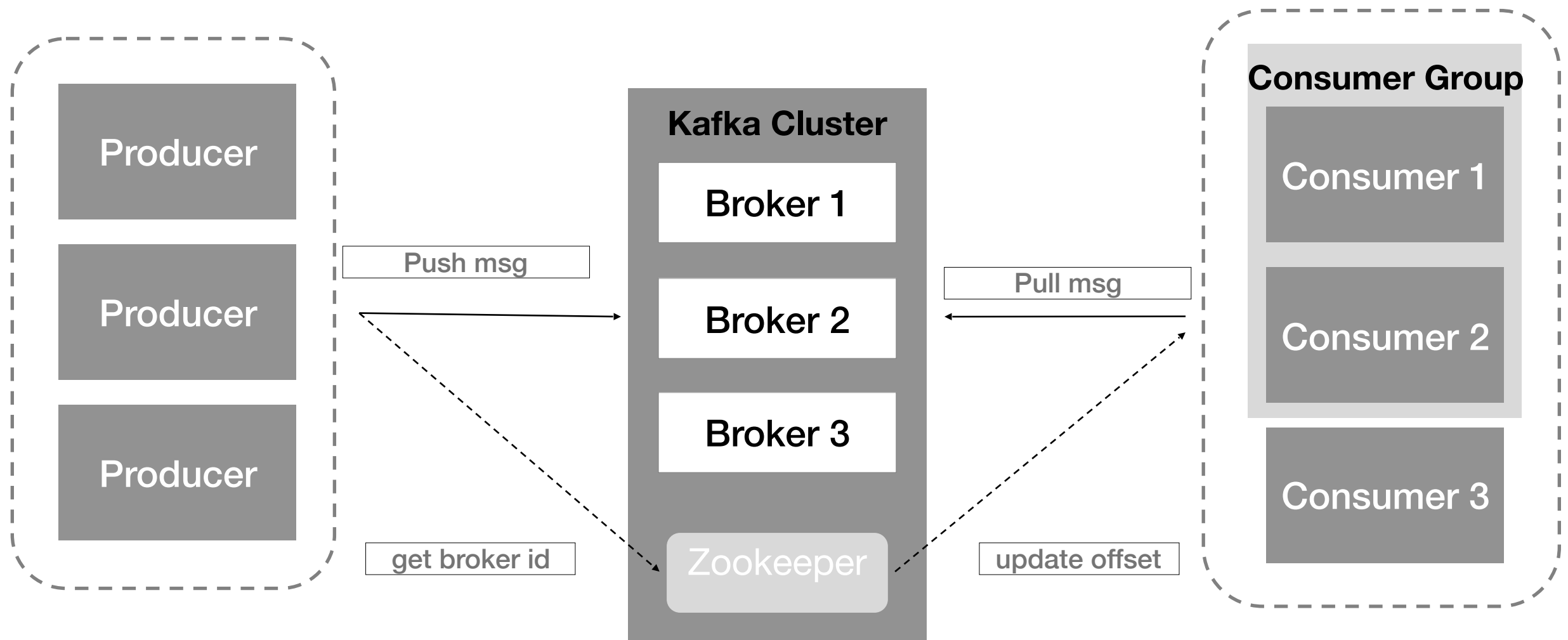
- Haut débit
 - Faible latence
 - Scalable
 - Distribué
 - Persistent
 - Tolérant à la panne
- Gestion de messages « légers » uniquement
 - Routage simple (basé sur les topics)

