Apache Kafka

Julien Romero

C’est quoi Apache Kafka ?

- Une plateforme distribuée pour les flux d’événements
- Basé sur publish-subscribe

Applications:

- *Suivi des activités :* Enregistrer les activités des utilisateurs sur un site web. Ces activités sont ensuite utilisées par d'autres applications pour générer des rapports, alimenter des modèles de machine learning, mettre à jour les résultats de recherche, etc.
- *Messagerie :* Envoyer des e-mails de manière asynchrone.
- *Métriques/Journalisation :* Les applications publient des métriques qui sont ensuite consommées par un système de surveillance.
- *Journal de commit :* Publier les modifications d'une base de données sur Kafka. Ces modifications peuvent ensuite être enregistrées dans une base de données, répliquées, ou traitées pour d'autres applications, etc.

C’est quoi Apache Kafka - Avantages/Inconvénients

Kafka :

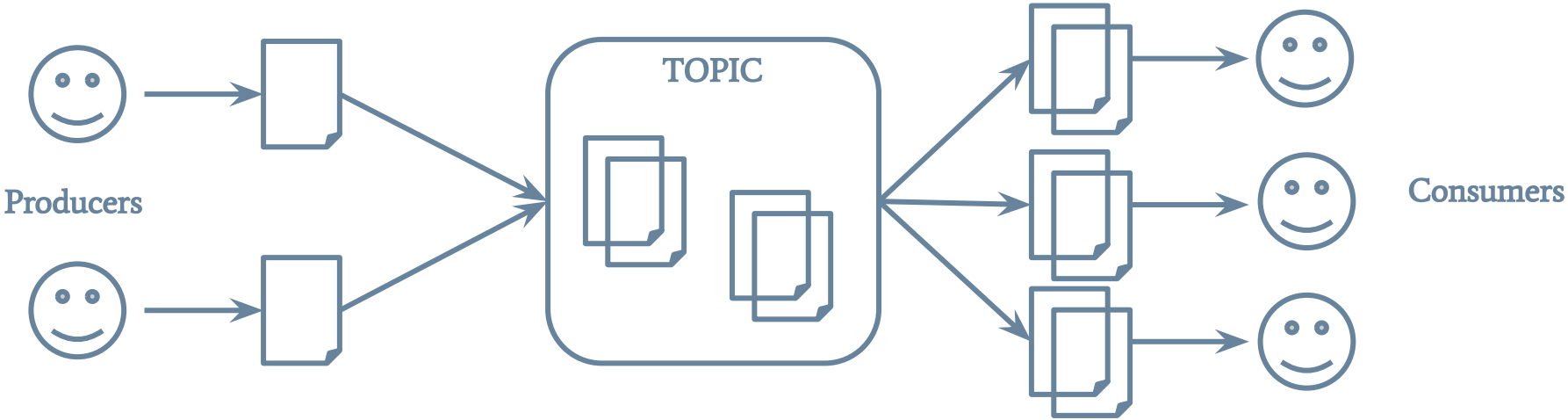
- Est scalable.
- A une faible latence et un débit élevé : les messages sont traités rapidement.
- Est tolérant aux pannes.
- Permet de créer un pipeline de données en temps réel.

Mais :

- Format des messages fixe : difficile de modifier les messages.
- La rétention est coûteuse : il faut utiliser un autre système pour le stockage à long terme (HDFS).
- Les consommateurs doivent lire l'intégralité des messages : ils ne peuvent pas récupérer un champ spécifique.

