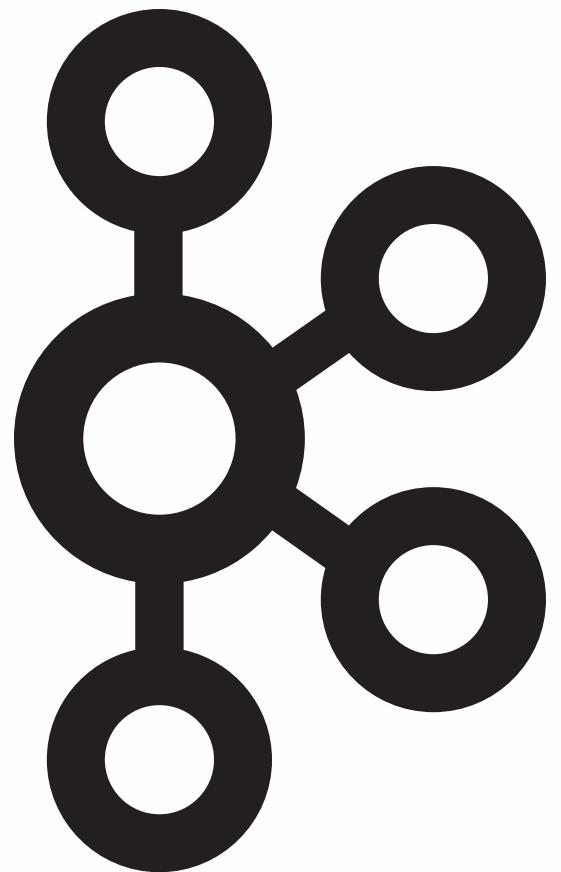


ENG221

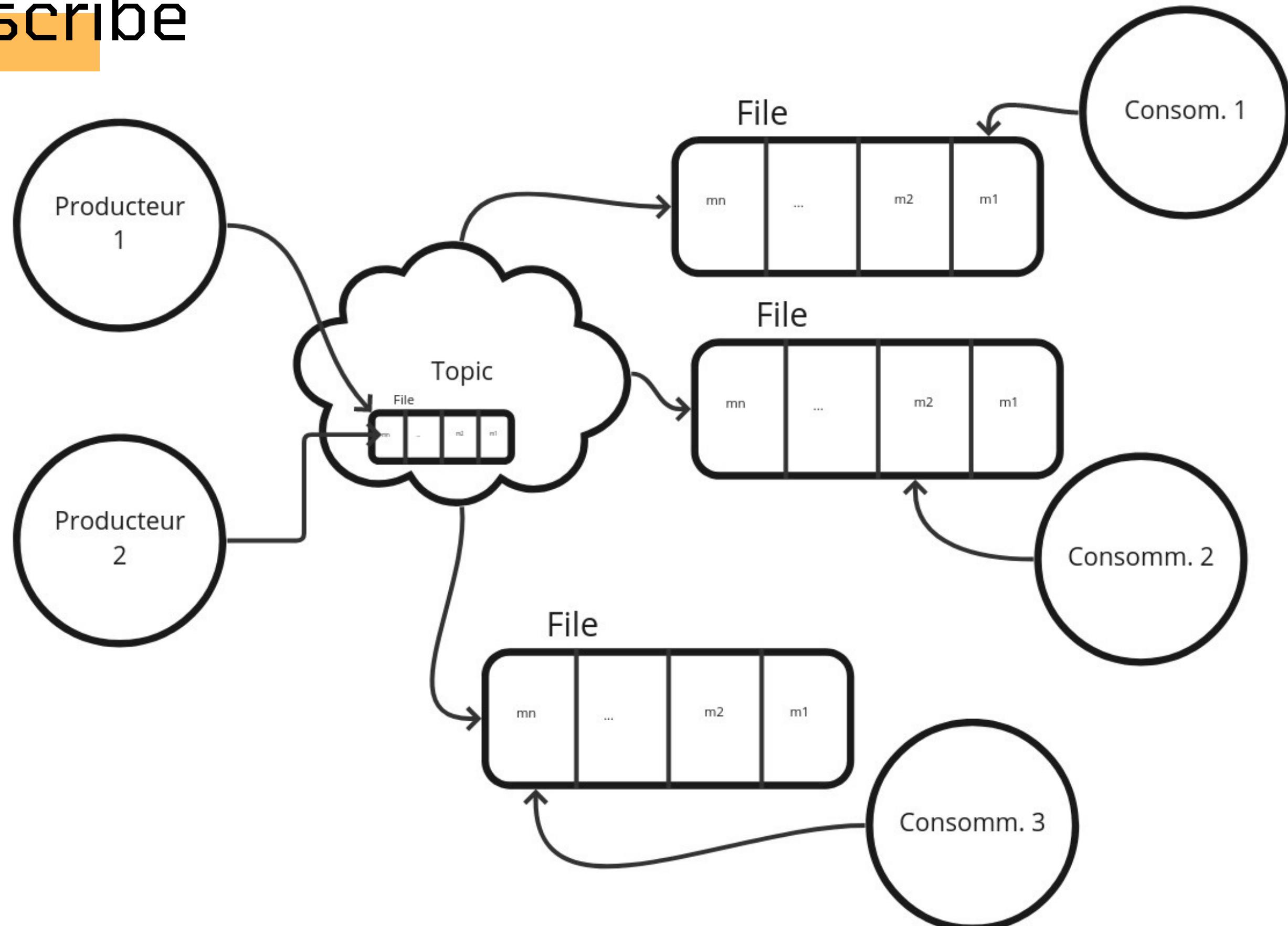
Mars 2023



APACHE KAFKA

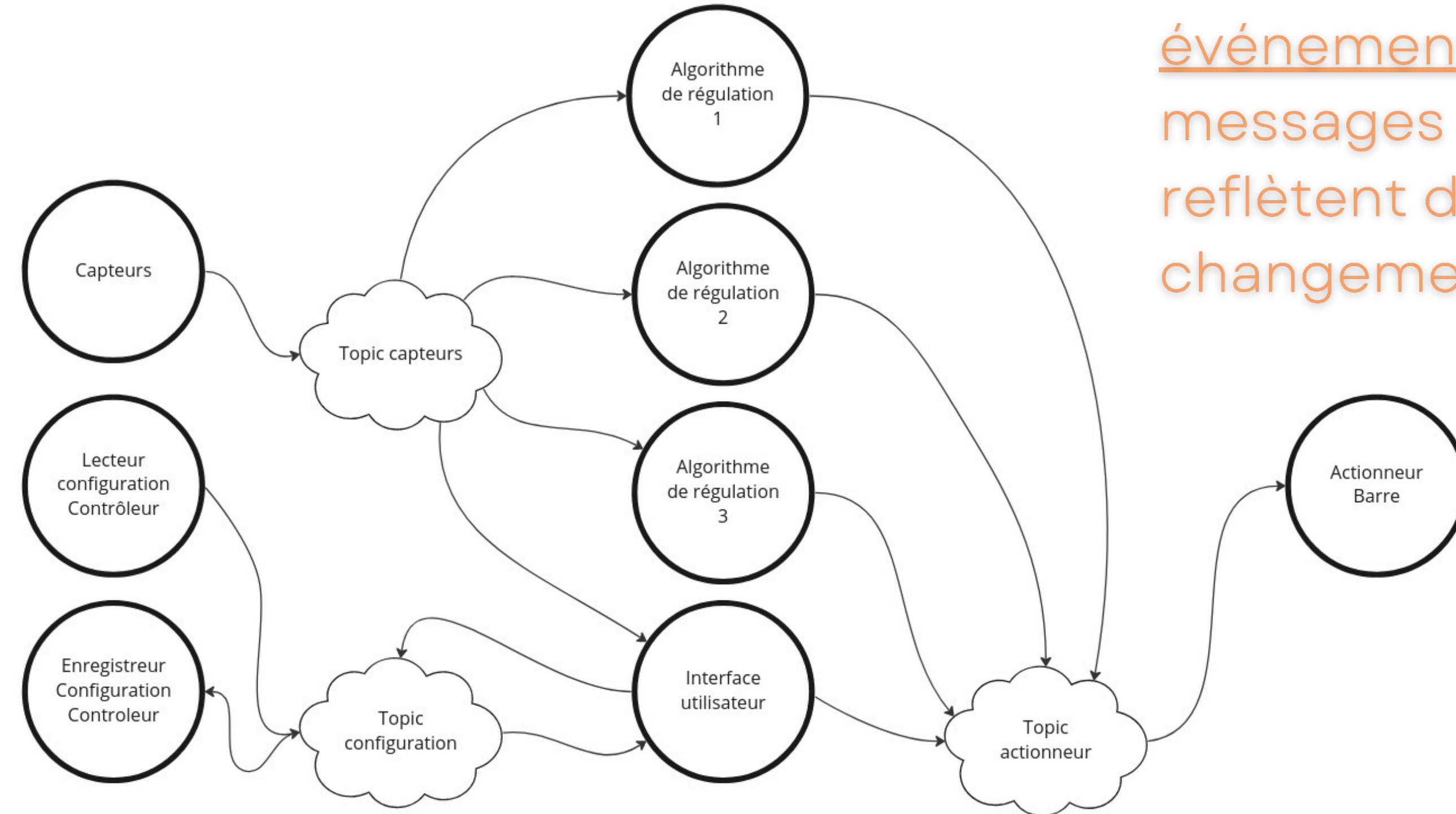
Principes et optimisation

Publish-Subscribe



Couplage faible
entre les agents

Architecture guidée par les événements

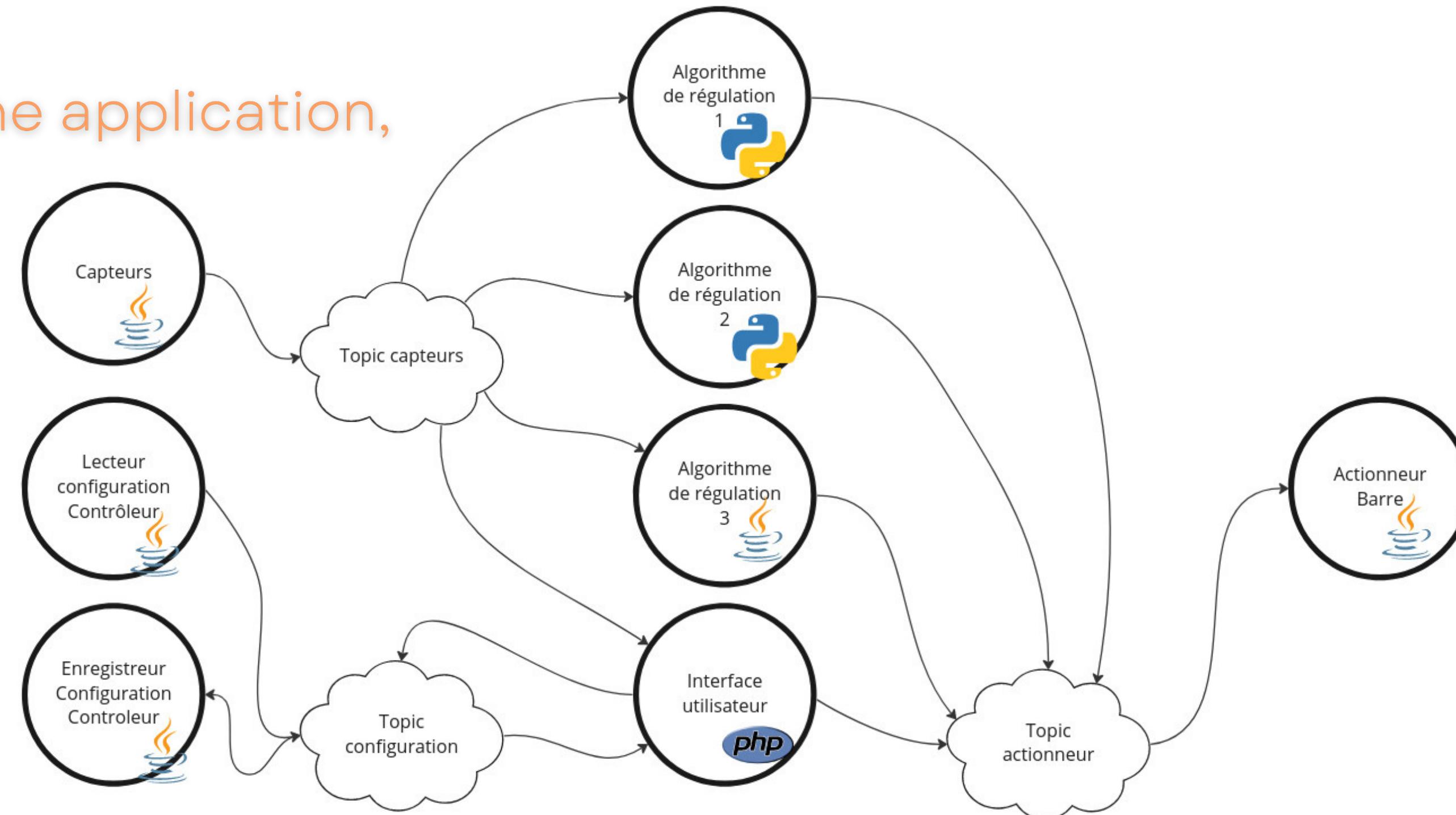


événements:
messages qui
réflètent des
changements

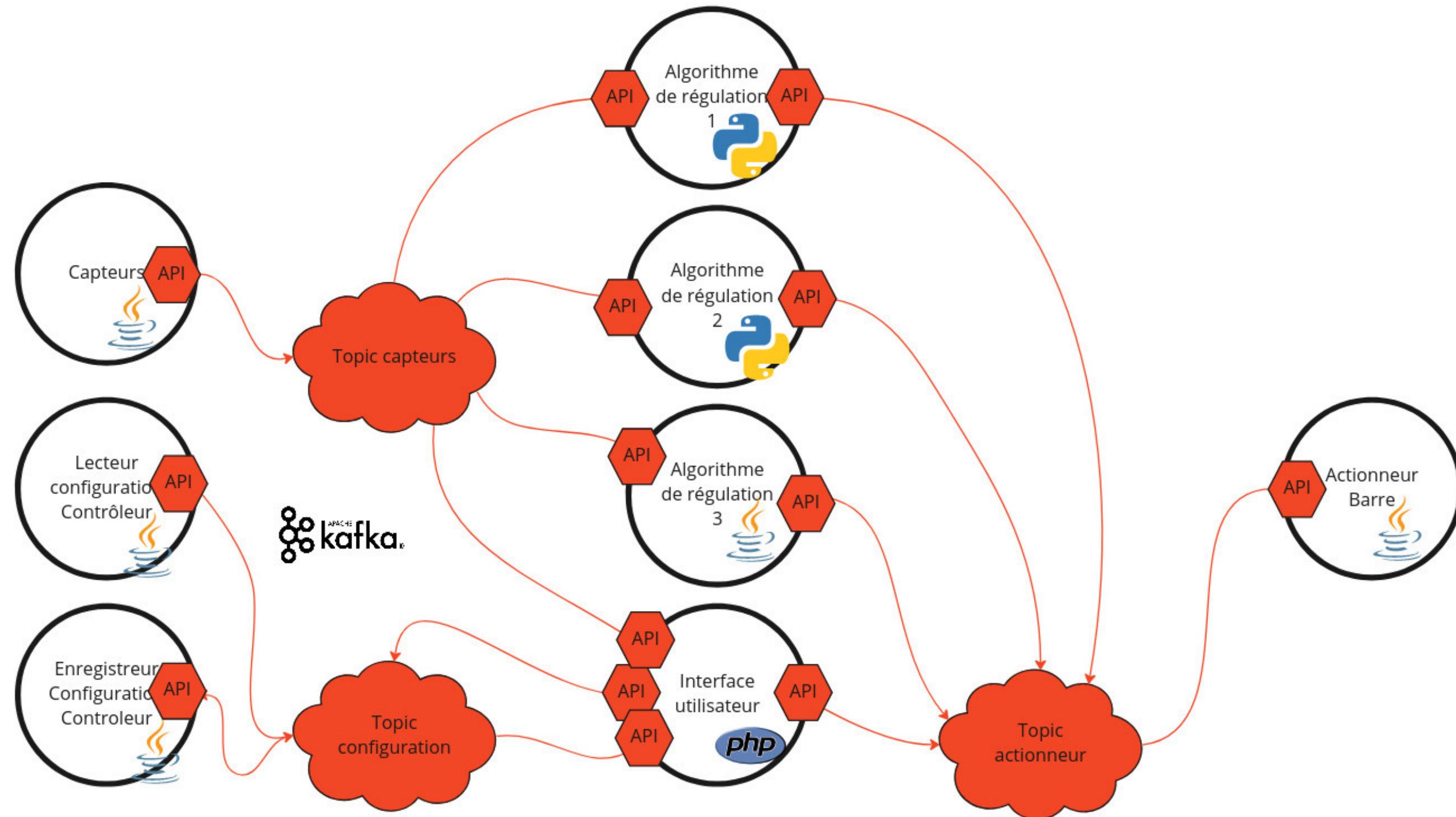
Architecture guidée par les événements orientée services

