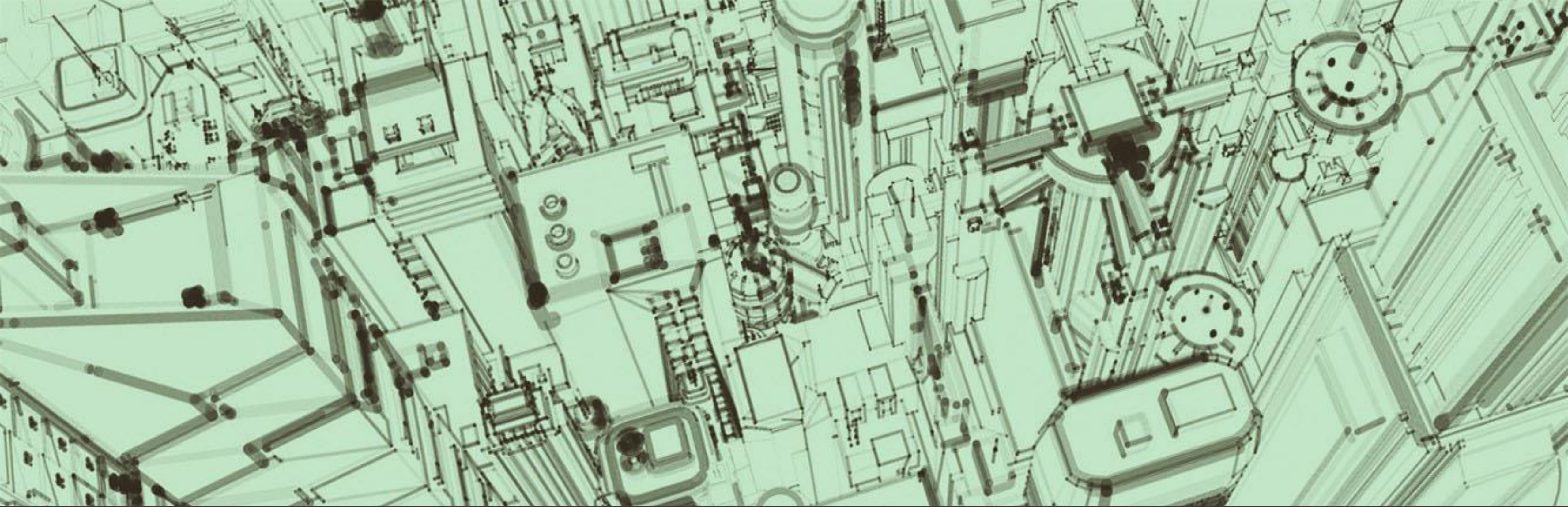
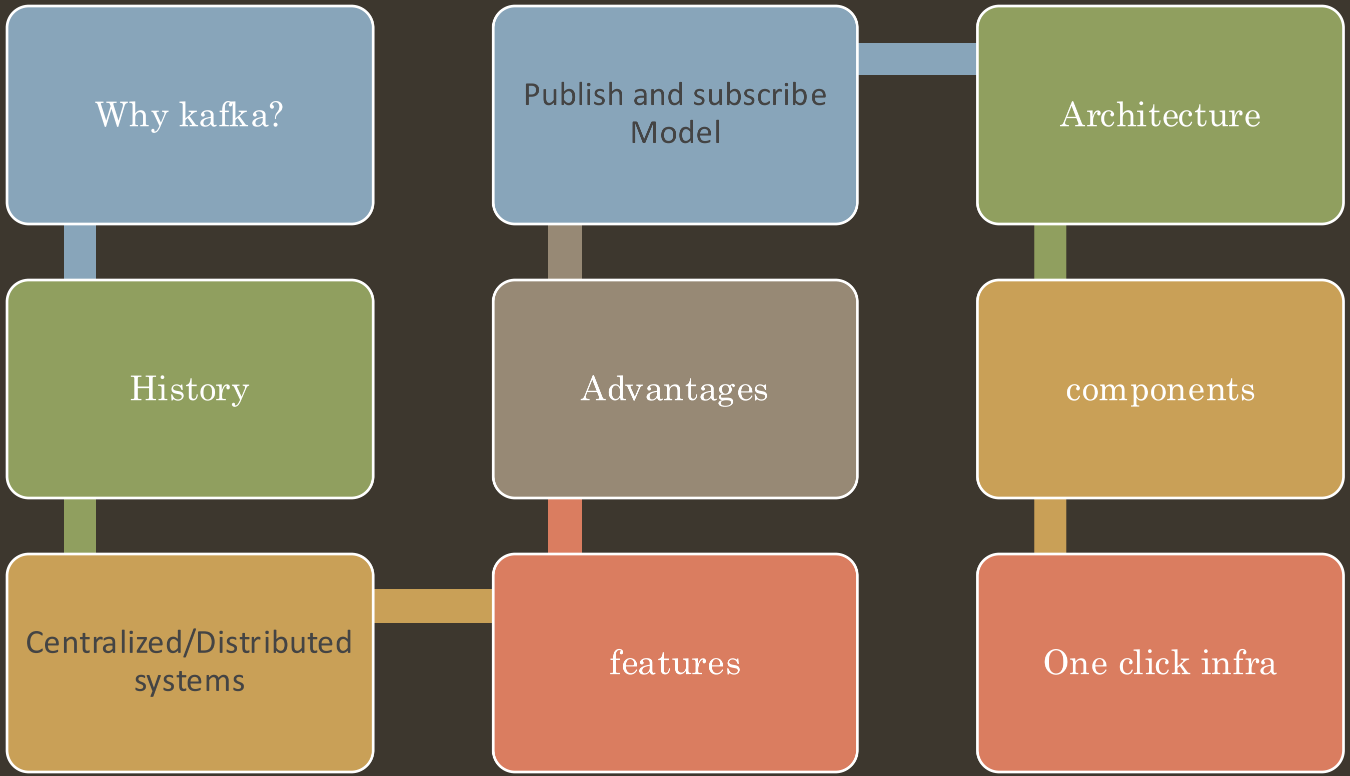


