Building your First Apache Kafka Application

Bogdan Sucaciu
SOFTWARE ENGINEER @ AXUAL

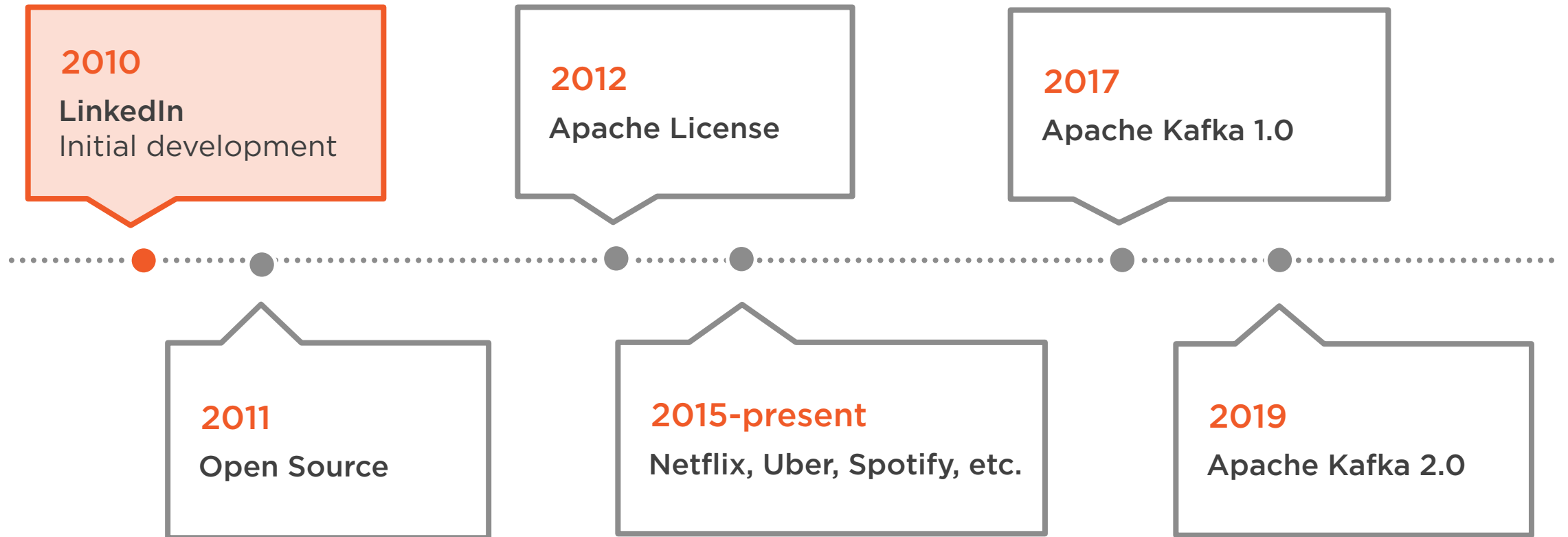@BSucaciu

Why Kafka?

Open source

Kafka History

2010
LinkedIn
Initial development

2012
Apache License

2017
Apache Kafka 1.0

2011
Open Source

2015-present
Netflix, Uber, Spotify, etc.

2019
Apache Kafka 2.0

Why Kafka?

Open source

Java

Classes

Compiler
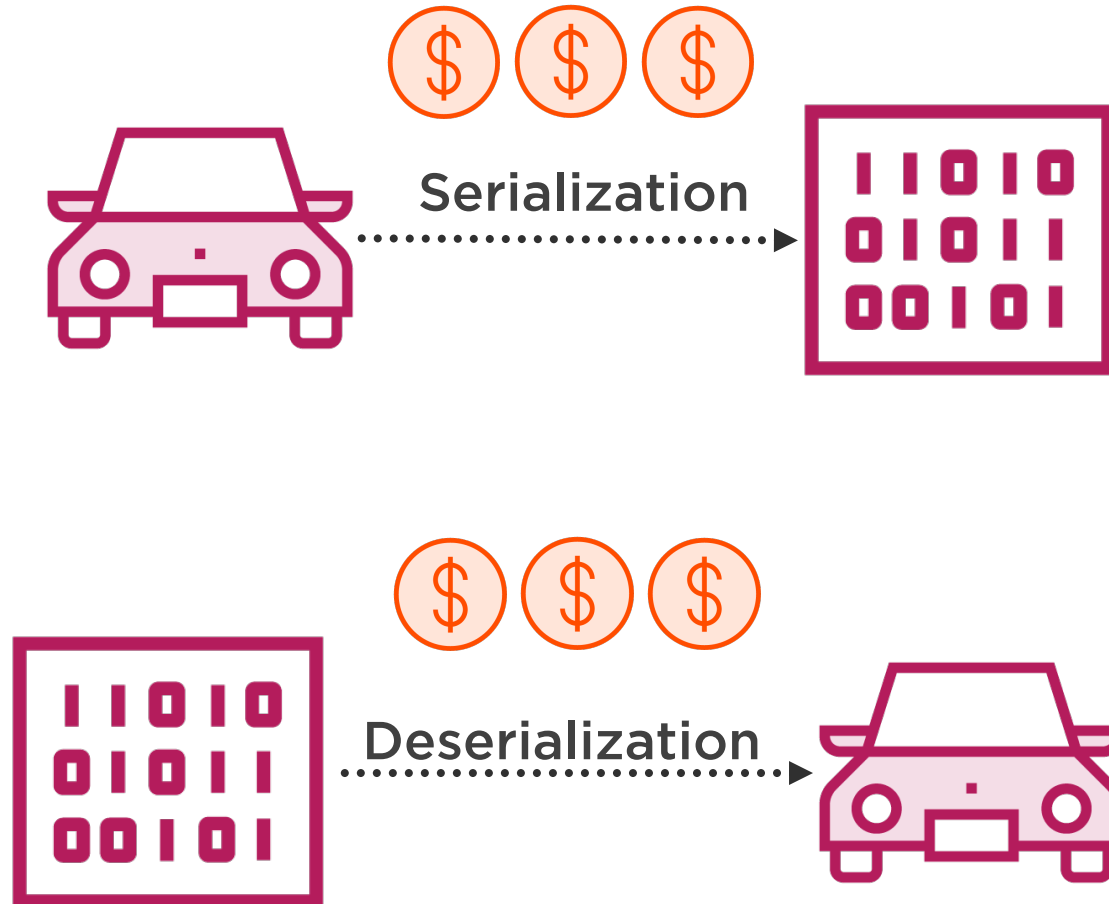

Scala

Bytecode

Java

Bytecode

JVM
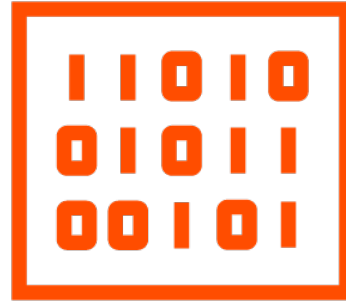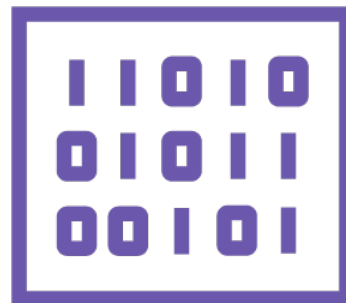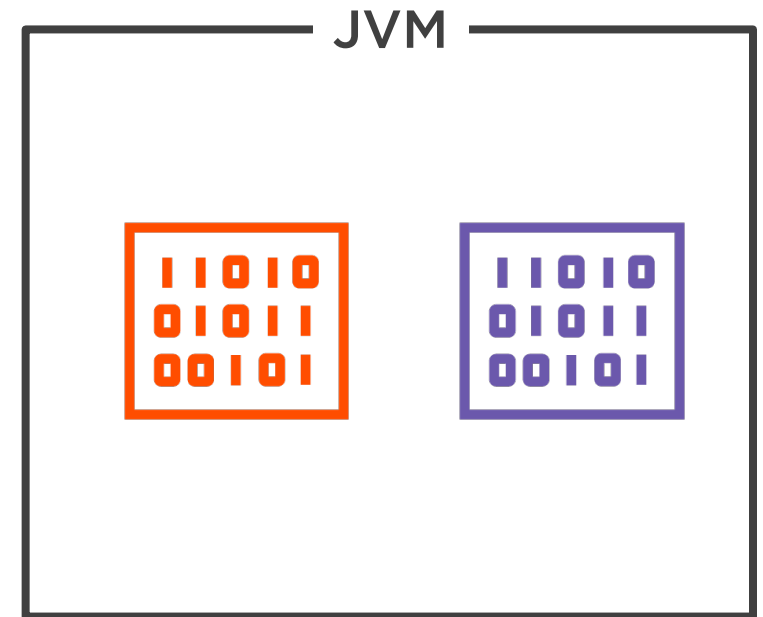
Why Kafka?