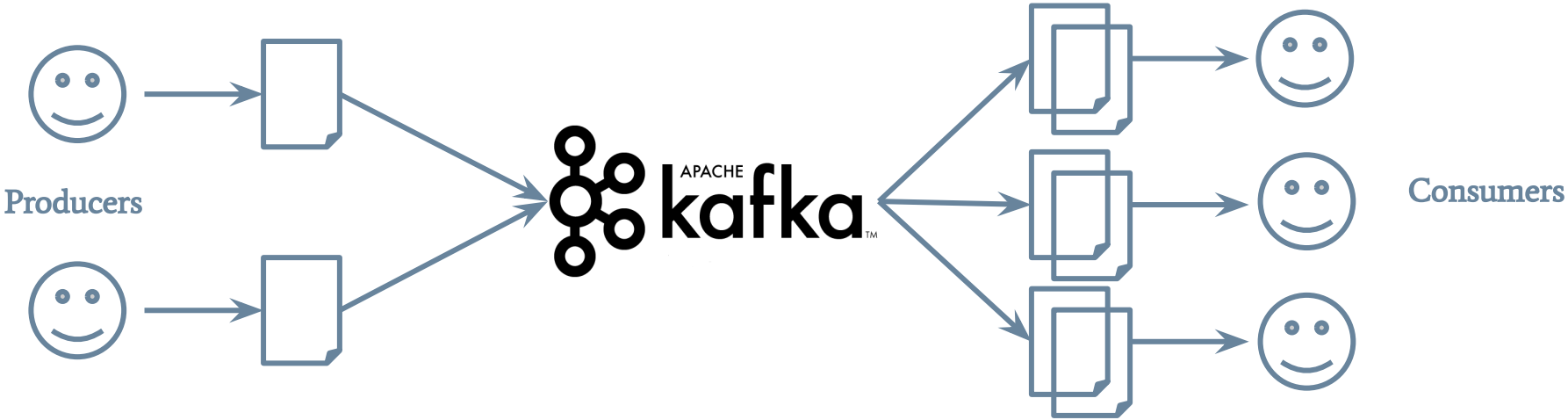
Producer/Consumer

- Aussi connu comme publish/subscribe
- Deux types d'acteurs:
  - Les publishers/producers: Ils peuvent écrire des messages dans des topics
  - Les subscribers/consumers: Ils sont notifiés quand il y a des nouveaux messages et peuvent les lire



Producer/Consumer Pattern

- Aussi connu comme publish/subscribe
- Deux types d'acteurs:
  - Les publishers/producers: Ils peuvent écrire des messages dans des topics
  - Les subscribers/consumers: Ils sont notifiés quand il y a des nouveaux messages et peuvent les lire



Topic Kafka

- Un topic est une catégorie de messages sur laquelle des producers peuvent écrire et des consumers lire

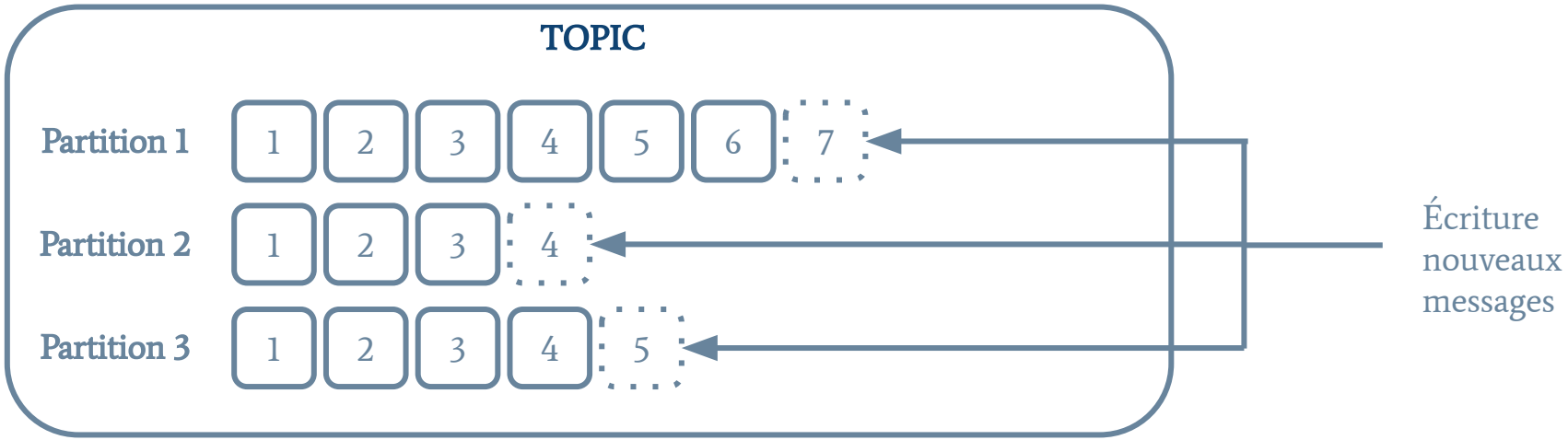
Topic Kafka - Opérations

```
admin = KafkaAdminClient(bootstrap_servers=['localhost:9092'])
```