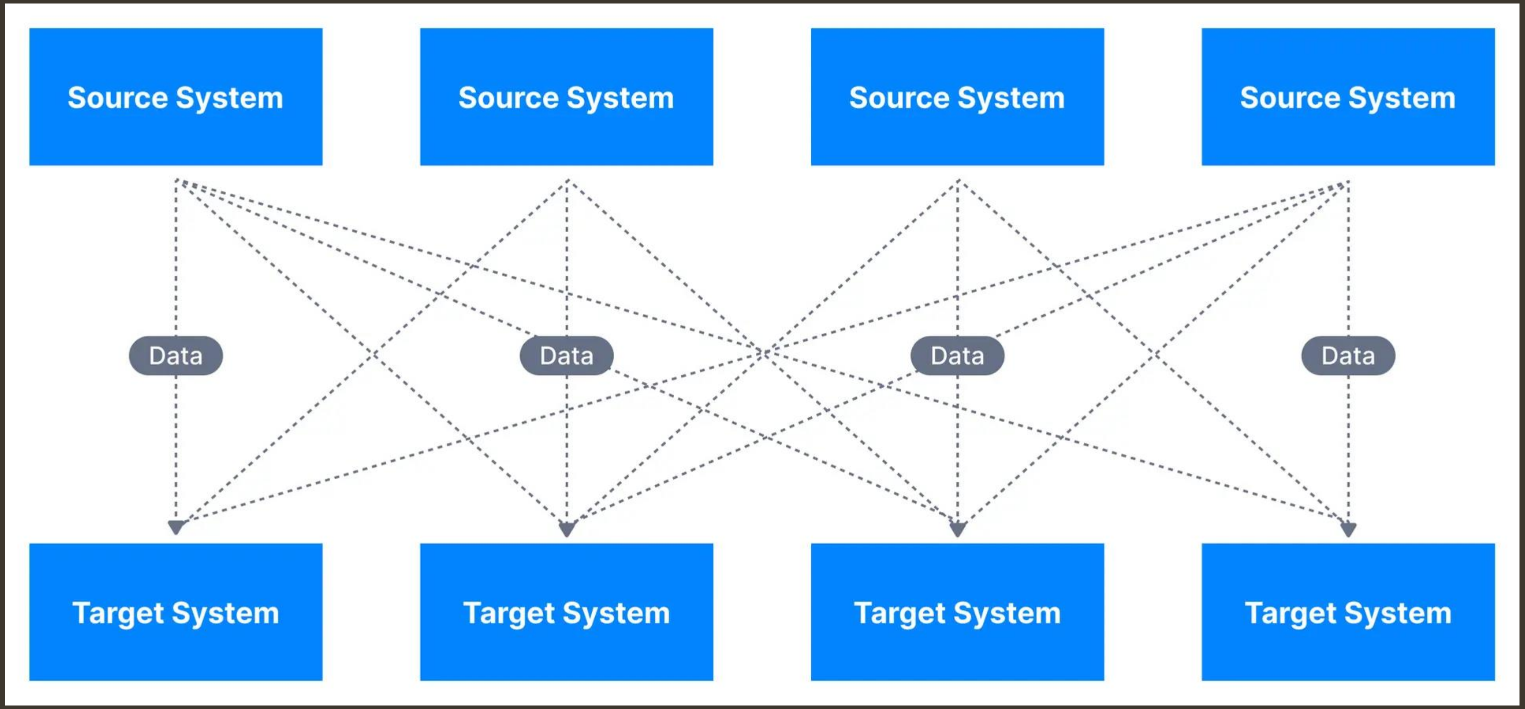
Apache Kafka

AGENDA

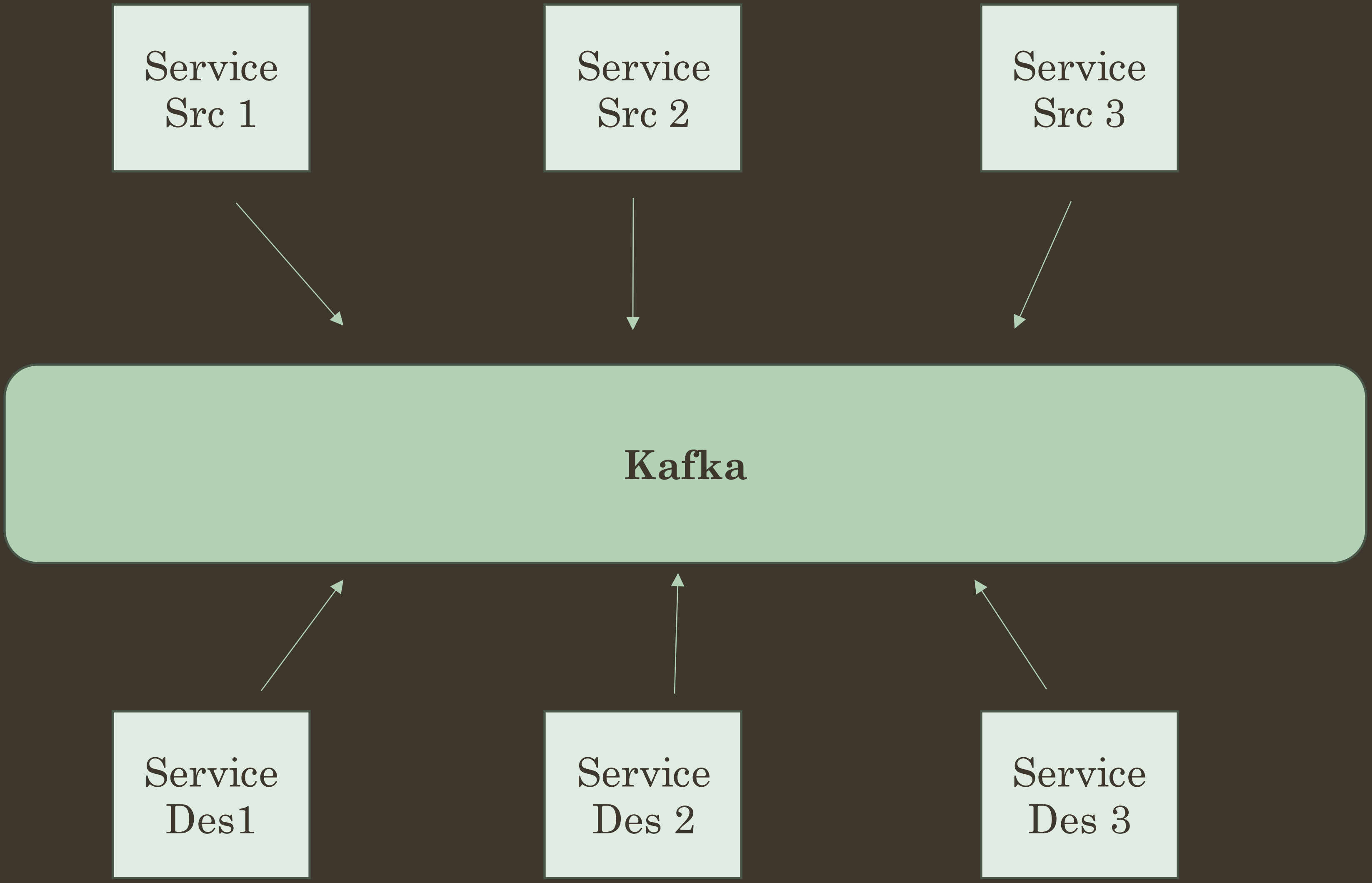


- Why kafka?
- History
- Centralized/Distributed systems
- features
- Advantages
- Publish and subscribe Model
- Architecture
- components
- One click infra

WHY Kafka?


Source System
Source System
Source System
Source System
Data
Data
Data
Data
Target System
Target System
Target System
Target System

Kafka

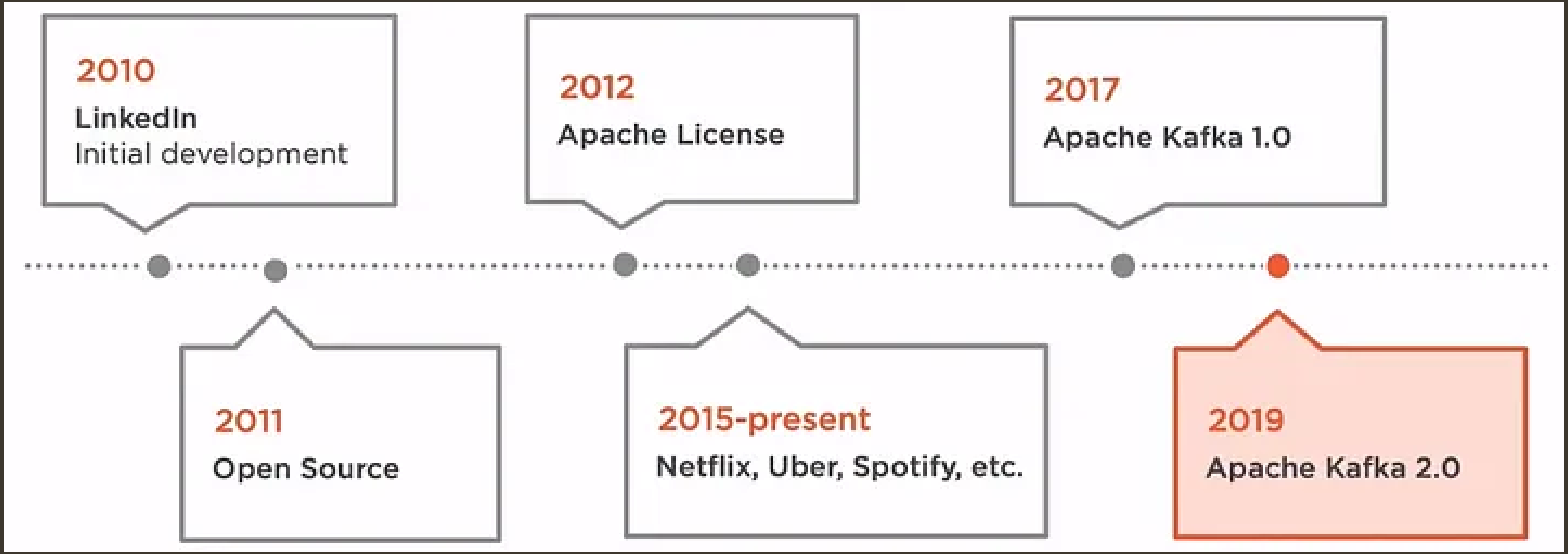

Service Src 1
Service Src 2
Service Src 3
Kafka
Service Des1
Service Des 2
Service Des 3

Kafka

It is a distributed,message streaming platform that uses the publish-subscribe model to stream records

Develop by LinkedIn and later donated to Apache Foundation.

History



2010
LinkedIn
Initial development

2011
Open Source

2012
Apache License

2015-present
Netflix, Uber, Spotify, etc.