Les concepts de Kafka



Architecture

Architecture

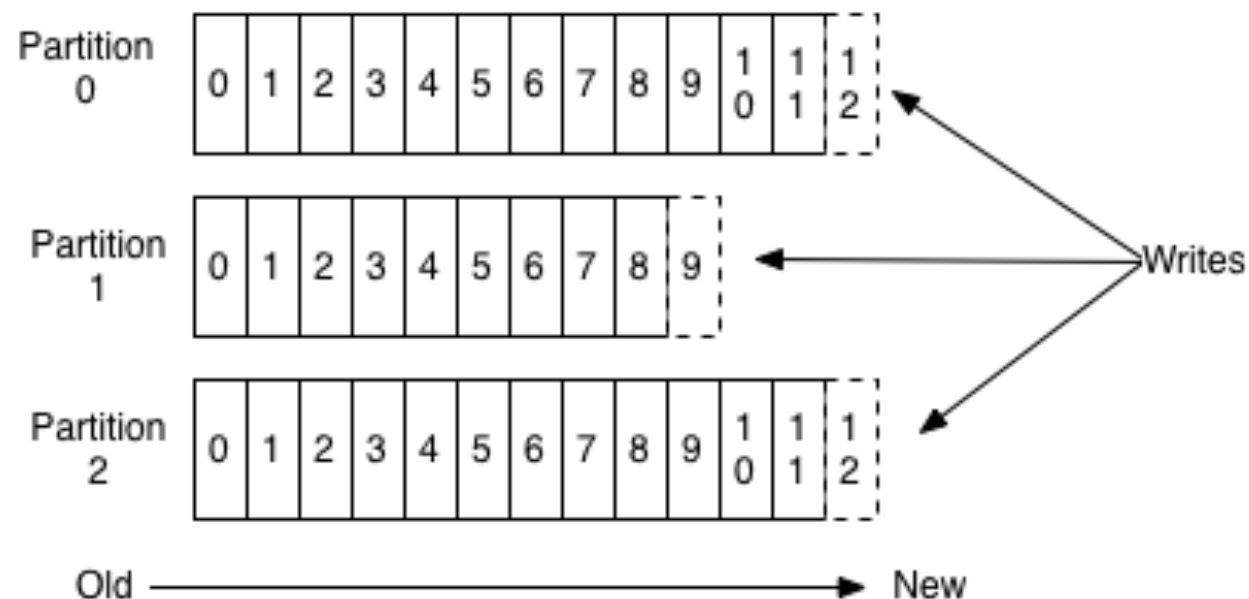


Topics

- Un topic est un « thème » / une « classification » définissant un stream particulier de donnée
- Les messages sont stockés au sein des topics
- Un topic comporte un nom unique
- Il n'y a pas de limitation de topics dans Kafka
- Les données dans les topics ne sont pas modifiables

