High-level Kafka Architecture

Paweł Kordek
DATA ENGINEER

@pawel_kordek https://kordek.github.io

Overview

- Minimal deployment
- Message flow
- Topics and partitions
- Basic settings

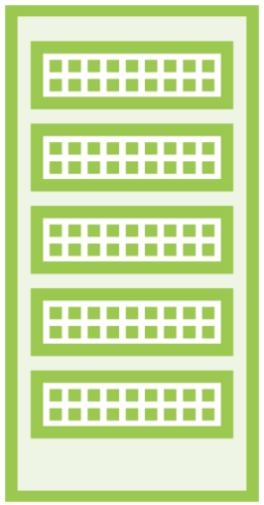
Demo

Minimal local Kafka deployment

Admin client

Producers and consumers

Minimal Kafka Deployment



ZooKeeper

Kafka broker

ZooKeeper

Coordination service

Used in distributed services

Consensus

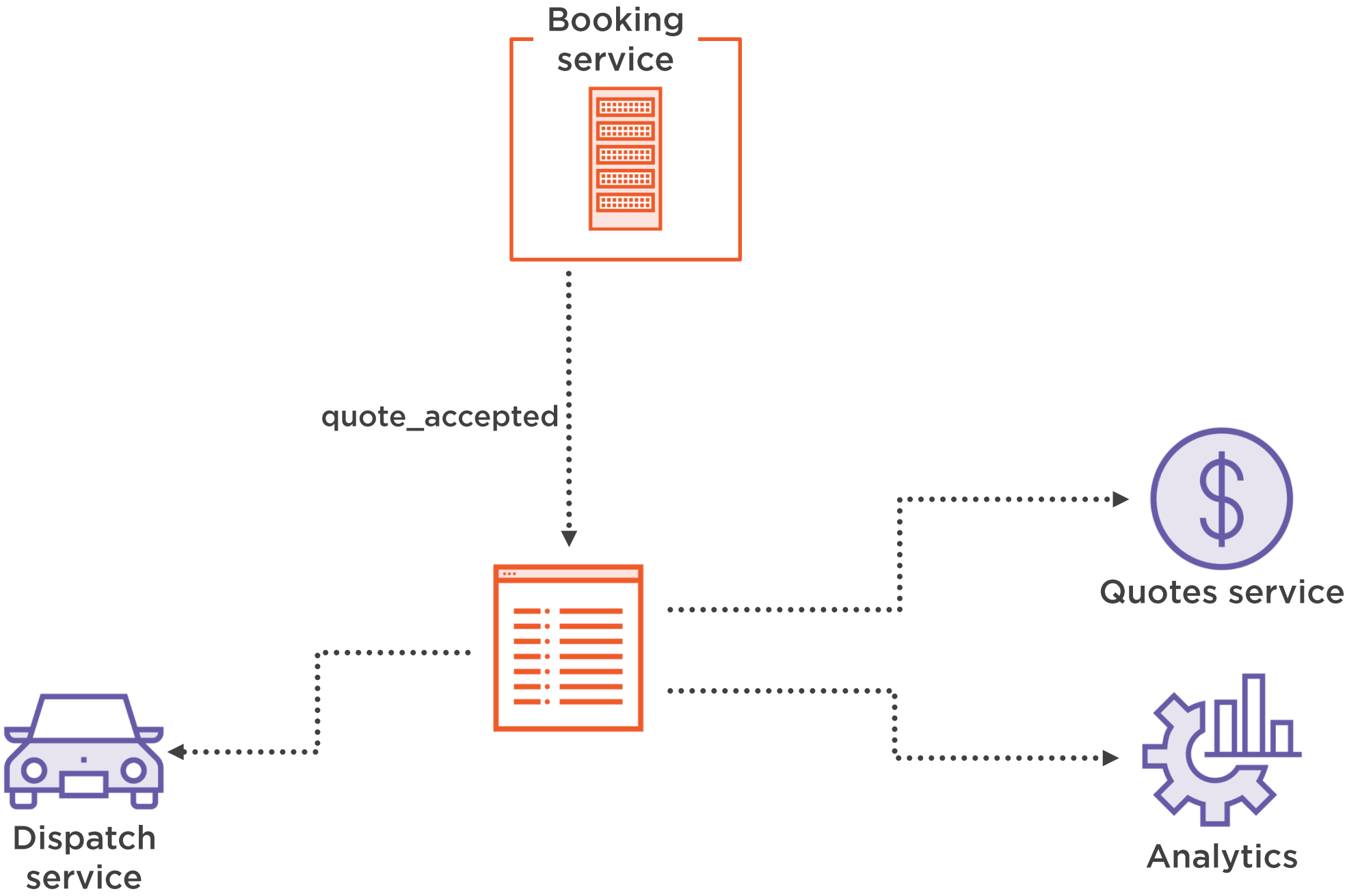
Kafka Broker

Manages topics and messages

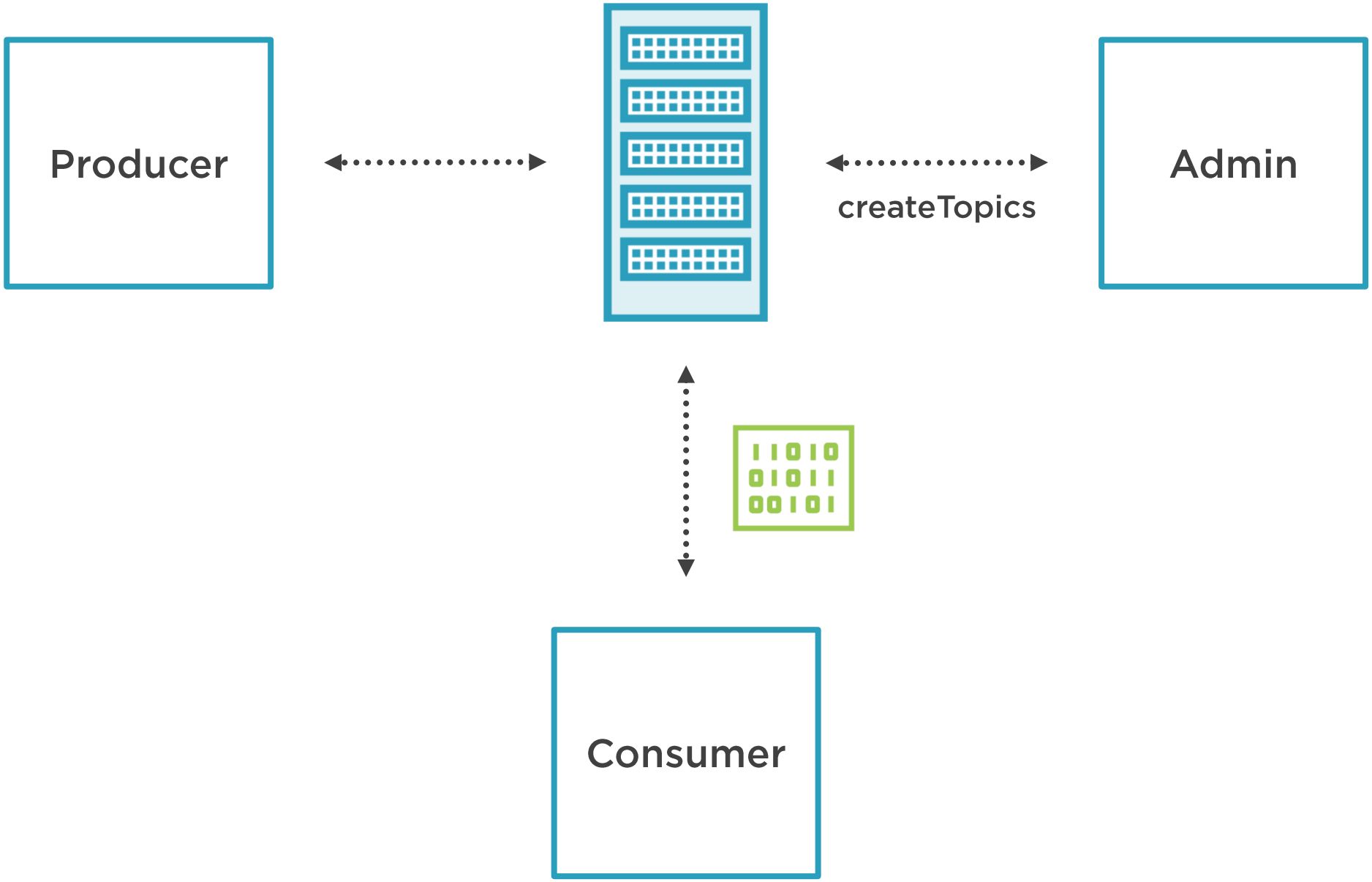
Clients connect to the broker

Core component

Interacting with Kafka Broker


Booking service
quote_accepted
Quotes service
Dispatch service
Analytics


Kafka
broker
quote-feedback


Producer
createTopics
Admin
Consumer

Client Libraries

docs.confluent.io/current/clients

Creating a New Topic

```
Admin admin = Admin.create(
        Map.of("bootstrap.servers", "localhost:9092")
);

NewTopic topic = new NewTopic("quote-feedback", ...)

admin.createTopics(List.of(topic))
```

Returns a KafkaFuture

Creating a New Topic

```
Admin admin = Admin.create(
        Map.of("bootstrap.servers", "localhost:9092")
);

NewTopic topic = new NewTopic("quote-feedback", ...)

admin.createTopics(List.of(topic)).get()
```