Service:

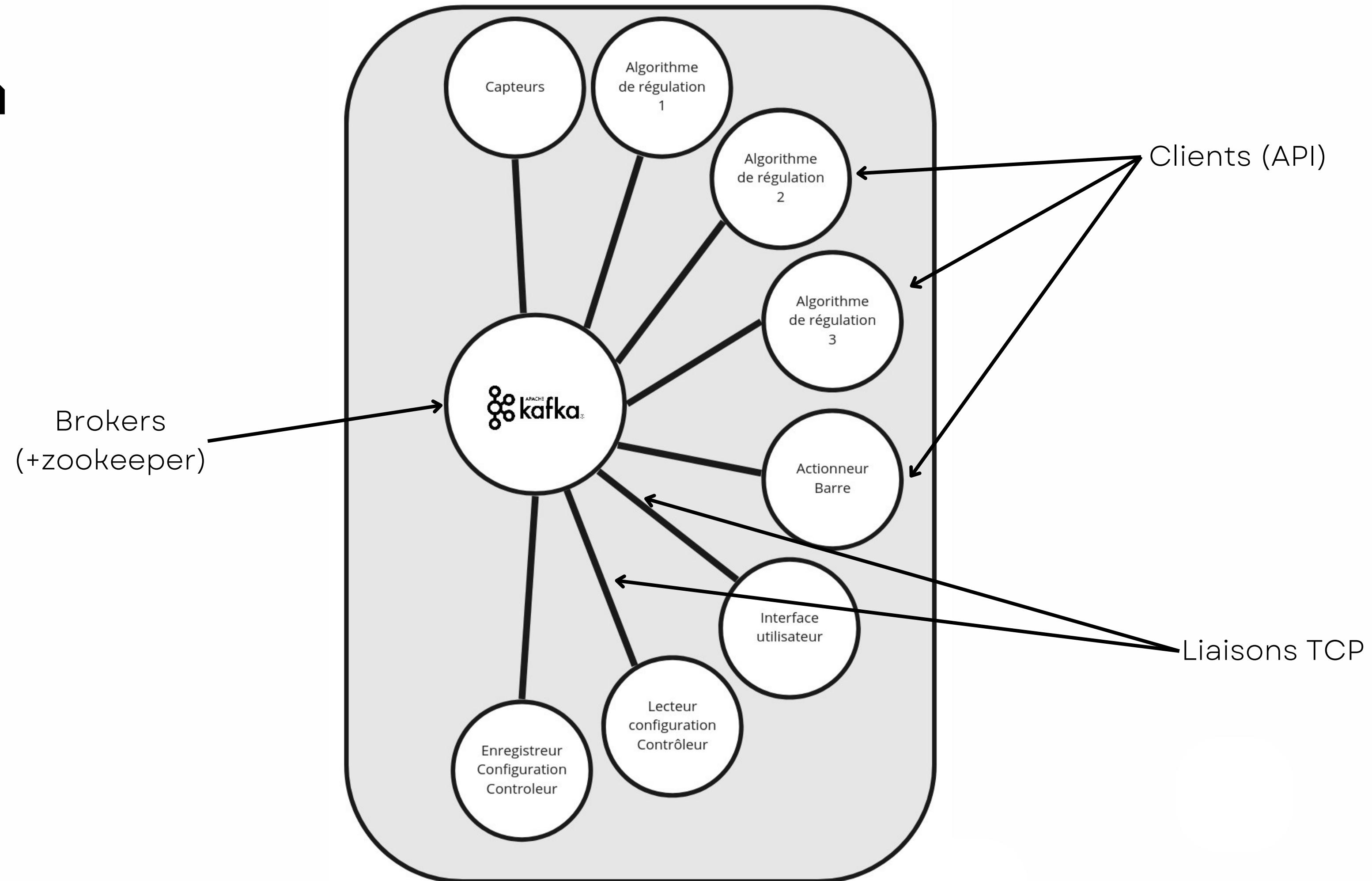
élément d'une application,
encapsulé,
découplé



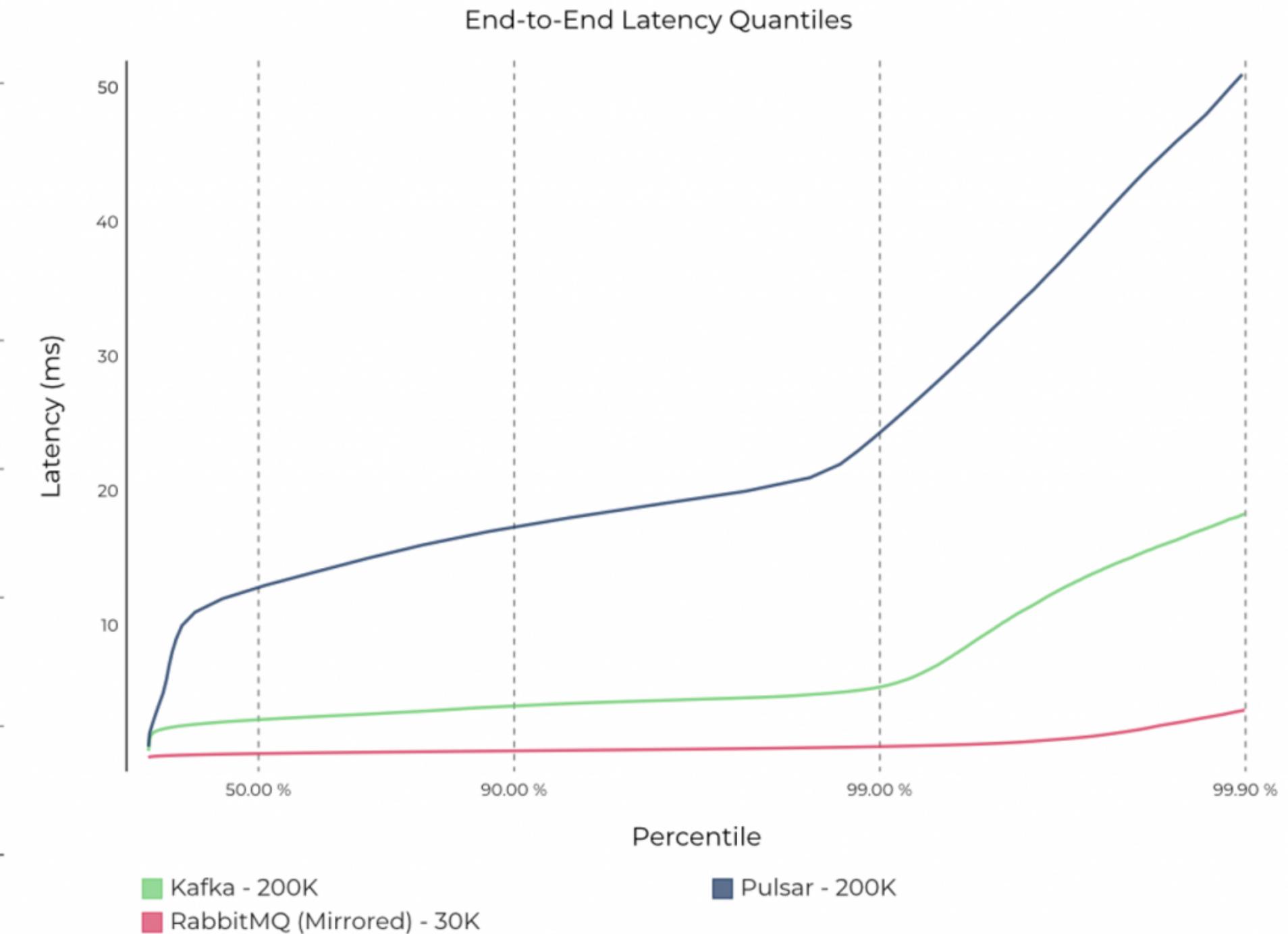
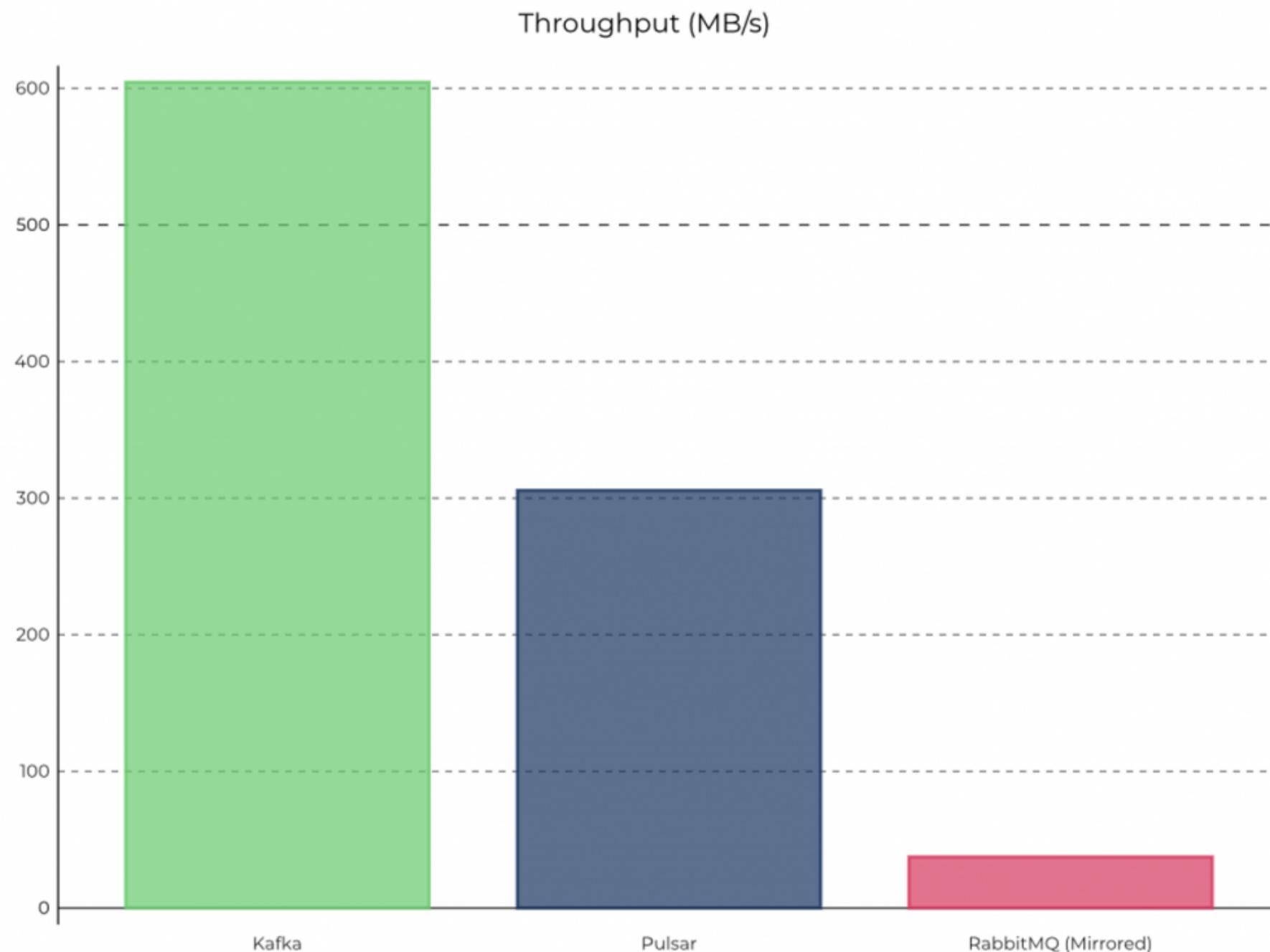
Kafka