Open source

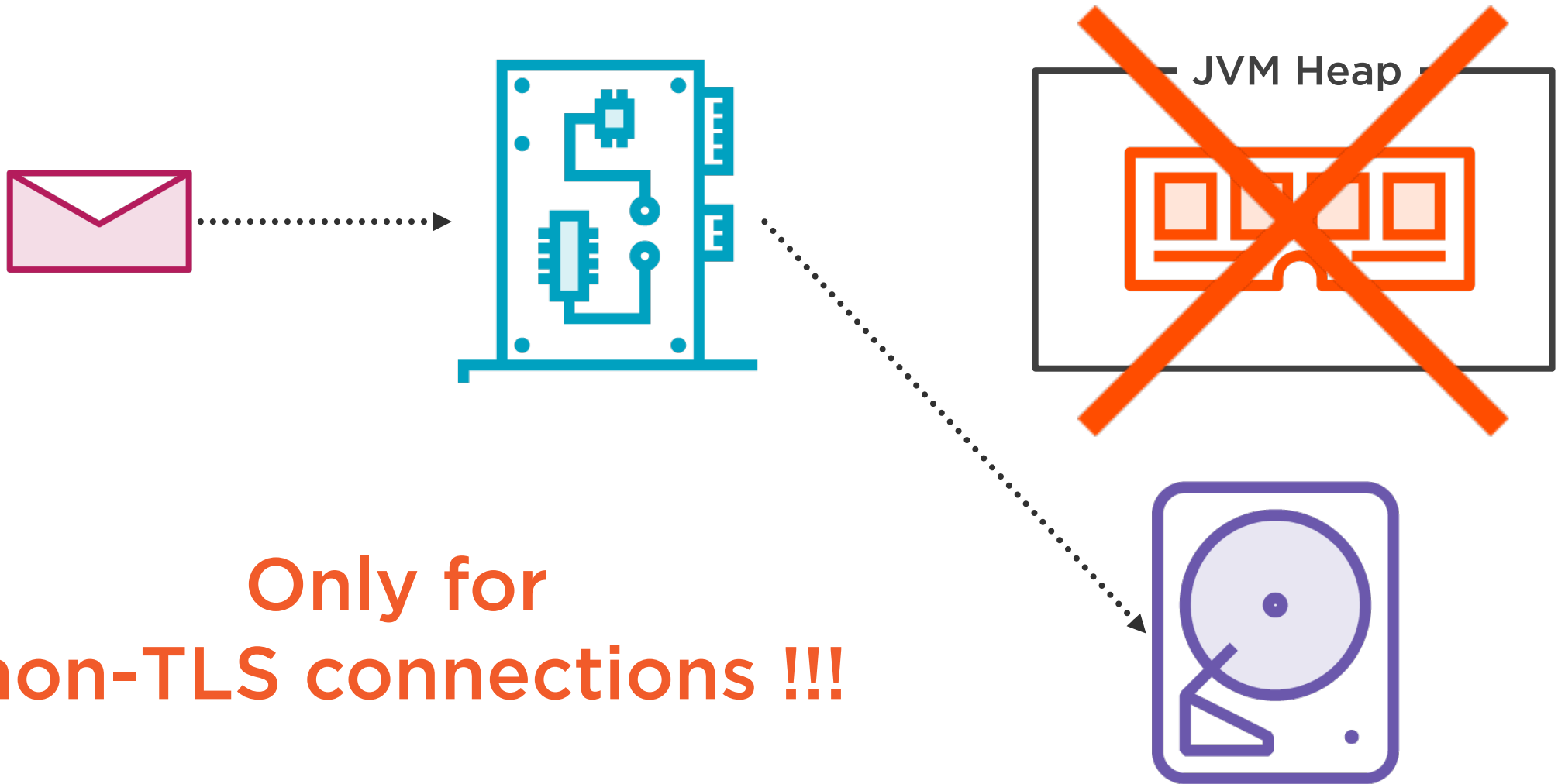Java

High throughput

No Serialization / Deserialization

No Serialization / Deserialization

Zero Copy

Zero Copy

JVM Heap

**Only for
non-TLS connections !!!**
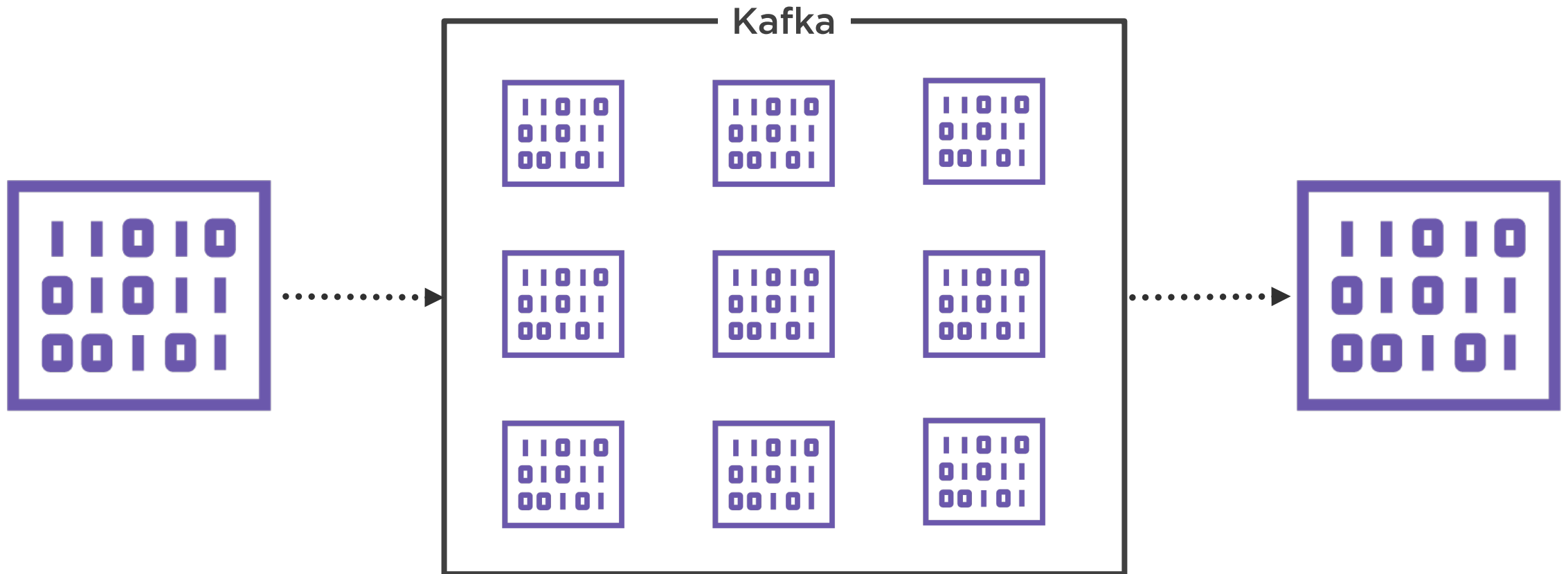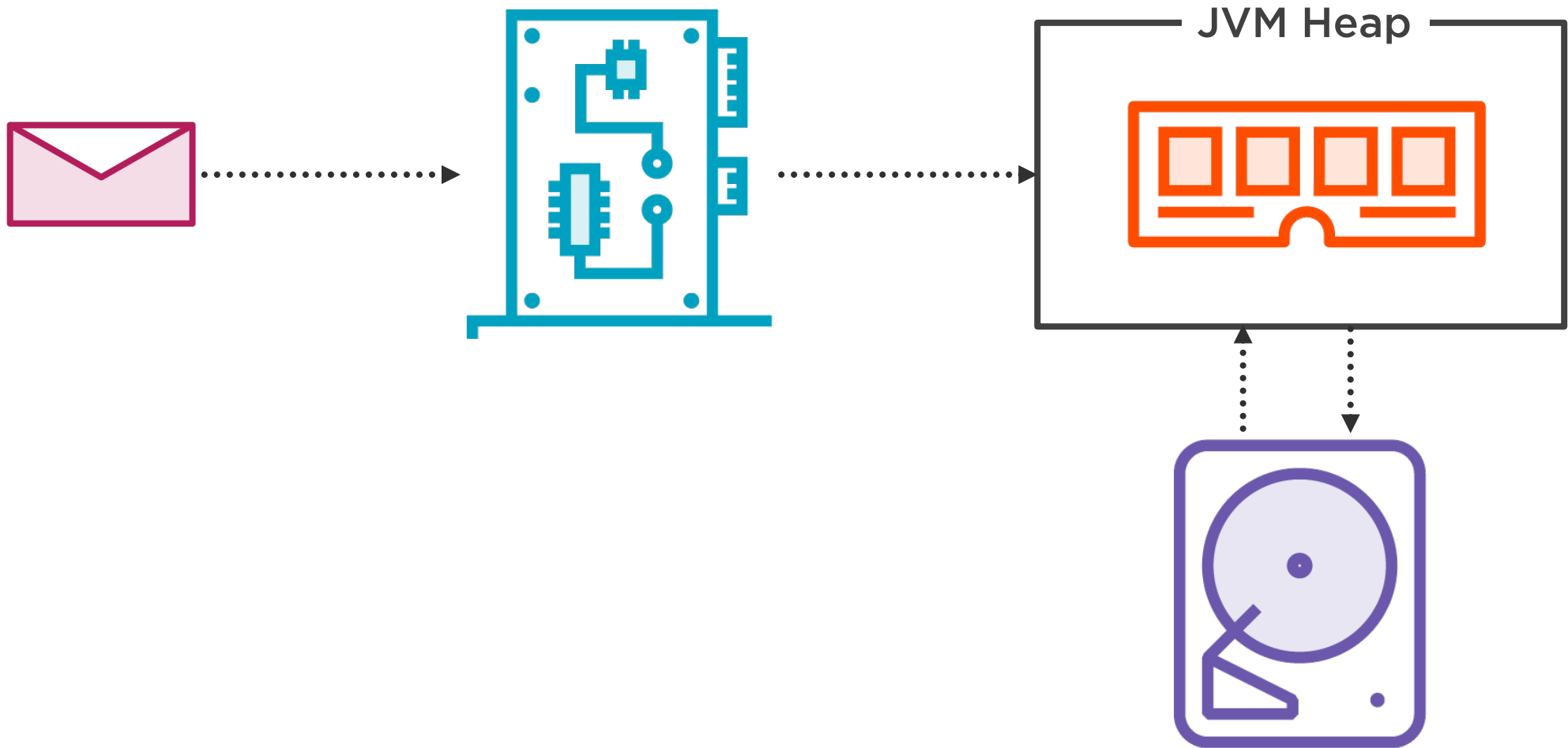
Why Kafka?