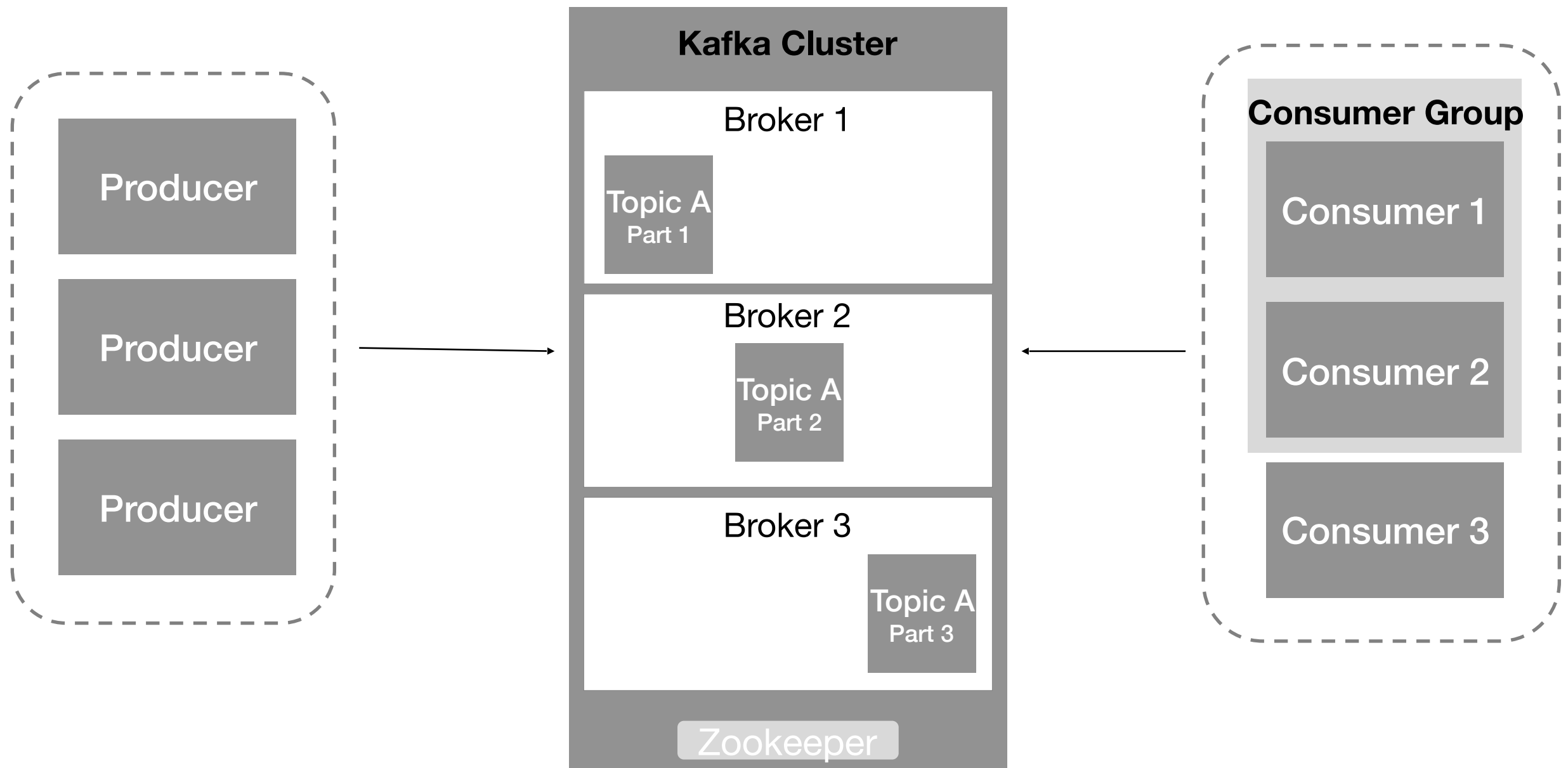
Partitionnement des topics

Anatomy of a Topic

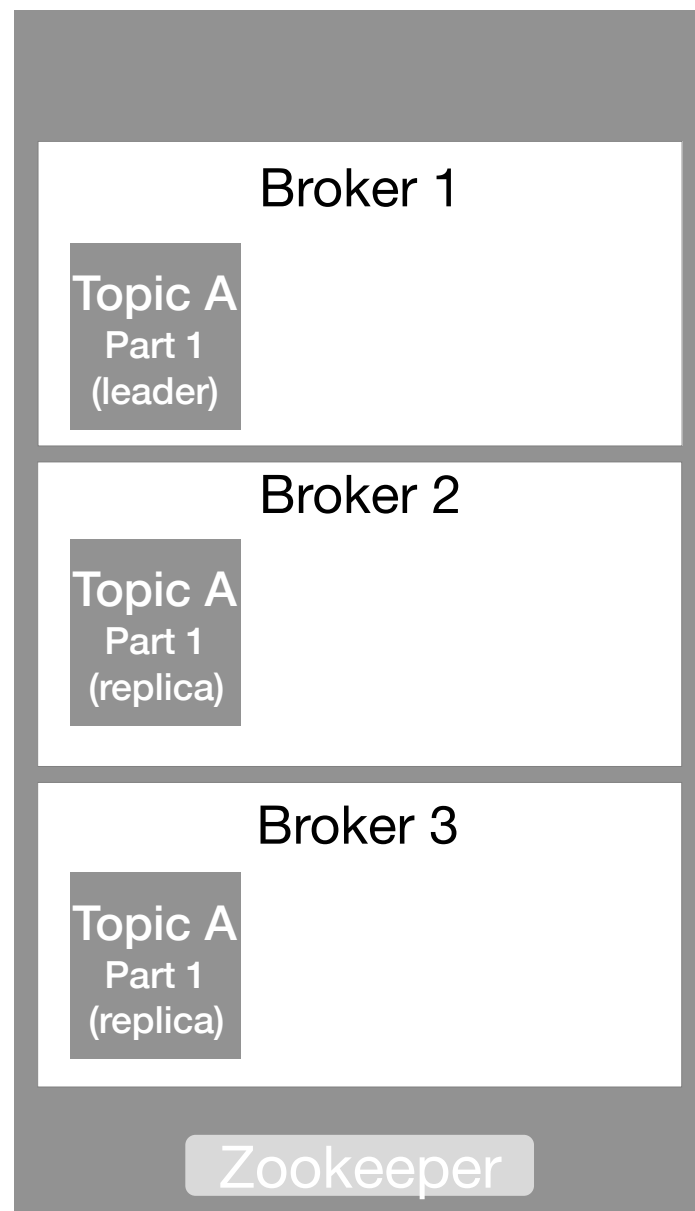


- Les messages sont écrits de manière aléatoire dans une partition
- Possibilité d'utiliser une clé dans un message pour garantir l'ordre de séquençement
- Chaque message possède un id unique dans une partition (offset)