Kafka

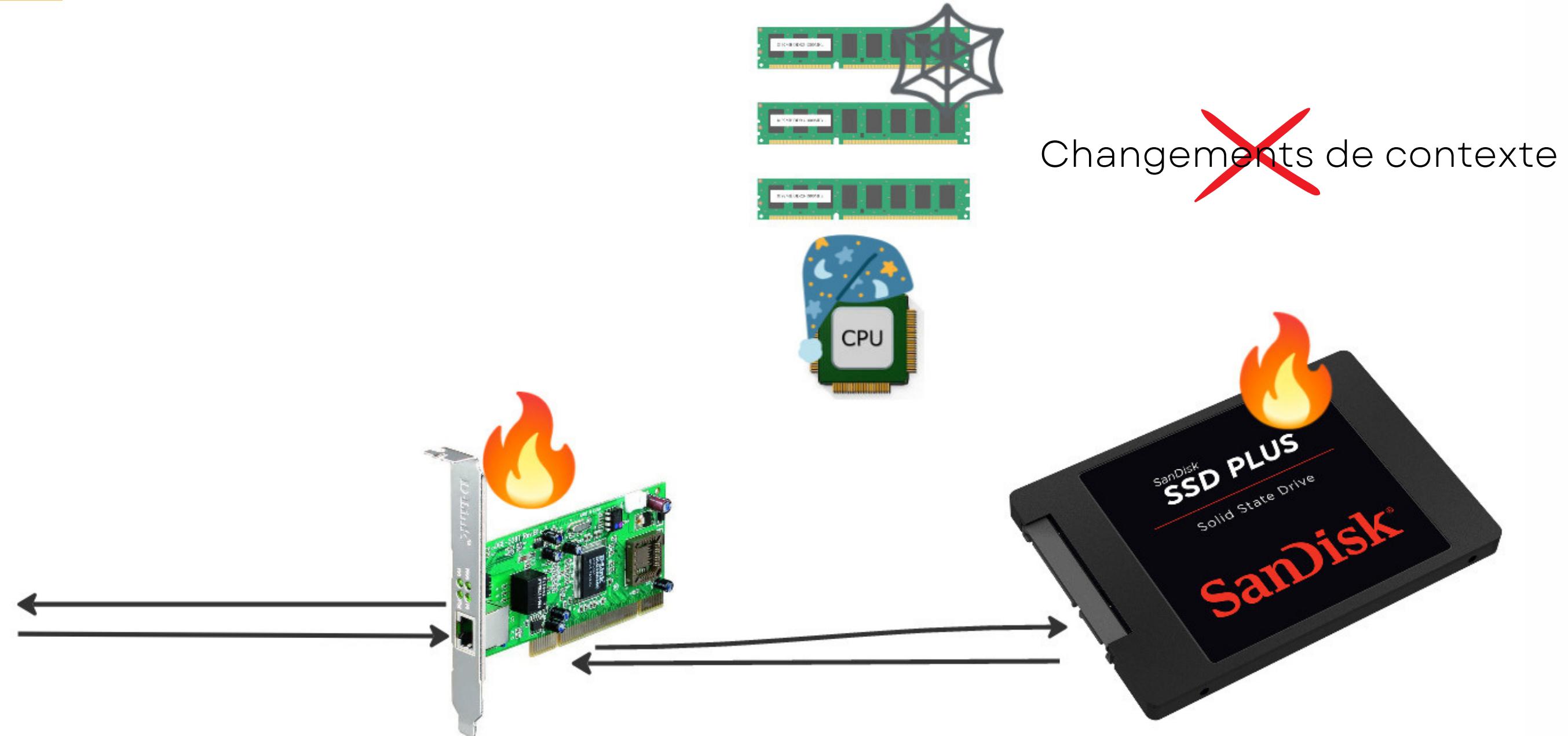


Kafka vs Pulsar vs RabbitMQ

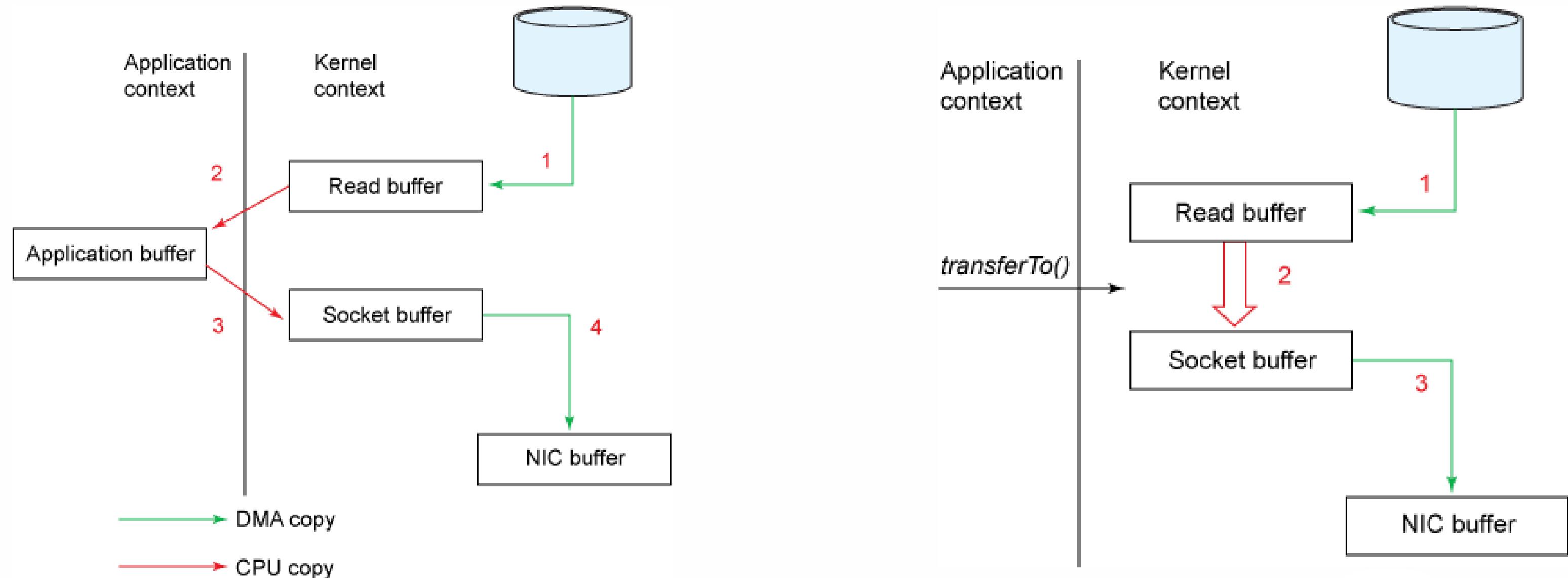


Crédit : Confluent

Zéro copie



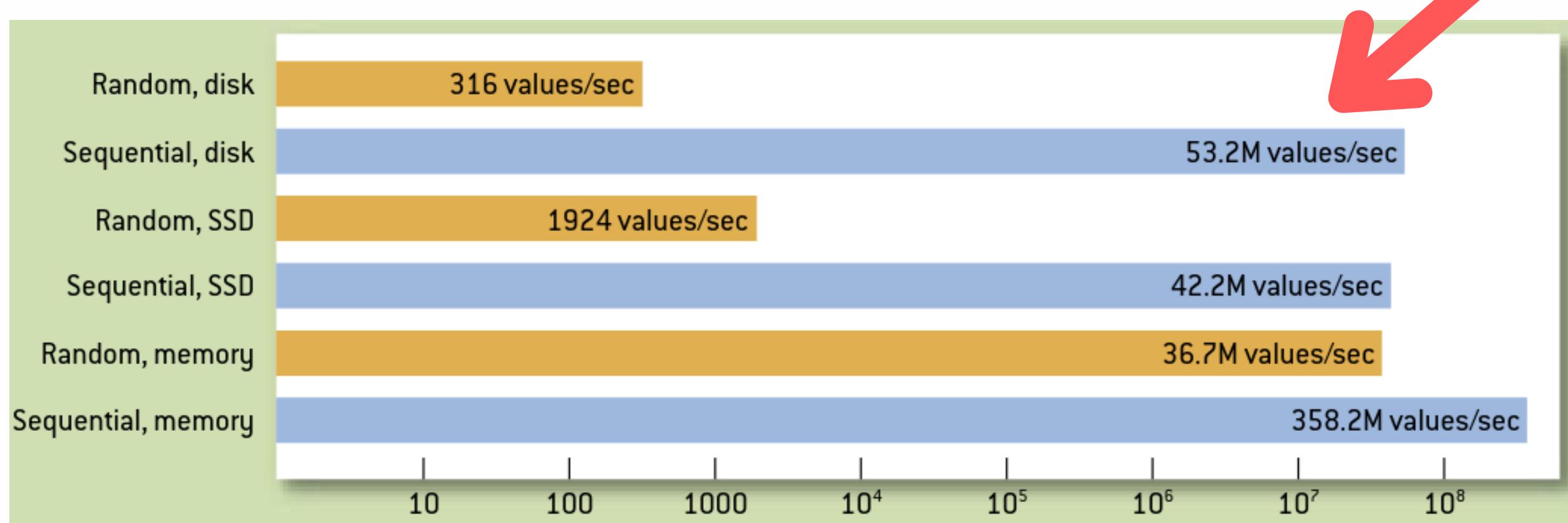
Zéro copie



```
File.read(fileDesc, buf, len);  
Socket.send(socket, buf, len);
```

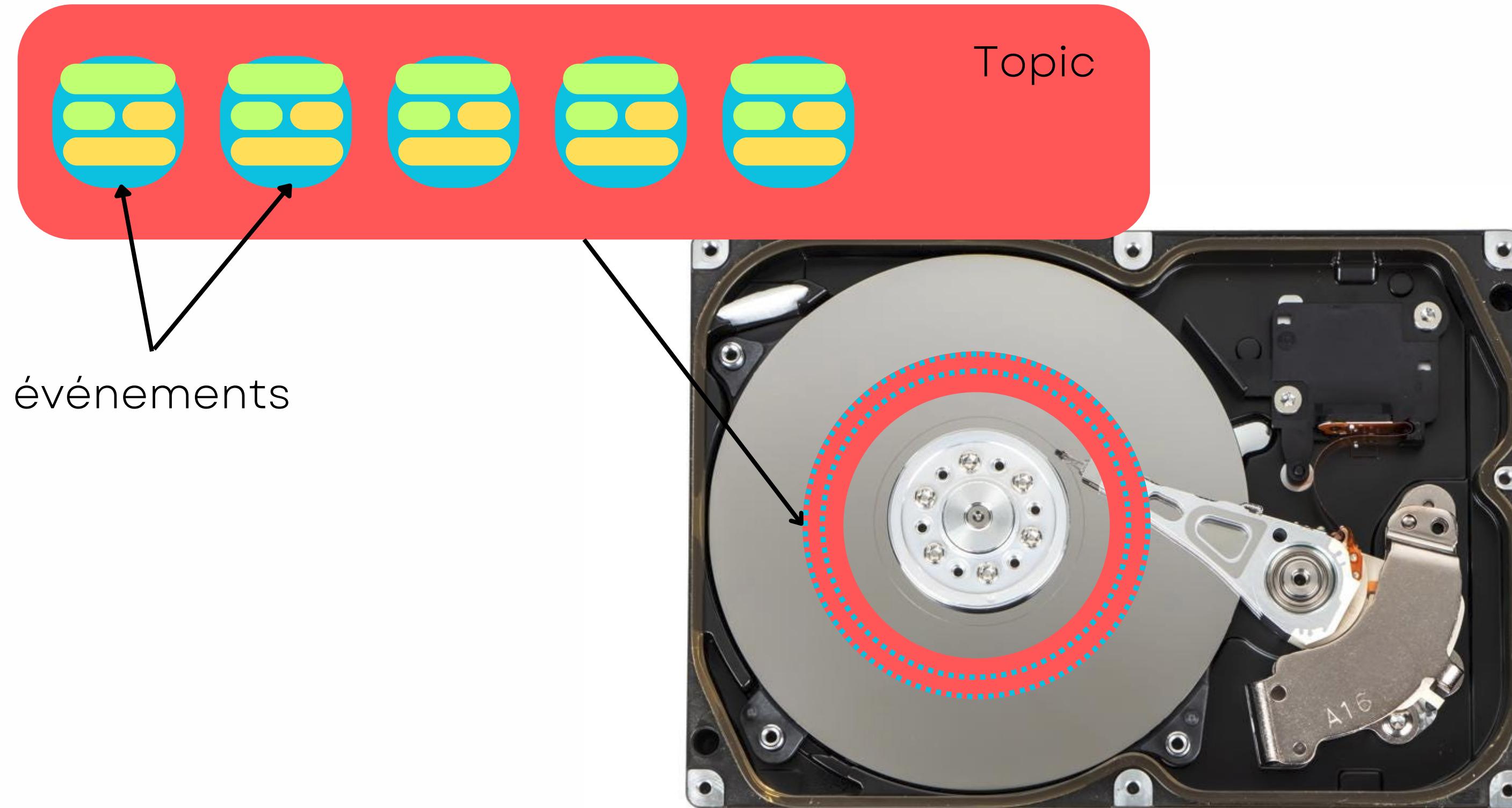
```
transferTo(position, count, writableChannel);  
(package java.nio.channels.FileChannel)
```