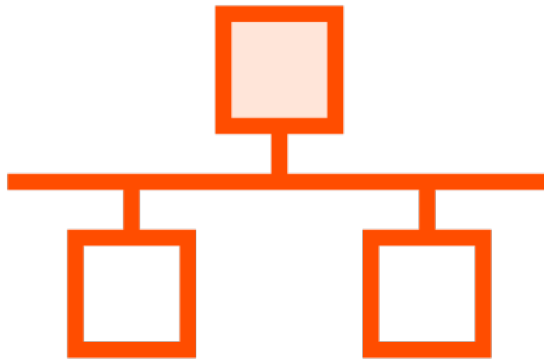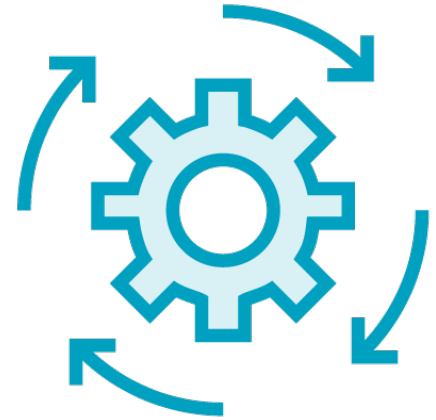|	
Open source	**Java**
High throughput	**More than a messaging system**

Distributed Streaming Platform

Messaging system

Publish / Subscribe pattern

Distributed storage

Store data in a fault-tolerant, durable way
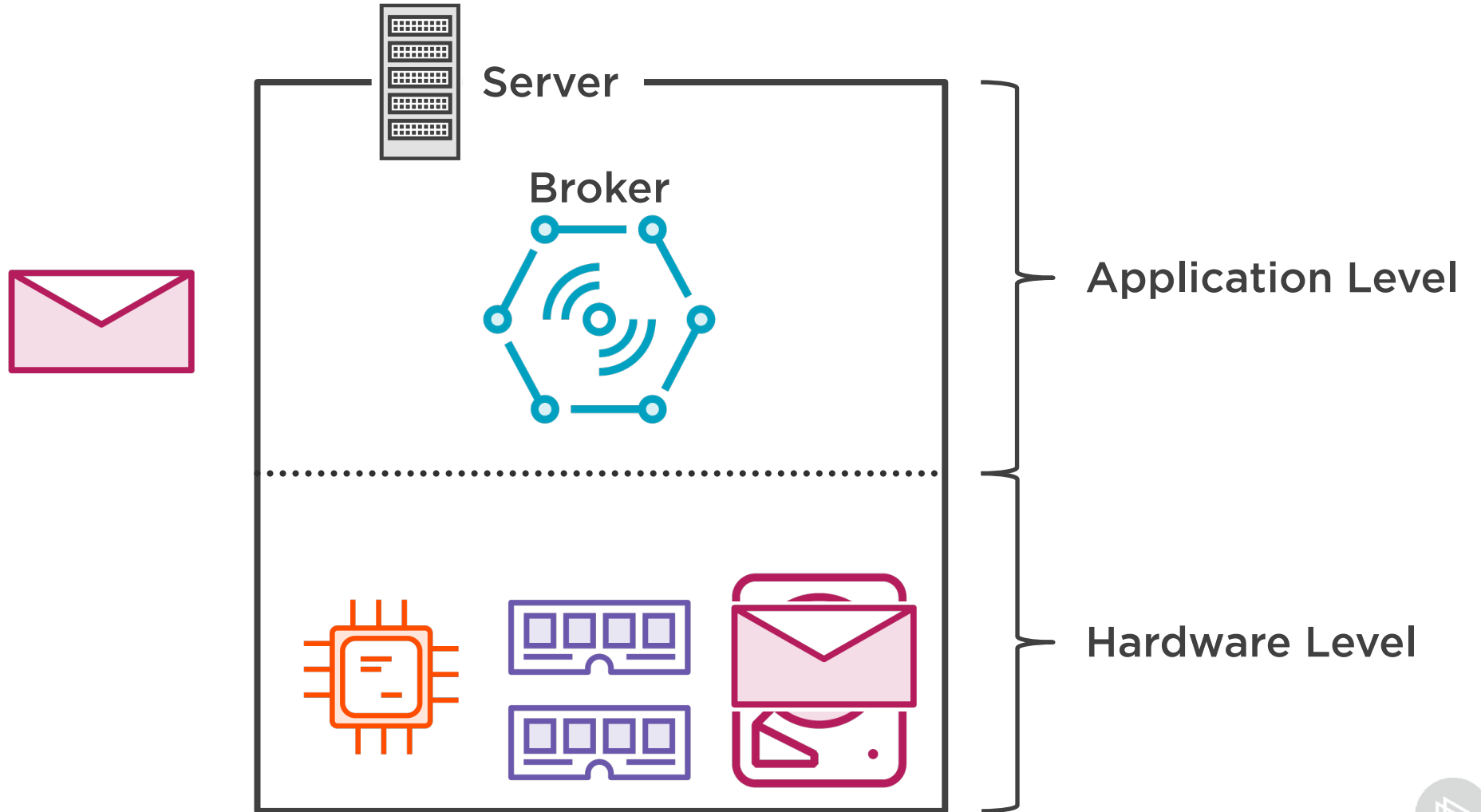
Data processing
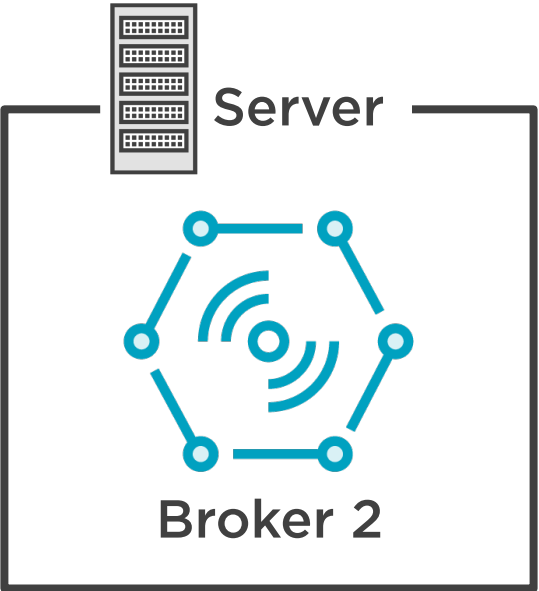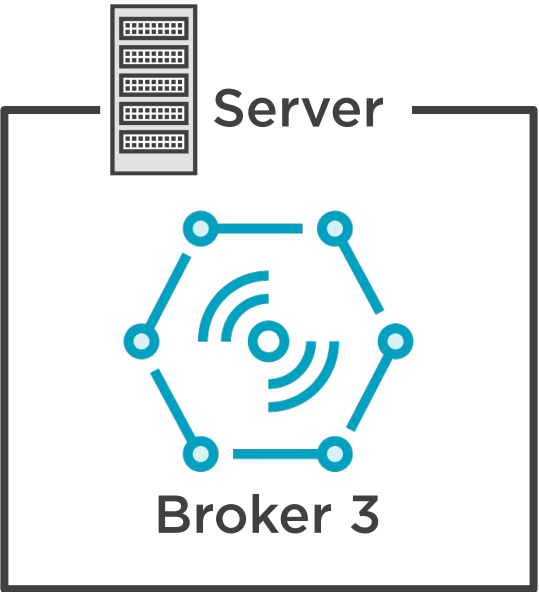
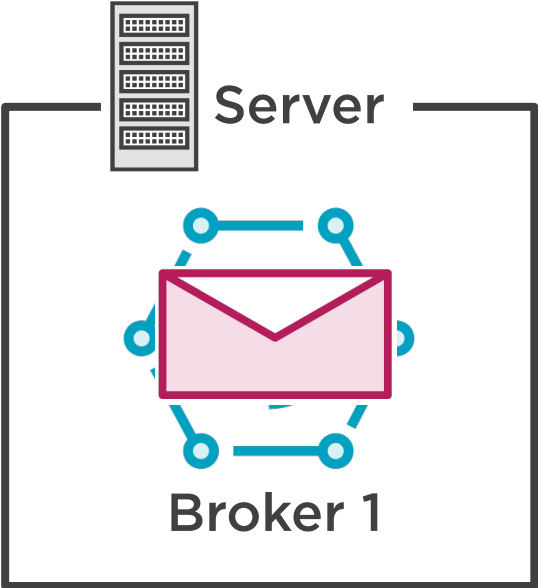Process events as they occur

