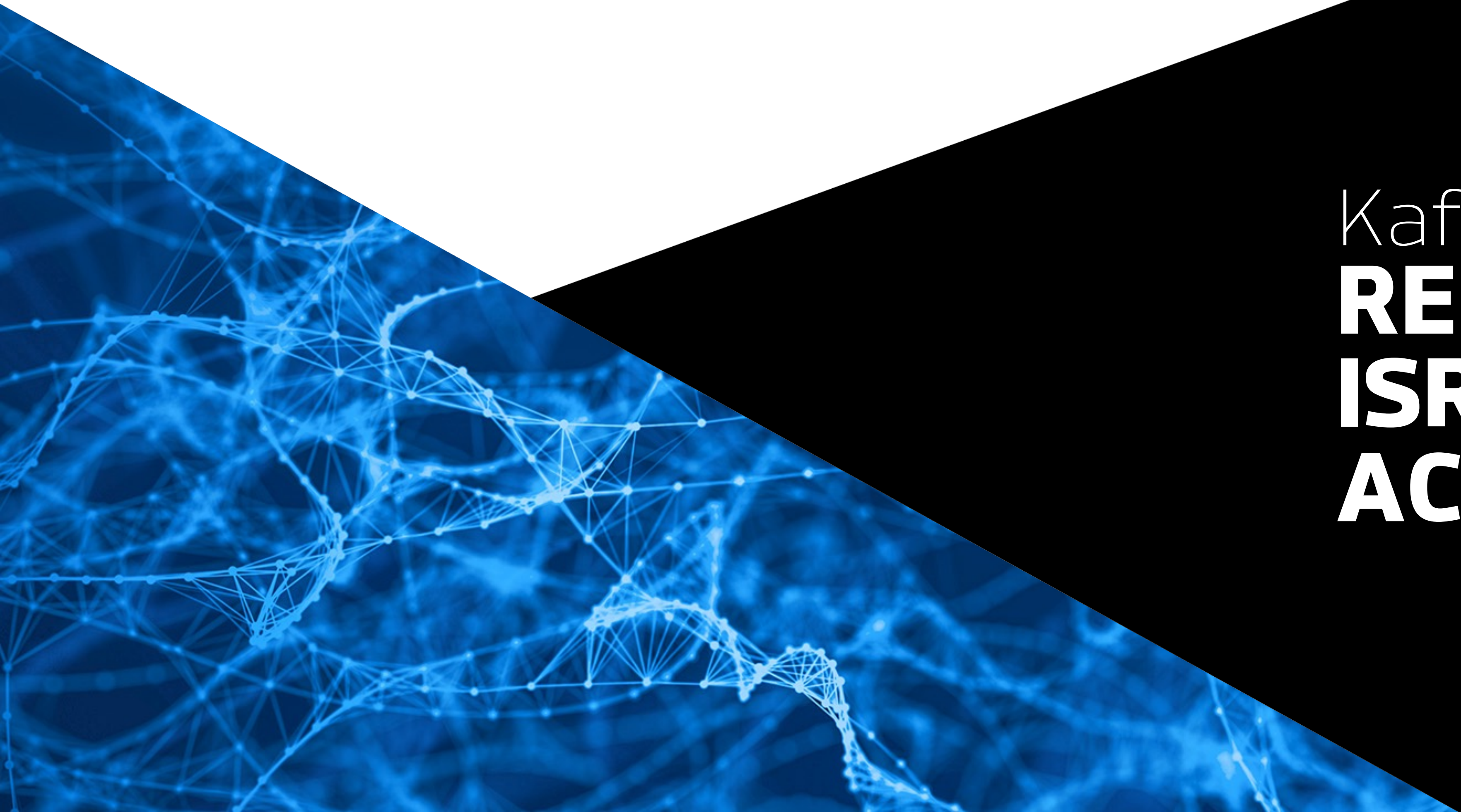
DEEPSHORE



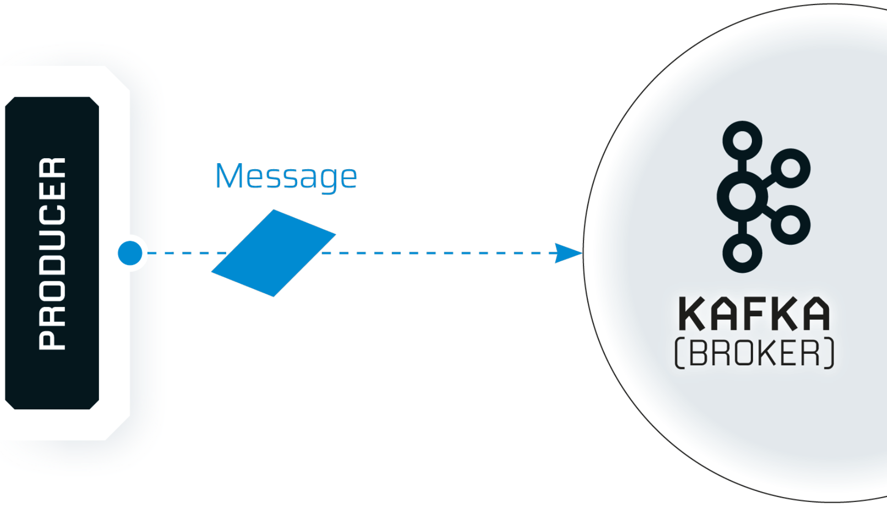
Kafka Basics:
REPLICATION
ISR
ACK()

IMPORTANT THINGS TO KNOW:
HOW DOES A PRODUCER WRITES DATA?