Lectures/écritures séquentielles

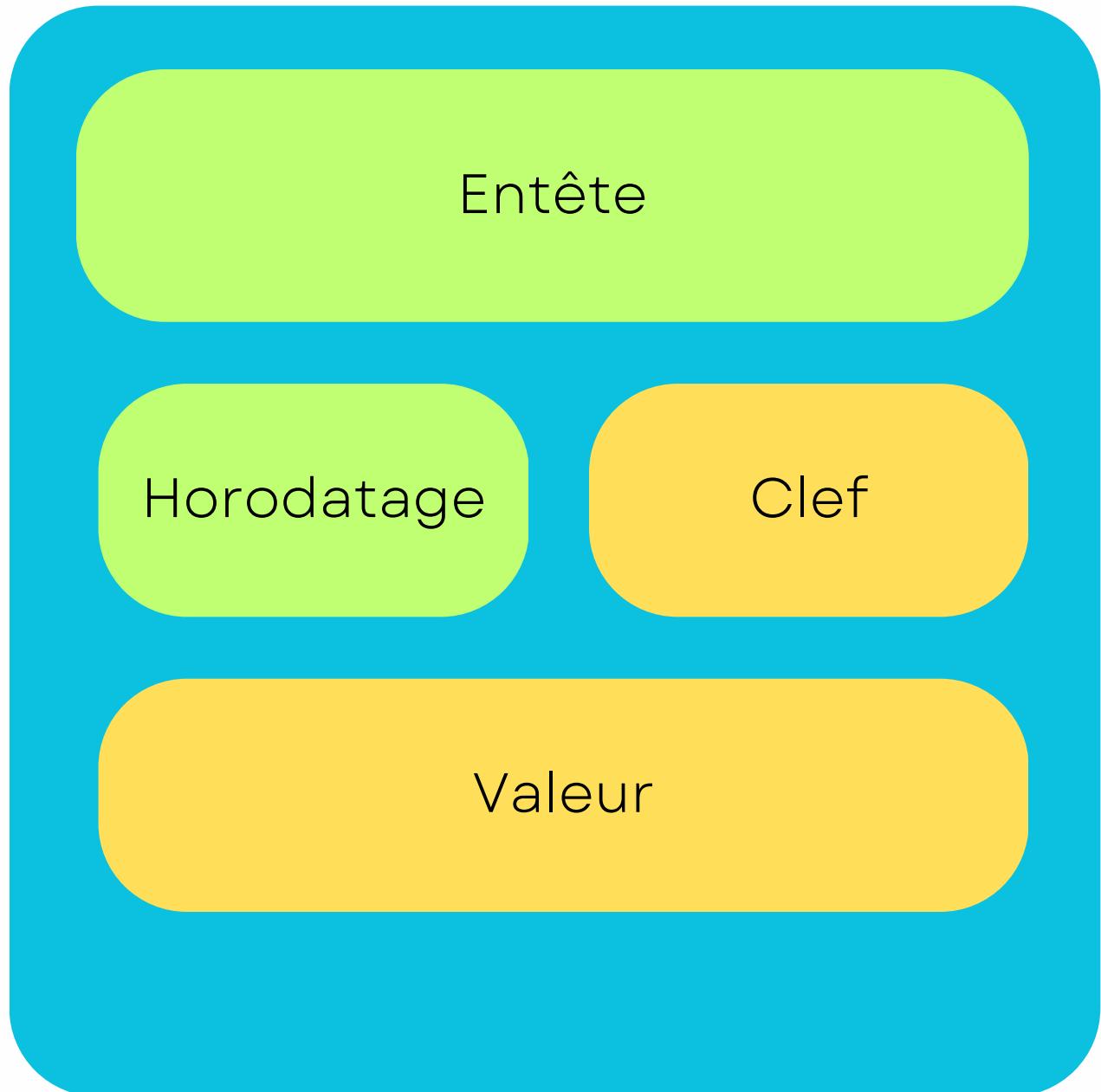


Crédit: A. Jacob

Structure de donnée "log" append-only

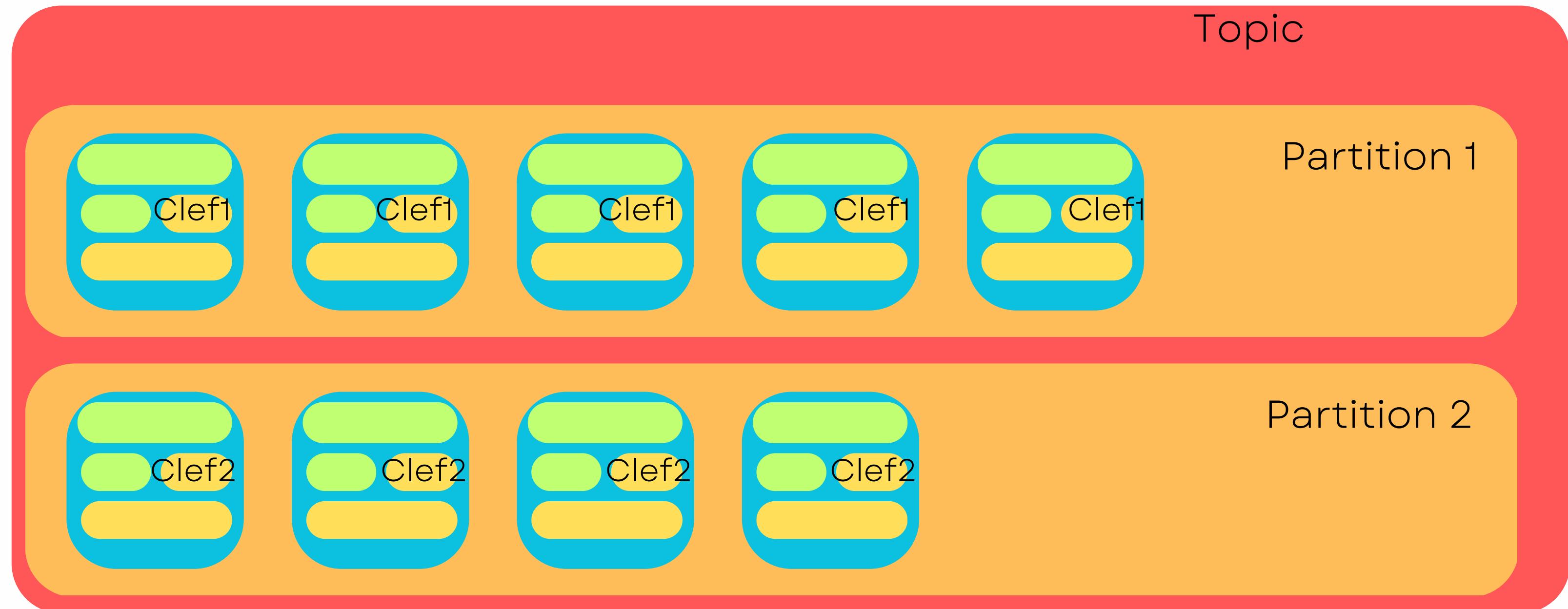


événement



Partitionnement des topics

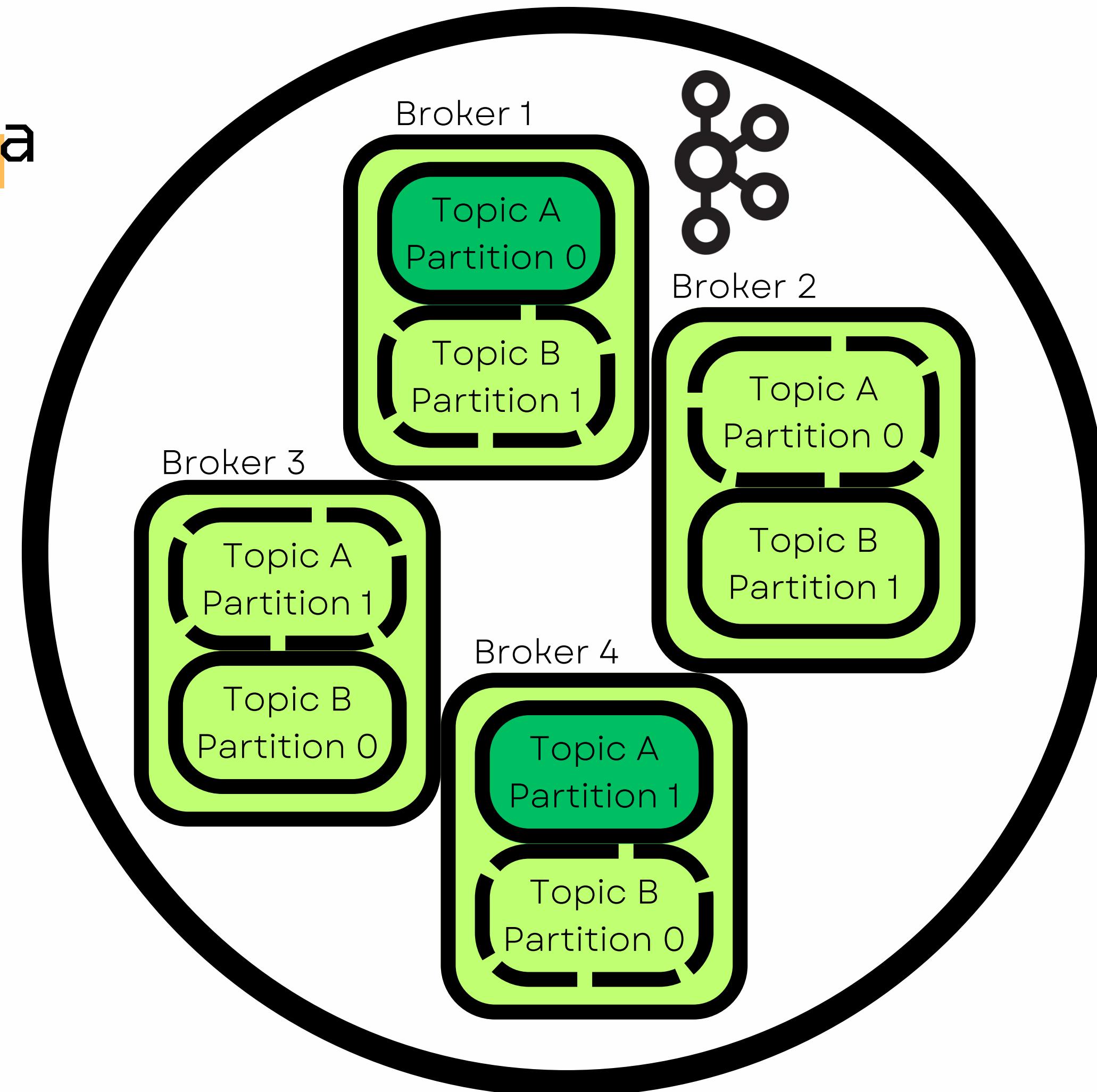
PARALLÉLISME



Cluster Kafka

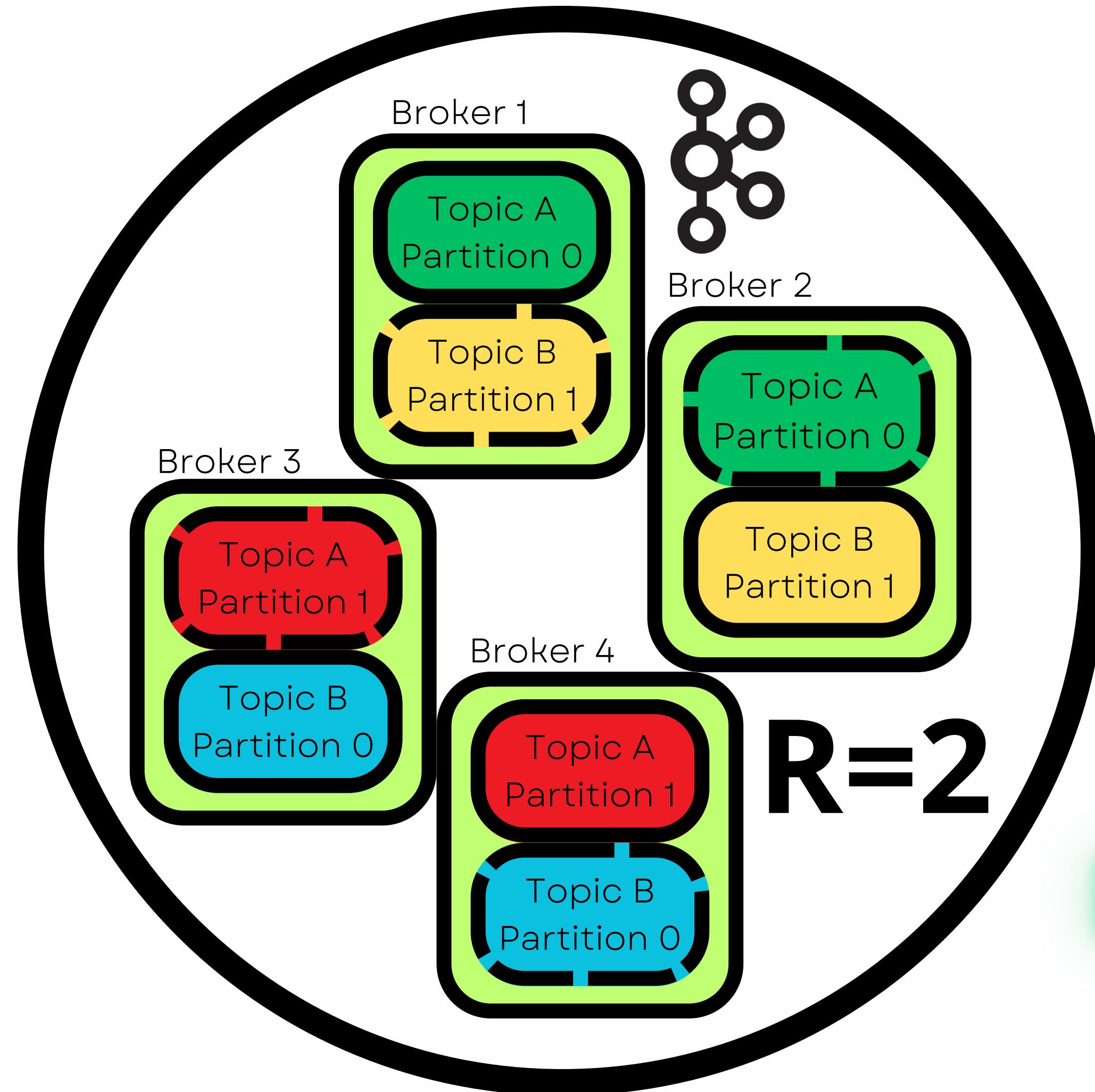
Exemple avec
4 brokers
2 topics

Facteur de réPLICATION 2



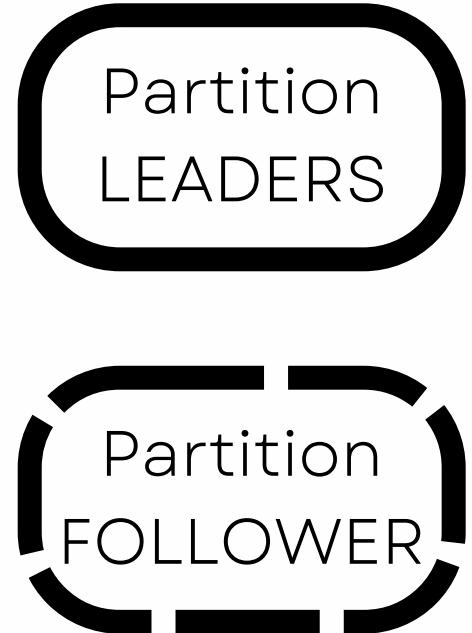
RéPLICATION

Facteur de réPLICATION
=
nombre de brokers
chargés de répliquer
chaque partition

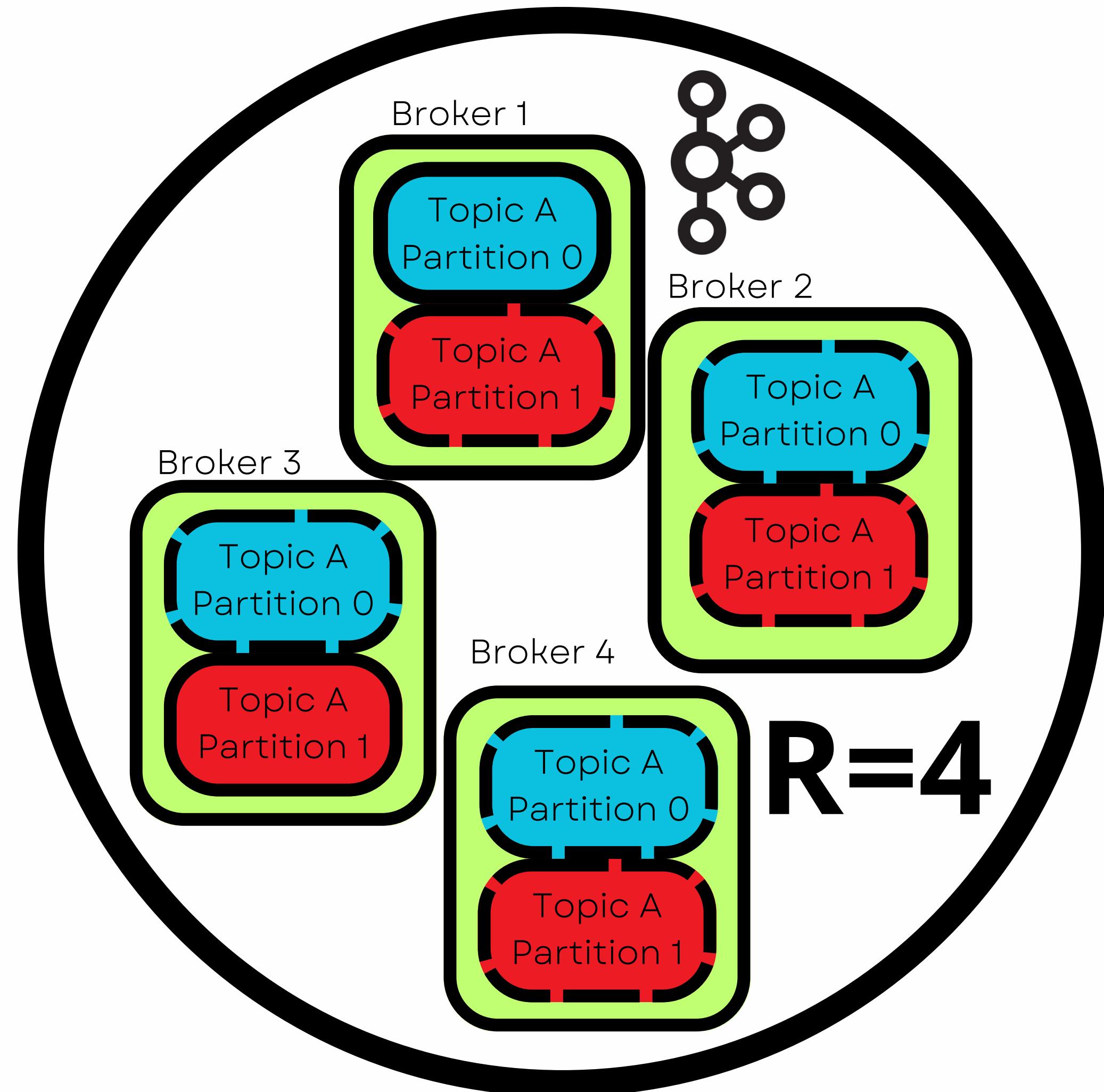