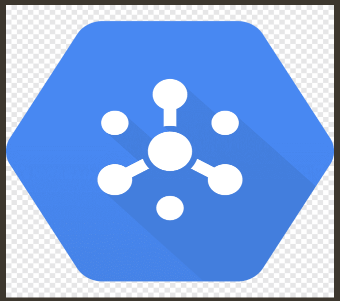
2017
Apache Kafka 1.0

2019
Apache Kafka 2.0

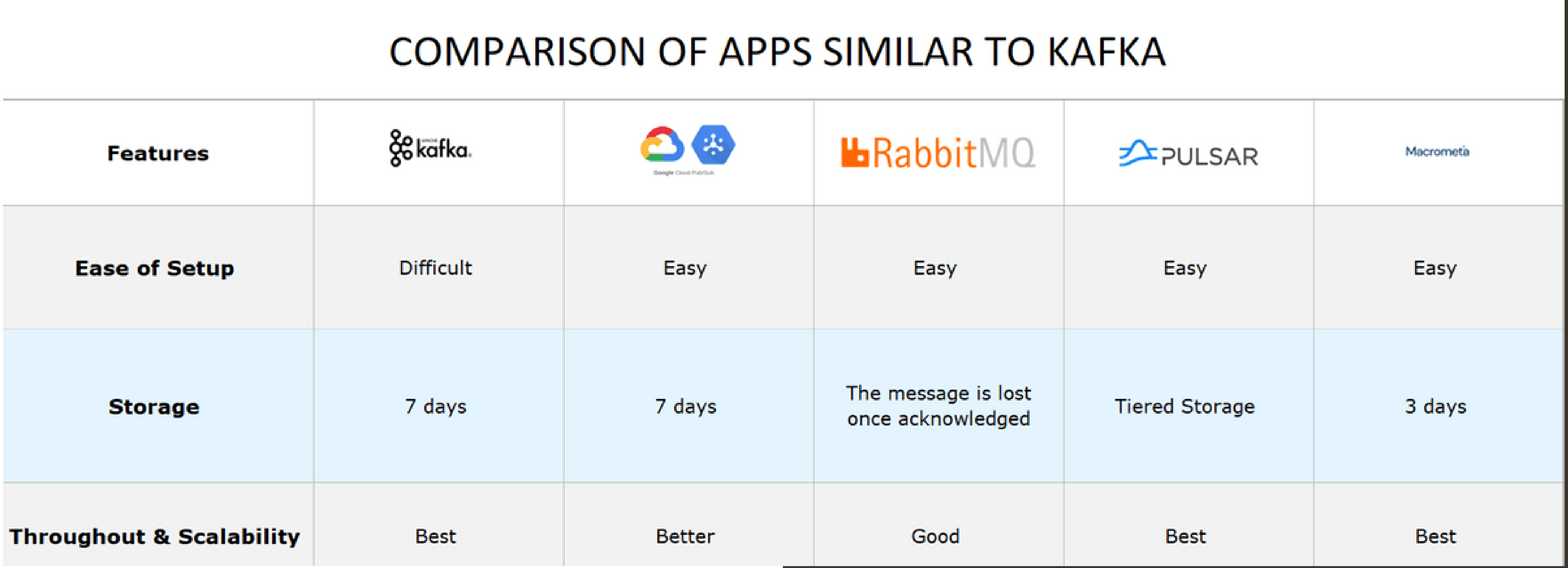
Alternative Tool

Rabbit MQ

Pulsar



Google Cloud
Pub Sub

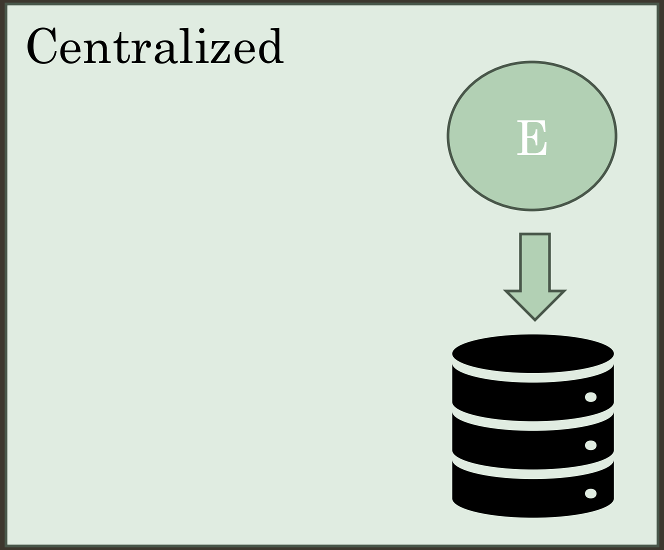


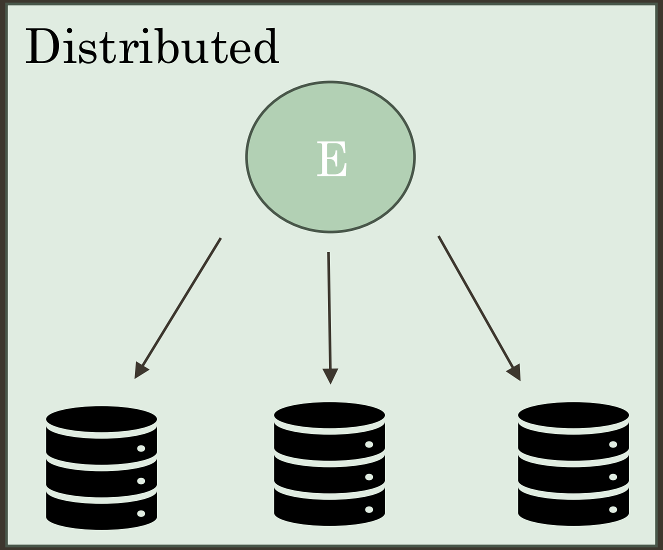
COMPARISON OF APPS SIMILAR TO KAFKA

Features	APACHE kafka	Google Cloud Pub/Sub	RabbitMQ	PULSAR	Macrometa
Ease of Setup	Difficult	Easy	Easy	Easy	Easy
Storage	7 days	7 days	The message is lost once acknowledged	Tiered Storage	3 days
Throughout & Scalability	Best	Better	Good	Best	Best

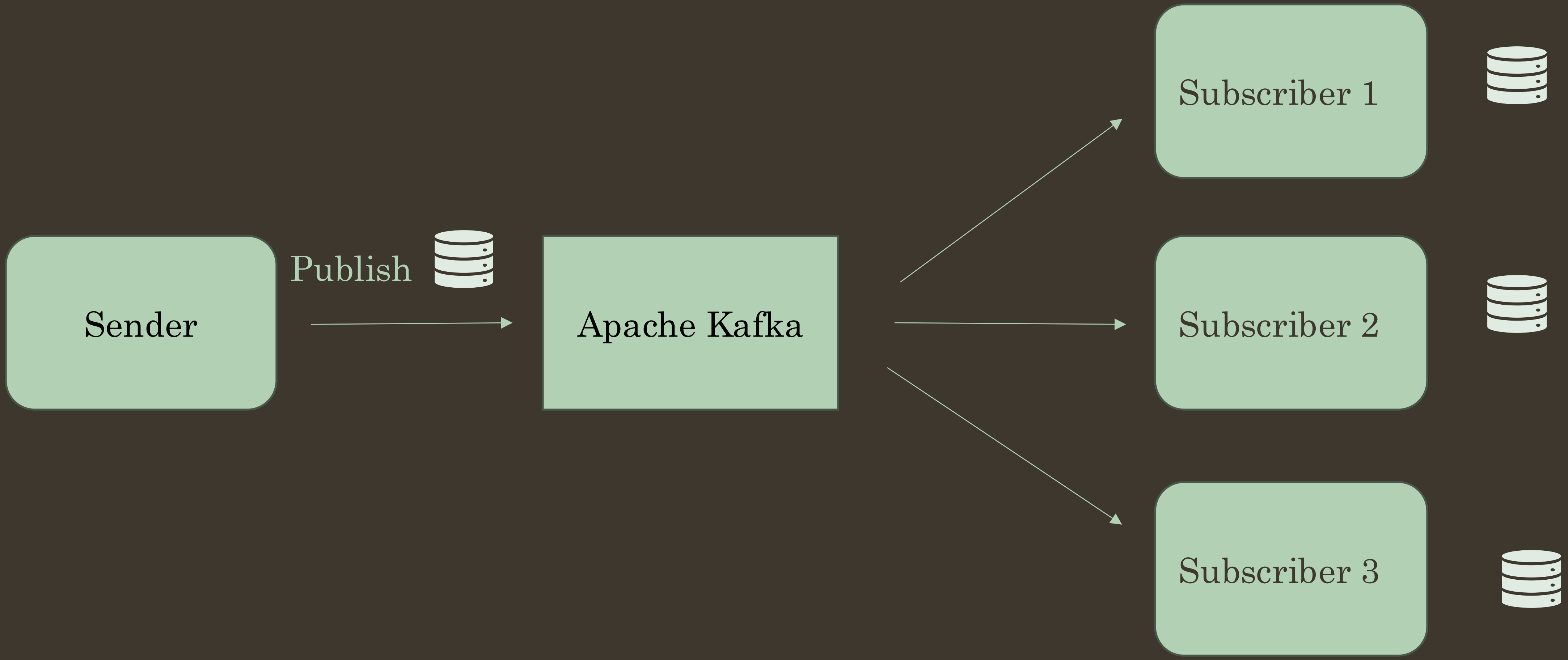
Alternatives

Centralized/Distributed systems

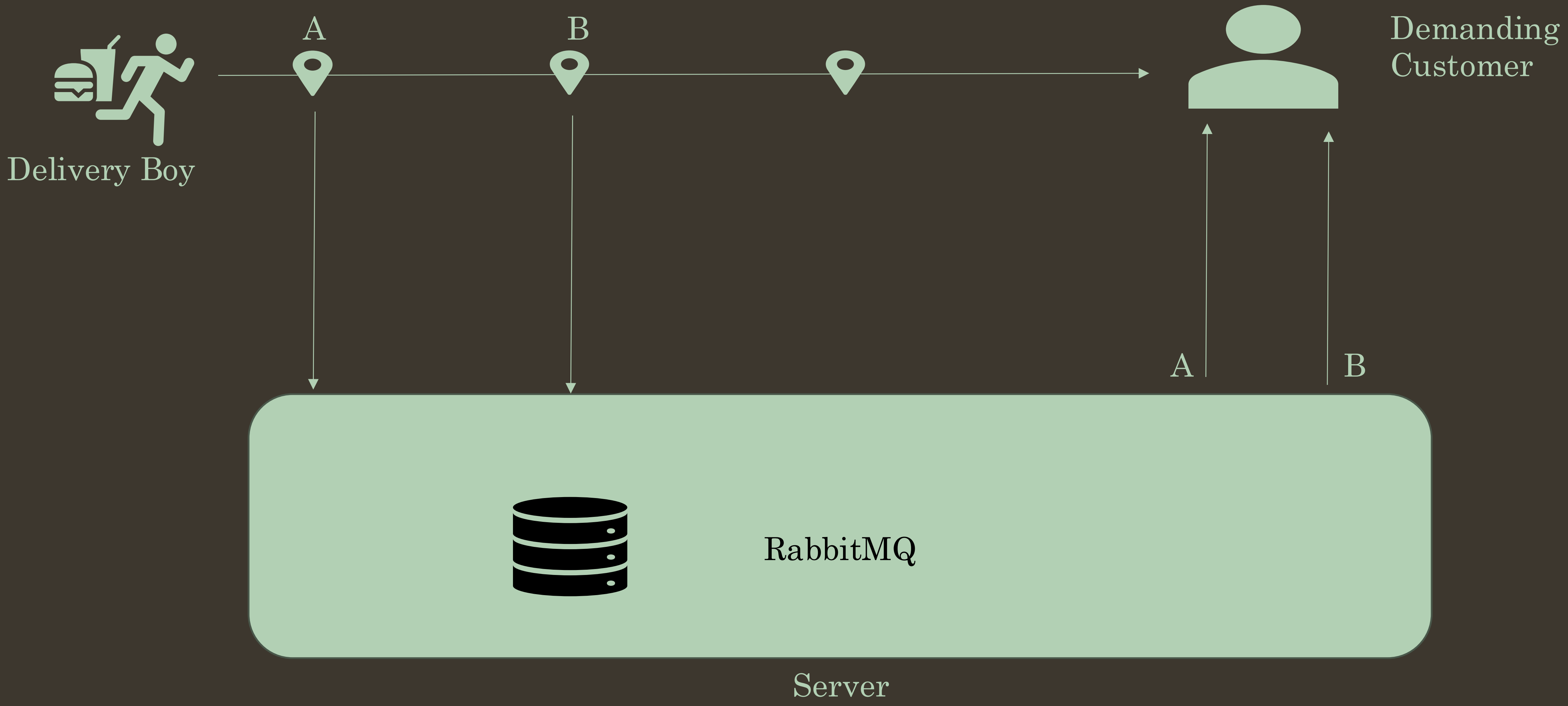




Publish and subscribe Model



EX: How Swiggy Worked Before Kafka



Before Kafka: Problems

Latency and performance

Scalability Challenges

Message Loss

Error Handling

Kafka features

High Throughput

Fault Tolerance (Replication)

