Kafka Theory Part 2

Kafka Theory
Part 2

- Brokers & Topics
- Topic replication Factor
- Producer Acks
- Topic Durablity
- Zookeeper
- KRaft

Brokers & Topics

Kafka Theory Part 2

- Topic replication Factor
- Producer Acks
- Topic Durablity
- Zookeeper
- KRaft

Brokers & Topics

Brokers & Topics

- Brokers
- Example: Topic A (3 partition)- and Topic B (2 partition)
- Broker discovery: each broker is bootstrap servers. that means you only need to connect to one broker and clients will know how to connect to entire cluster
- Each broker knows all brokers, topic and partition

Brokers & Topics

- Brokers

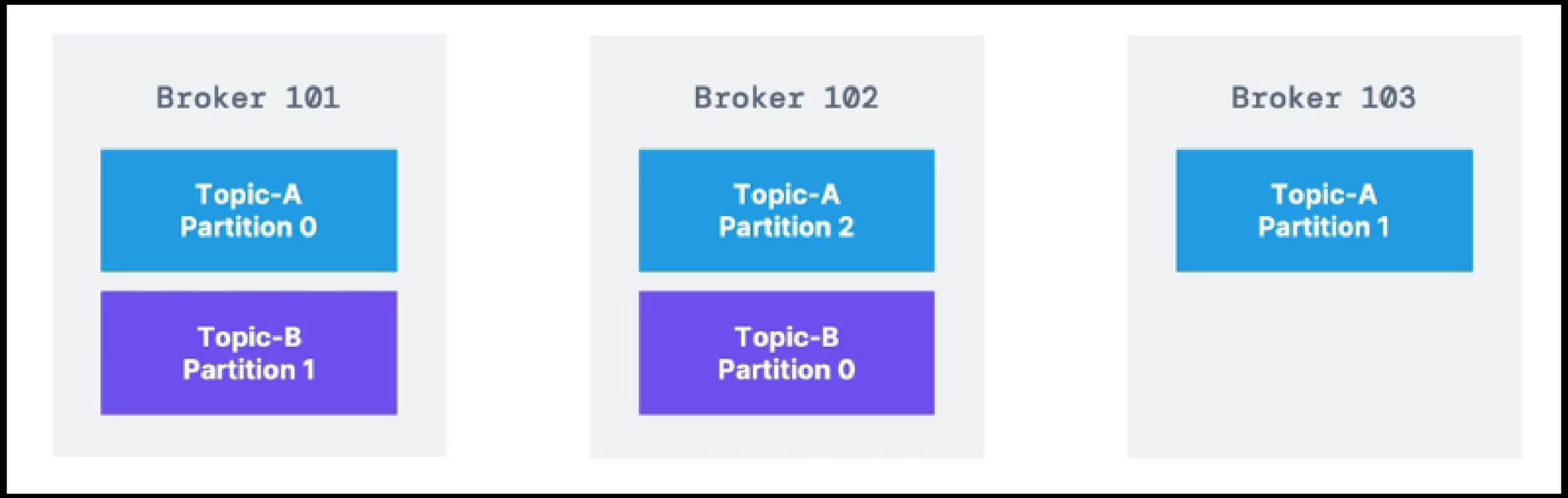
- Example: Topic A (3 partition)- and Topic B (2 partition)

- Broker discovery: each broker is bootstrap servers. that means you only need to connect to one broker and clients will know how to connect to entire cluster