Let´s understand, how …

- **REPLICATION FACTOR**
- **ISR (IN SYNC REPLICA)**
- **ACK()**

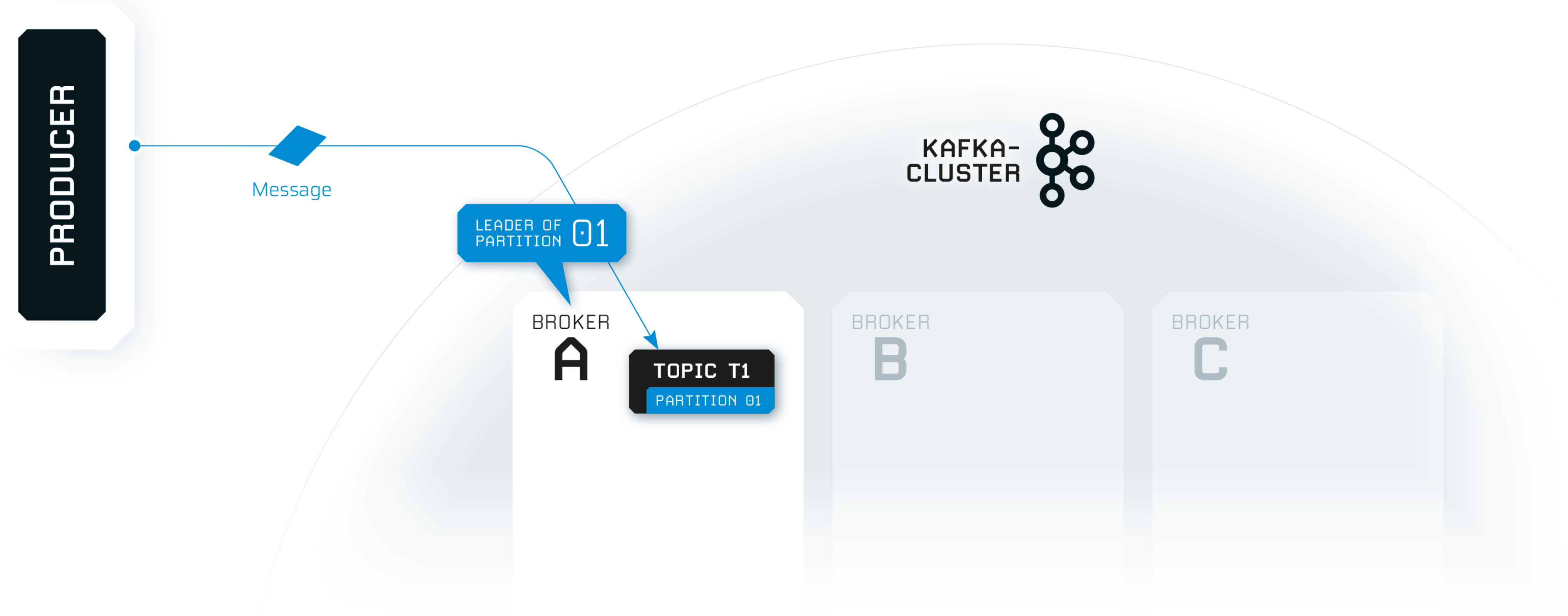
… affect each other.



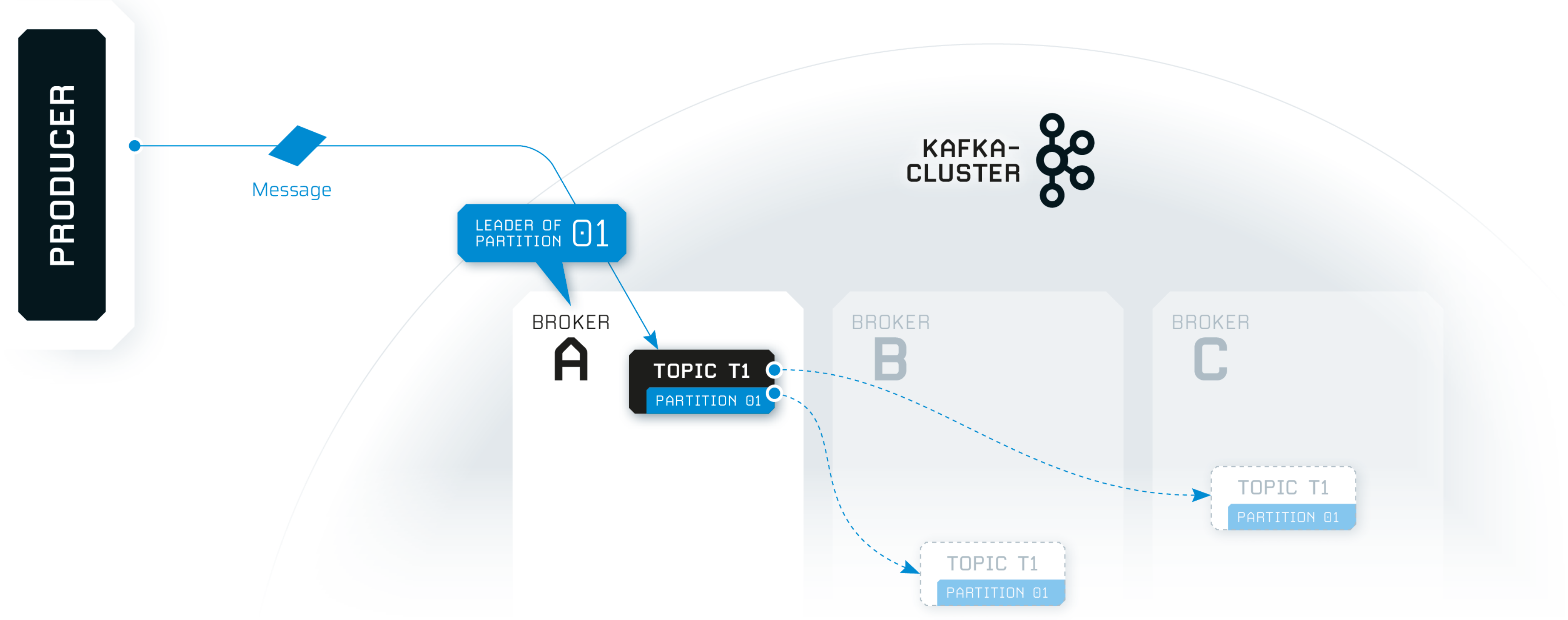
01

REPLICATION FACTOR IN KAFKA

A PRODUCER WILL ALWAYS ... TALK TO „THE LEADER"!