- Création : `admin.create_topics([NewTopic("new-topic", num_partitions=1, replication_factor=1)])`
- Description d'un topic : `admin.describe_topics(["new-topic"])`
- Suppression : `admin.delete_topics(["new-topic"])`
- Liste : `admin.list_topics()`

Partitions

- Les messages sont stockés dans un topic sous forme de partitions en **ajout seulemen** t (append-only).
- Cela permet le traitement groupé des messages, la réplication, la tolérance aux pannes, l'écriture parallèle, un débit élevé, etc.
- La partition d'un message peut être choisie automatiquement par Kafka ou définie via une fonction basée sur une clé.



Partitioning - Operations

- Ajout de partitions :

```
admin.create_partitions(
    {"new-topic": NewPartitions(2)}
)
```

```
admin.describe_topics(["new-topic"])
```

```
[{'error_code': 0, 'topic': 'new-topic', 'is_internal': False, 'partitions':
[{'error_code': 0, 'partition': 0, 'leader': 0, 'replicas': [0], 'isr': [0],
'offline_replicas': []}, {'error_code': 0, 'partition': 1, 'leader': 0,
'replicas': [0], 'isr': [0], 'offline_replicas': []}]}]
```

On ne peut pas réduire le nombre de partitions !

Producers

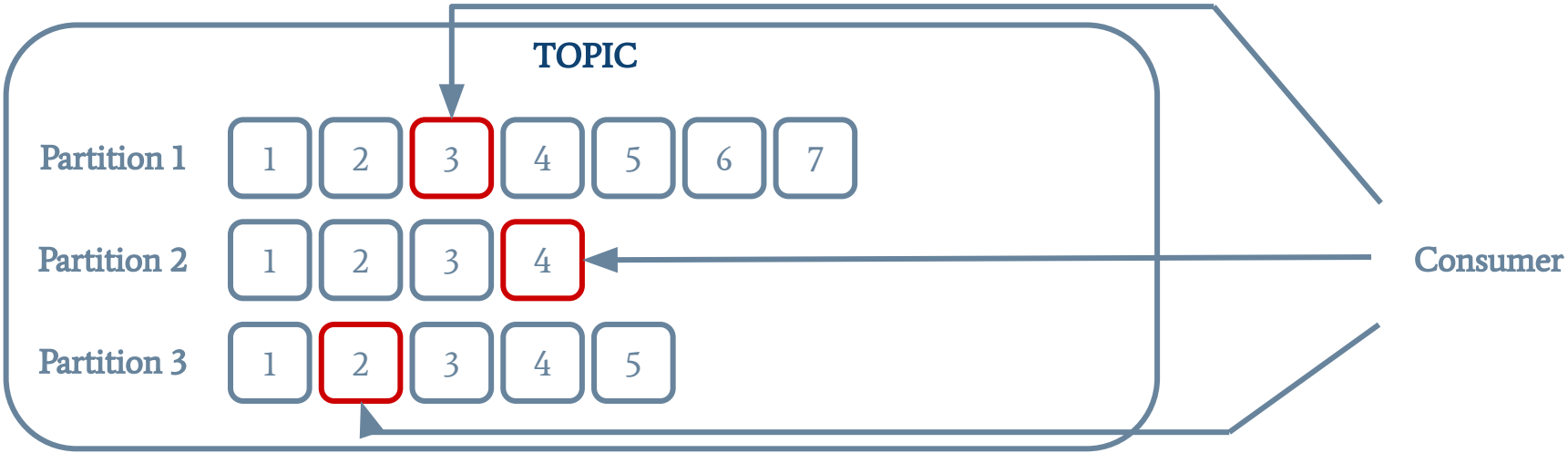
- Les producteurs écrivent des messages dans un topic.
- Qu'est-ce qu'un message ?
  - N'importe quelle suite d'octets (byte array).
  - Généralement au format clé/valeur.
  - On essaie de structurer la valeur avec un format simple (JSON, CSV, XML) ou une bibliothèque plus avancée.
  - Les données doivent être transférées, donc elles sont sérialisées/désérialisées.
  - Les messages sont indexés avec un compteur appelé offset pour chaque partition.