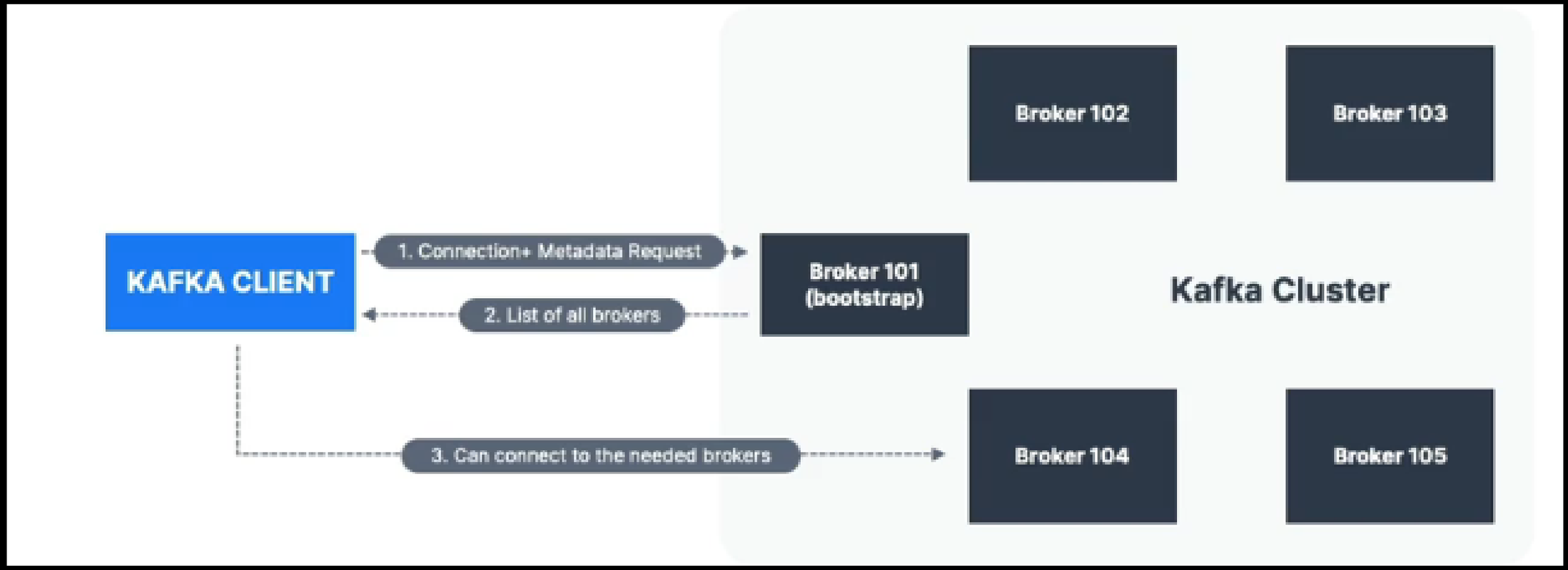
- Each broker knows all brokers, topic and partition

Brokers

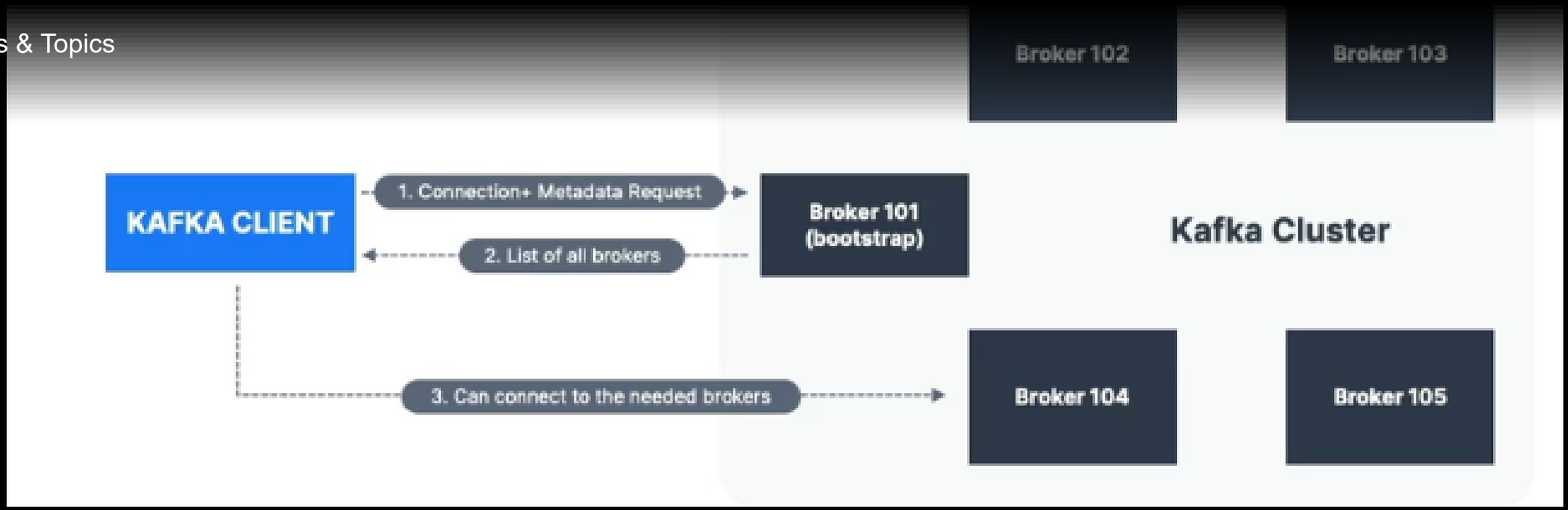
- Kafka cluster = multiple brokers(servers)
- Each Broker is identified with ID
- Each broker contains certain topic partition
- After connection to any broker, you are connected to entire cluster