Kafka Components

Broker

Zookeeper

Broker

Kafka Cluster

Zookeeper

Zookeeper
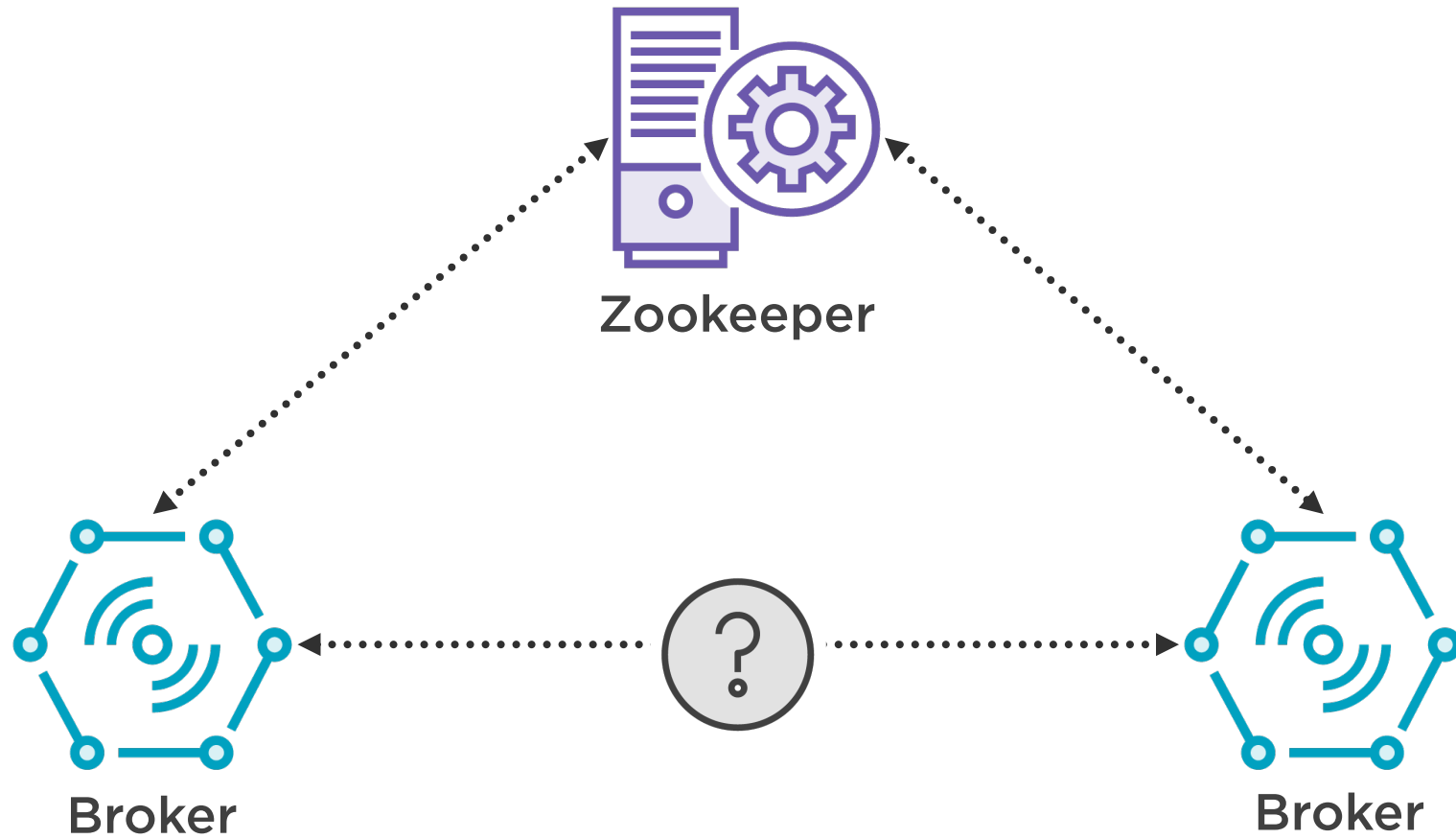
Broker

Broker

Demo

Setup a 2 node Kafka cluster:
- 2x Broker