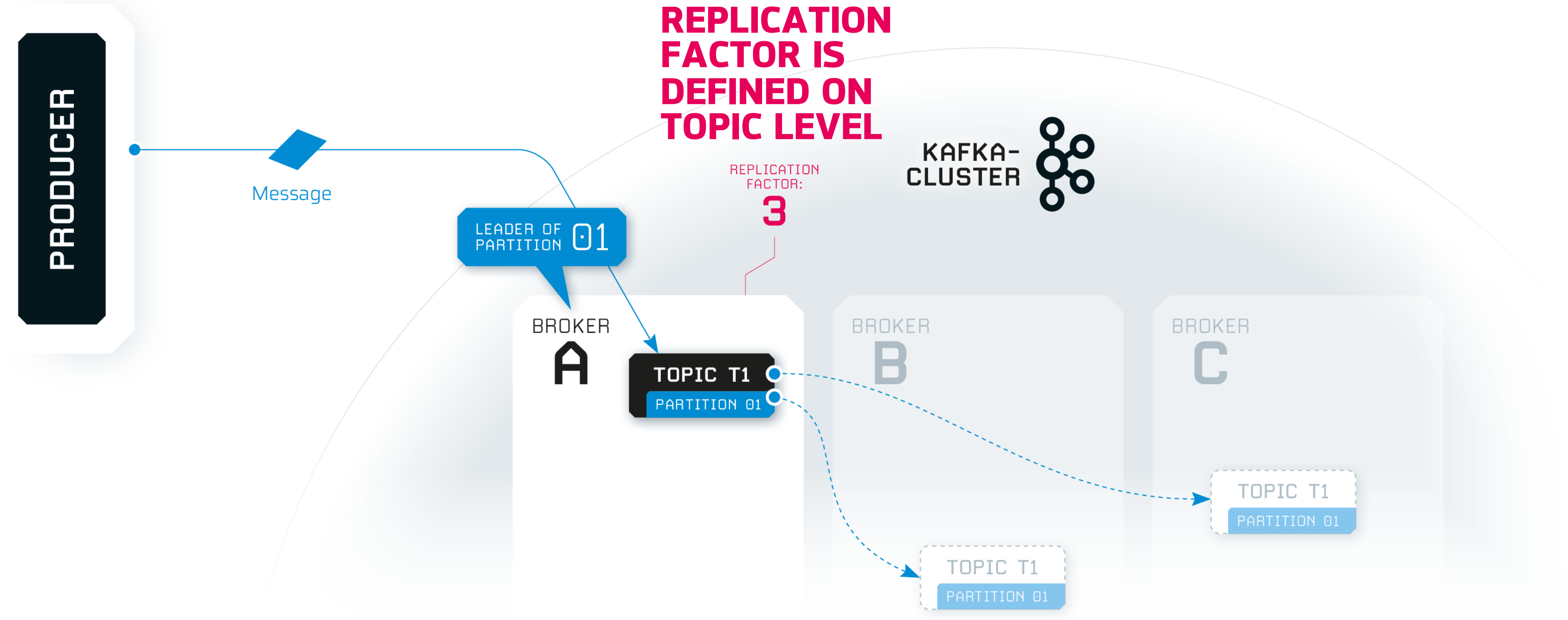


THE LEADER WILL PROVIDE MESSAGE DATA TO IT´S FOLLOWERS

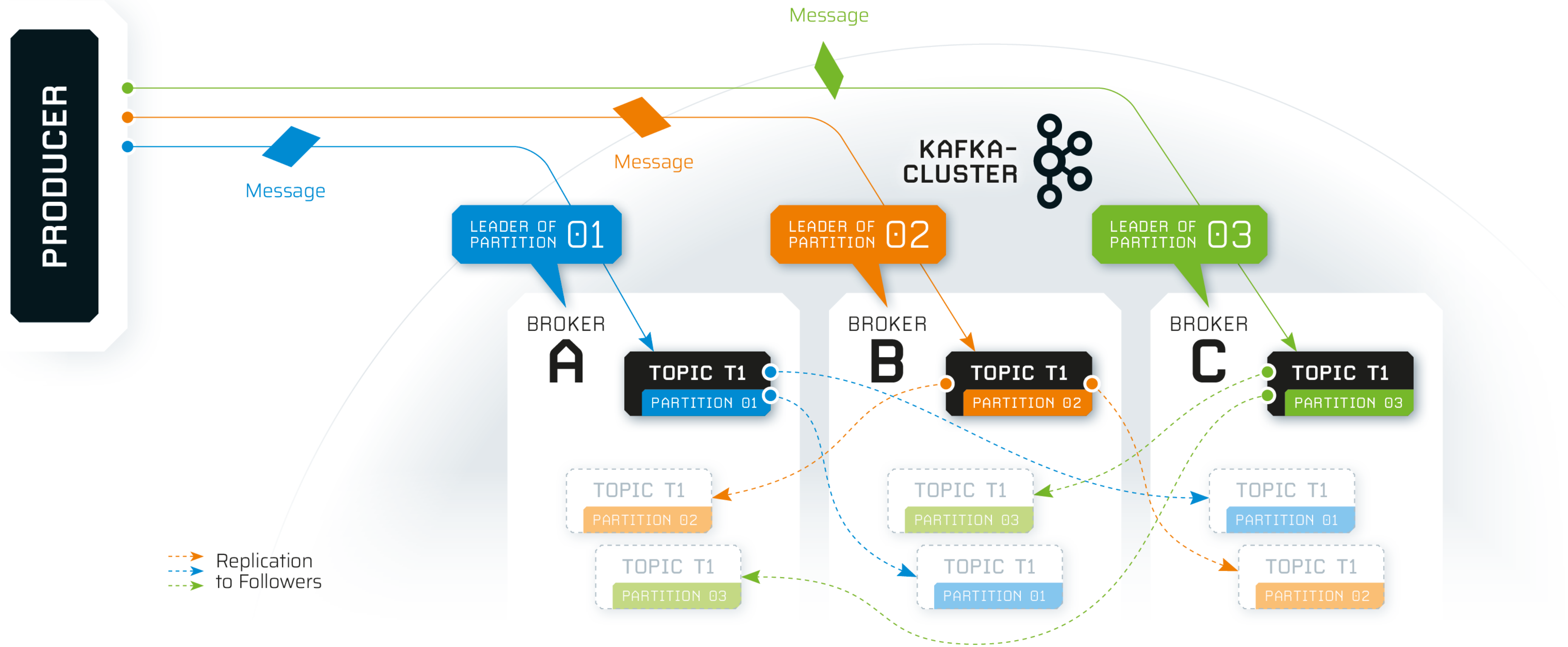


REPLICATION FACTOR = NUMBER OF MESSAGE COPIES

REPLICATION FACTOR IS DEFINED ON TOPIC LEVEL



THERE IS A LEADER FOR EACH PARTITION