No Data Loss

Durable

Kafka with Examples

OLA Driver location update.

Zomato live food tracking

Notification system to users

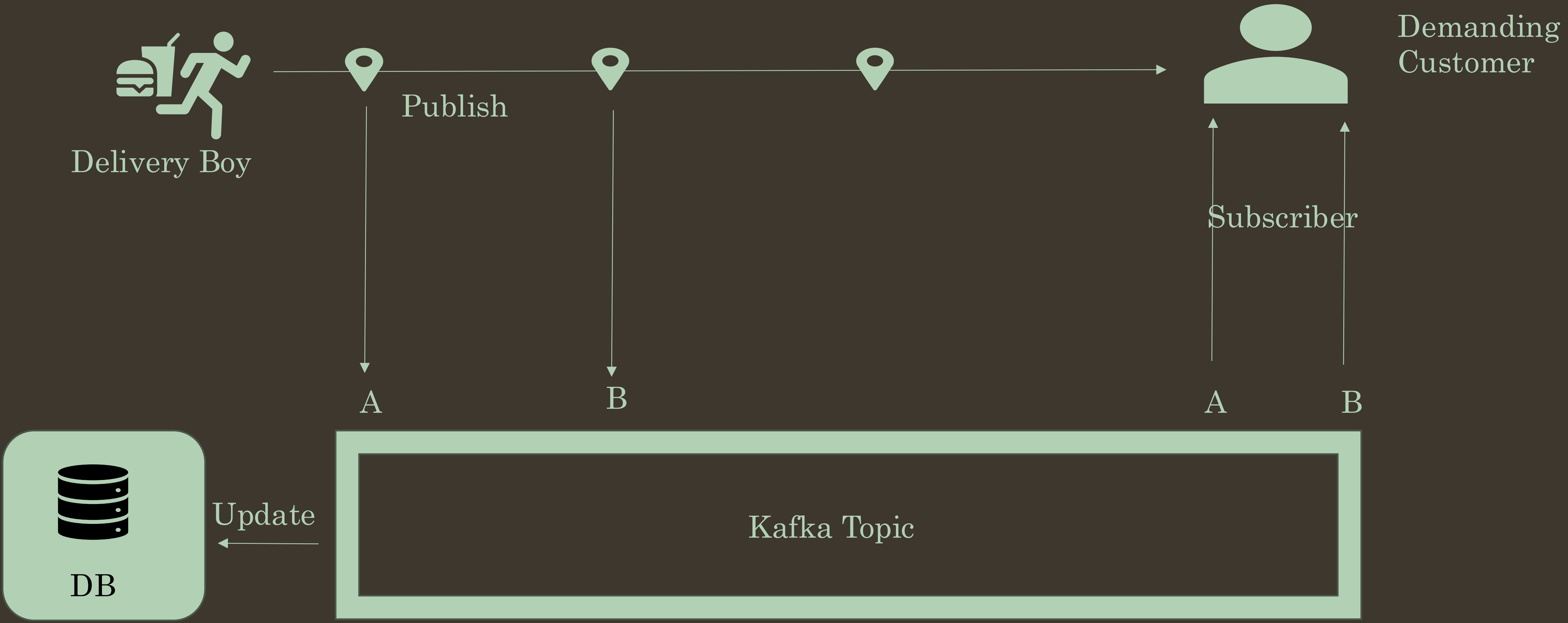
Use Case

Event streaming

Message brokering

Data integration

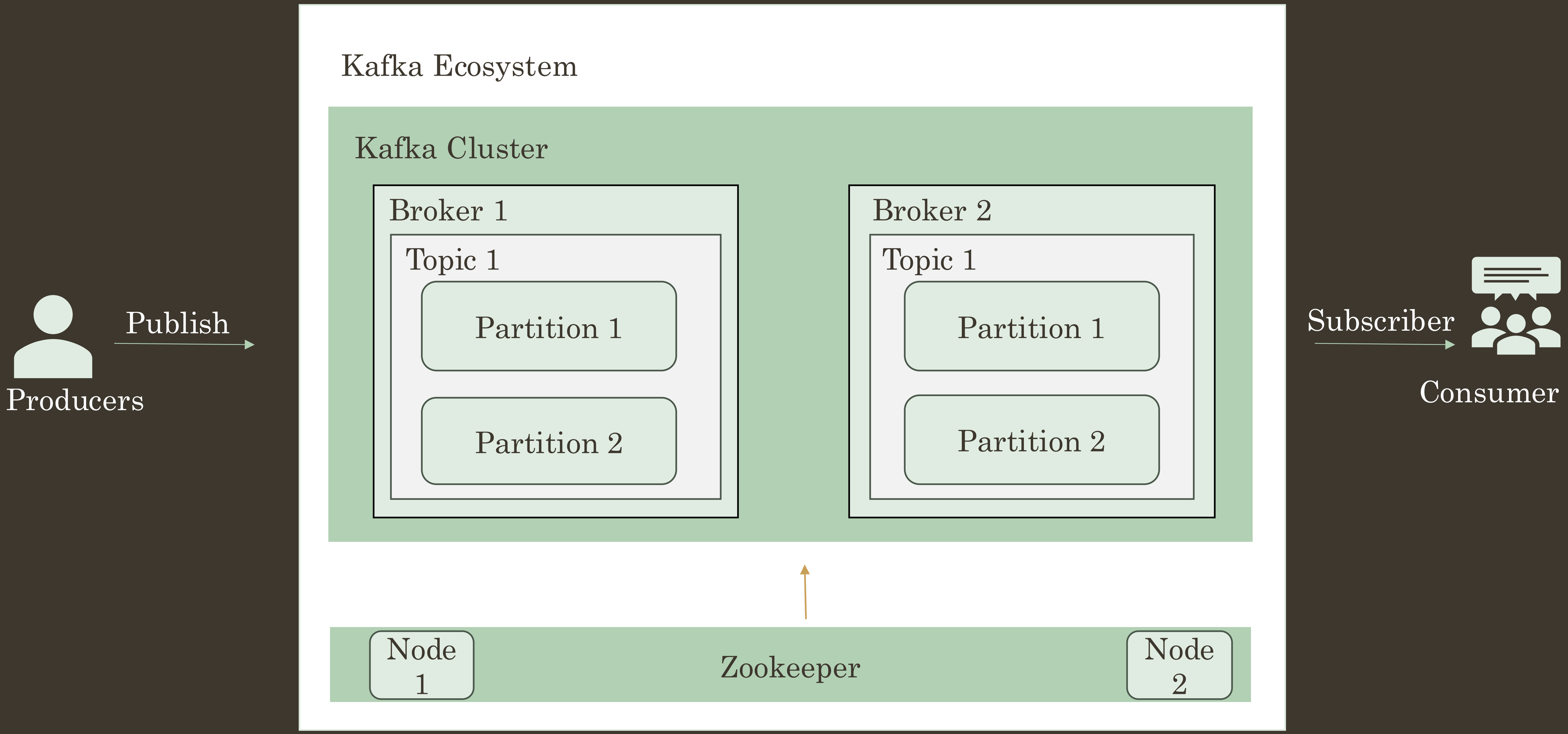
EX: How Swiggy Worked after Kafka



Advantages

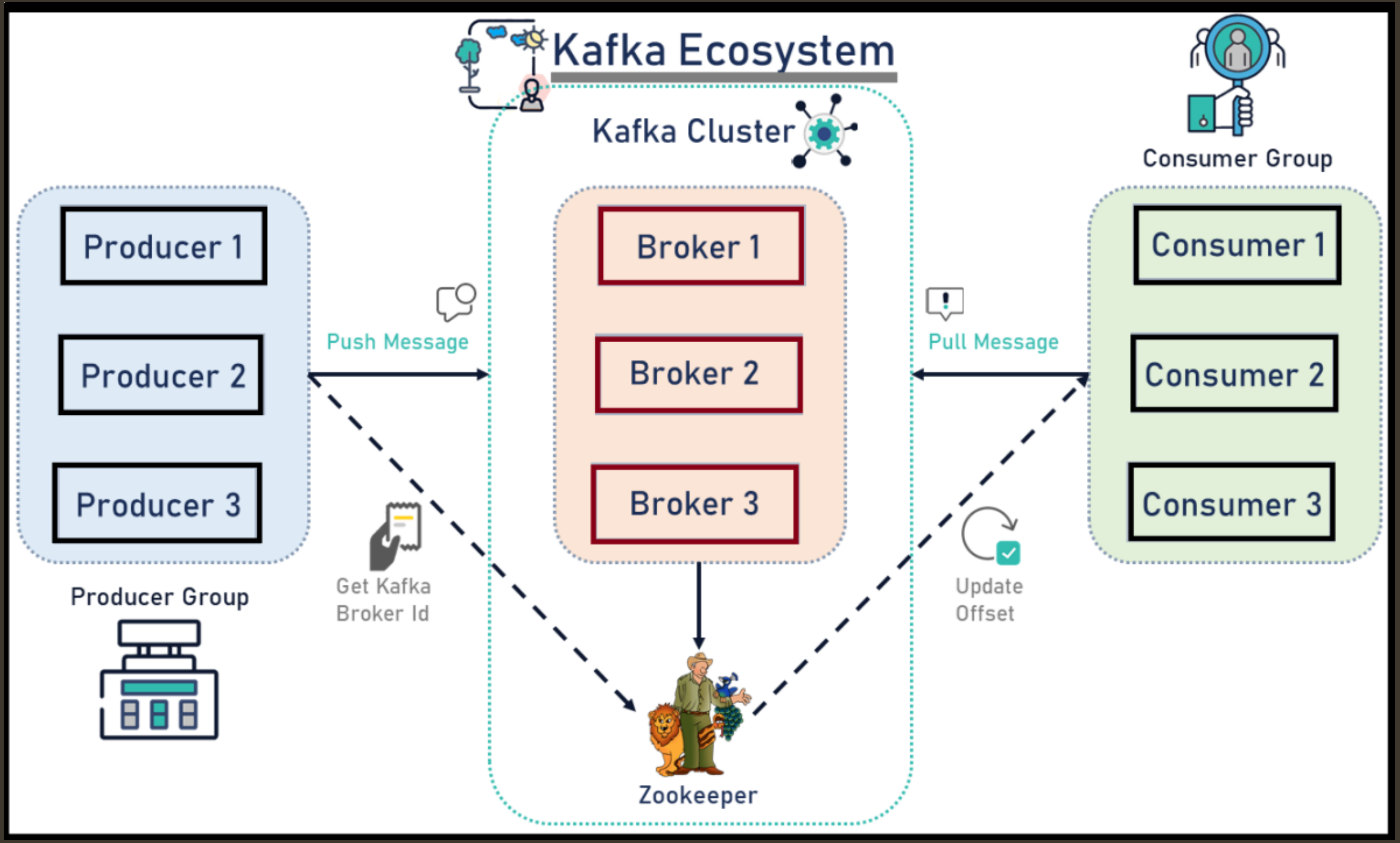
- High Throughput and Scalability
- Fault Tolerance and Reliability
- Real-time Processing
- Message Ordering

Architecture