REDONDANCE

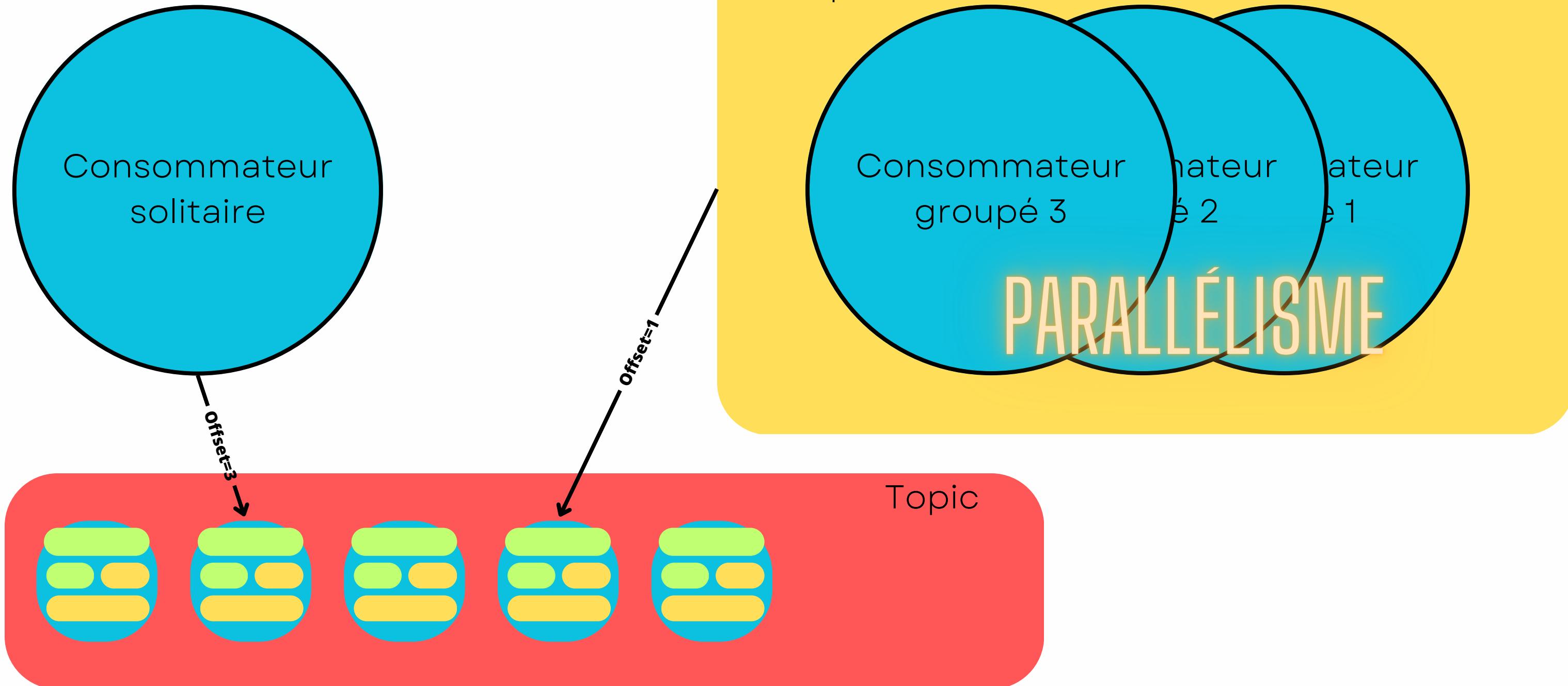
RéPLICATION



Leader élue

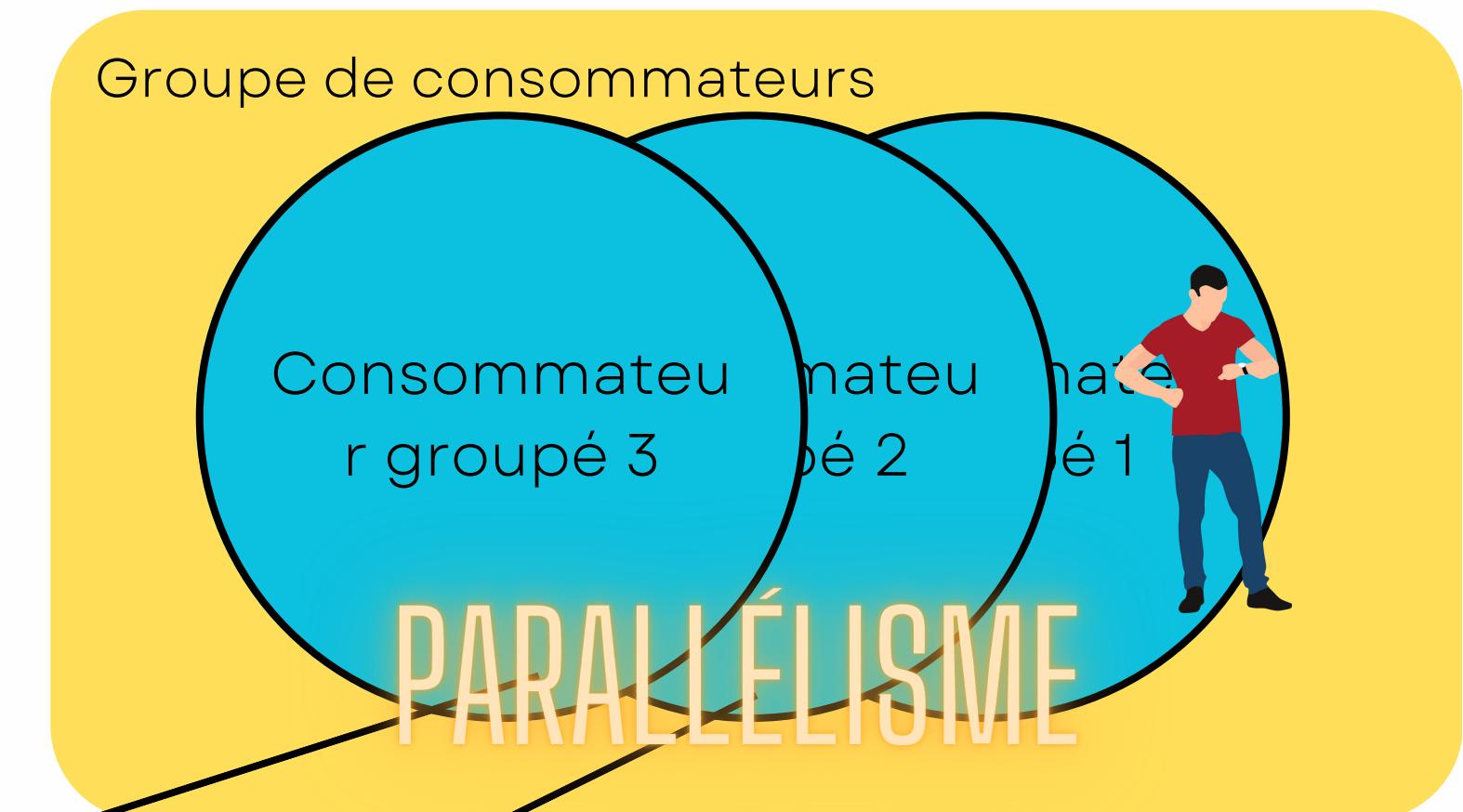
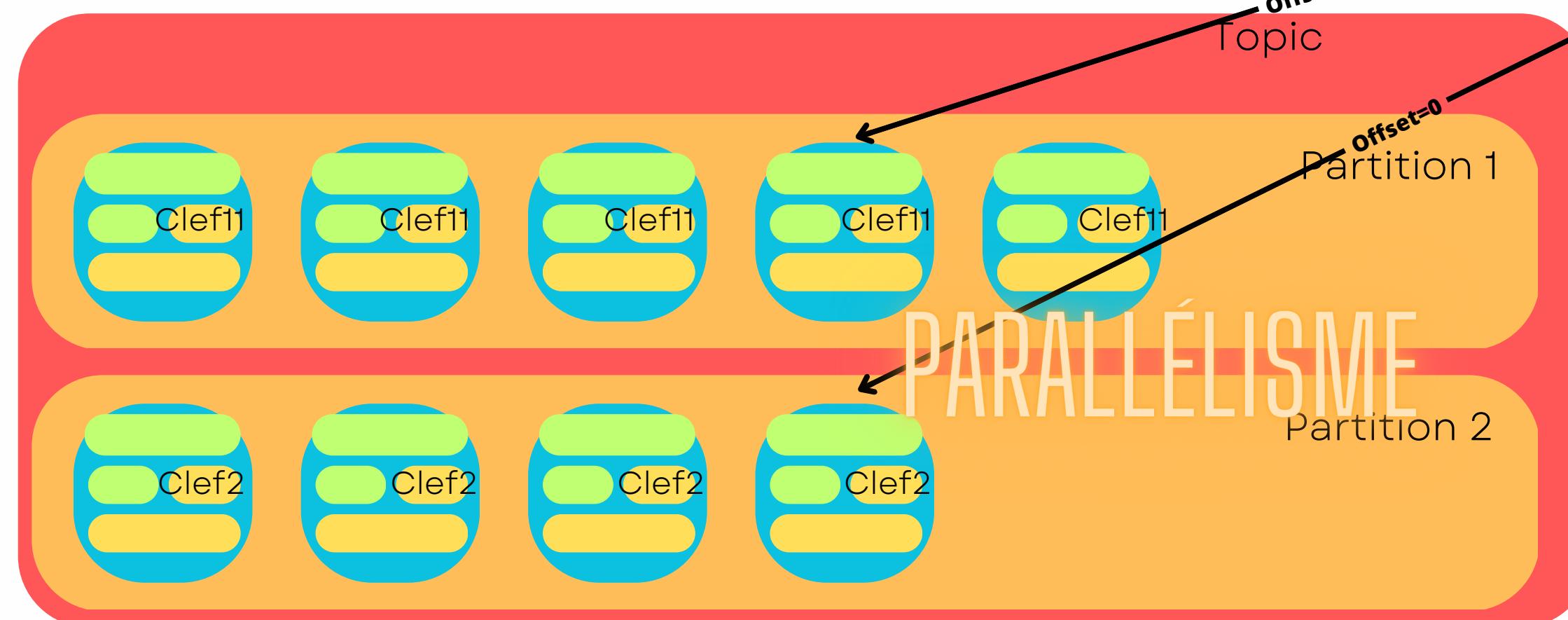


Groupe de consommateurs



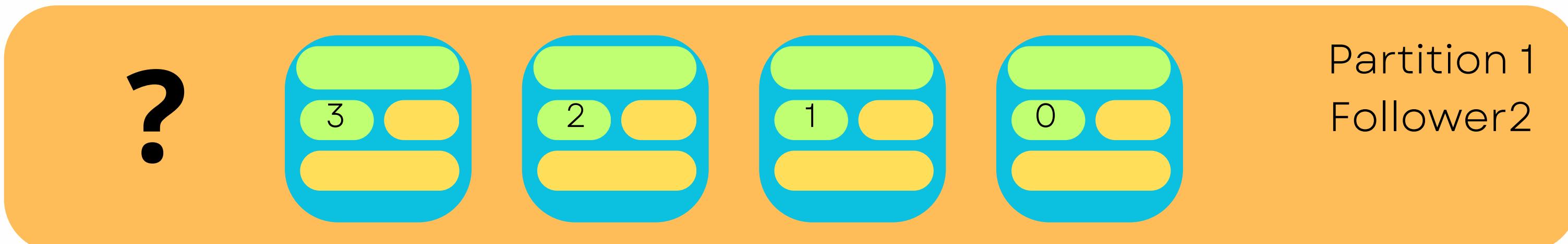
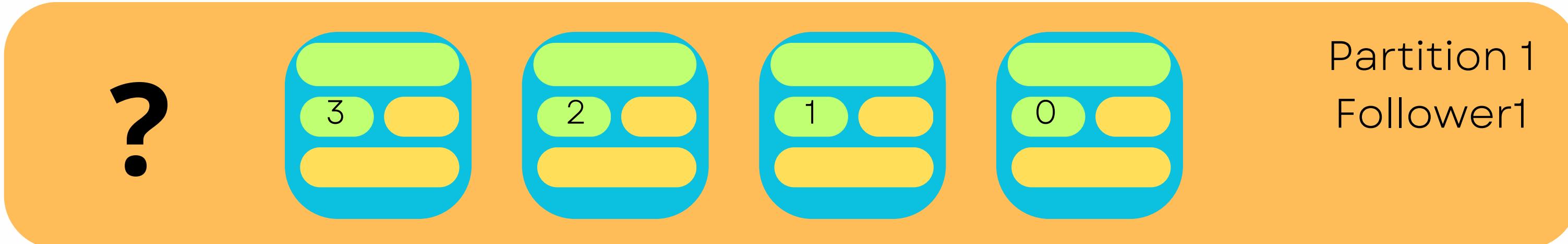
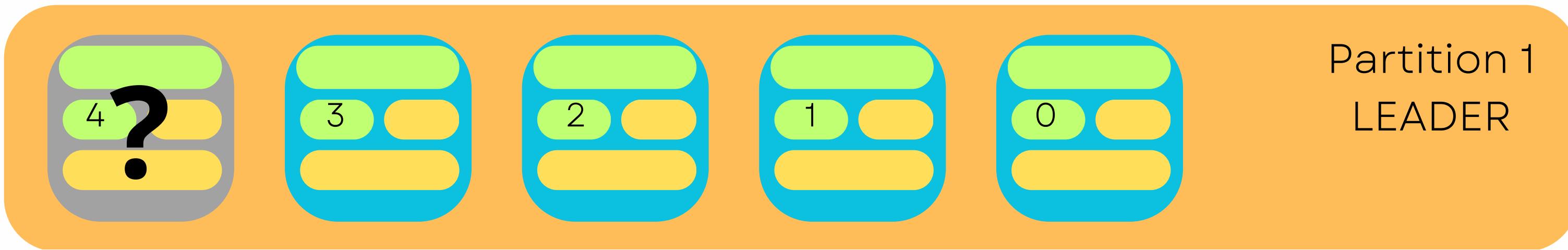
Partitionnement + groupage

NPartitionsMax = 100 x Nbrokers x Nrépliques
(empirique)



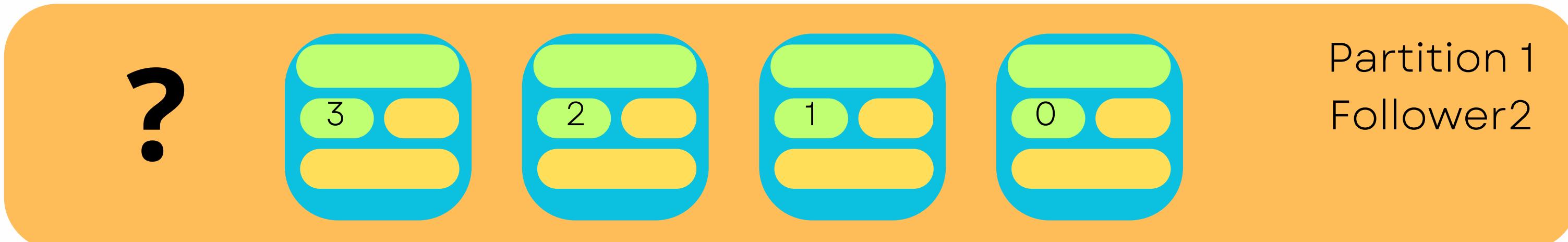
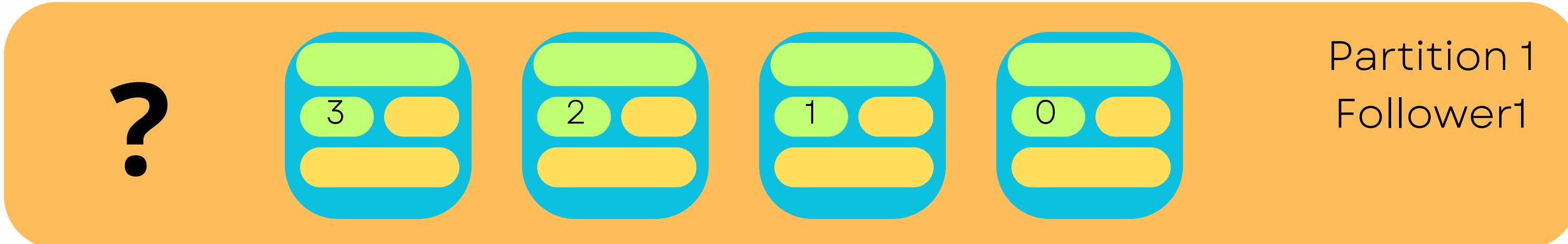
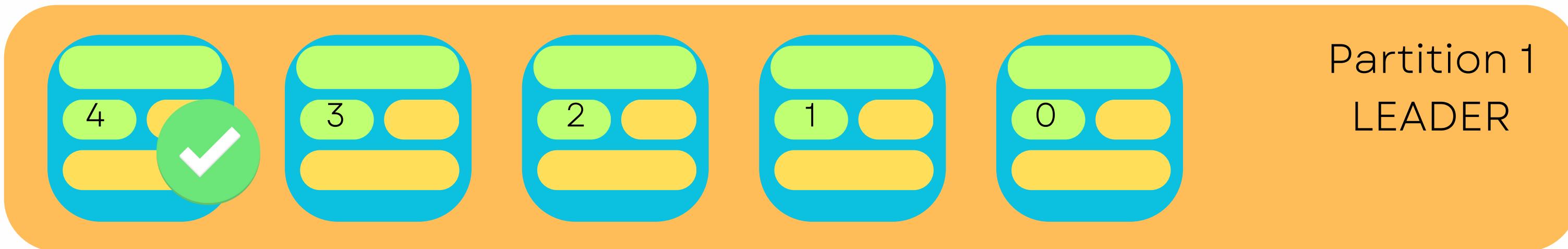
Synchronisation