Architecture - Topics

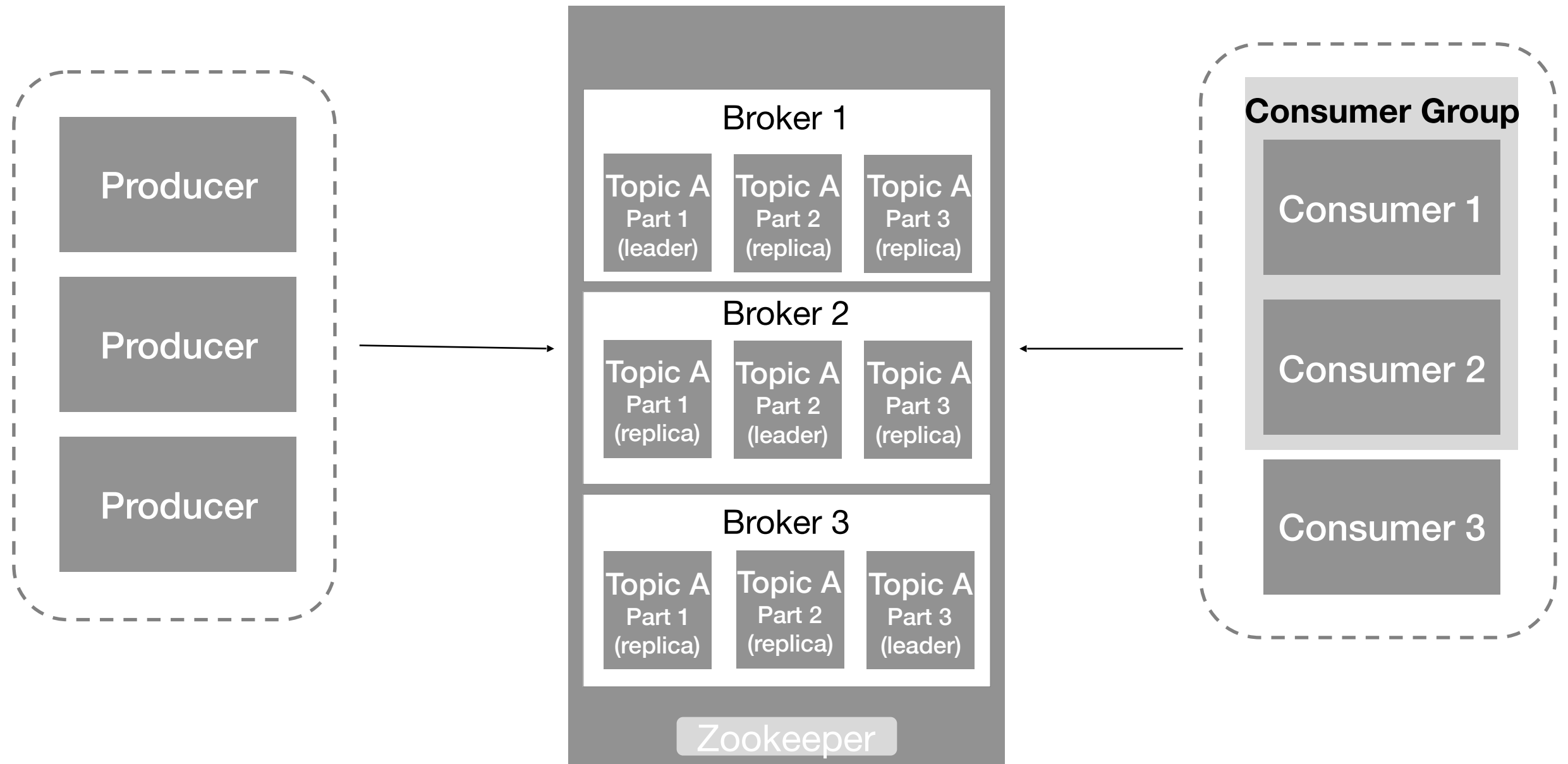


Réplication des topics



- Les répliquions se font au niveau des partitions
- Pour une partition donnée, il n'y a qu'un leader et plusieurs replicas
- Non possible d'avoir plus de répliques que de nombre de brokers

Architecture - Topics



Producer

- Possibilité de publier un message dans un topic particulier
- possibilité de faire du « semantic partitionning » basé sur une clé d'un message

Consumer

- Plusieurs consumer peuvent lire un même topic