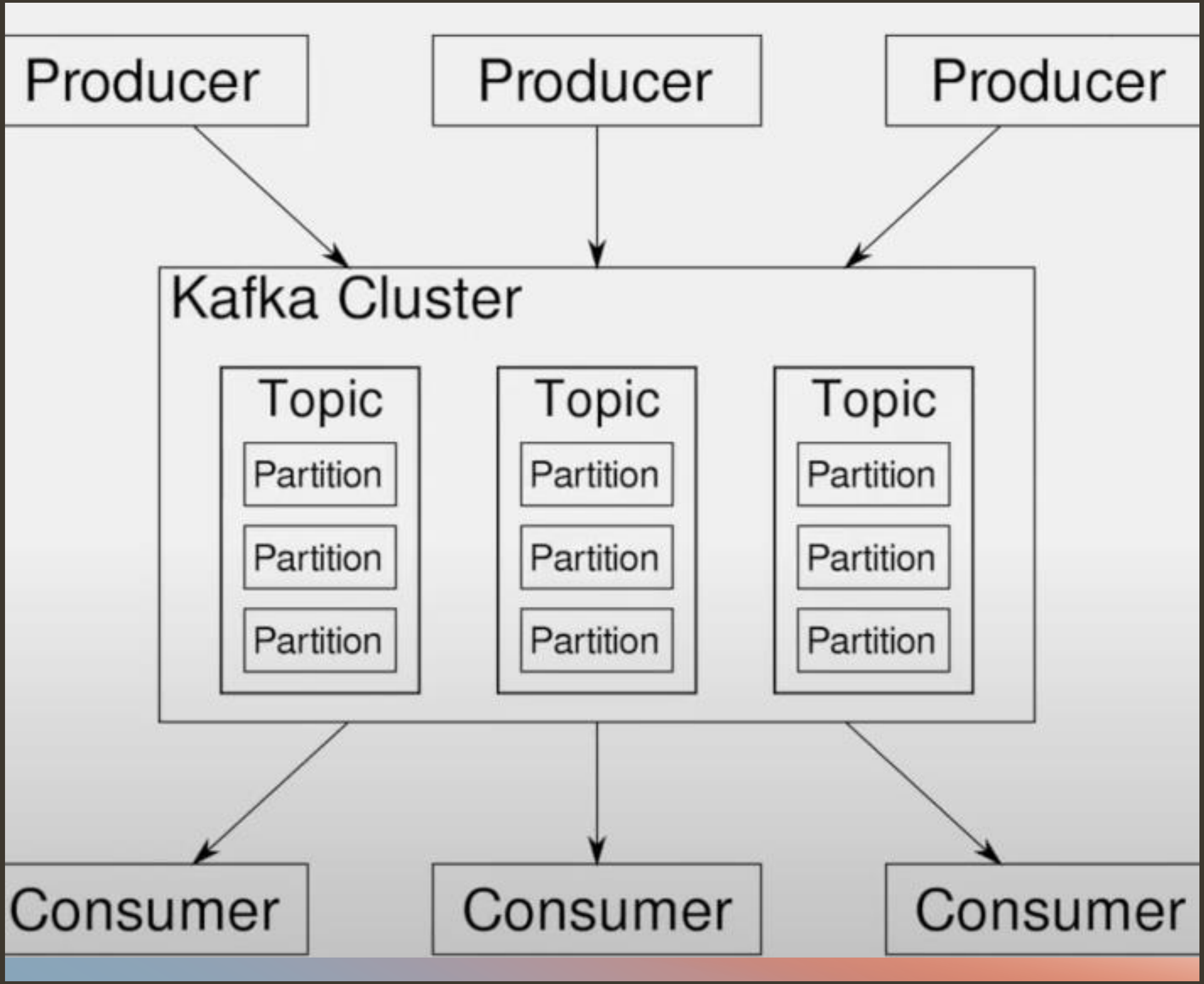
Zookeeper

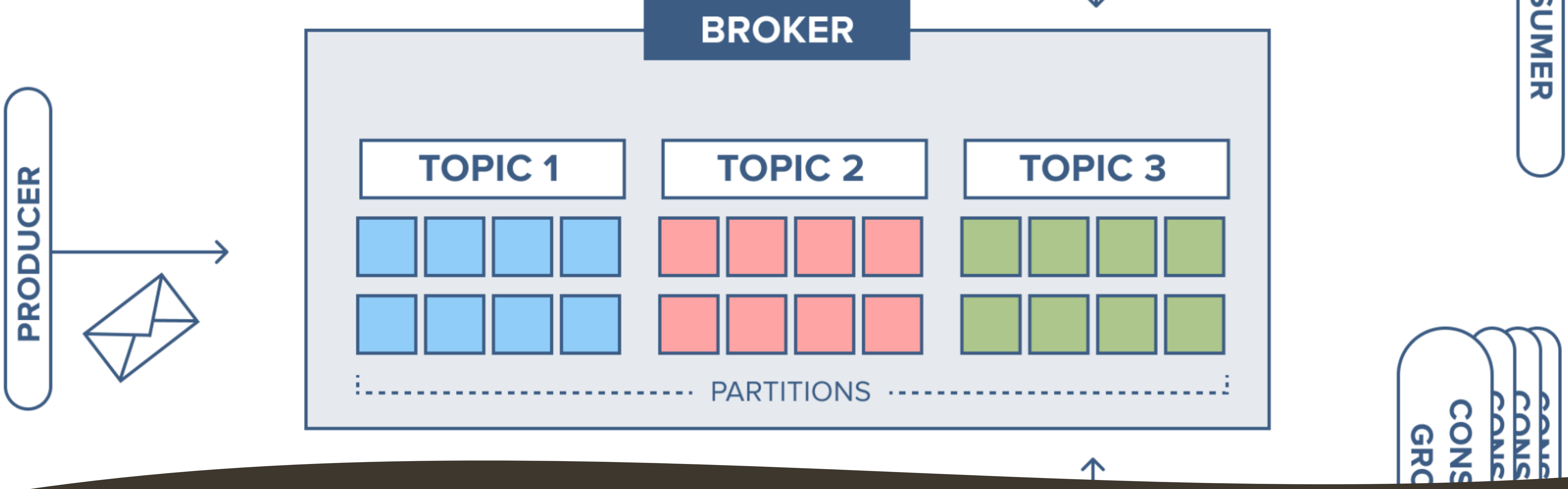
- Zookeeper is used to monitor kafka cluster and co-ordinate with each broker.
- Keeps all the meta-data related to kafka cluster in the form of a key-value pair.
- Meta Data Include:
  - Configuration information
  - Health Status of each broker



Topics

- A stream of messages belonging to a particular category.
- similar to a table in a database.
- We can create as many topics as we need.