02

ISR: IN-SYNC REPLICATIONS IN KAFKA

IMPORTANT THINGS TO KNOW: HOW DOES A PRODUCER WRITES DATA?

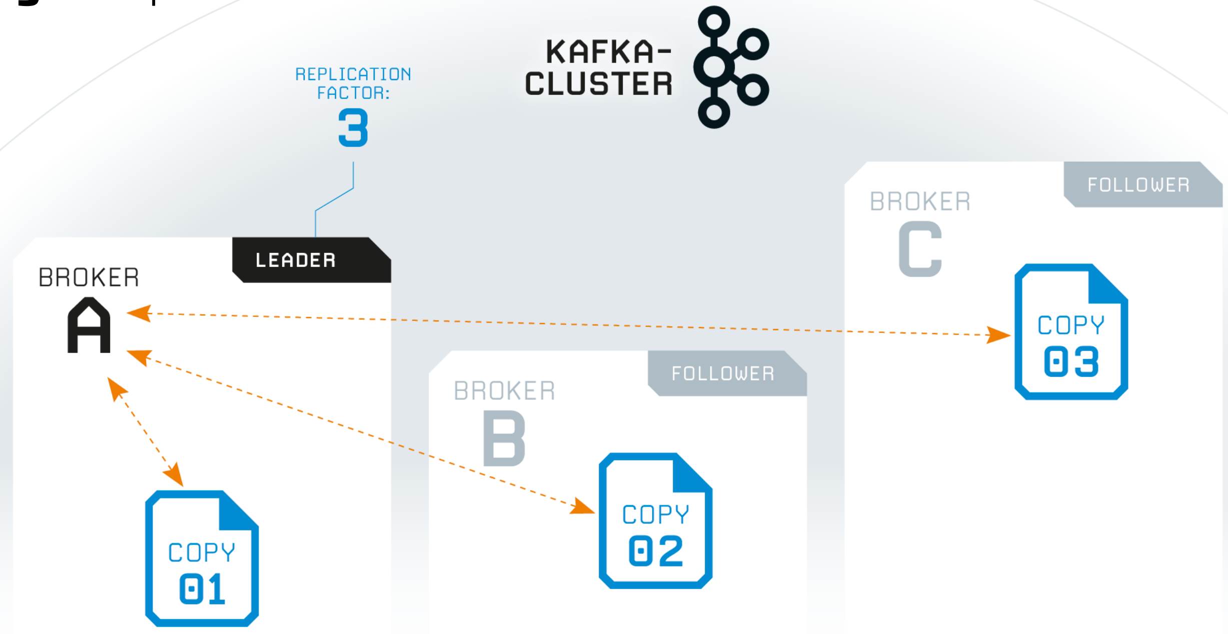
Let´s understand …

- **REPLICATION FACTOR** = number of message copies in the cluster
- **ISR (IN SYNC REPLICA)**