Example: Topic A (3 partition)- and Topic B (2 partition)



Broker discovery: each broker is bootstrap servers. that means you only need to connect to one broker and clients will know how to connect to entire



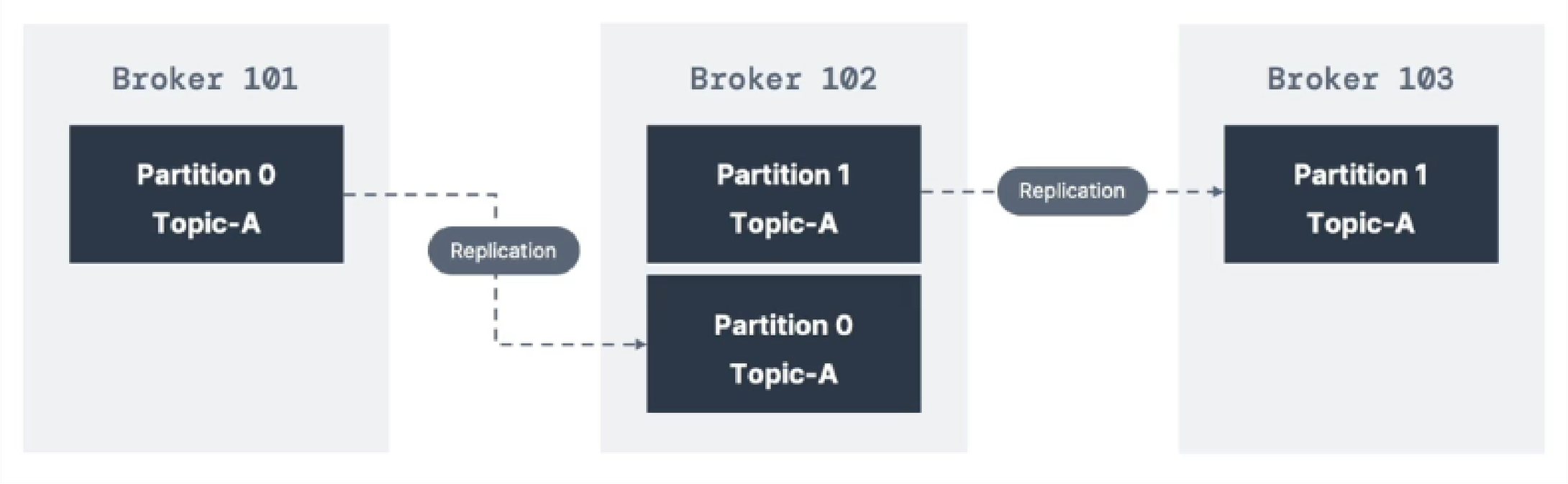
Broker discovery: each broker is bootstrap servers. that means you only need to connect to one broker and clients will know how to connect to entire cluster

Topic replication Factor

Topic replication Factor

- Topics should have rep factor > 1
- If the broker is down, another broker servers
- Ex: Topic A with 2 part & repl factor 2
- Leader for a partition

Topic replication Factor



Ex: Topic A with 2 part & repl factor 2

Leader for a partition

- One broker can be a leader at a given time
- Producer write data to leader broker (by default)
- Each replica is called ISR (in sync replica)
- Kafka consumer will read from leader broker (by default)
- Kafka replica fetching (new Feature) Kafka 2.4 - Allows Consumer to read from closest replica (To improve latency and decrease network costs)

Leader for a partition

- Producer write data to leader broker (by default)
- Each replica is called ISR (in sync replica)
- Kafka consumer will read from leader broker (by default)
- Kafka replica fetching (new Feature) Kafka 2.4 - Allows Consumer to read from closest replica (To improve latency and decrease network costs)

Producer Acks

Producer Acks

- acks=0: Producer wont wait for acks (possible data loss)
- acks=1: Producer will wait for leader acknowledgement (limited data loss)
- acks=all: Leader + replicas acknowledgement (no data loss)

Topic Durablity

Topic Durablity

Generally, repl factor of N => Topic durablity = N-1 Ex: Topic with Repl factor: 3, topic can withstand 2 brokers loss

Zookeeper

Zookeeper

- Zookeeper manages brokers
- Zookeeper helps in performing leader election for partitions
- Zookeeper sends notification in case of changes (new topic, broker dies, broker comes up, etc)
- Kafka 2.x (Zookeeper) Vs Kafka 3.x (Zookeper or KRaft) Vs Kafka 4.x (No Zookeeper)