ACKS=0 : FIRE & FORGET



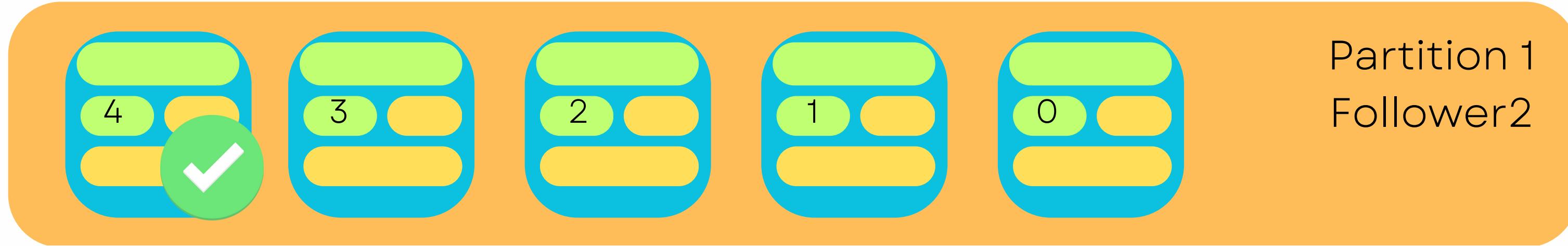
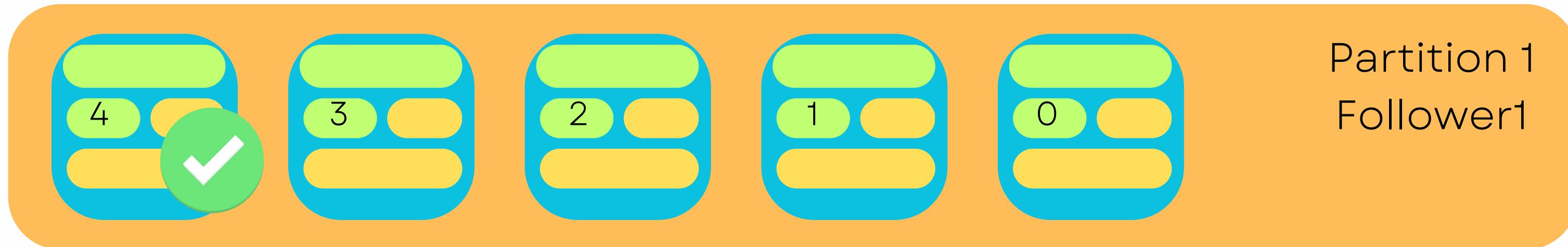
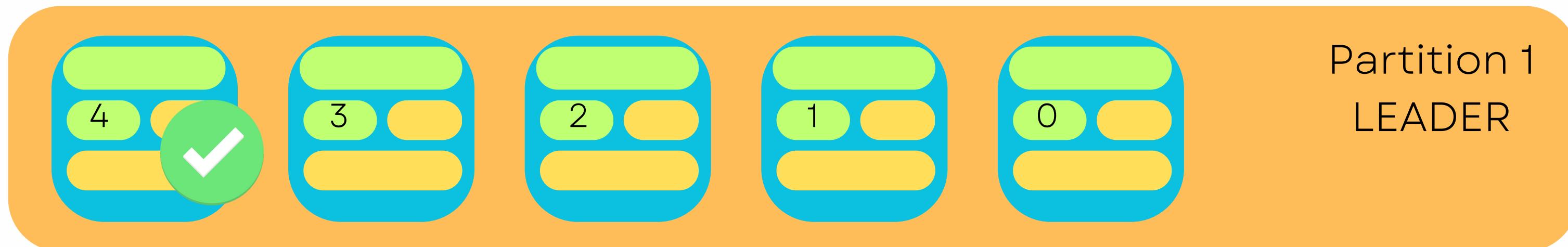
Synchronisation

ACKS=1 : LEADER OK



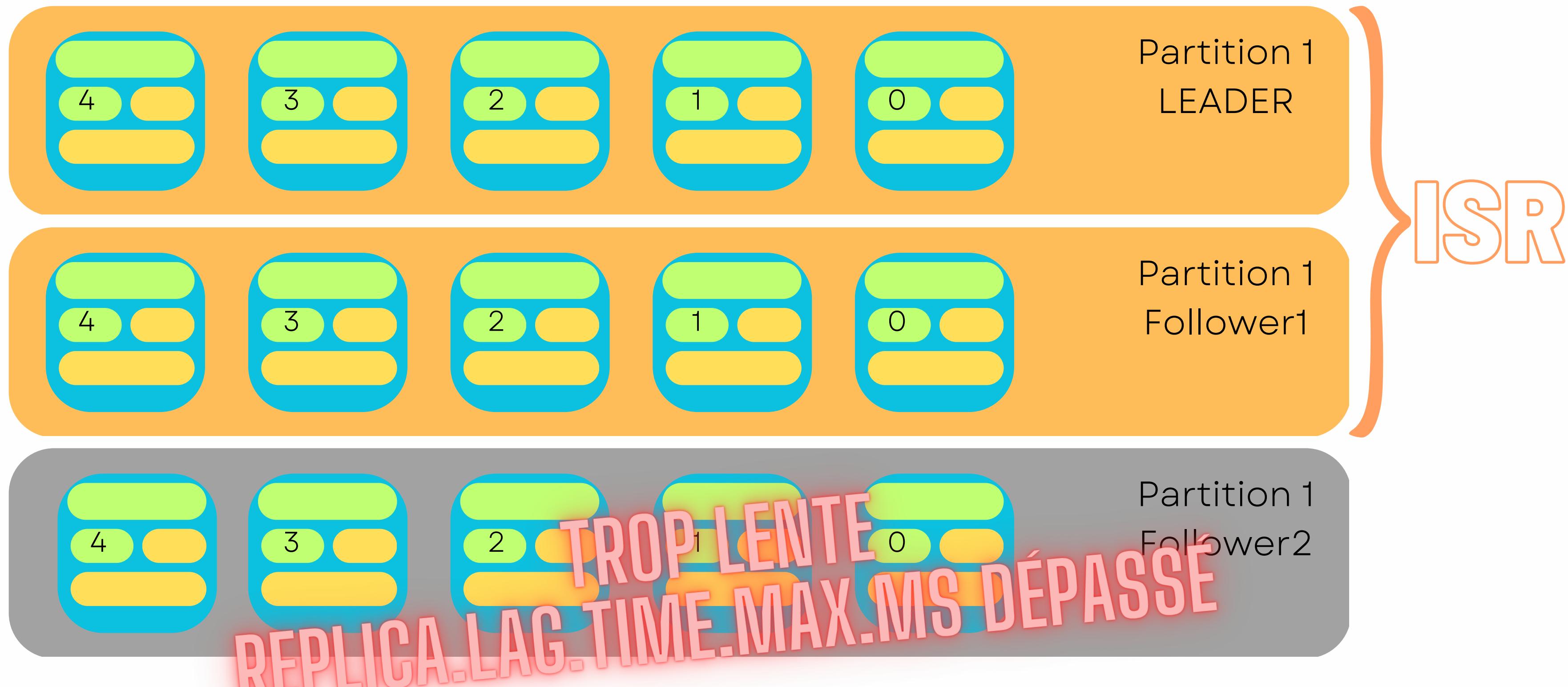
Synchronisation

ACKS=ALL : IN SYNC REPLICAS OK



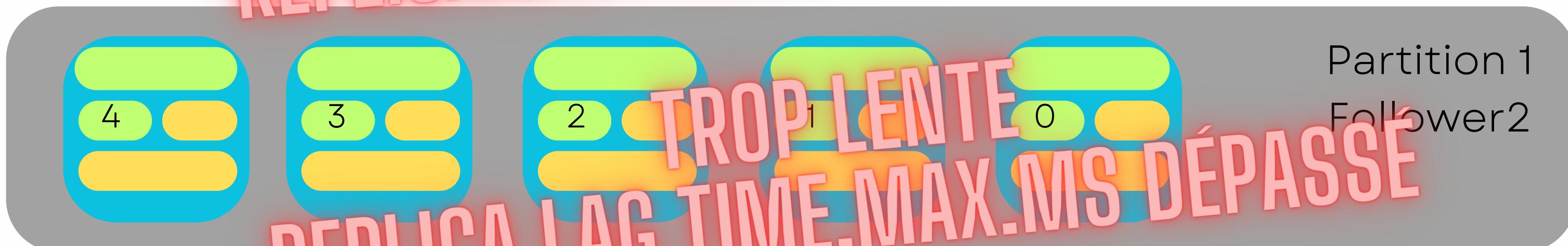
Synchronisation / ISR

Exemple : min.insync.replicas = 2

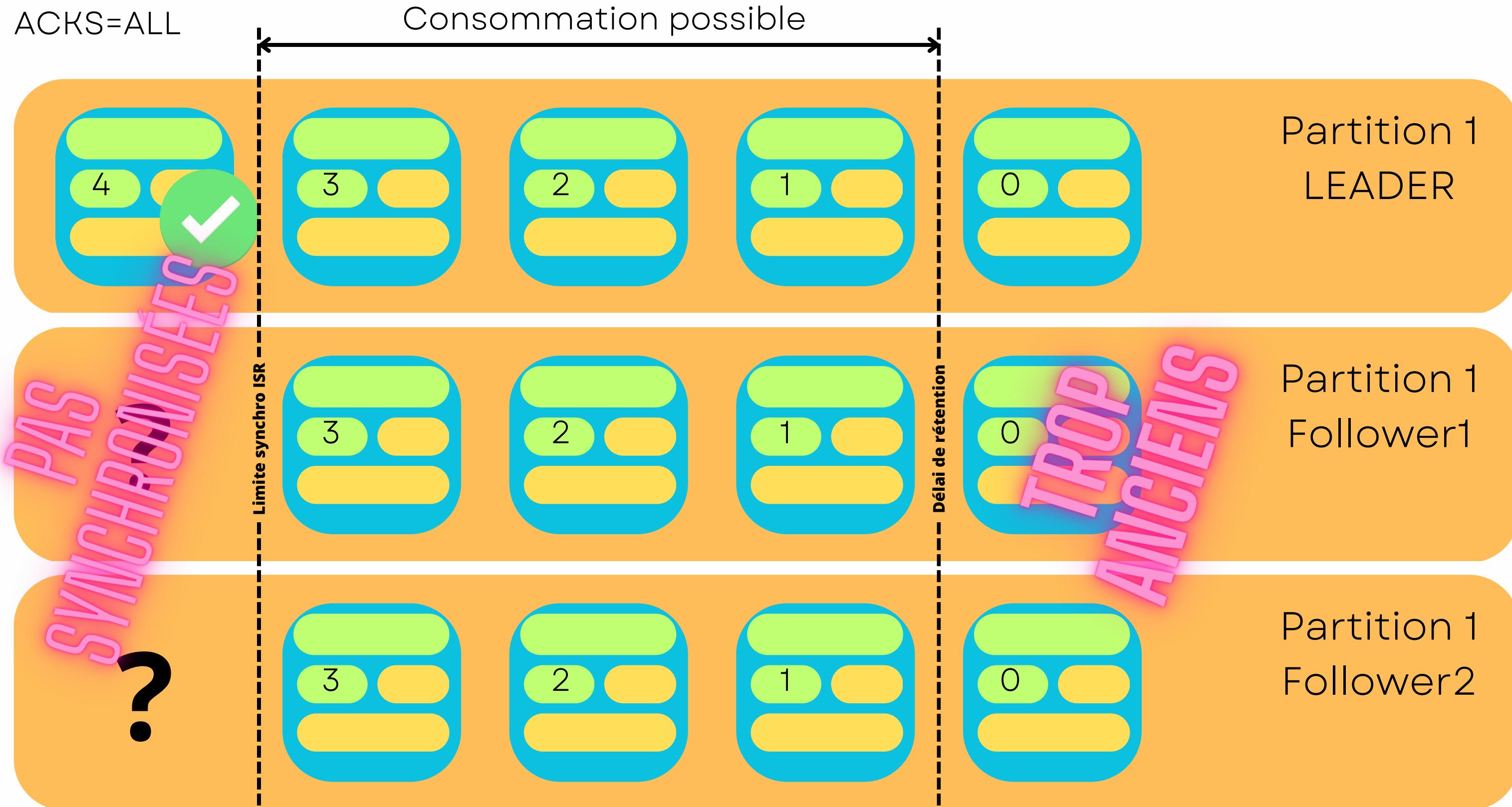


Synchronisation / ISR

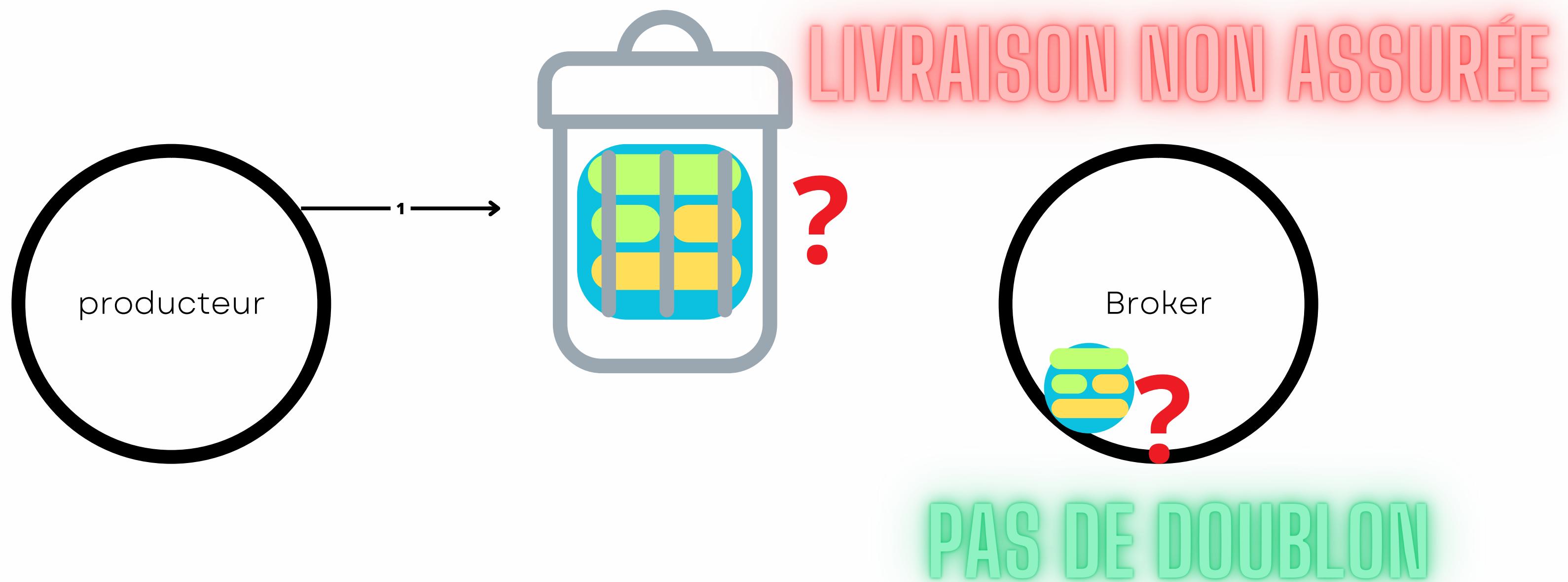
Exemple : `min.insync.replicas = 2`



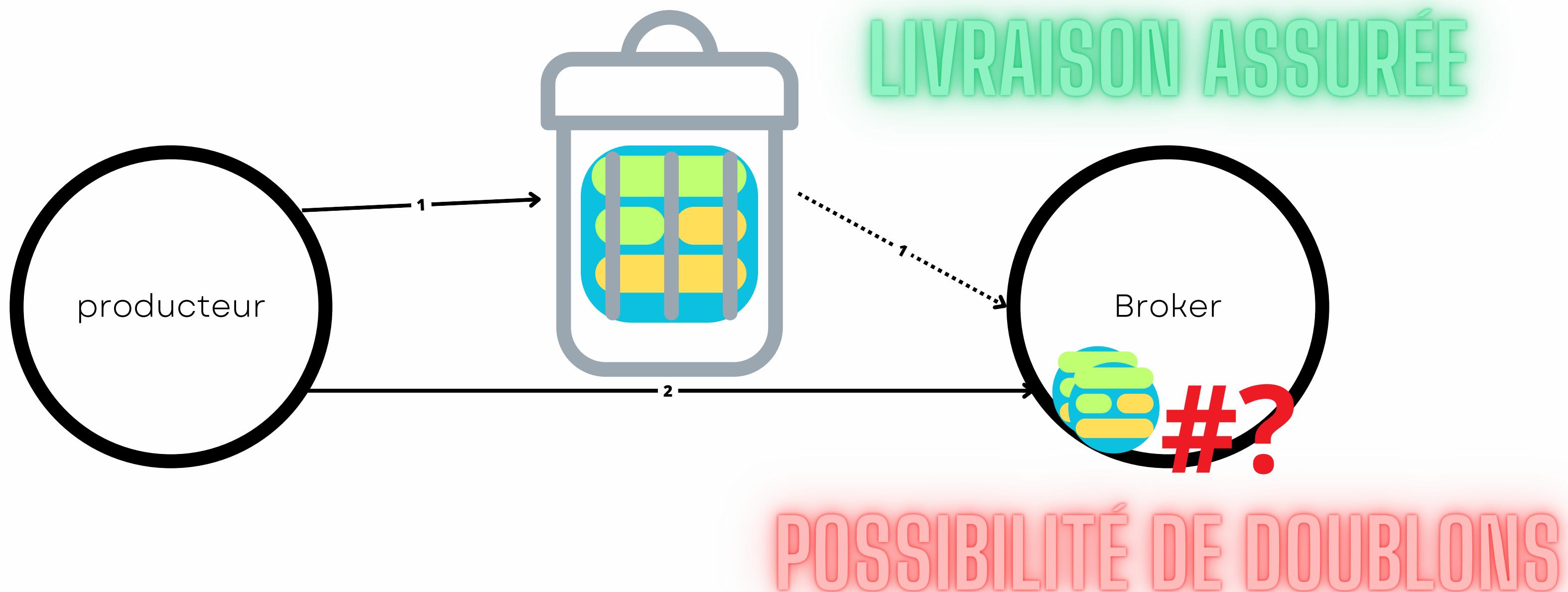
rétention & synchronisation



Mode de livraison "at most once"