EXAMPLE: A PARTITION LEADER WITH RF=3 GENERATES 2 COPIES OF A MESSAGE AND KEEPS THEM SYNCHRONISED

- Total number of replicas: 3
- Number of **in-sync** replicas: 3

ISR=3

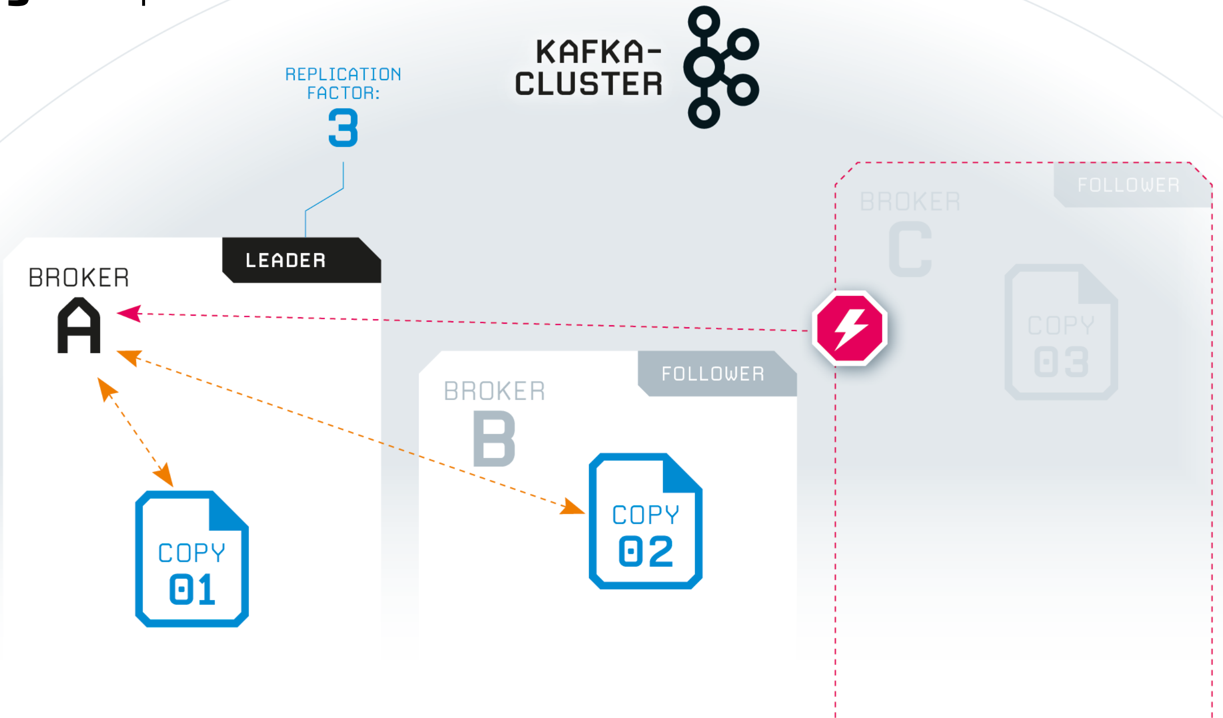


BROKEN CONNECTIONS TO FOLLOWERS DECREASE THE ISR

- Total number of replicas: 3
- Number of **in-sync** replicas: 2

ISR=2

MINIMUM ISR REQUIREMENTS ARE DEFINED ON TOPIC LEVEL!



ISR: TAKE AWAY

- Kafka dynamically maintains a set of in-sync replicas (ISR) that are caught-up to the leader
- A node (Broker) is in sync when:
  - It is able to maintain its session with ZooKeeper (via ZooKeeper's heartbeat mechanism)
  - If the replicated writes (happening on the leader) will not fall "too far" behind In-Sync Replicas are equal or a subset of all replicas in the cluster
- If a follower dies, gets stuck, or falls behind, the leader will remove it from the list of In-Sync replicas (replica.lag.time.max.ms)

03

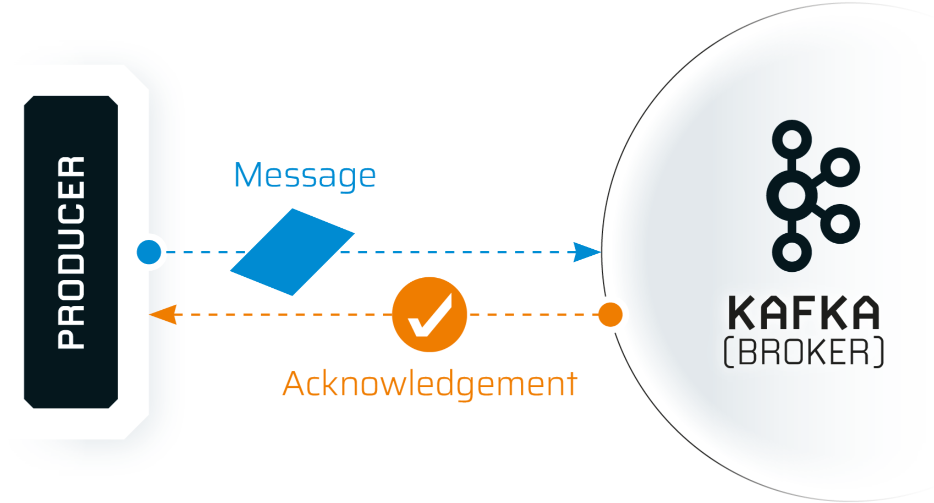
ACK(): ACKNOWLEDGEMENT IN KAFKA

IMPORTANT THINGS TO KNOW: HOW DOES A PRODUCER WRITES DATA?

Let´s understand …

- **REPLICATION FACTOR** = number of message copies in the cluster
- **ISR (IN SYNC REPLICA)** = available replicas in a cluster / …
- **ACK()**

WHEN DID A PRODUCER SUCCESSFULLY SEND A MESSAGE?



LEVELS OF ACKNOWLEDGMENT

ACK(0)

The producer never waits for an ack from the broker

> fast, but unsafe

ACK(1)