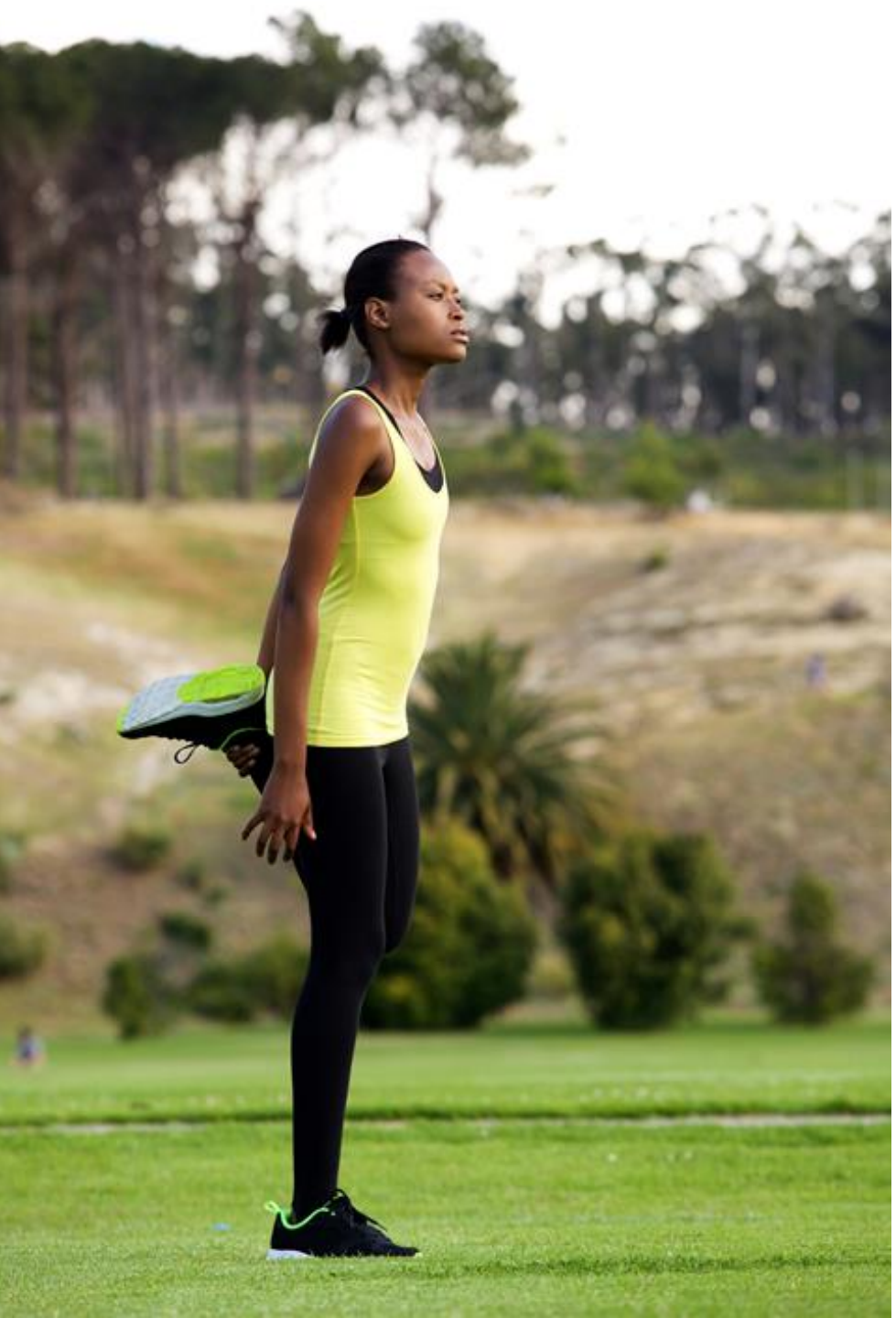
Creating a New Topic

```
Admin admin = Admin.create(
        Map.of("bootstrap.servers", "localhost:9092")
);

NewTopic topic = new NewTopic("quote-feedback", ...)

admin.createTopics(List.of(topic)).all().get()
```