- 1x Zookeeper

Record

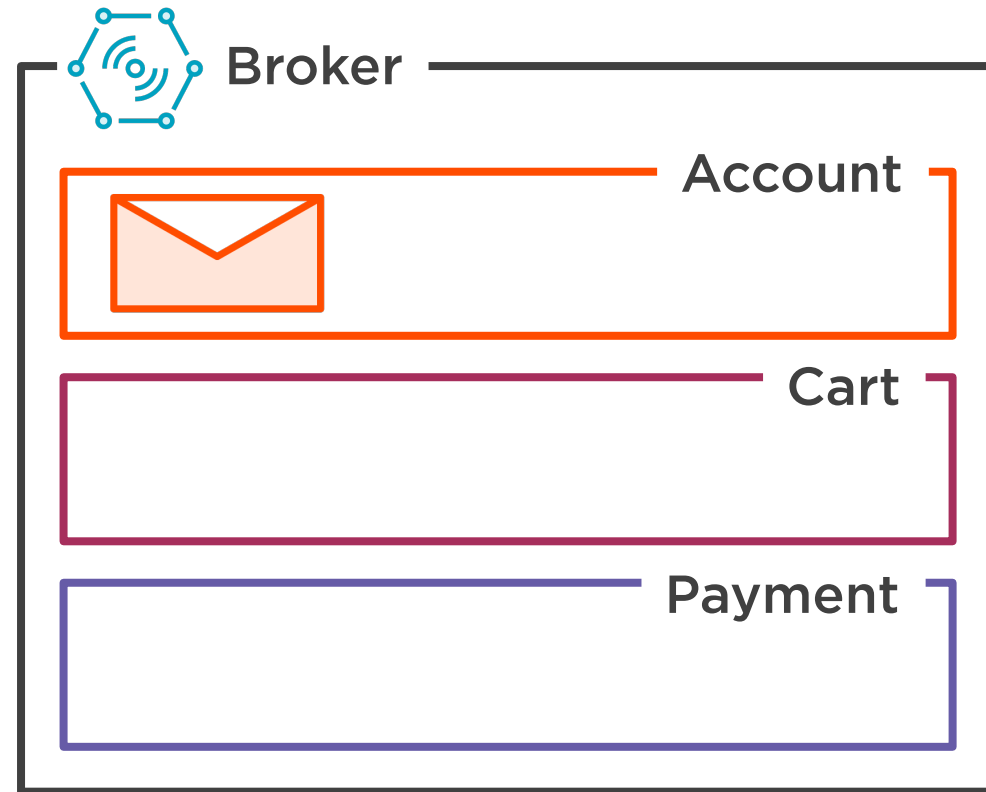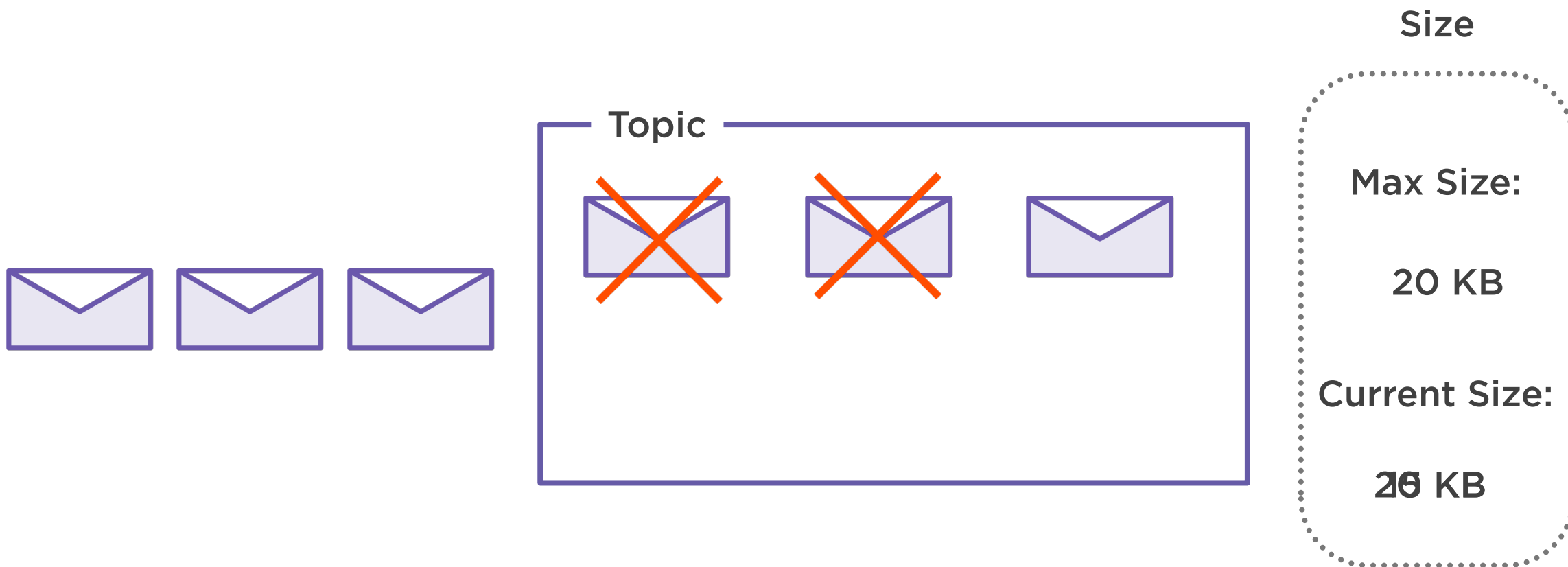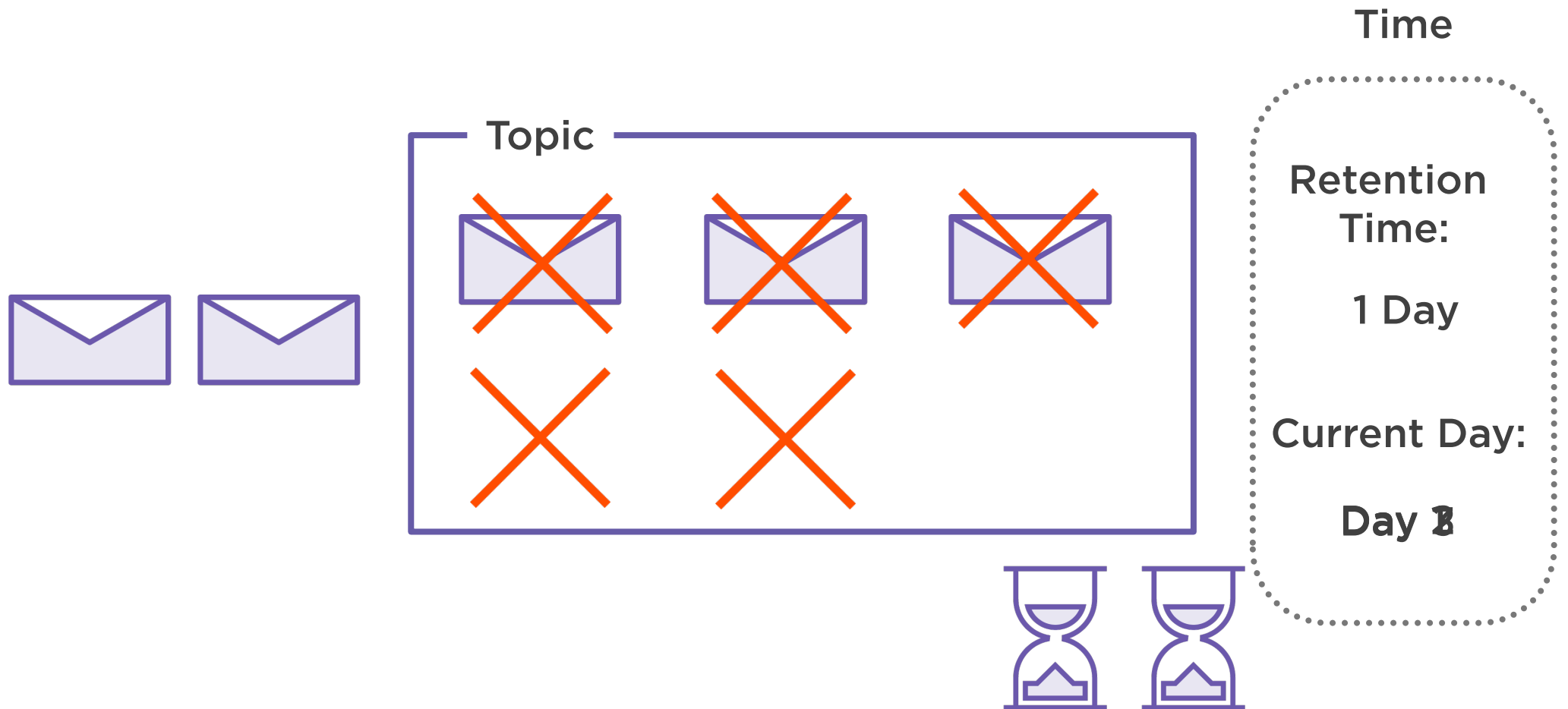| Key	Value	Timestamp
Anything	Anything	Timestamp format