The producer gets an ack after the leader has received the record

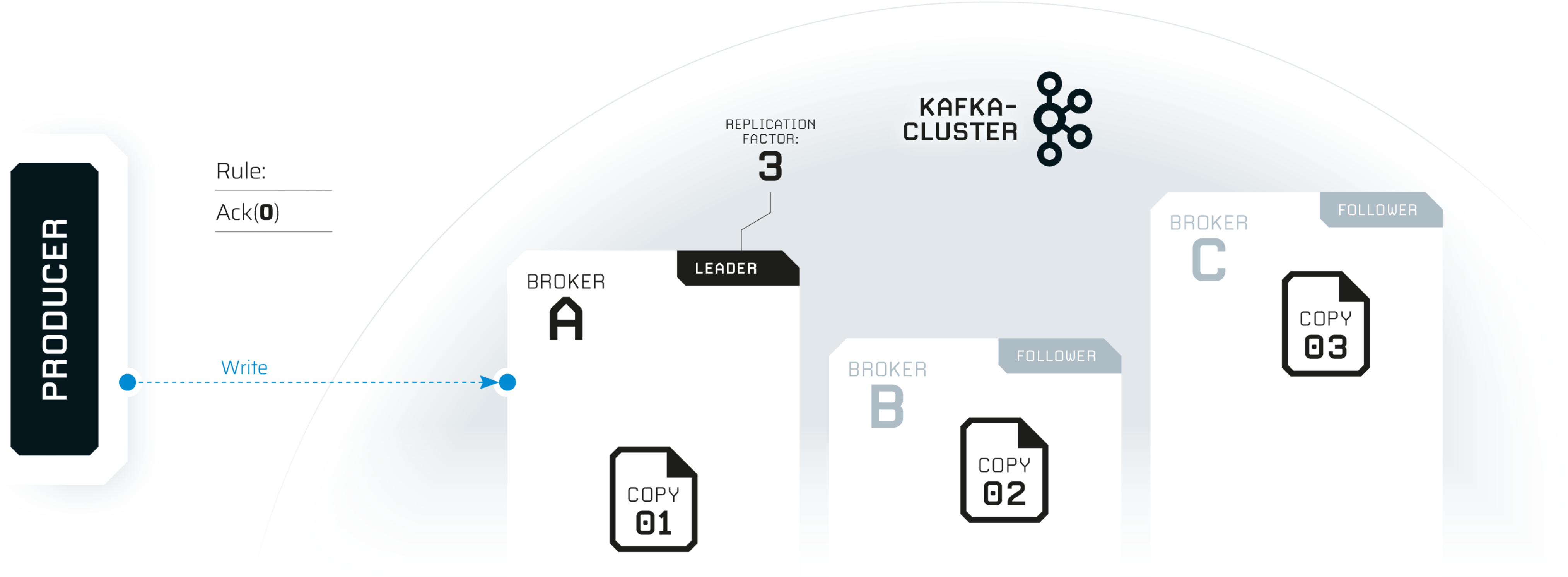
ACK(ALL)

The producer gets an ack when all in-sync replicas have received the record

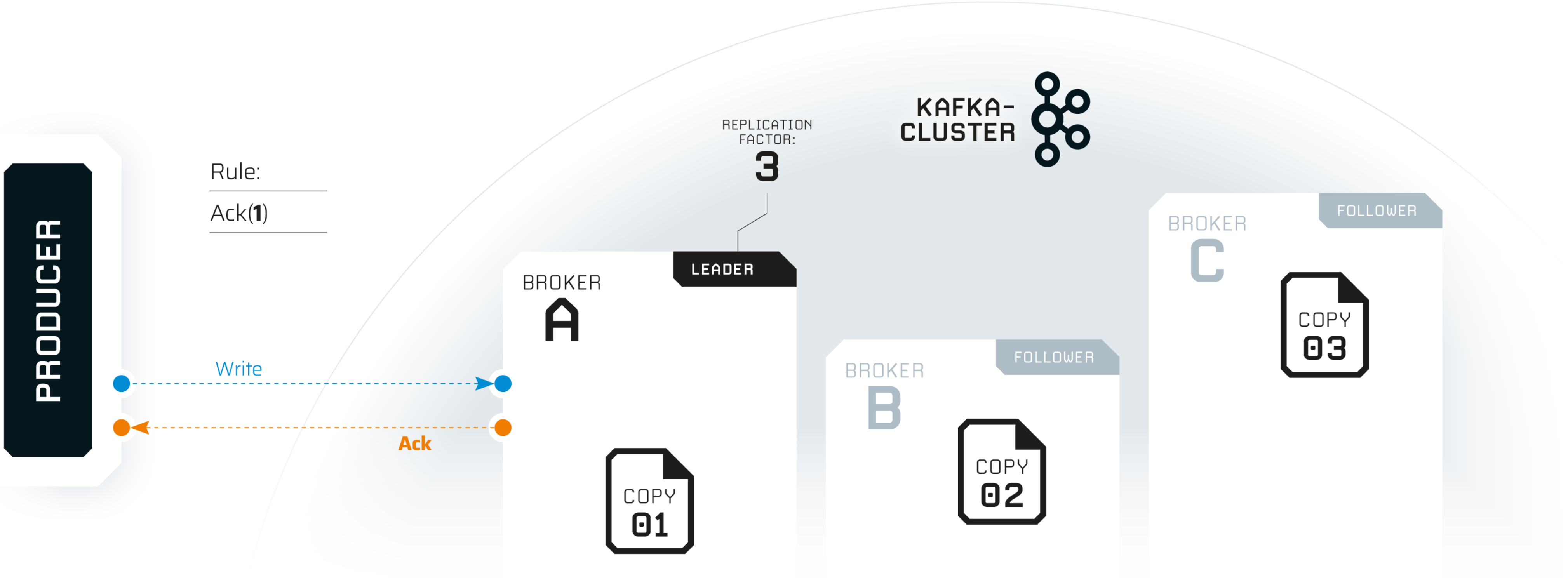
> safe, but might be slow

ACK() LEVEL IS A PRODUCER CONFIGURATION TO AFFECT THROUGHPUT AND DATA CONSISTENCY!