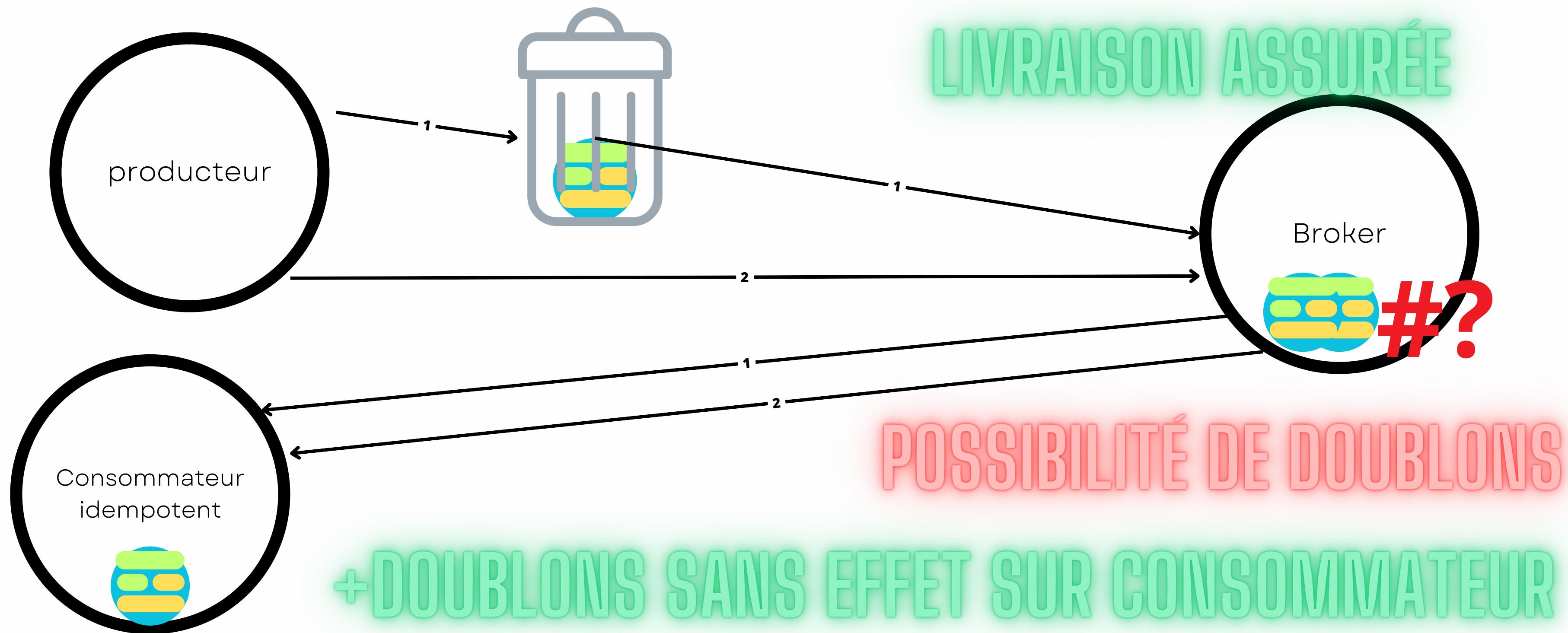


Mode de livraison "at least once"



Mode de livraison "exactly once"

=AT LEAST ONCE + IDEMPOTENCE



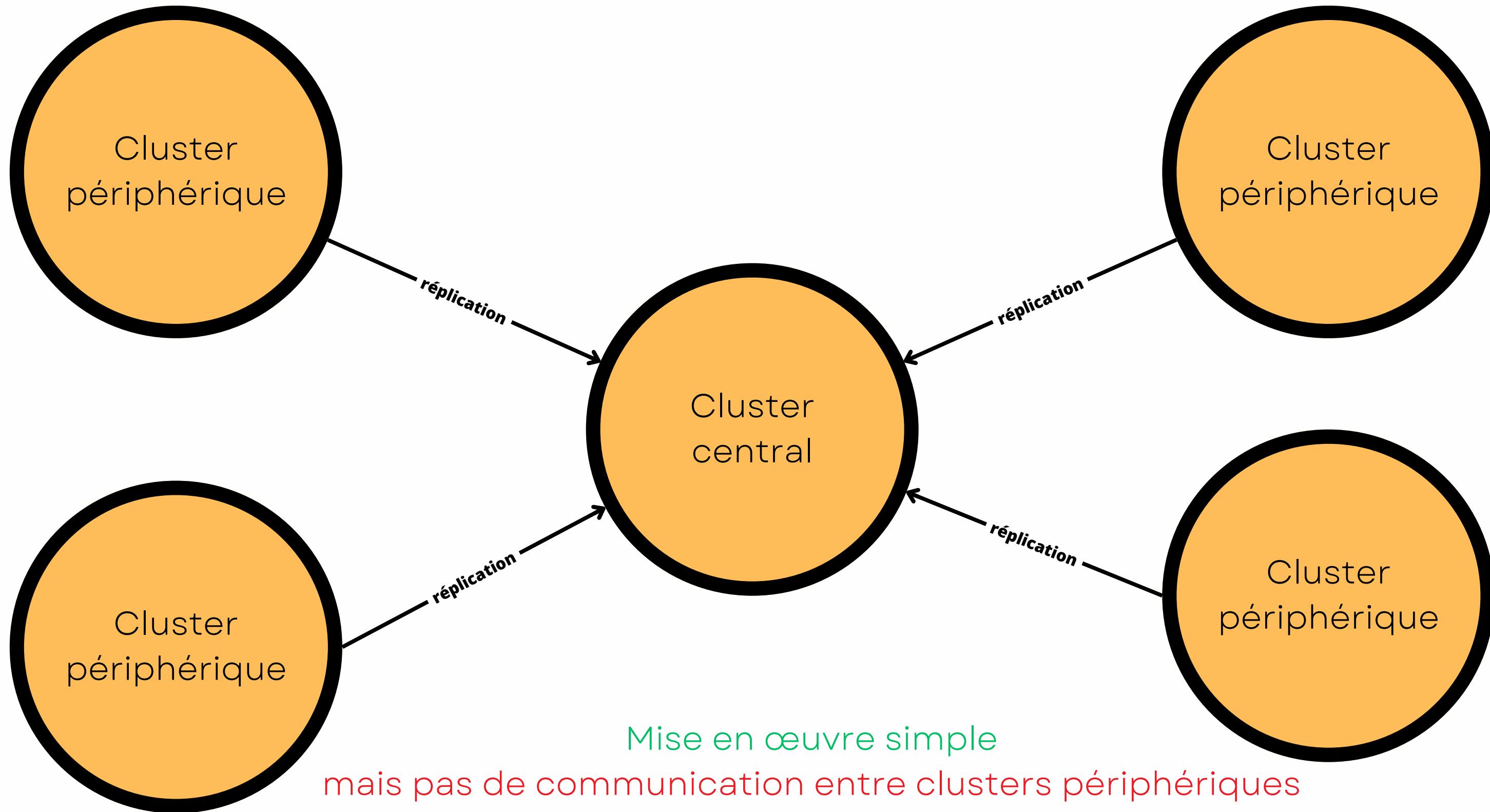
Transactions

```
producer.beginTransaction();
try {
    ...
    producer.send(...);

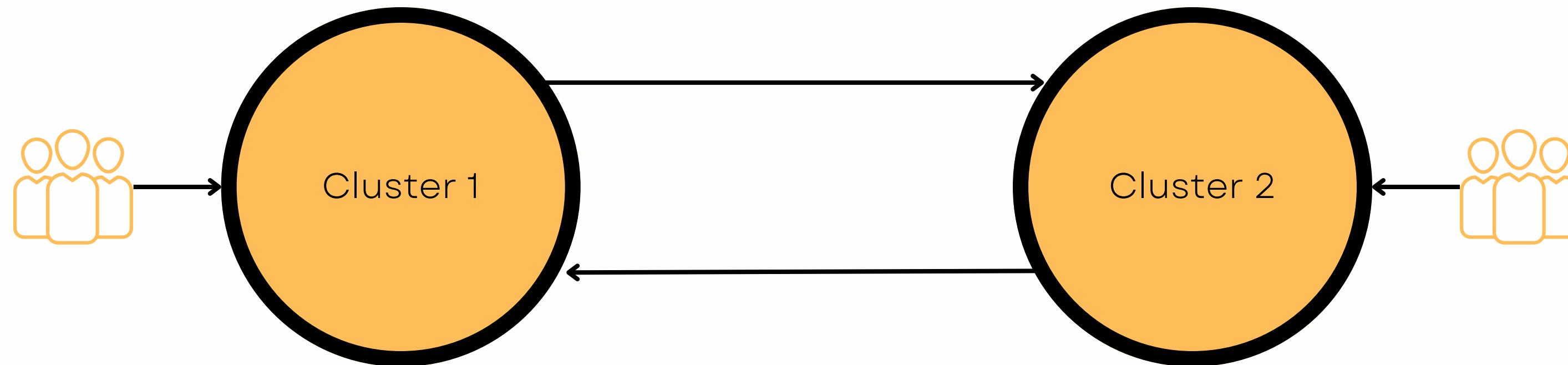
    producer.commitTransaction();

} catch (KafkaException e) {producer.abortTransaction();throw e;}
```

Architecture multi-DC "en étoile"

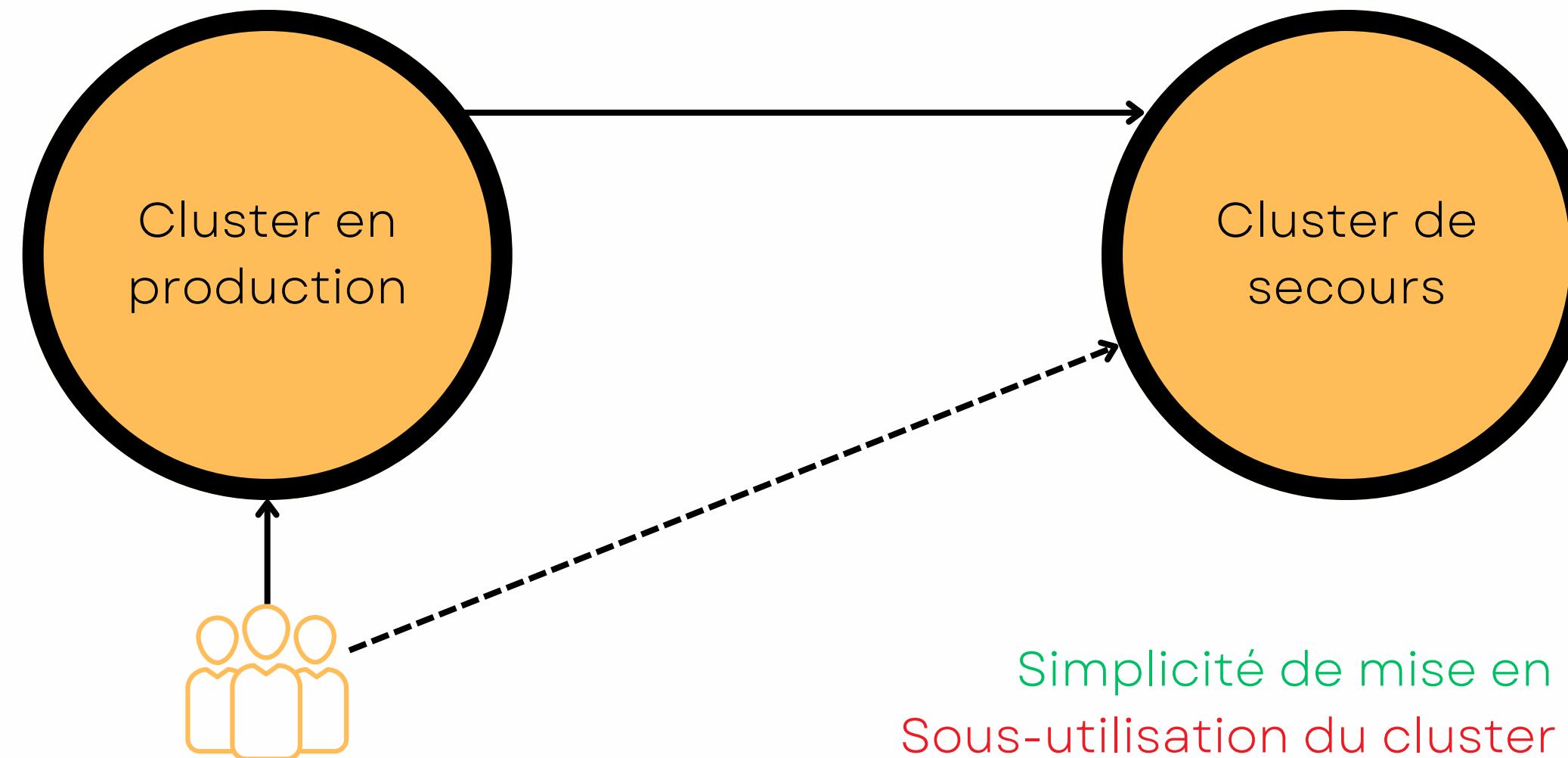


Architecture multi-DC "actif-actif"