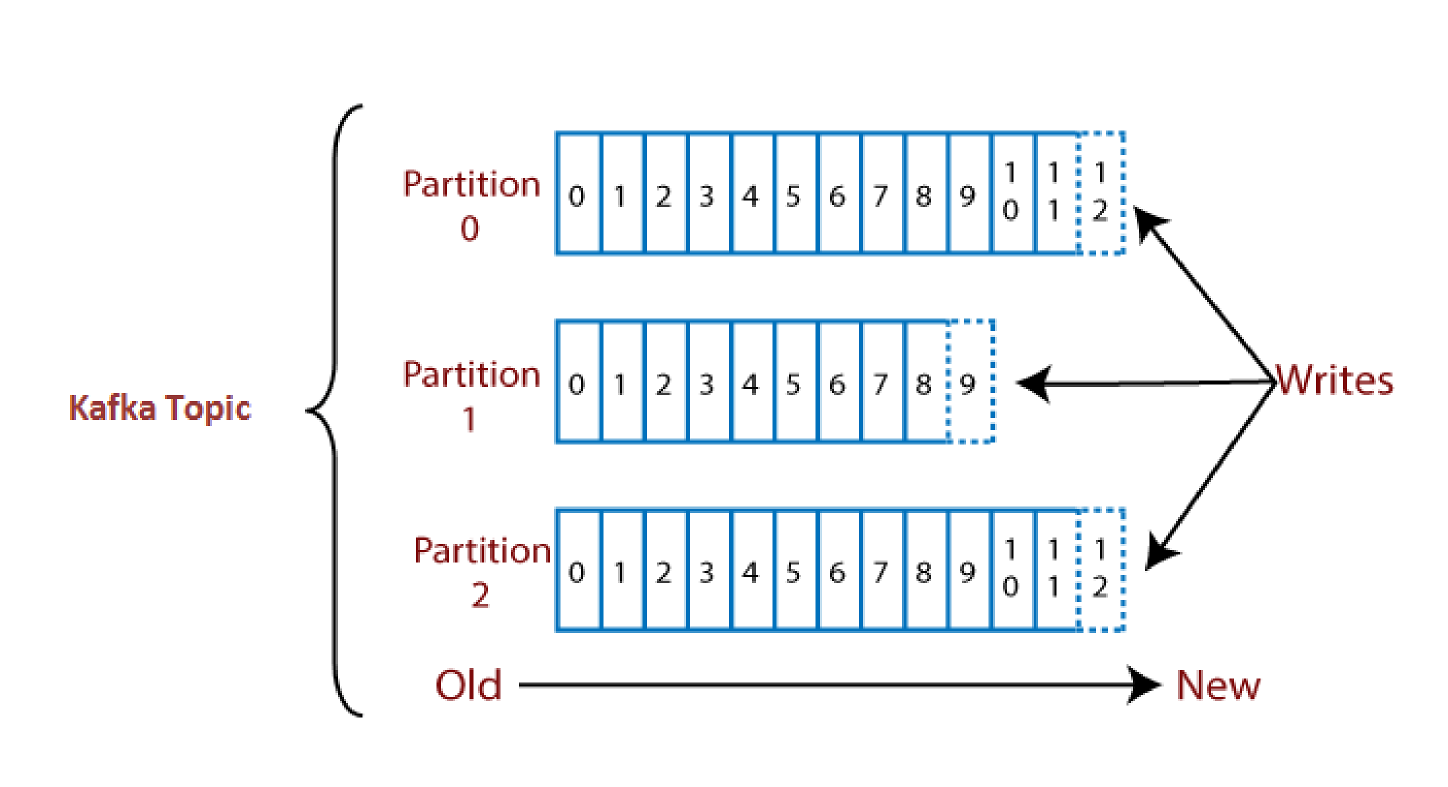


Partition

- Topics are split into partitions.
- All messages within a partition are ordered and immutable.
- Each message within a partition has a unique ID, known as an OFFSET.

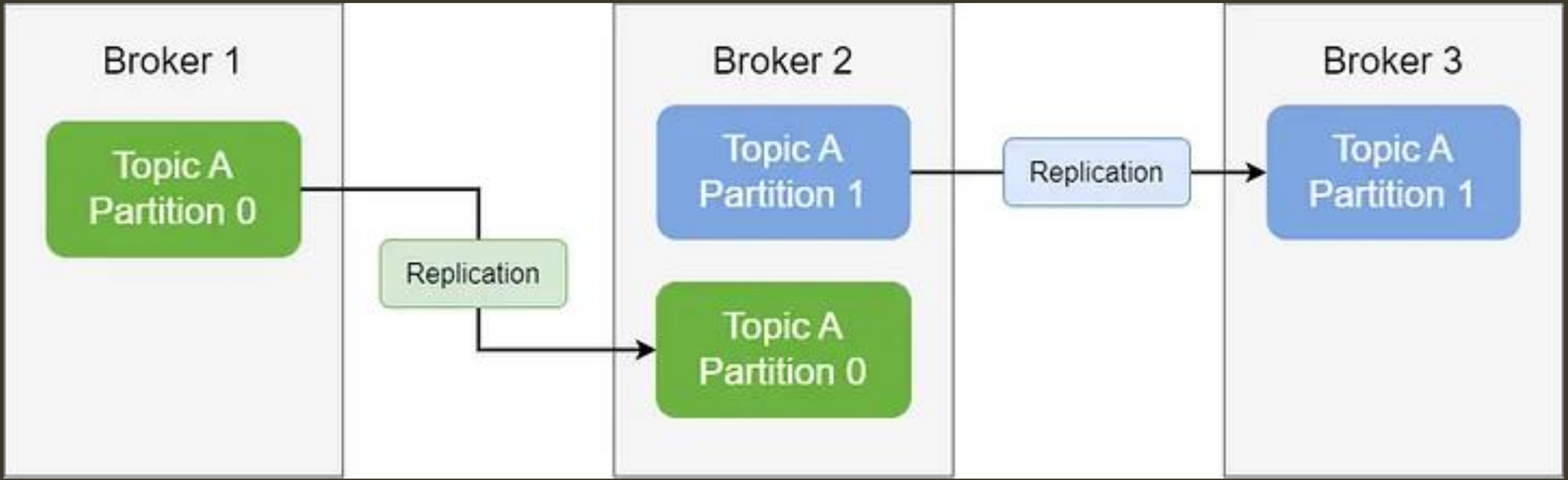
OFFSET

- The records in the partitions are each assigned a sequential ID number called an offset, which uniquely identifies each record within the partition



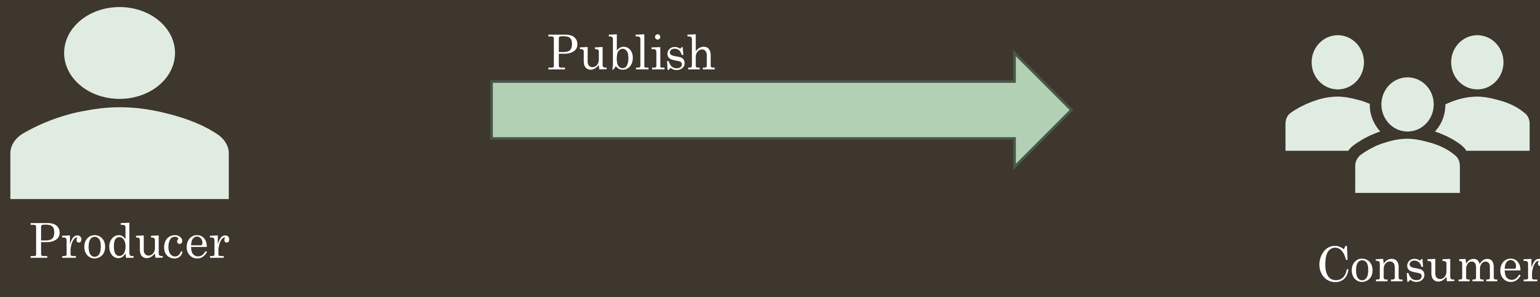
Replica & Replication

- Replica are backups of partitions.
- Replica do not read or write data.
- They are used to prevent data loss (fault tolerance).



Producer

- Producer are applications that write or publish data to topics within a cluster using the Producer API.