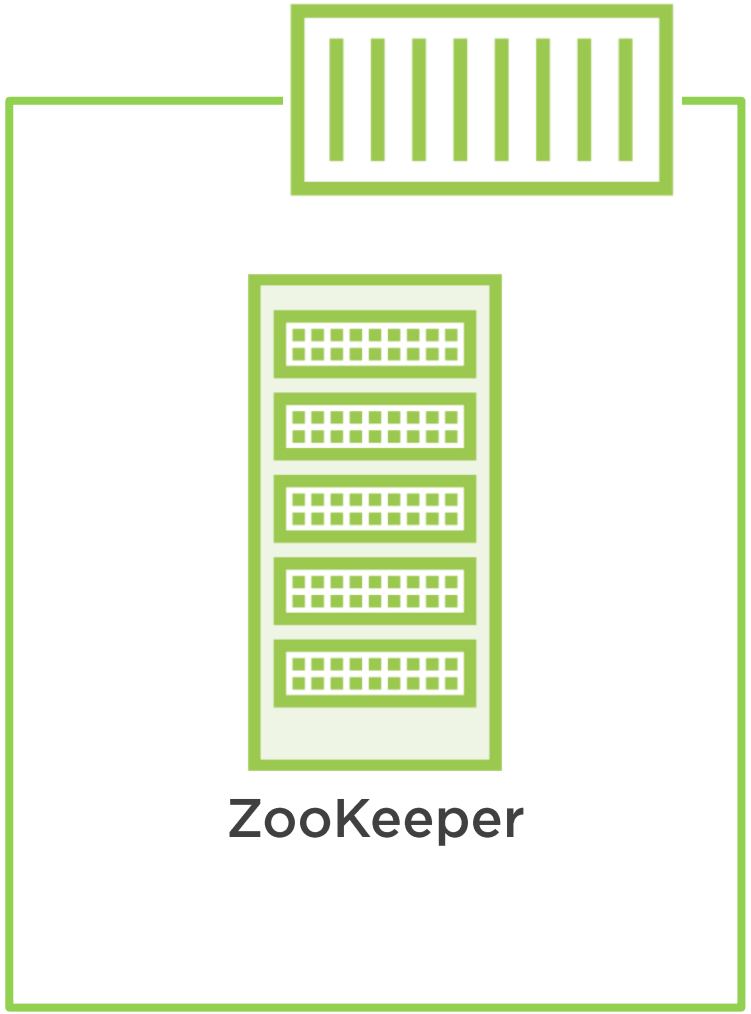
Running Kafka

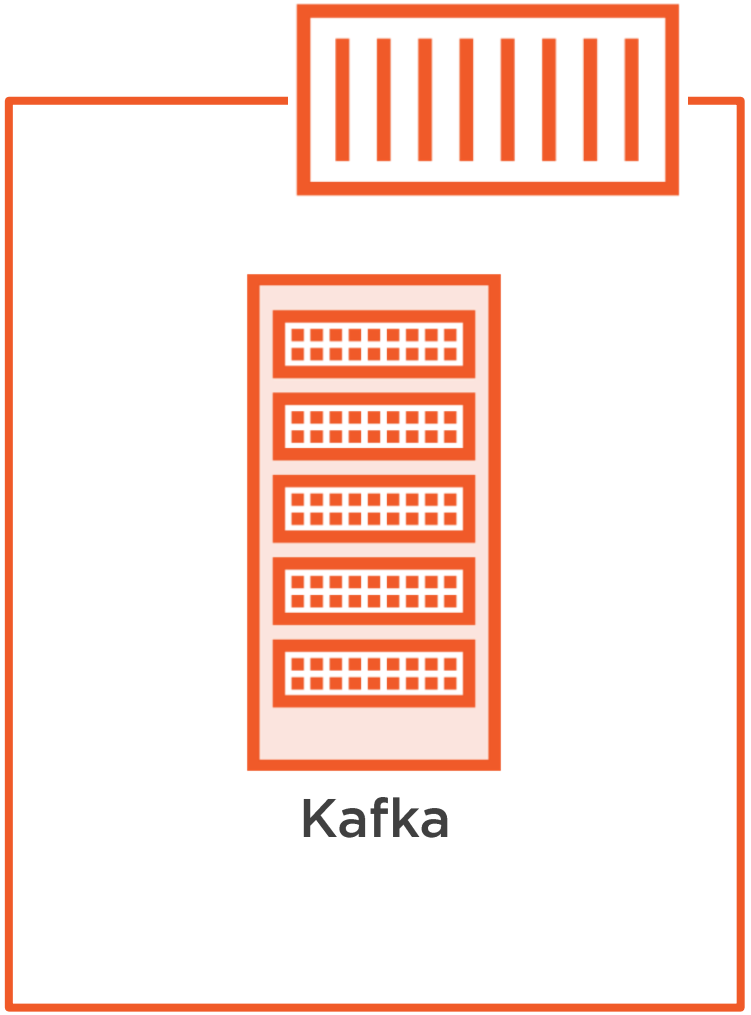


Make sure you have read the instructions

I will code in my preferred editor

But you will have no problem using whatever you like

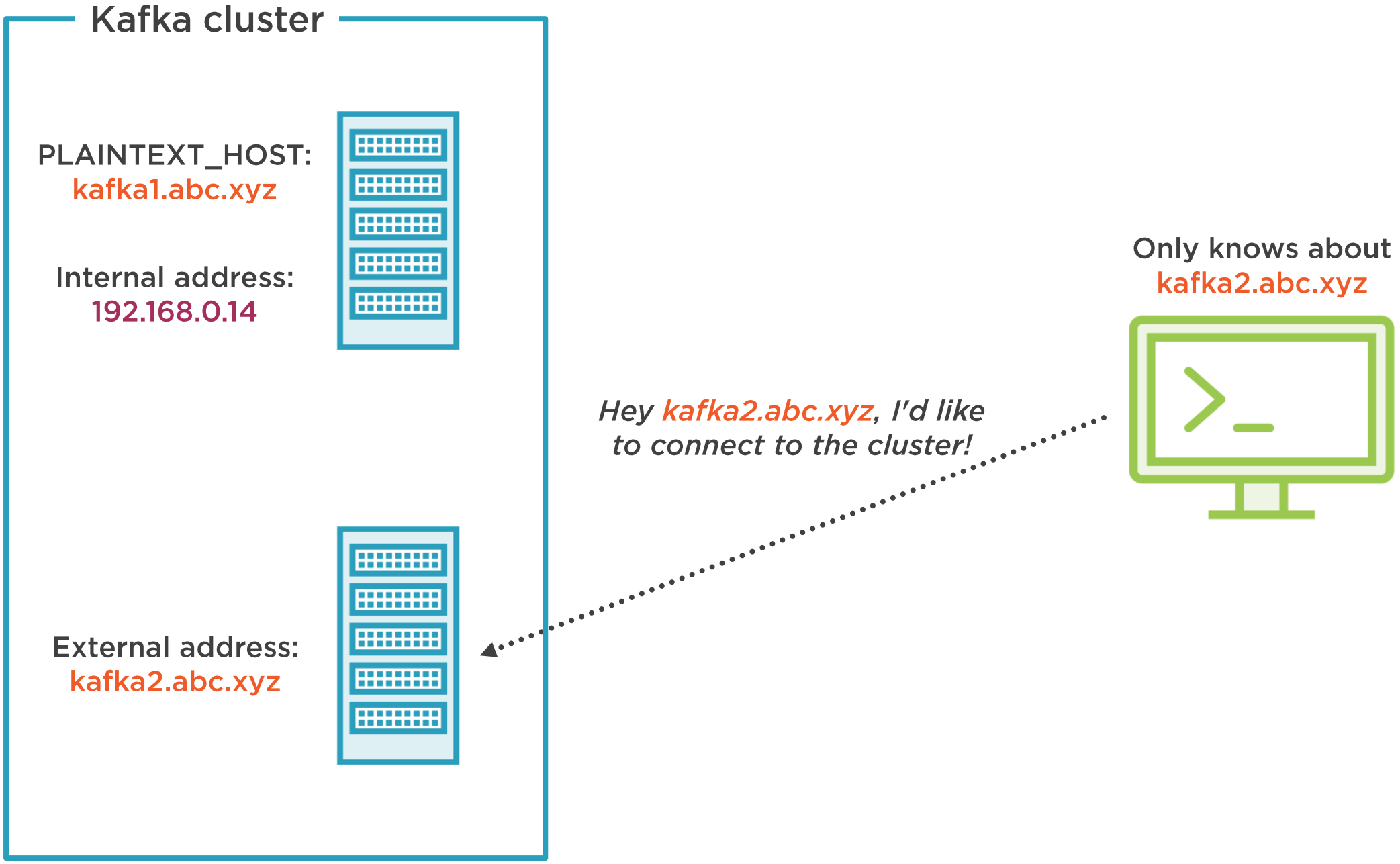

ZooKeeper

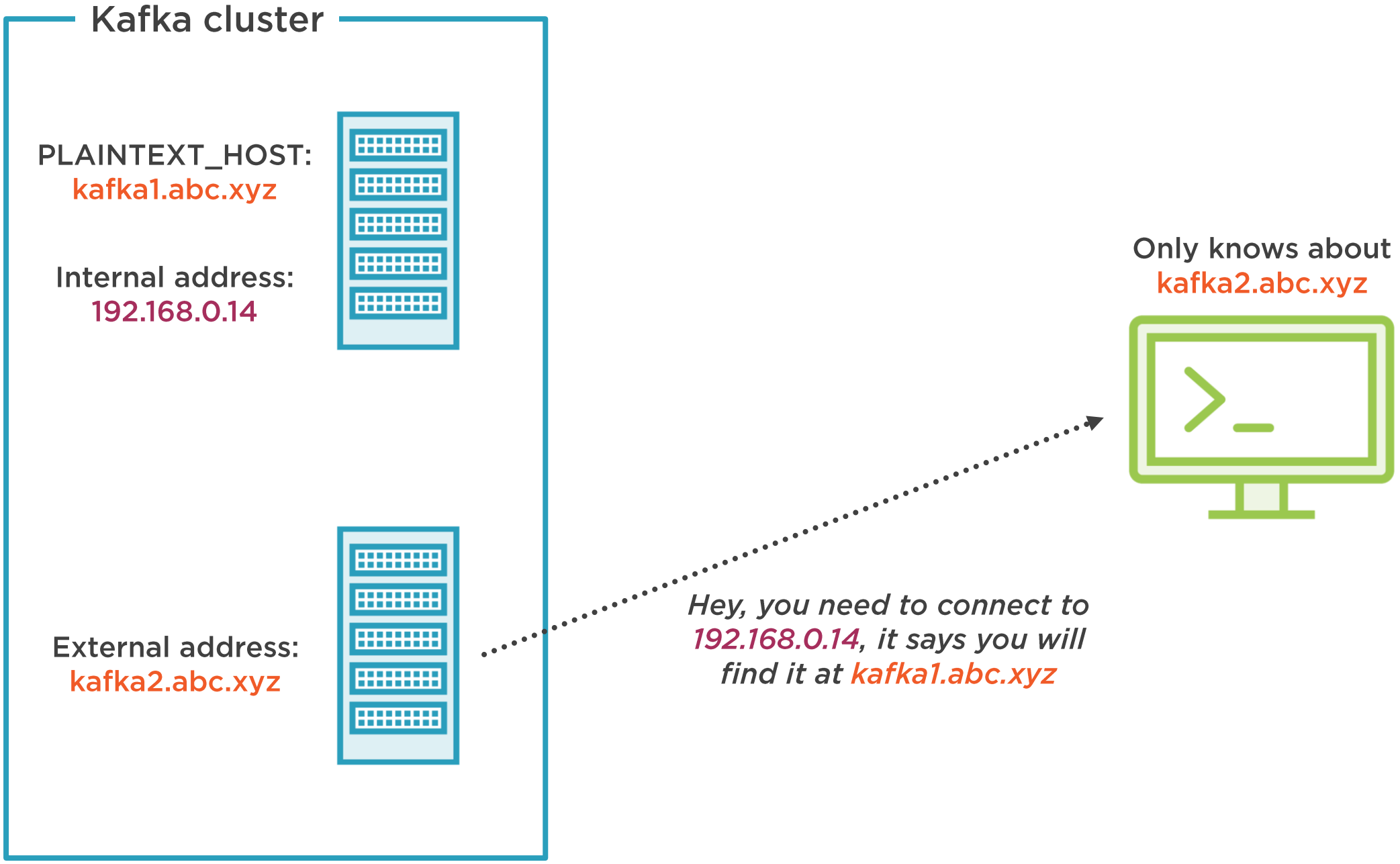

Kafka

Orchestrates individual containers

Deployments defined using YAML files

Docker Compose


Kafka cluster
PLAINTEXT_HOST:
kafka1.abc.xyz
Internal address:
192.168.0.14
External address:
kafka2.abc.xyz
Only knows about
kafka2.abc.xyz
Hey kafka2.abc.xyz, I'd like
to connect to the cluster!