- Un message consommé reste dans le broker (possibilité d'obtenir un historique des messages même après consommation)
- Une panne de consumer n'est pas problématique (rebalancing automatique)
- Il est possible de former des groupes de consumer

Modes de livraison

- at least once
(défaut)
- at most once
- exactly once



Langages de programmation

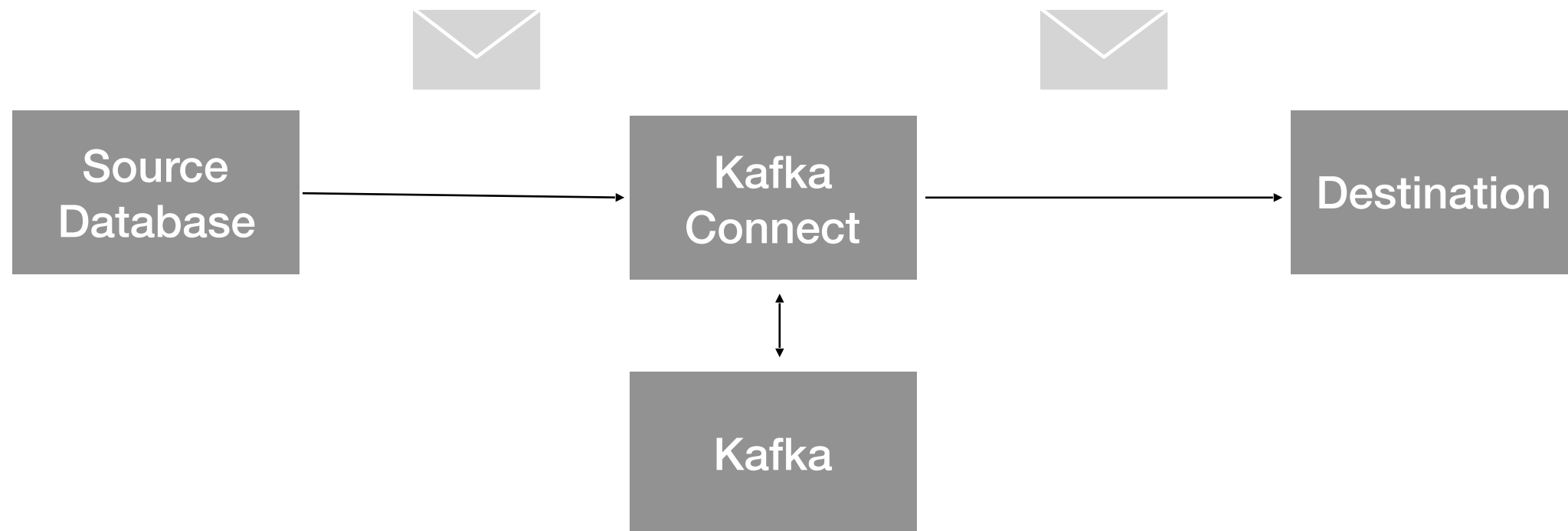
- **C/C++**
- **Python**
- **Go (AKA golang)**
- **Erlang**
- **.NET**
- **Clojure**
- **Ruby**
- **Node.js**
- **Perl**
- **PHP**
- **Rust**
- **Java**
- **Storm**
- **Scala DSL**
- **Clojure**
- **Swift**