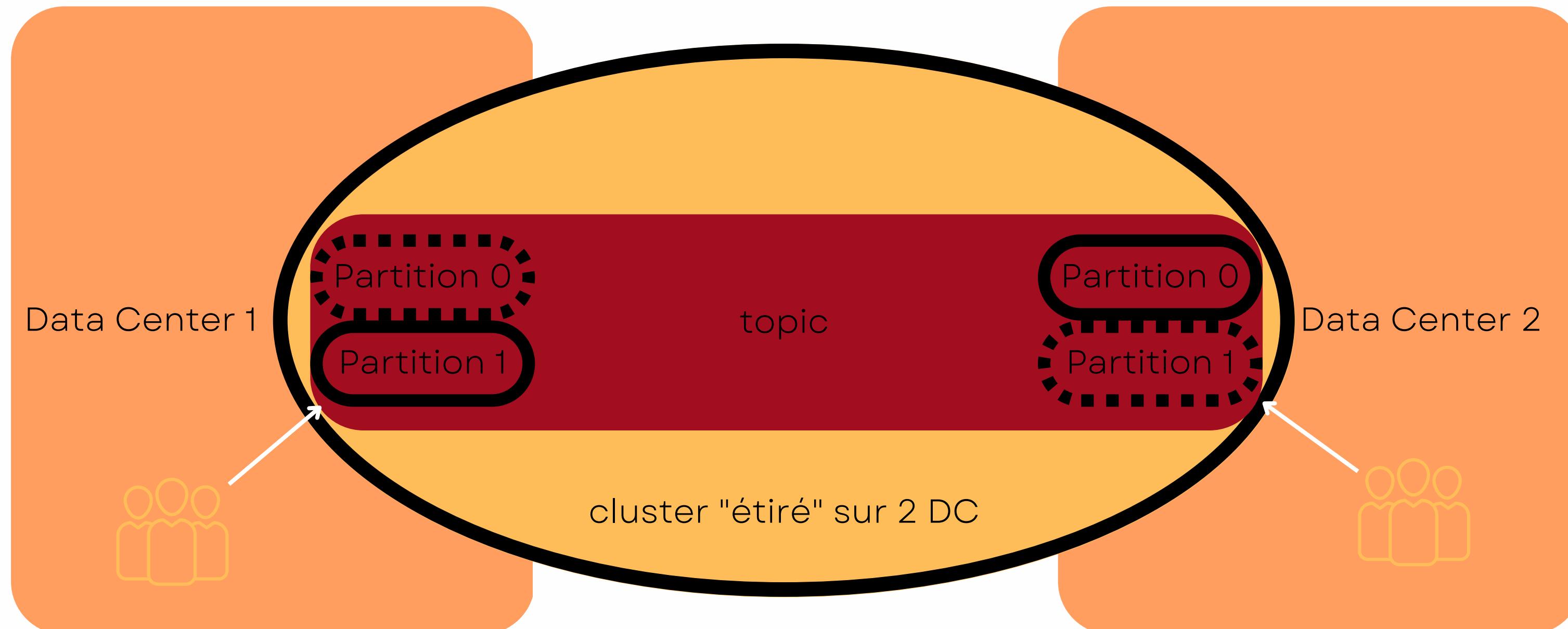
+de centres => +de résilience et +redondance
Performances pour les clients proches
mais complexité R/W async et cohérence données

Architecture multi-DC "actif-passif"



Simplicité de mise en œuvre
Sous-utilisation du cluster de secours
Possibilité de perte de qq événements

Architecture multi-DC "cluster étiré"



RéPLICATION gérée par Kafka

Nécessite une très bonne connexion inter-DC
si possible au moins 3 DC pour R=3

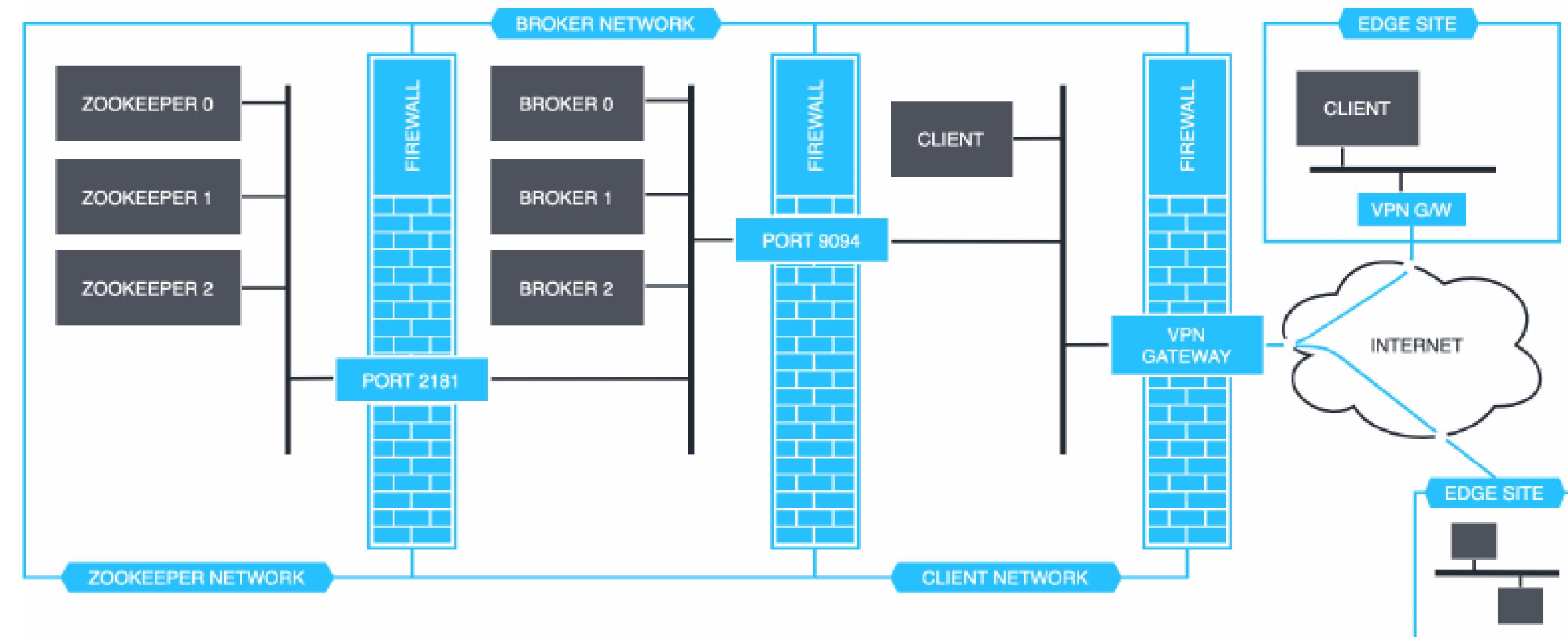
Pas de protection si pbm interne clients ou Kafka

Sécurité

- Inexistante par défaut

Sécurité

- Inexistante par défaut
- Ségrégation réseau

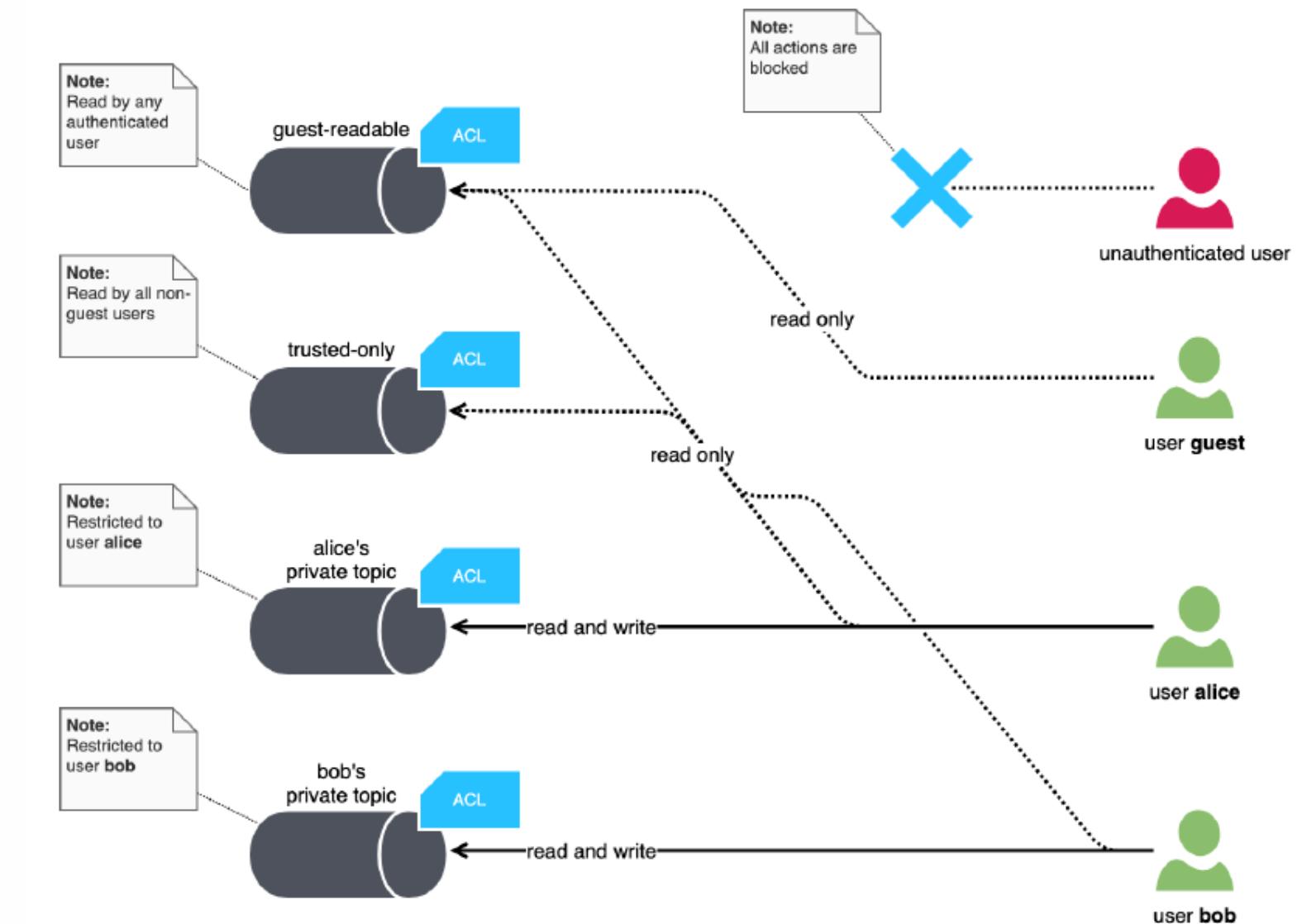


Sécurité

- Inexistante par défaut
- Ségrégation réseau
- Chiffrement p2p

Sécurité

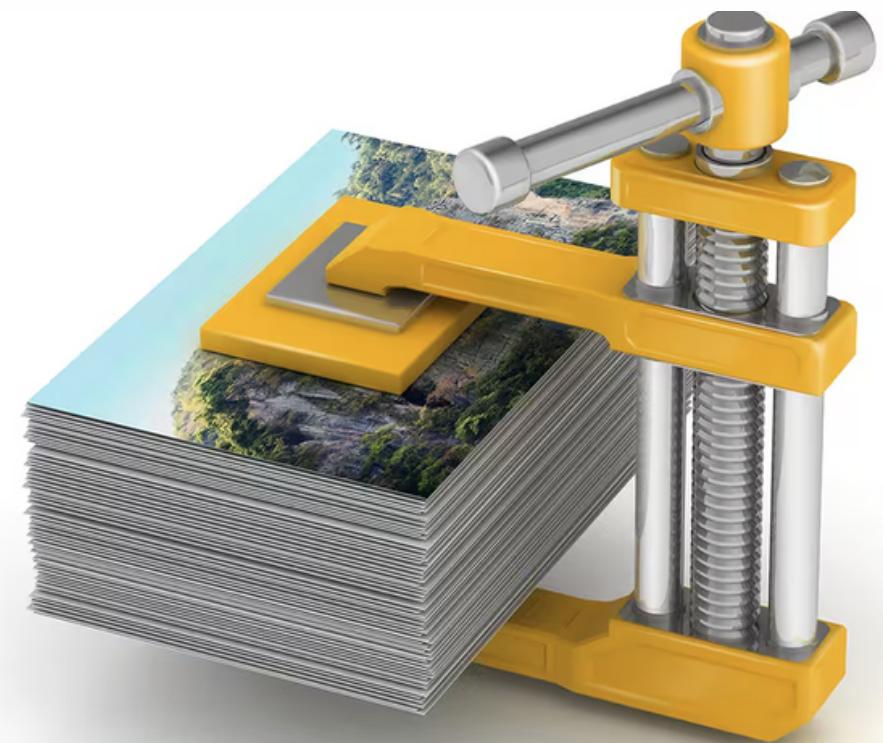
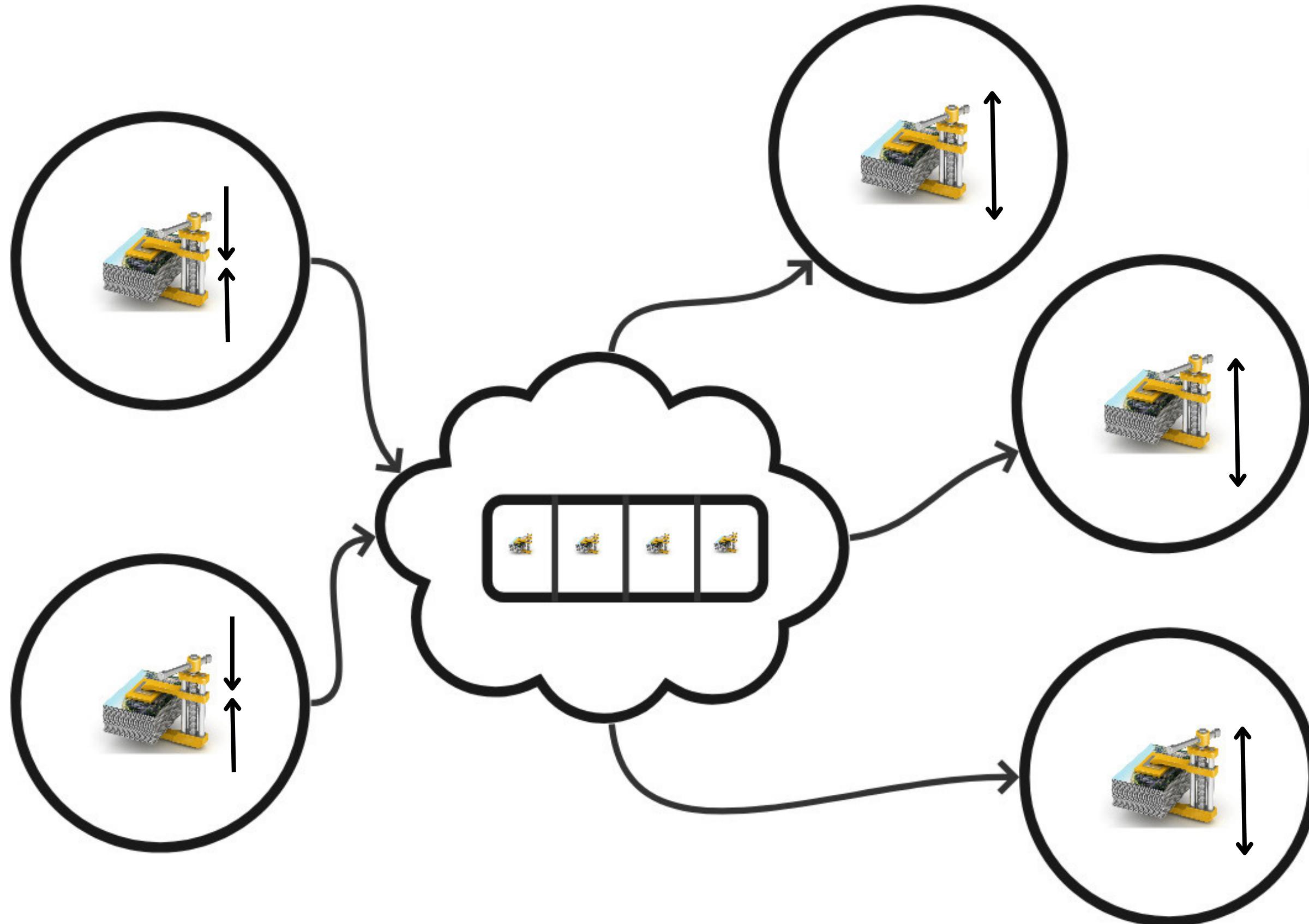
- Inexistante par défaut
- Ségrégation réseau
- Chiffrement p2p
- Authentification
- Autorisations (ACL)



Sécurité

- Inexistante par défaut
- Ségrégation réseau
- Chiffrement p2p
- Authentification
- Autorisations (ACL)
- Quotas (DoS)

Compression de point à point



5 < Q < 7

Si faible entropie
=> pas de chiffrement



Merci