EXAMPLE ACK(0)

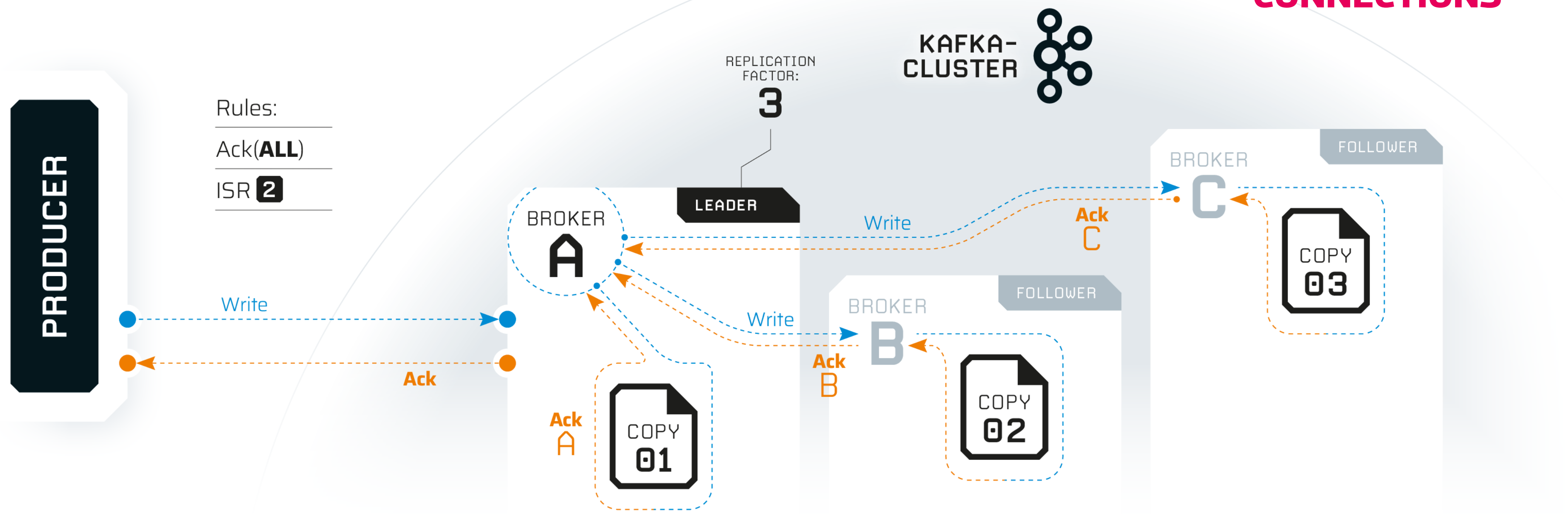


EXAMPLE ACK(1)



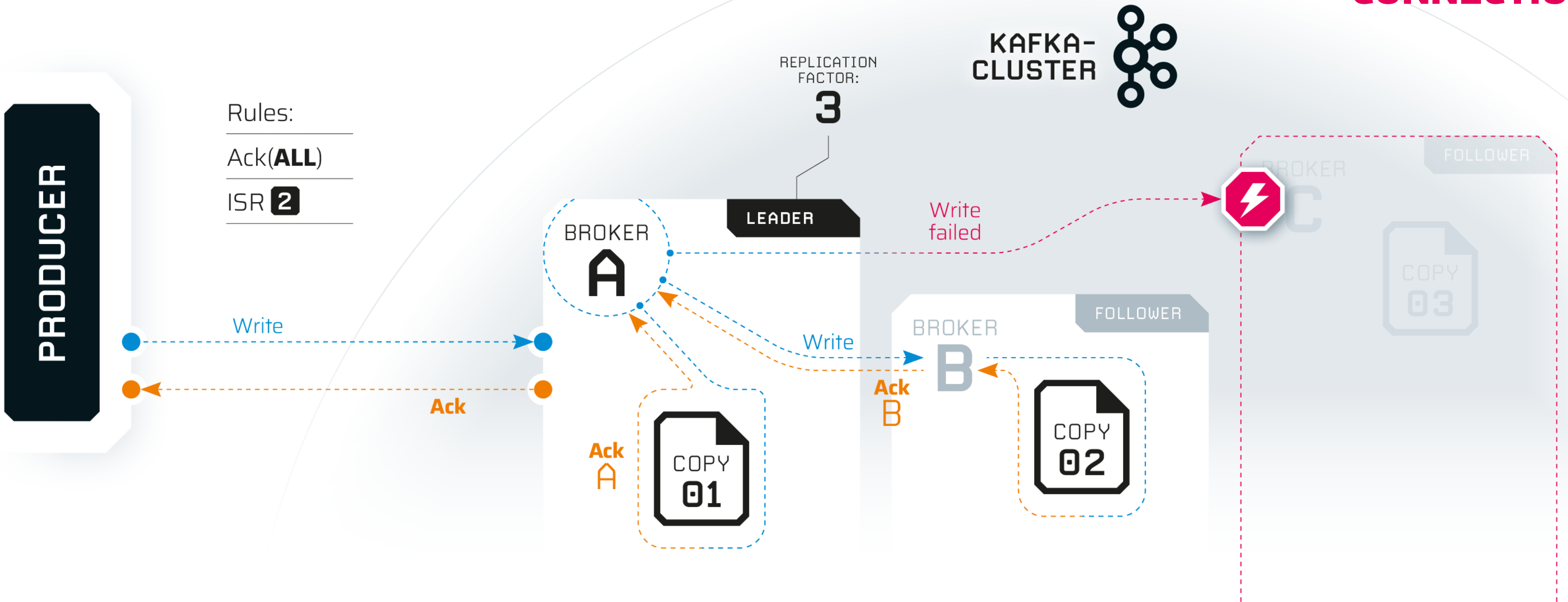
EXAMPLE ACK(ALL)