Kafka cluster
PLAINTEXT_HOST:
kafka1.abc.xyz
Internal address:
192.168.0.14
External address:
kafka2.abc.xyz
Only knows about
kafka2.abc.xyz
Hey, you need to connect to
192.168.0.14, it says you will
find it at kafka1.abc.xyz

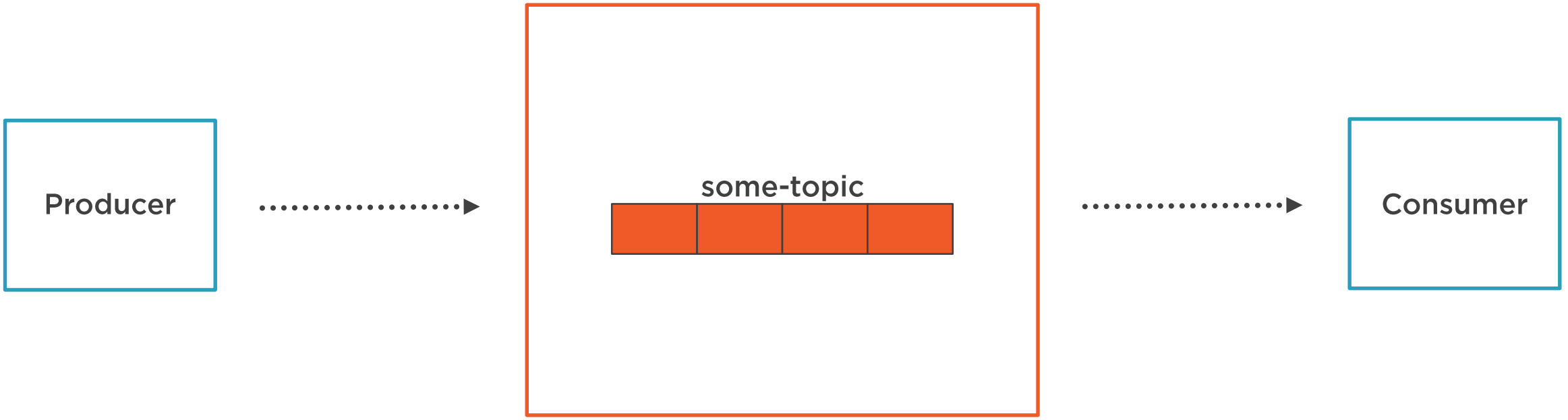
Partitions, Consumers, and Producers

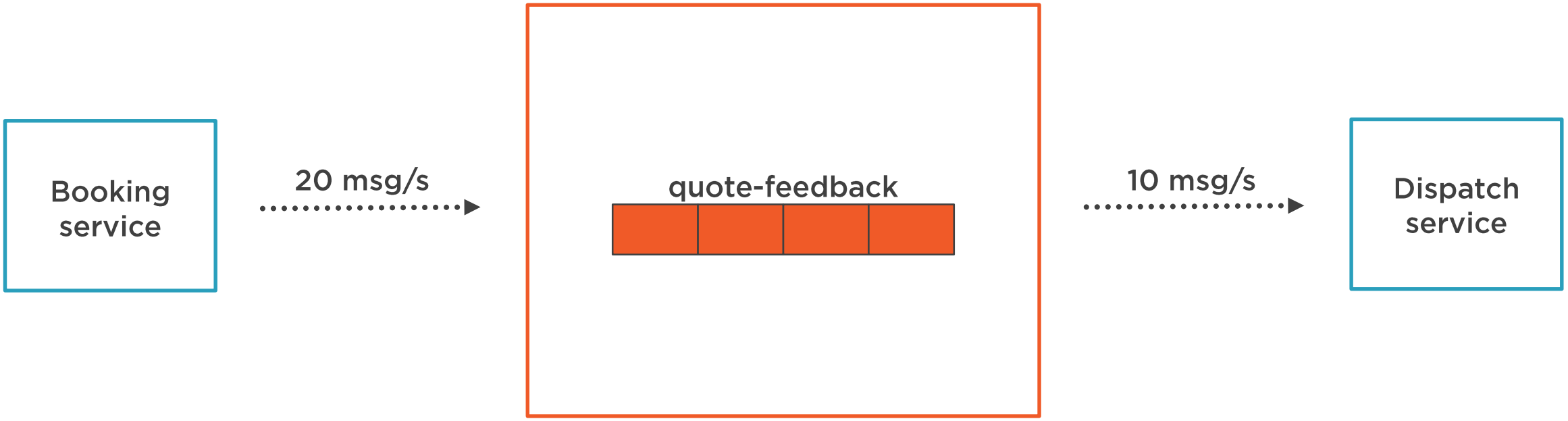
Partitions

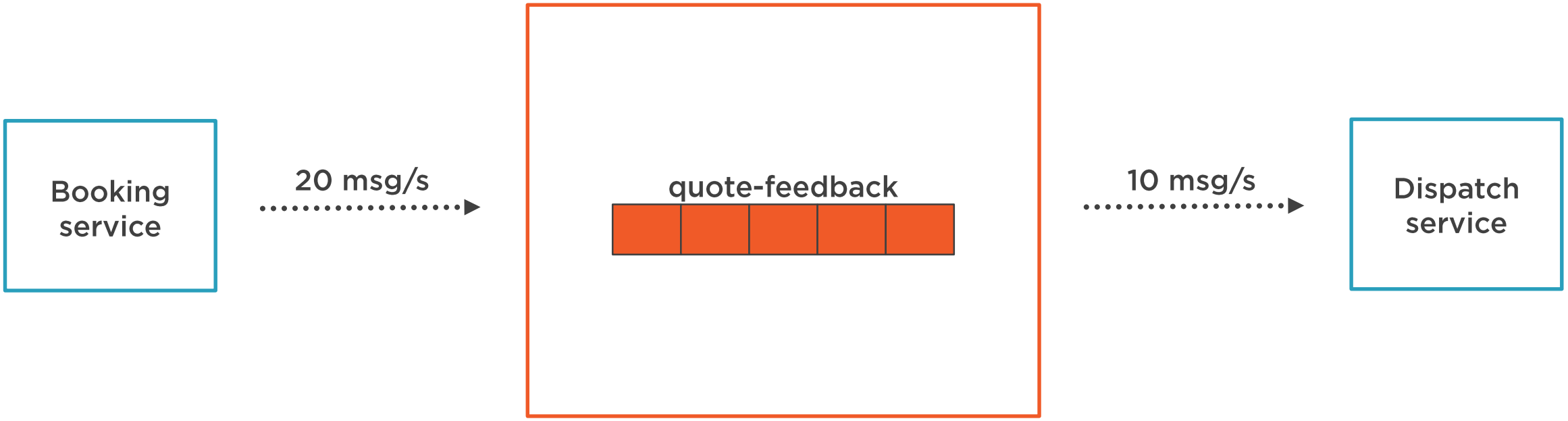
Logical and physical topics' component