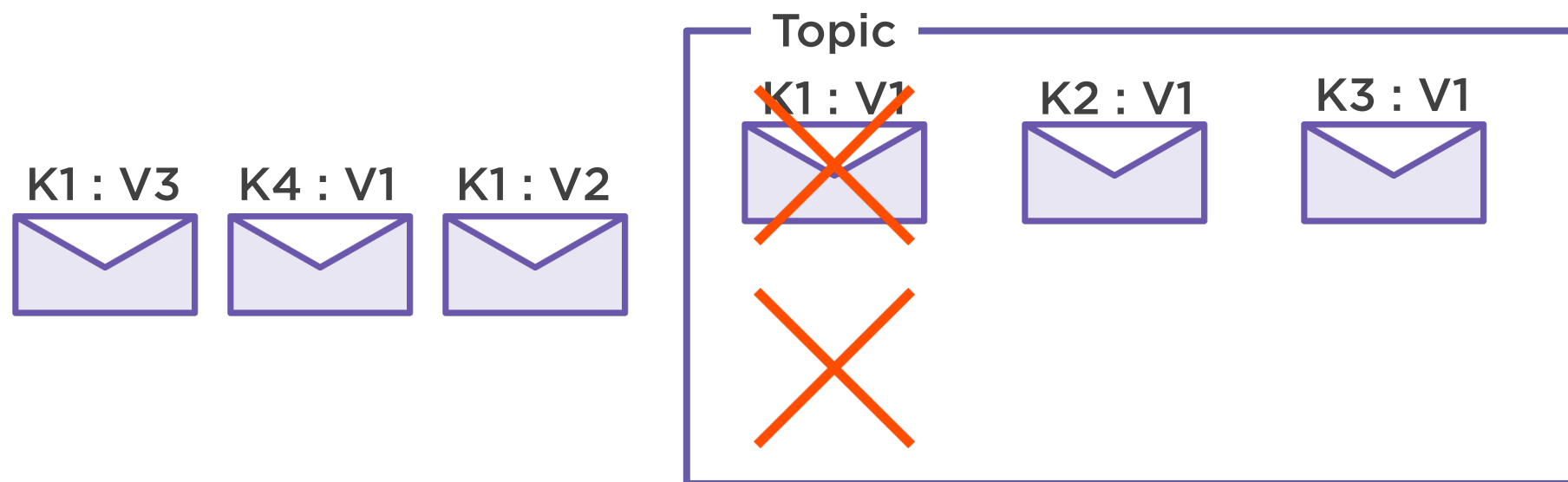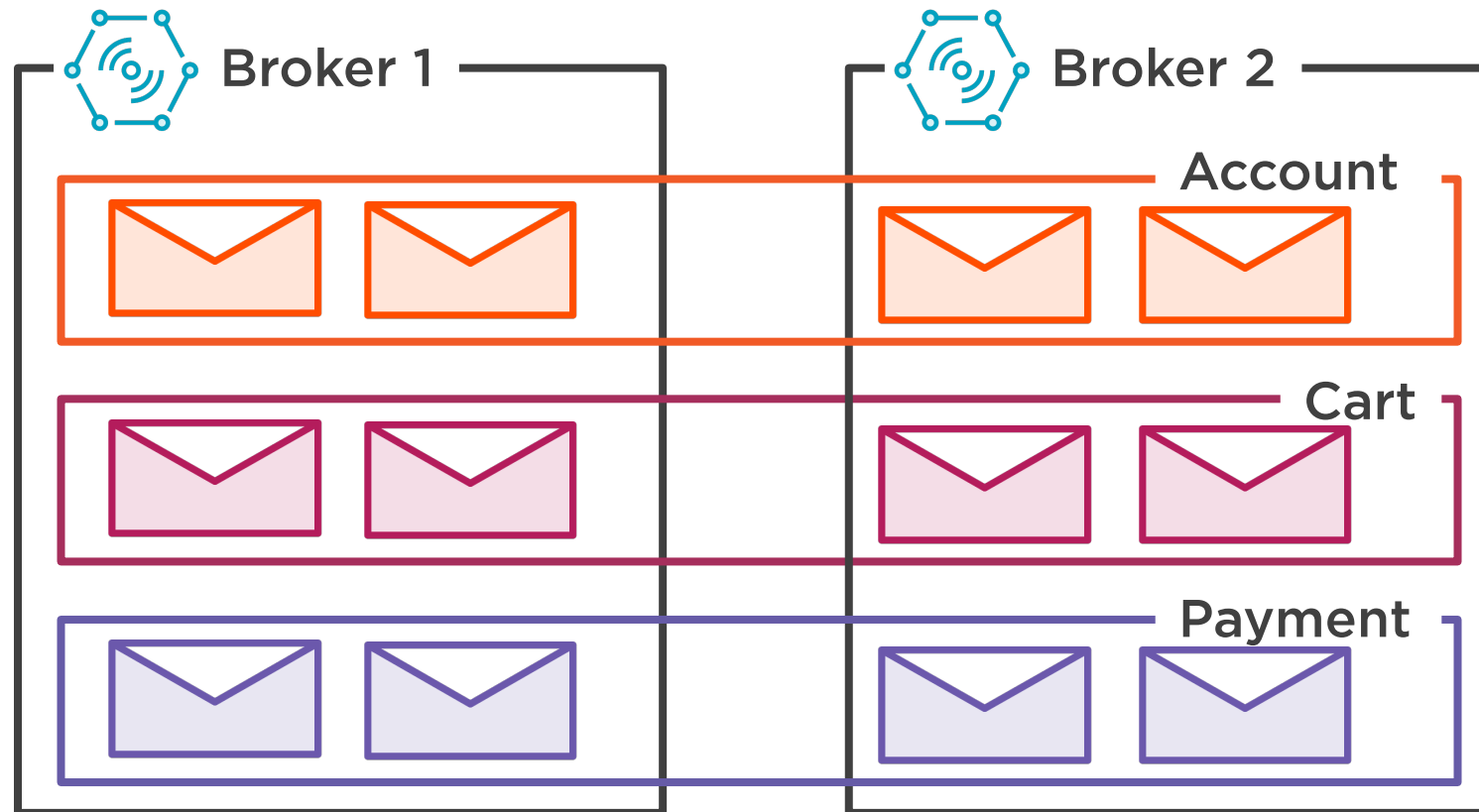# Topics
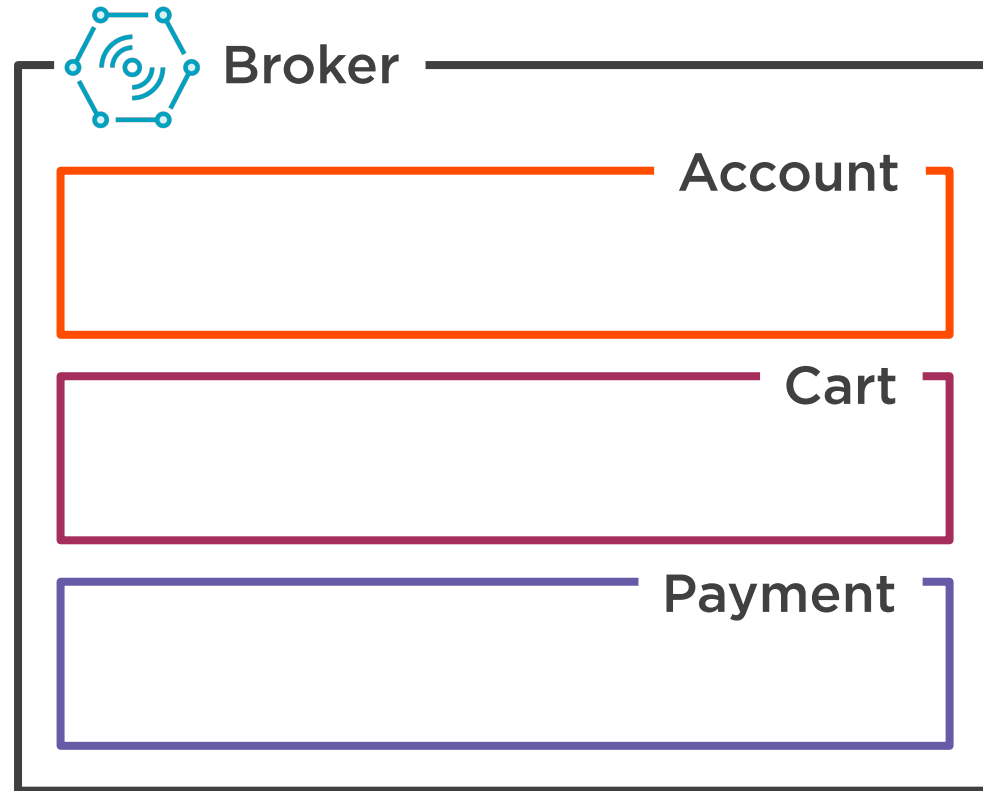
Topics

Topics
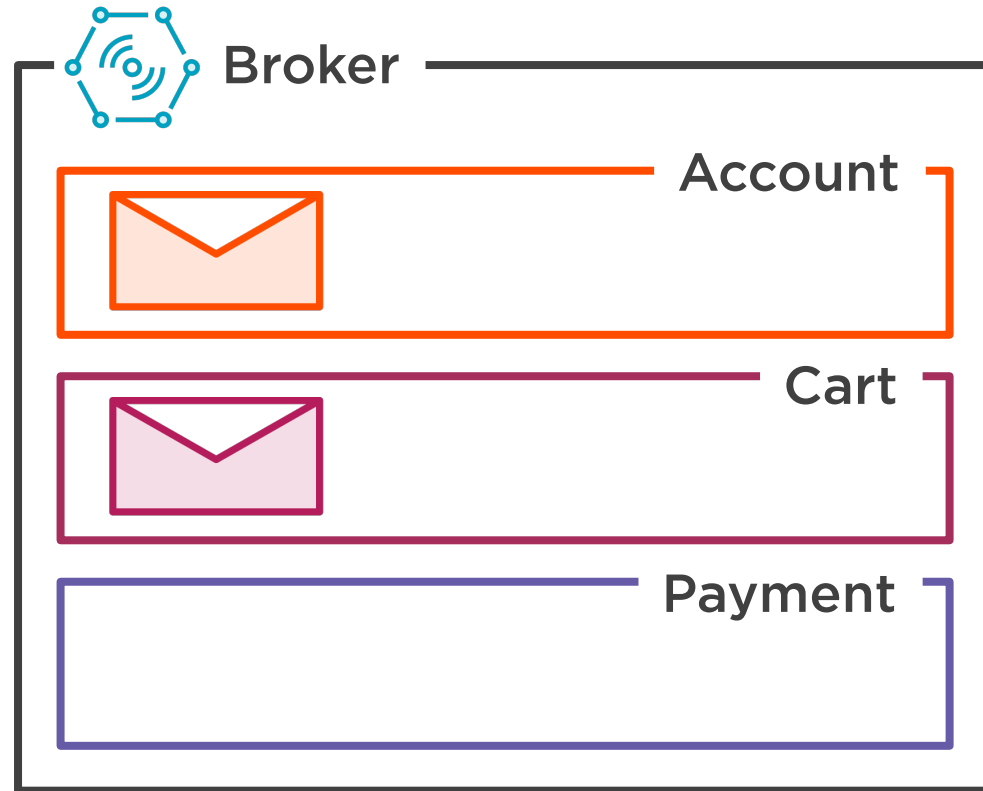
Topics

Delete

Compact

Delete Topics

Delete Topics
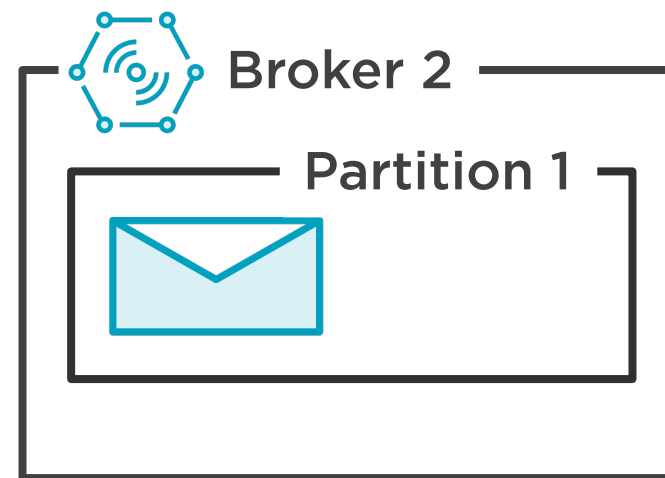

Topic

Time

Retention Time:

1 Day

Current Day:

Day 2

Compaction Topics
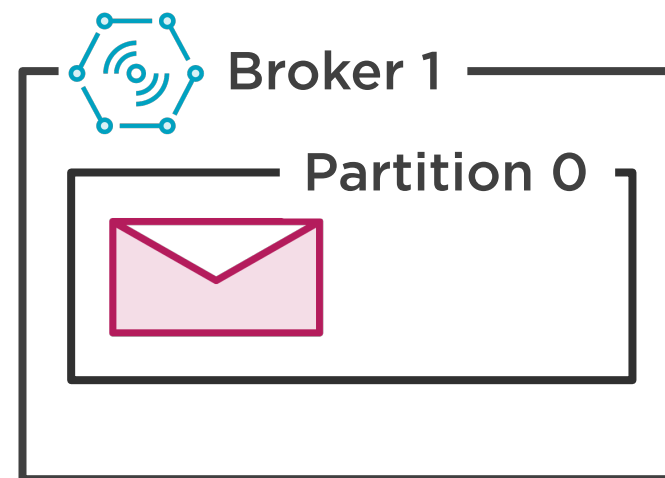
K1 : V3

K4 : V1

K1 : V2
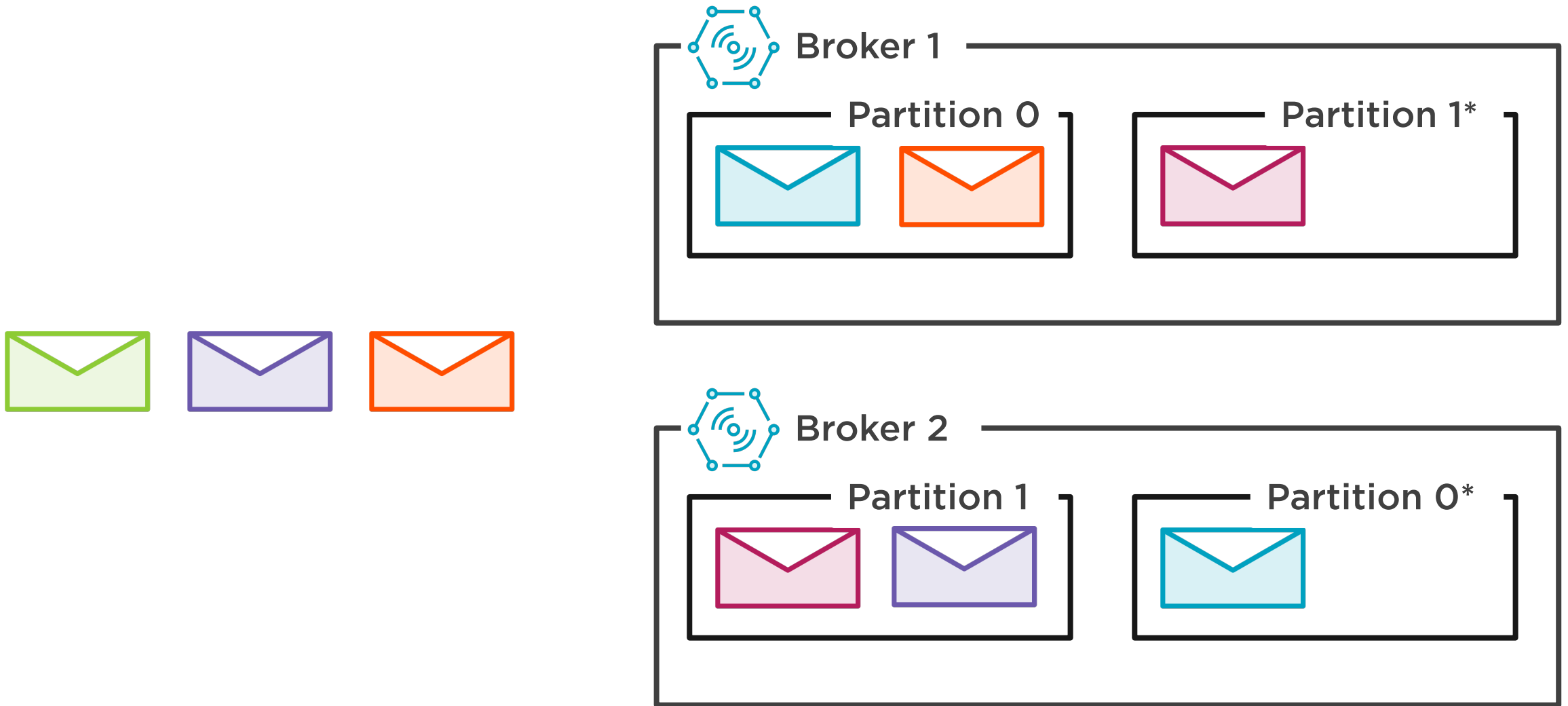
Topic

K1 : V1

K2 : V1

K3 : V1

Topics

Partitions

Replicated Partitions

Know Your Data !