SUCCESSFUL WRITE WITH INTACT SYNC CONNECTIONS



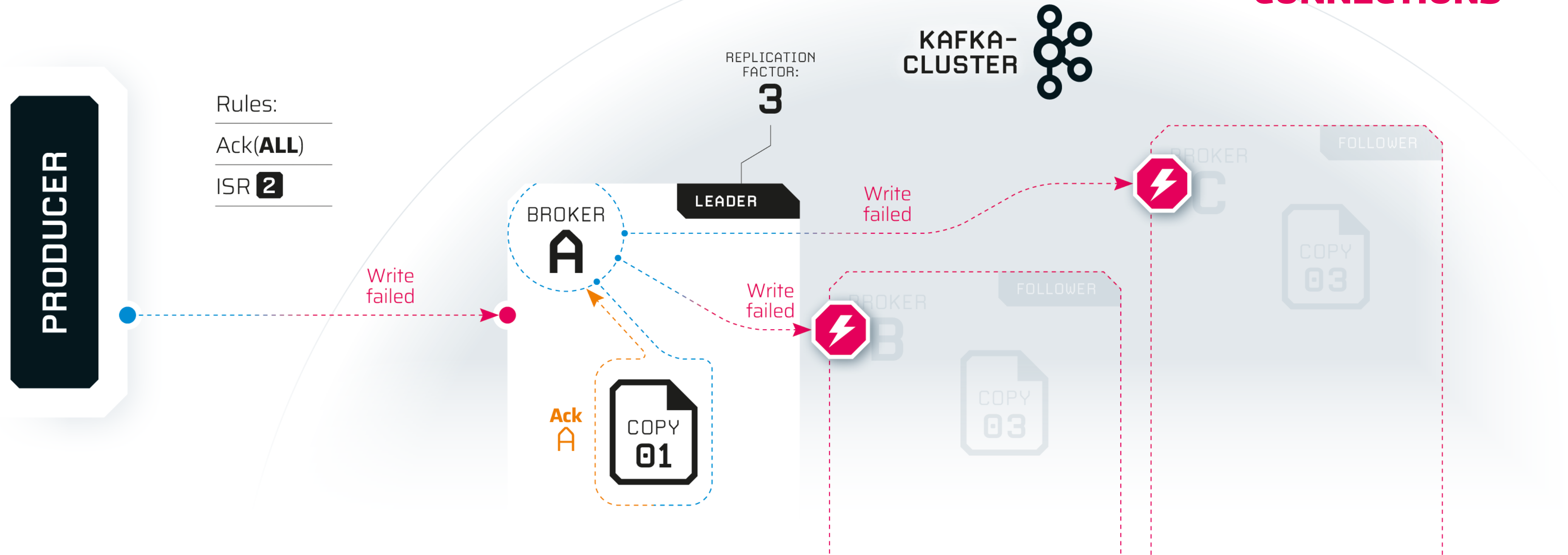
EXAMPLE ACK(ALL)

SUCCESSFUL WRITE DESPITE BROKEN SYNC CONNECTION



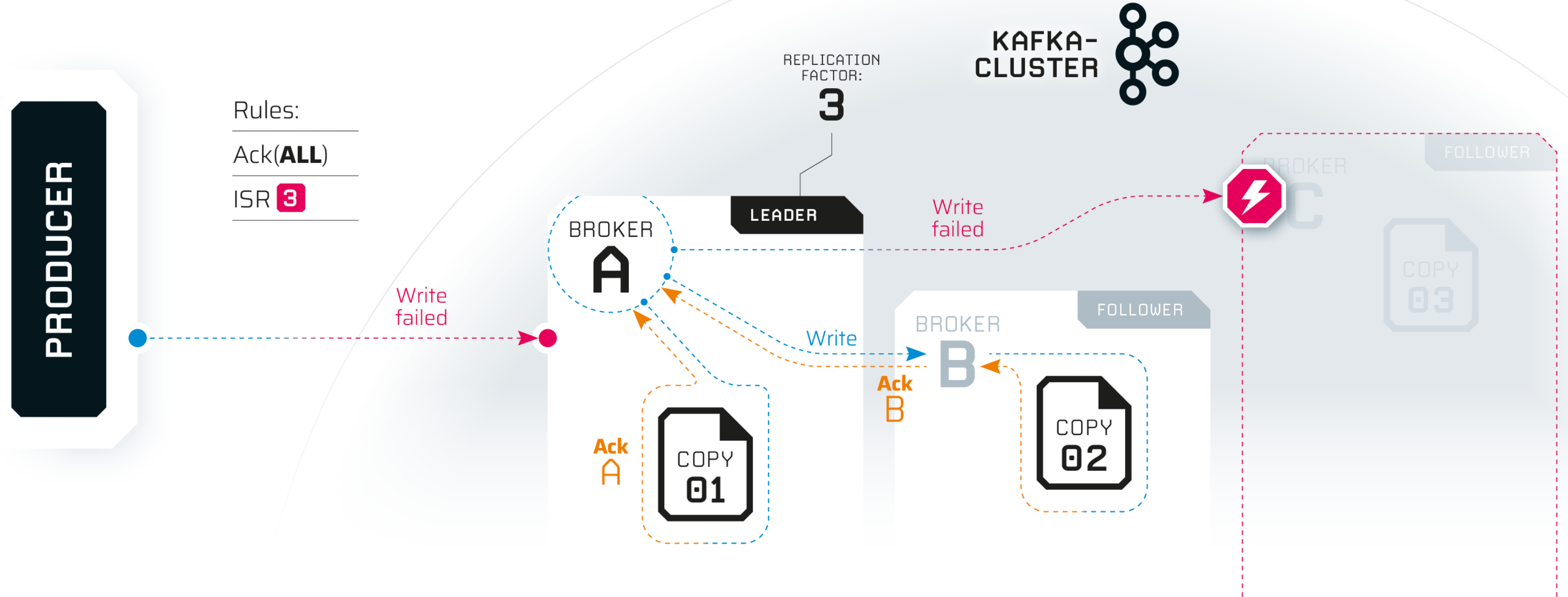
EXAMPLE ACK(ALL)

FAILED WRITE DUE TO TOO MANY BROKEN SYNC CONNECTIONS



EXAMPLE ACK(ALL)

FAILED WRITE DUE TO ZERO ISR TOLERANCE



Who said:

KAFKA WRITE PASS IS EASY?

CONTACT

Deepshore GmbH · Baumwall 3, D-20459 Hamburg
Telefon +49 40 46664-296 · Fax +49 40 46664-299
E-Mail info@deepshore.de · **www.deepshore.de**