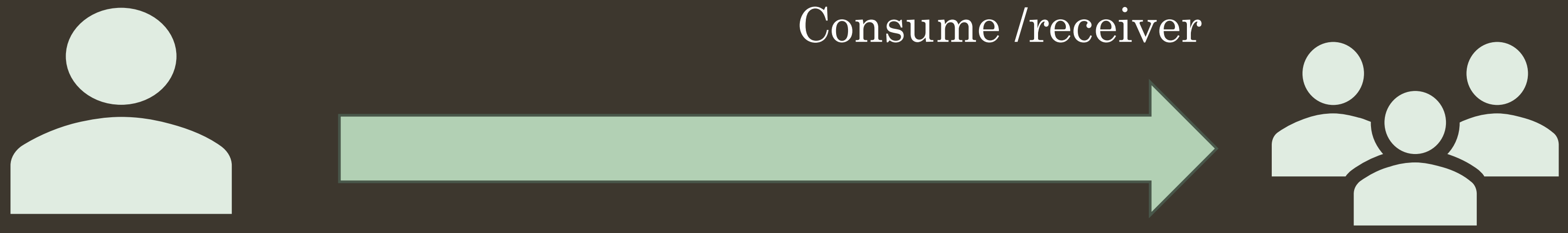
Publish

Producer

Consumer

Consumers

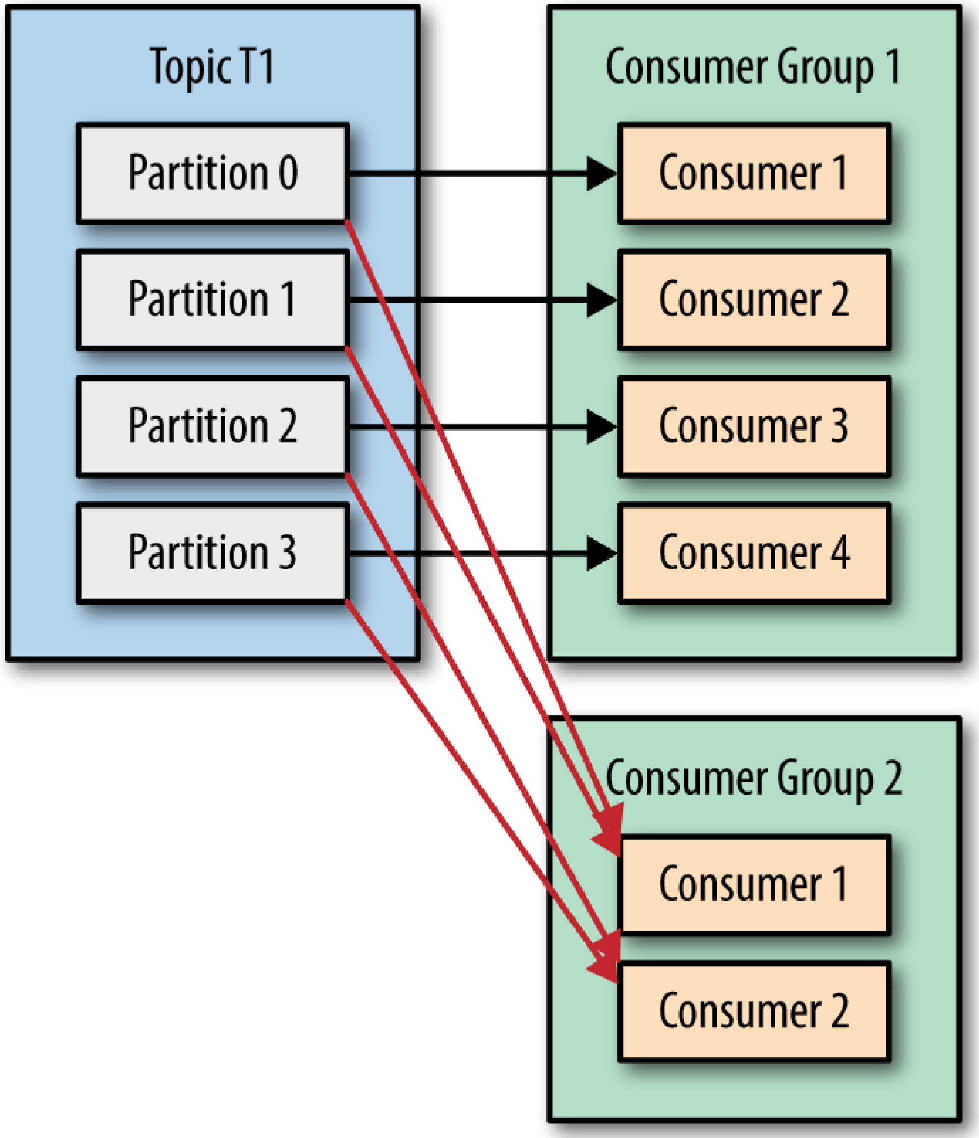
- Consumers are applications that read or consume data from topics within a cluster using the Consumer API.
- Consumers are always associated with exactly one consumer group.
- A consumer group is a group of related consumers that perform a task.



Consume /receiver

Producer

Consumers

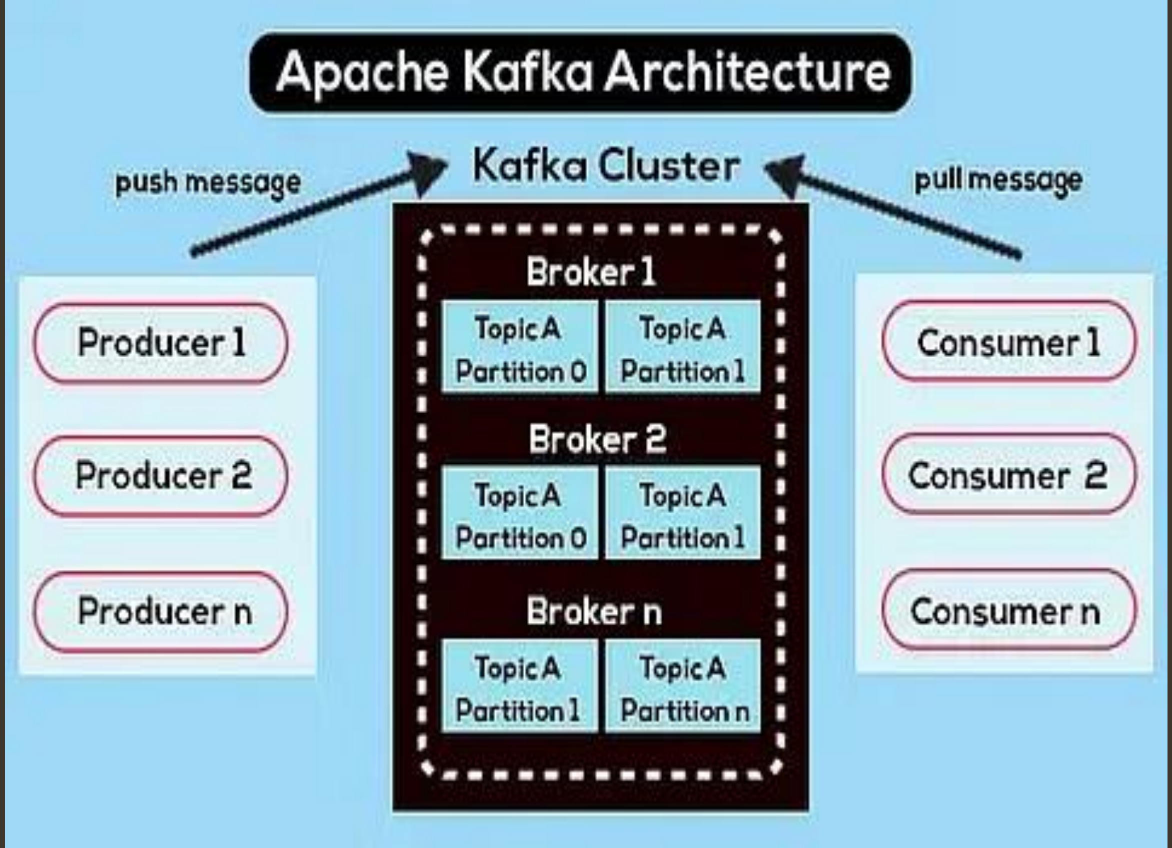


Consumer Group

- A collection of consumers that work together to process data from a topic

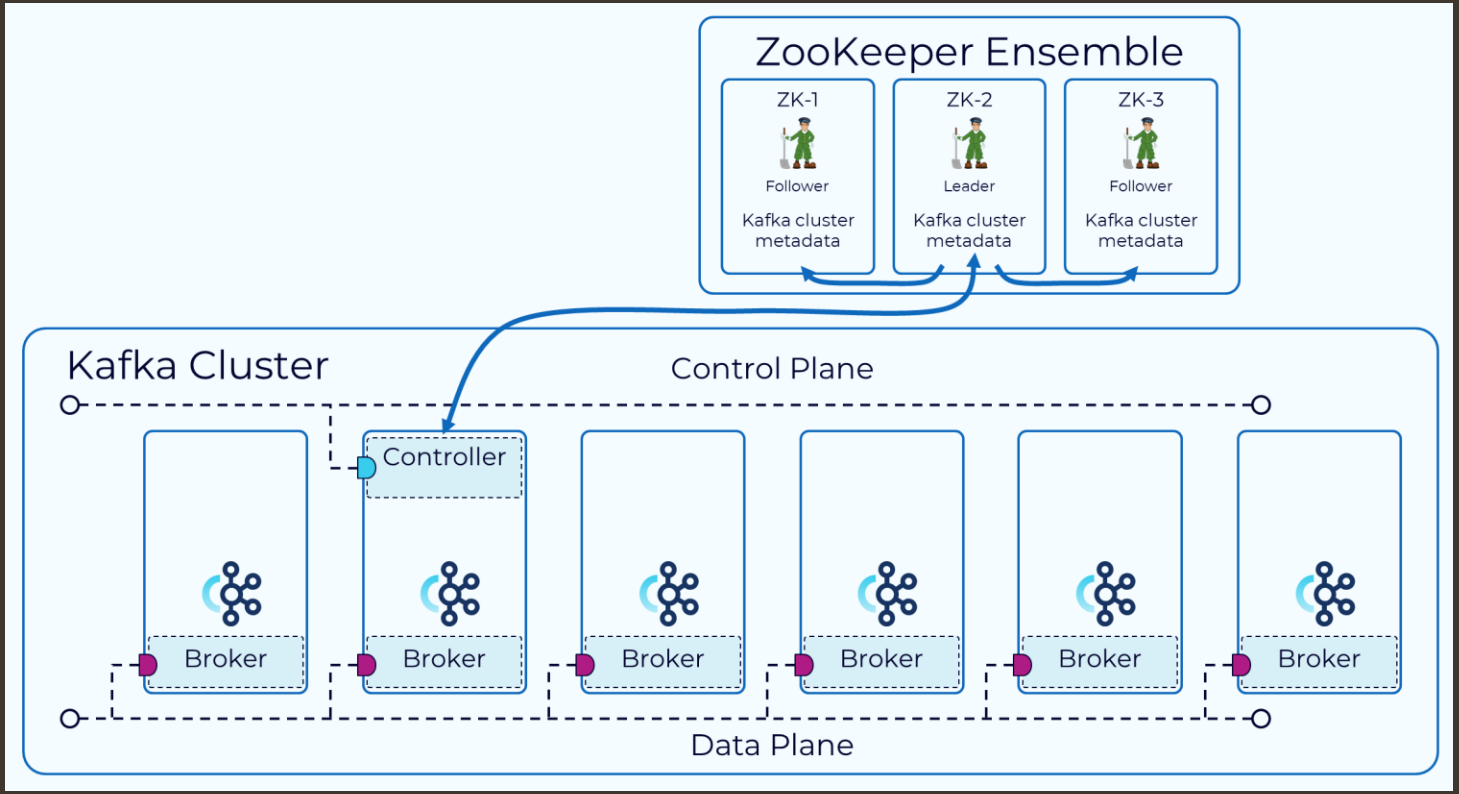
Broker

- Brokers are simple software processes that maintain and manage the published messages.
- They are also known as Kafka servers.
- We can add more brokers to an already running Kafka cluster without any downtime.