Producers

- Création : `producer = KafkaProducer(bootstrap_servers=['localhost:9092'])`
- Envoi asynchrone (valeur) : `producer.send('my-topic', b'raw_bytes')`
- Envoi clé/valeur : `producer.send('my-topic', key=b'foo', value=b'bar')`
- Attendre l'envoi des messages : `producer.flush()`

Consumers

- Les consommateurs s'abonnent à un topic donné et lisent les messages.
- Un consommateur garde une trace de son offset actuel pour chaque partition. Il peut ainsi reprendre le traitement plus tard en cas de crash.



Consumers

- Création :

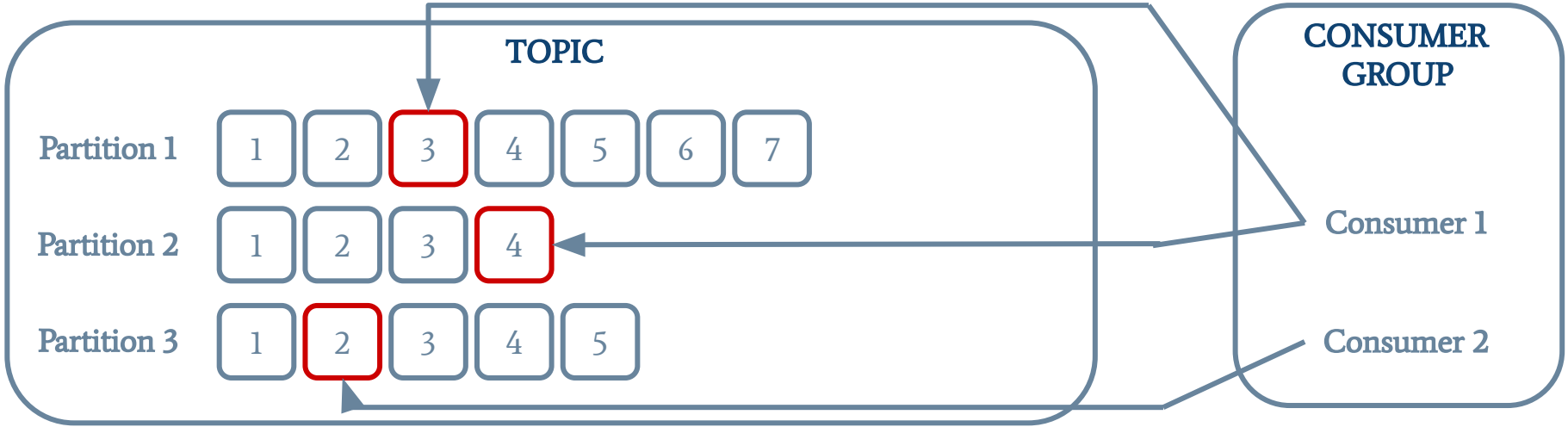
```
consumer = KafkaConsumer('my-topic',
                         auto_offset_reset='earliest',
                         bootstrap_servers=['localhost:9092'])
```