Modules Kafka

Kafka Connect

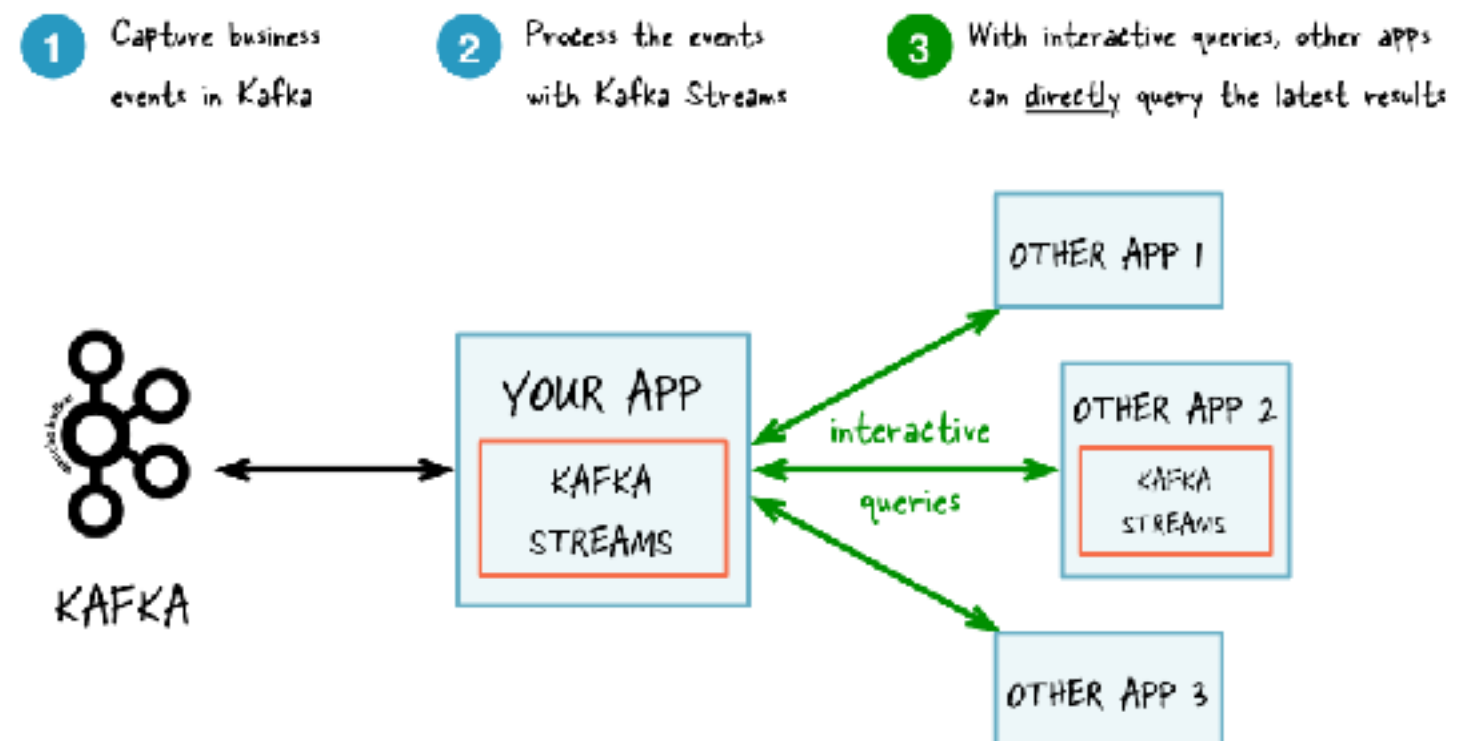
- Interfaçage de Kafka avec diverses sources
- La donnée est produite depuis une « source » et consommée dans un « sink »
- exemple :
 - publish an sql tables
 - consume kafka into hdfs
 - consume kafka into elasticsearch ...

Kafka Connect



Kafka Streams

- Module permettant le développement d'application stream dans Kafka
- Léger mais puissant
 - windowing,
 - join,
 - aggregation,
 - filter,
 - map...



KSQL

- Manipulation des streams comme en sql