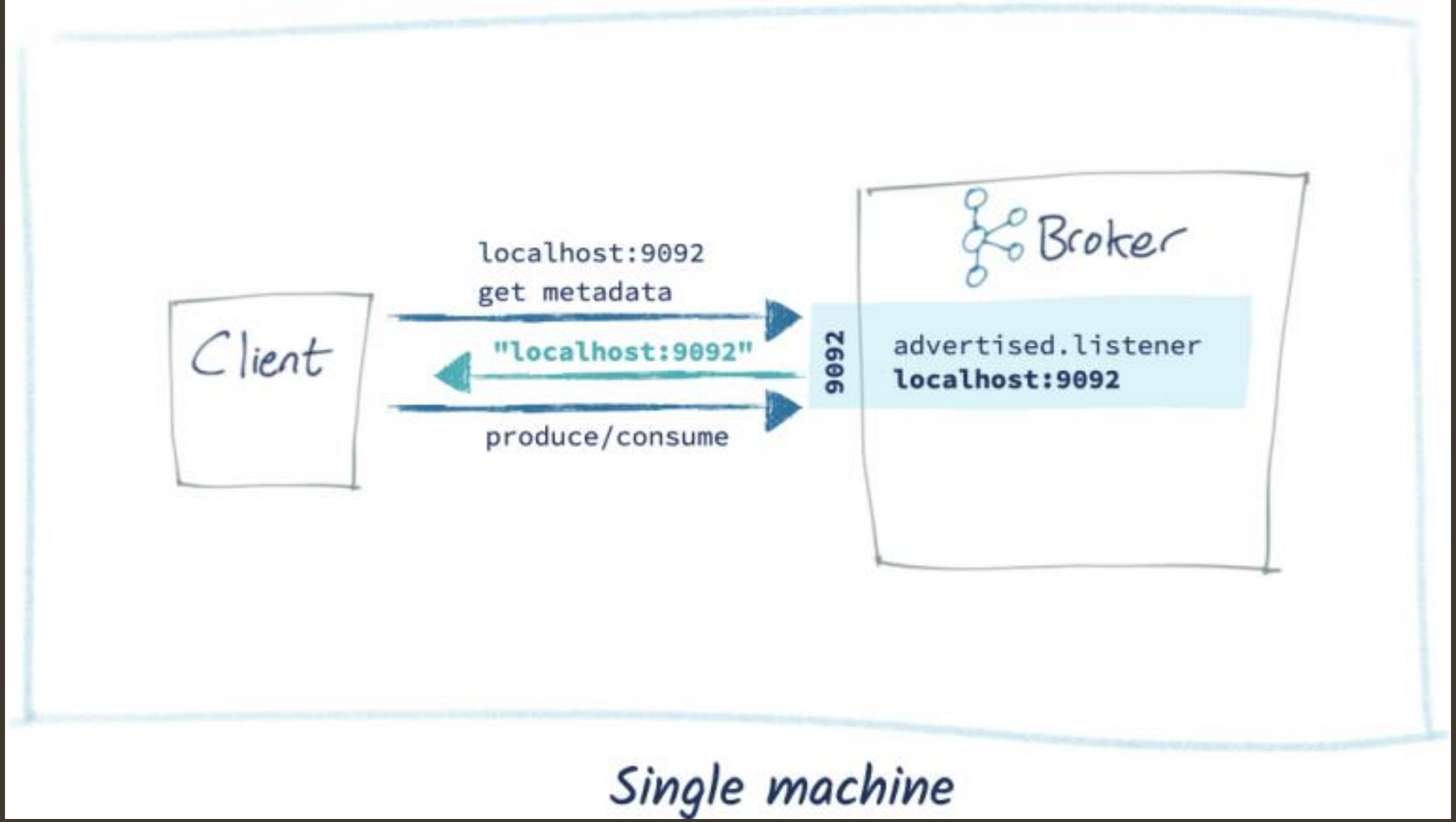
Kafka Controller Node

- Kafka cluster, one of the broker servers acts as the controller, responsible for managing the states of partitions and replicas, and performing administrative tasks like reassigning partitions.



BootStrap Server

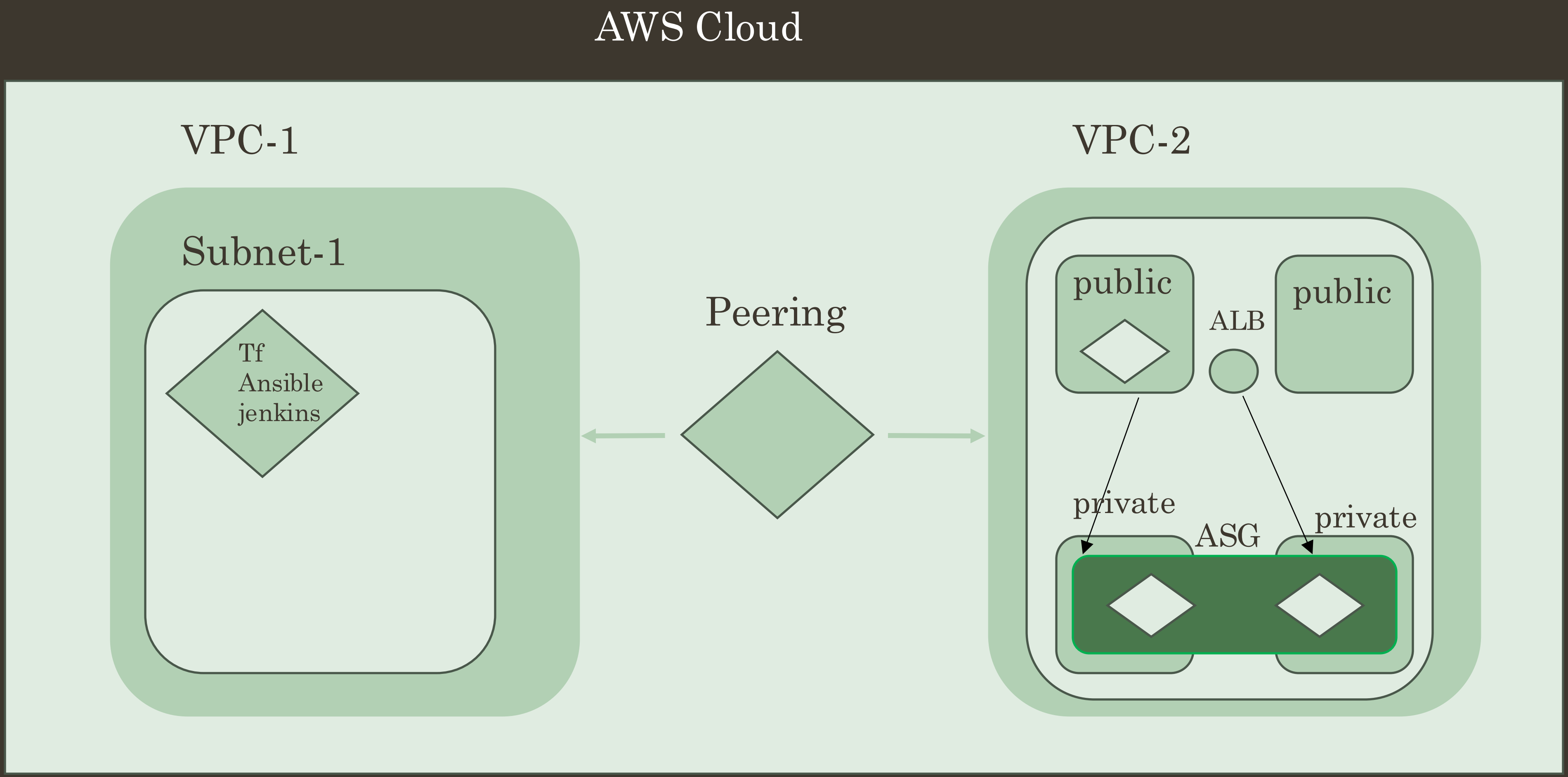
- Bootstrap: Used to connect to the Kafka cluster and establish a connection for both consumers and producers.



Installation of Apache Kafka

- Download Kafka Zip file from official website (wget https://packages.confluent.io/archive/7.0/confluent-community-7.0.0.tar.gz)
- Extract file(tar -xf confluent-community-7.0.0.tar.gz)
- Install Java version 11 (sudo apt install openjdk-11-jdk)
- cd /path/to/kafka_2.13-3.0.0
- Start Kafka Server

One-Click Infra



THE END
THANK FOR YOUR ATTENTION
makeameme.org

Thanks For Joining