  - earliest = on commence par les plus vieux message, puis on lit jusqu'aux plus récents
  - latest = on commence par les message les plus récents, puis on lit seulement les nouveaux message
- Lecture des messages

```
for message in consumer:
    print("%s:%d:%d: key=%s value=%s"% (message.topic, message.partition,
                                        message.offset, message.key,
                                        message.value))
```

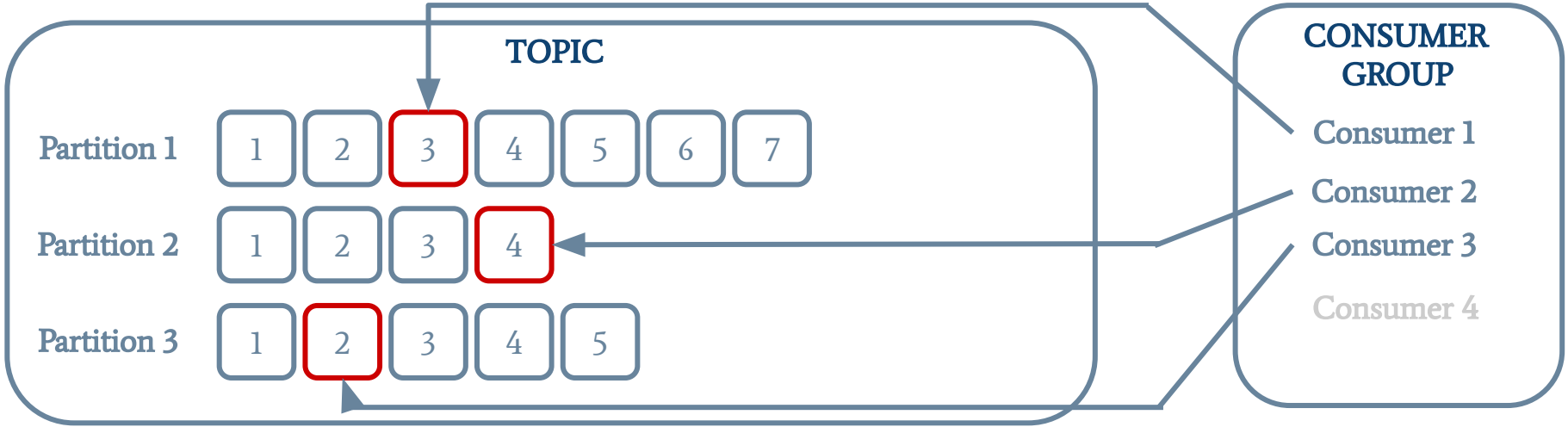
Consumer Groups

- Un groupe de consommateurs est composé de plusieurs consommateurs qui partagent les mêmes offsets.
- Une partition est toujours attribuée à un seul consommateur.
- S'il y a plus de consommateurs que de partitions, les consommateurs supplémentaires attendent.



Consumer Groups

- Un groupe de consommateurs est composé de plusieurs consommateurs qui partagent les mêmes offsets.
- Une partition est toujours attribuée à un seul consommateur.
- S'il y a plus de consommateurs que de partitions, les consommateurs supplémentaires attendent.