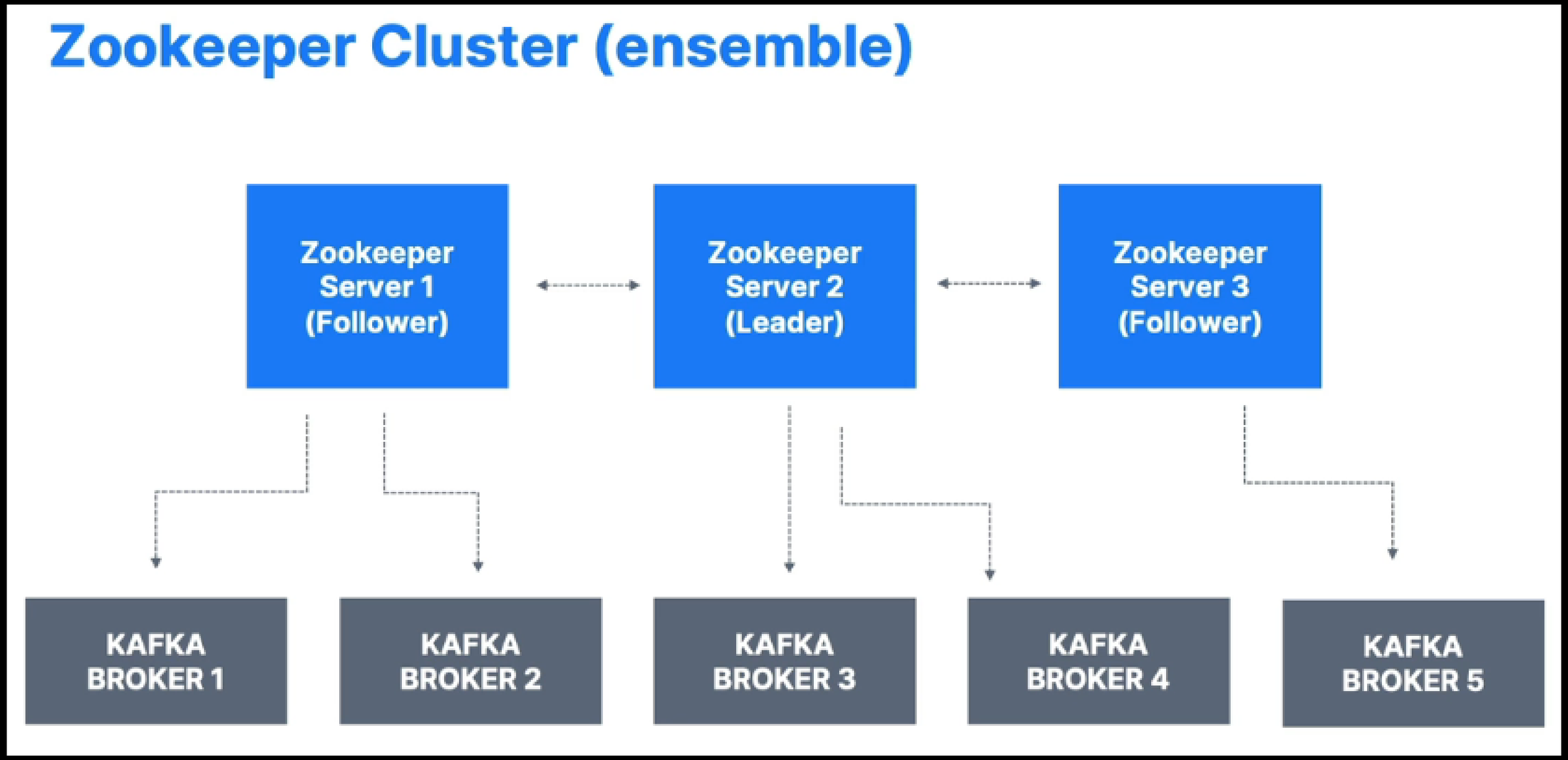
Zookeeper

Kafka 4.x (No Zookeeper)

- Zookeeper by design operates with odd number of servers (1,3,5,7)
- 
- Never use Zookeeper as config in kafka clients and other programs that connect to Kafka

Zookeeper

- Zookeeper by design operates with odd number of servers (1,3,5,7)
- 
- Never use Zookeeper as config in kafka clients and other programs that connect to Kafka



KRaft

KRaft

- 2020, To remove Zookeeper dependency
- Scaling issues with Kafka + Zookeeper > 100000 paritition
- Without Zookeeper, Scales to Millions
- Security Model, Stablity, Single process to start, Faster controller shutdown and recovery time

Thank you