Producer

Producer　　　　　　　　　　　　　　**Kafka**

Producer

Web Page

Kafka

Producer

Information

Producer Record

123ABC

Key Serializer
StringSerializer

Producer Record

01001010

TShirt, Purple, None

Value Serializer
StringSerializer

11100010
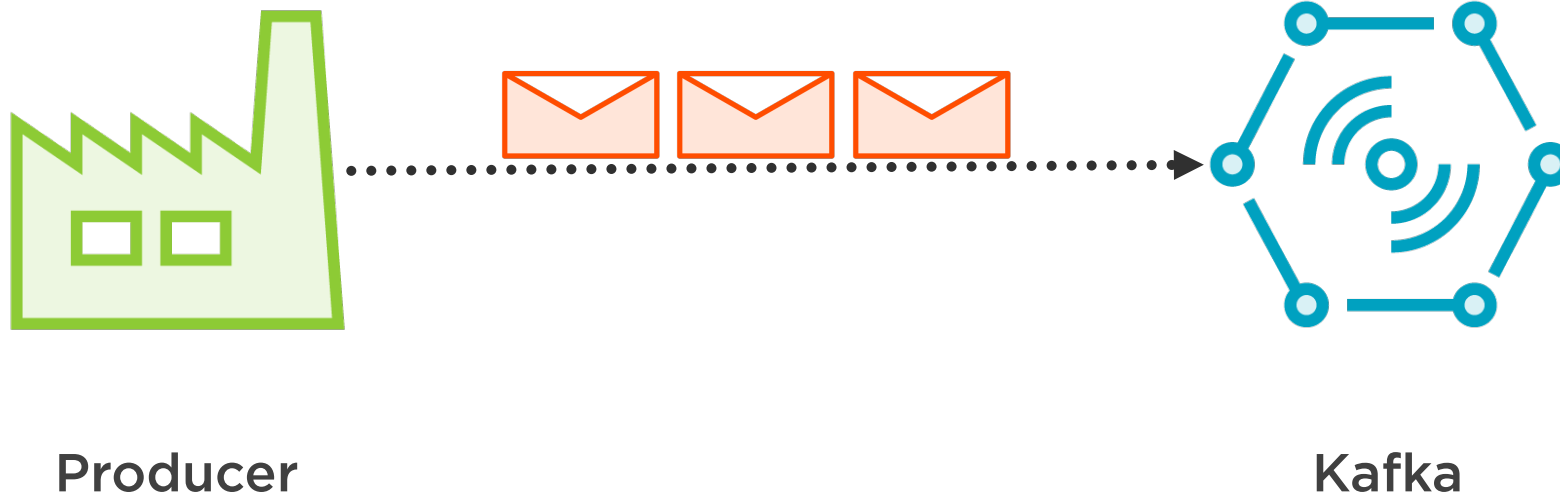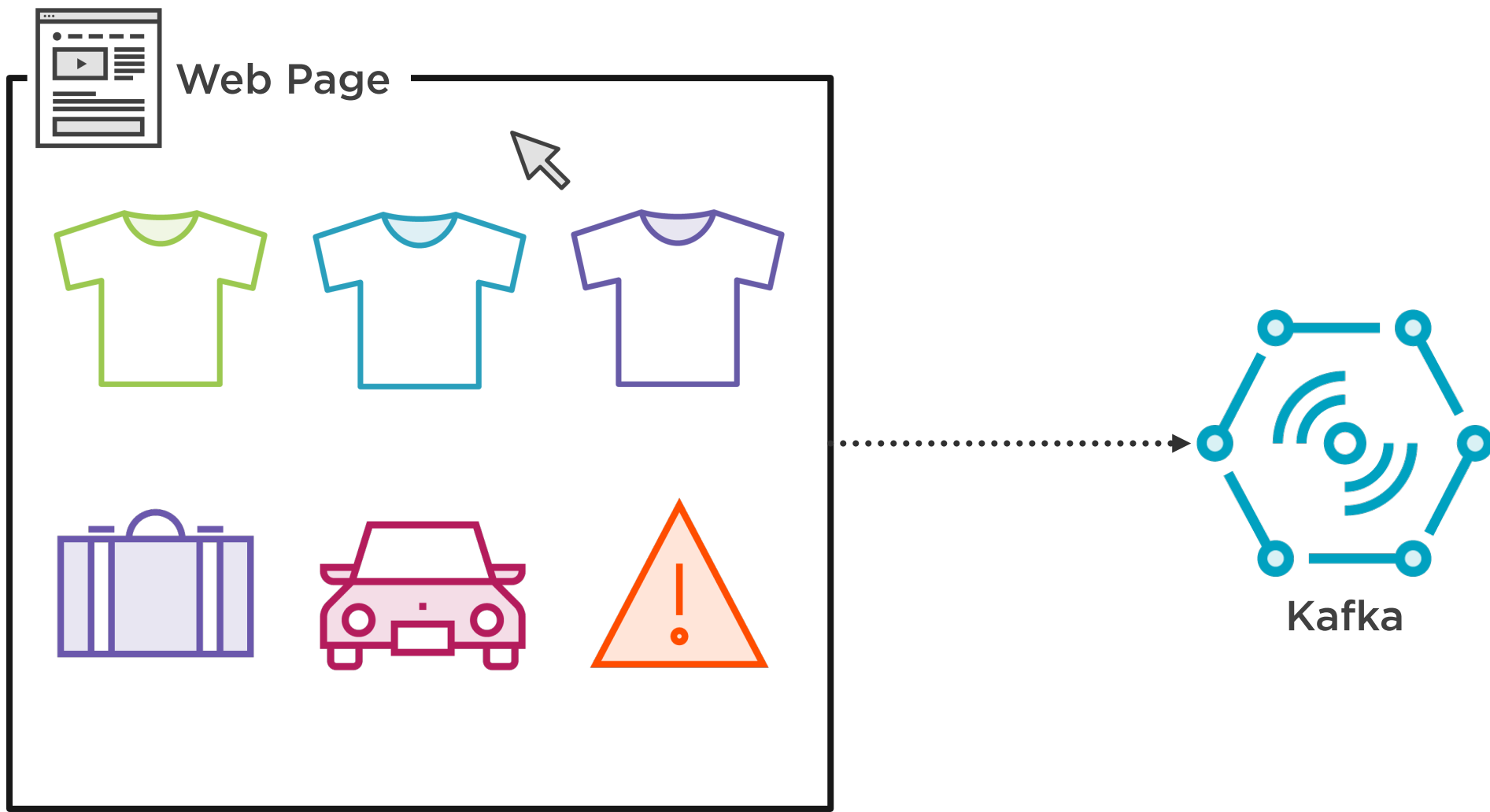
Kafka
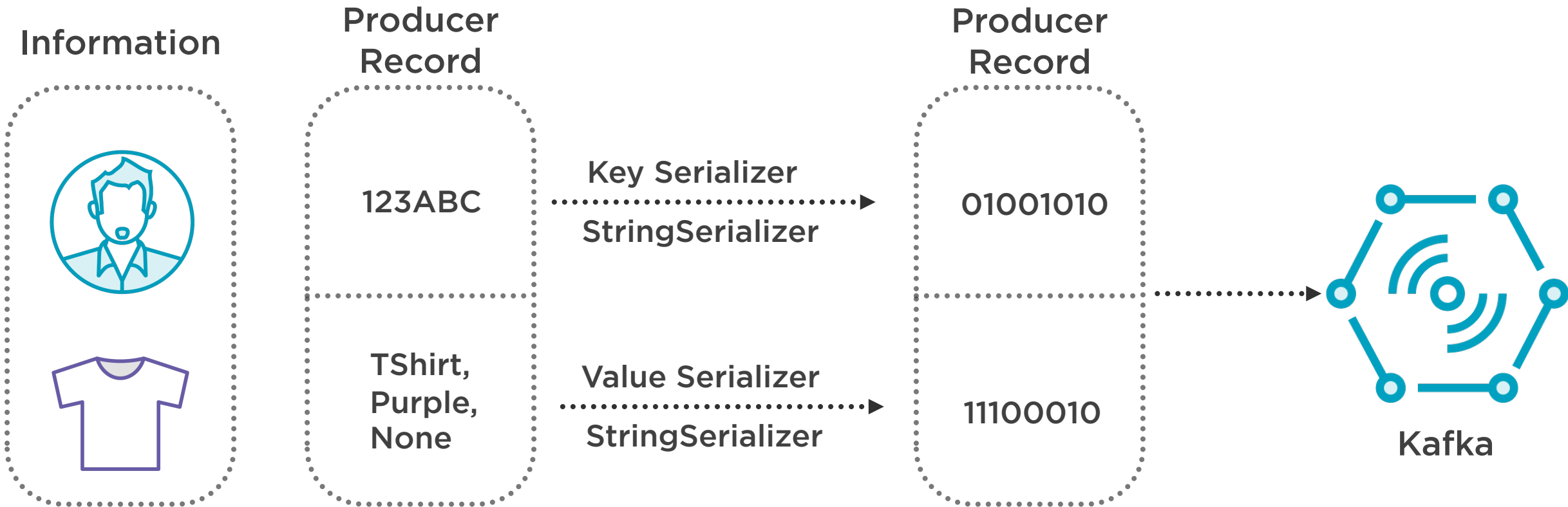
Demo

Basic producer application

Prerequisites:
- Java 8+
- Maven

Consumer

Kafka

Consumer

Consumer
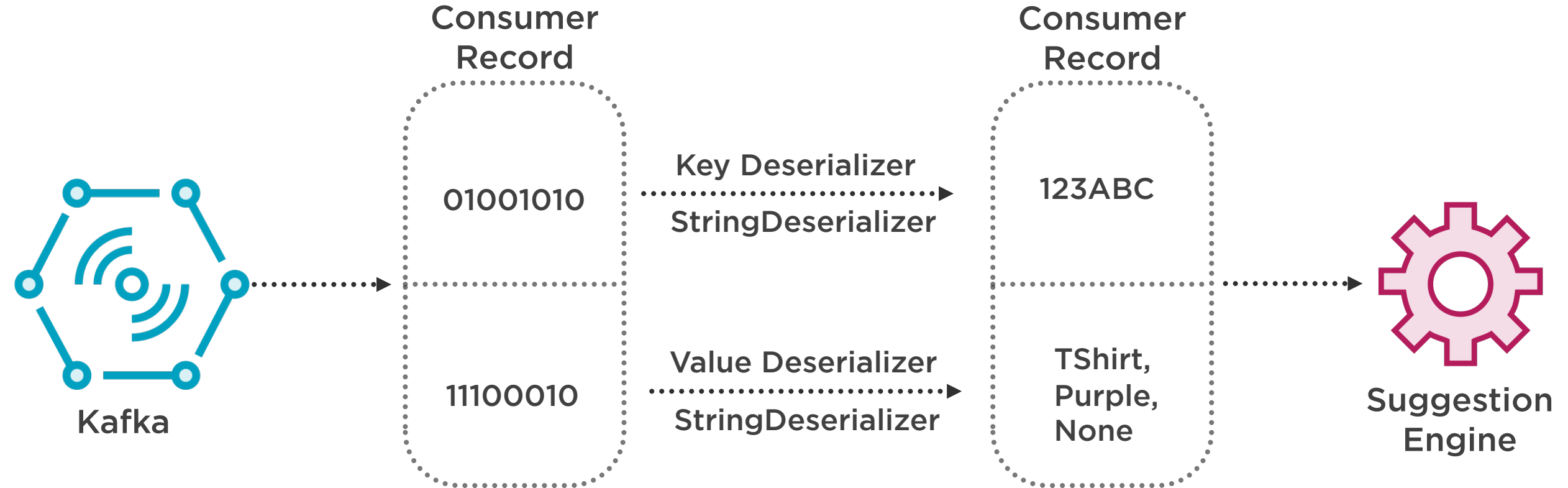
Demo

Basic consumer application

Summary

Why Kafka?

Kafka Architecture

Topics

Kafka Clients
- Producer
- Consumer