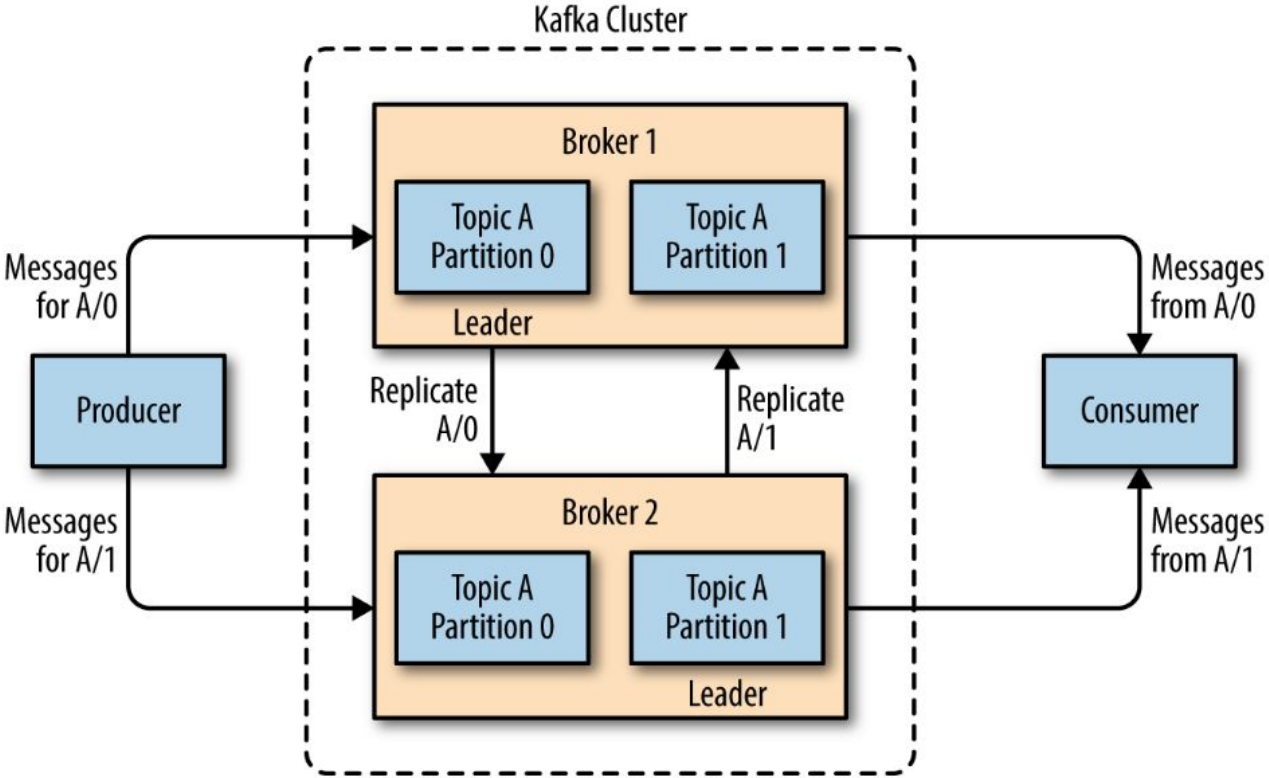
Consumer Groups - Operations

- Liste : `admin.list_consumer_groups()`
- Description : `admin.describe_consumer_groups("my-group")`
- Suppression : `admin.delete_consumer_groups(["my-group"])`
- Rejoindre un groupe : `consumer = KafkaConsumer('my-topic', group_id="my-group",...)`

Brokers

- Un serveur Kafka unique est appelé un broker.
- Un broker est chargé de répondre aux requêtes des producteurs et des consommateurs et de gérer ses partitions.
- Un cluster est composé de plusieurs brokers.
  - L'un d'eux est le contrôleur, responsable des tâches administratives comme l'attribution des partitions.
  - Une partition appartient à un seul broker (le leader), mais elle peut être répliquée sur plusieurs brokers.

Brokers



From Kafka - The definitive guide

En résumé

- Kafka est un système pour faire passer des messages entre des producteurs et des consommateurs
- Les consommateurs peuvent être agrégés en groupes
  - Le traitement est alors distribué dans le groupe