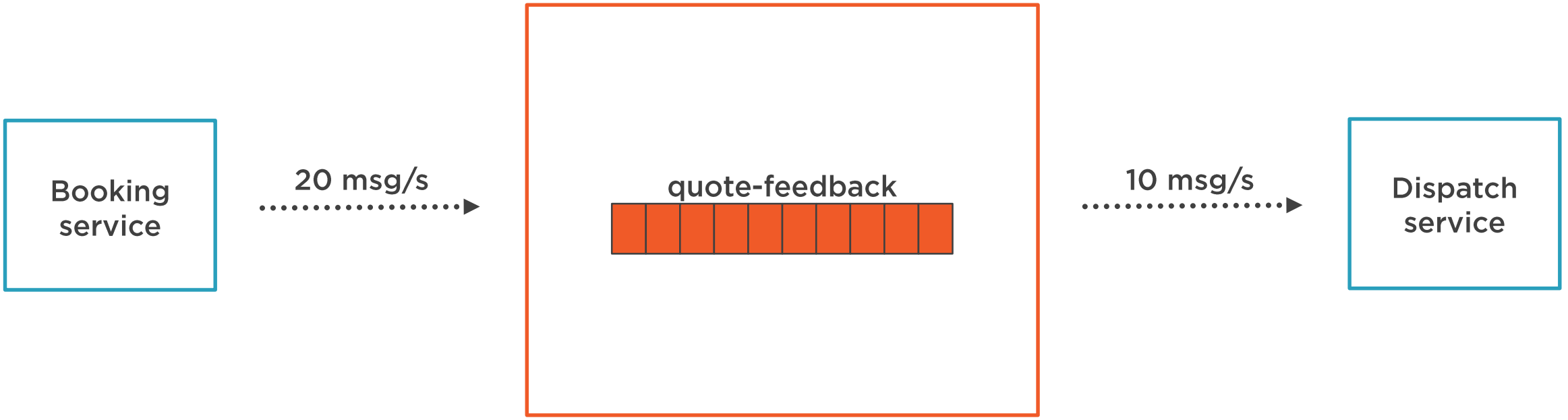
Each message belongs to one partition

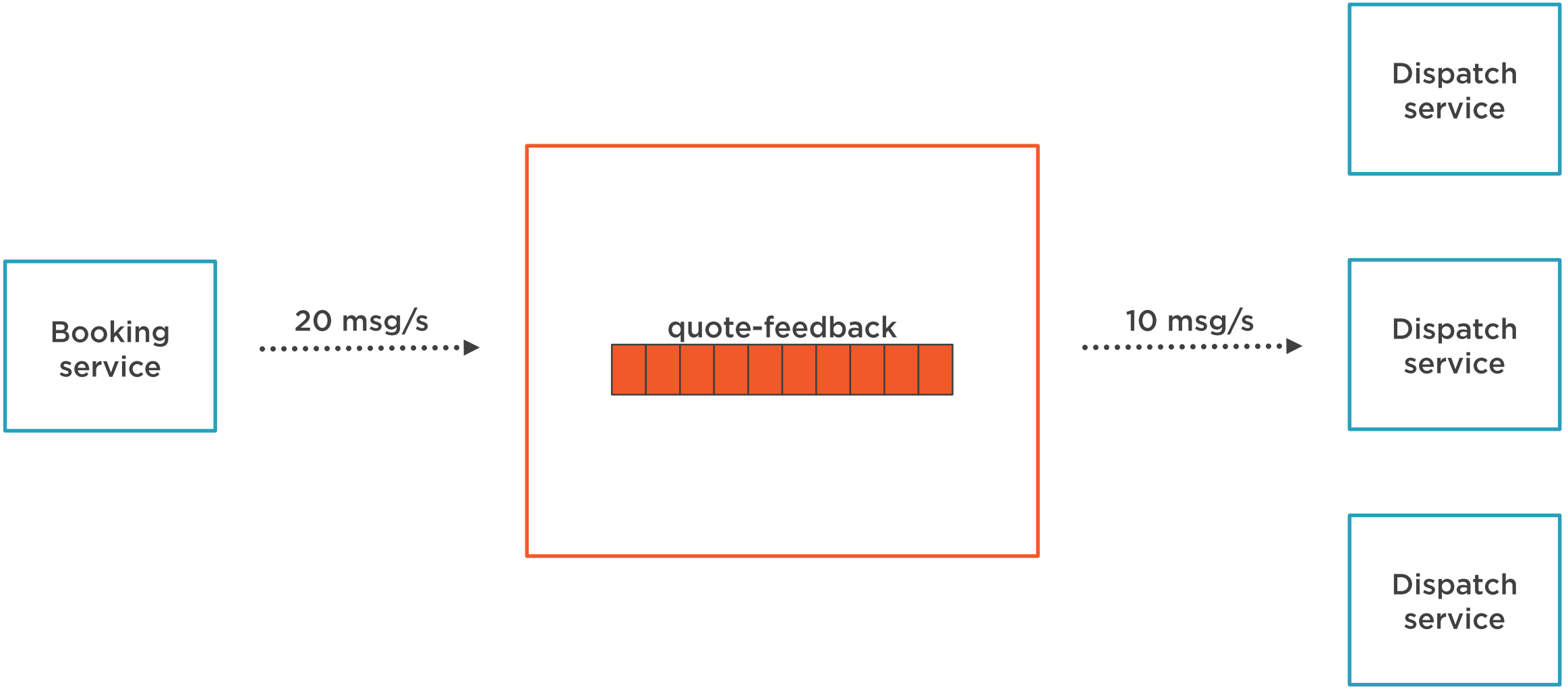
Key to scalability

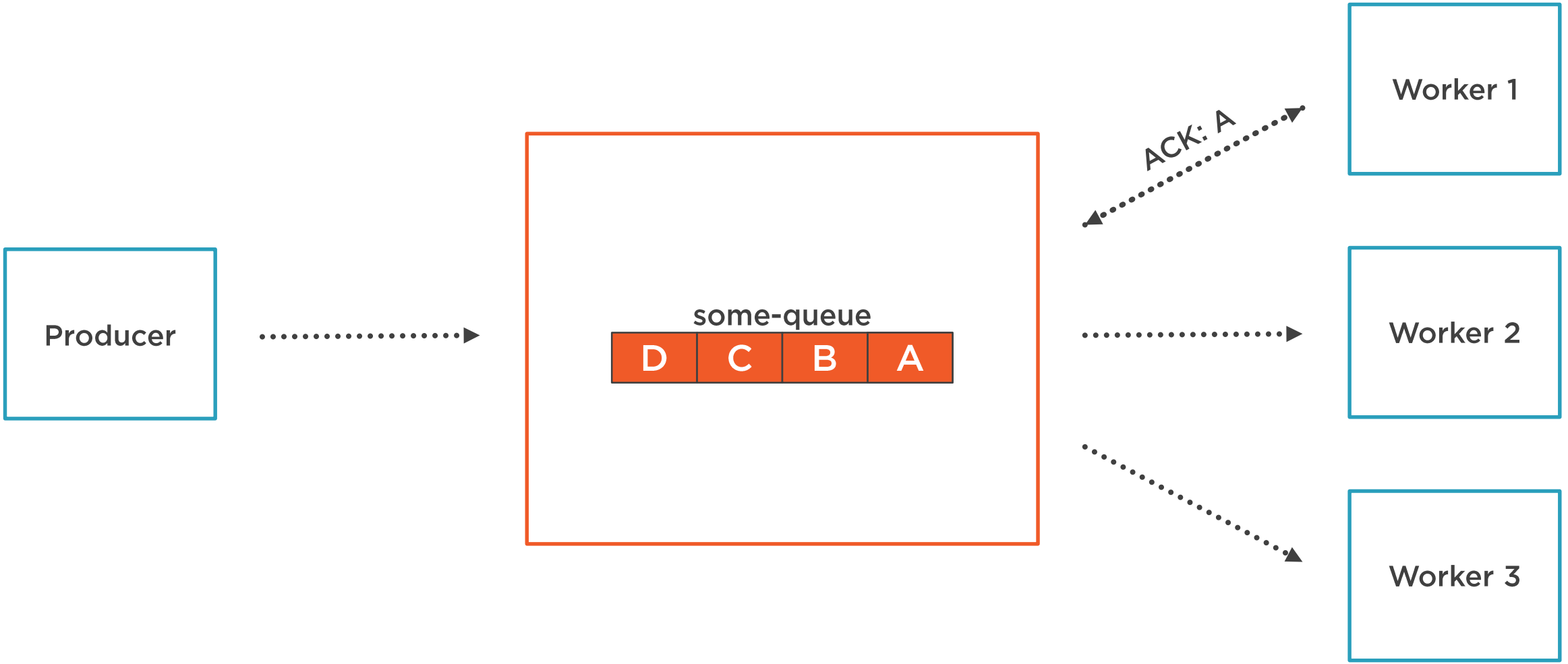

Producer
some-topic
Consumer

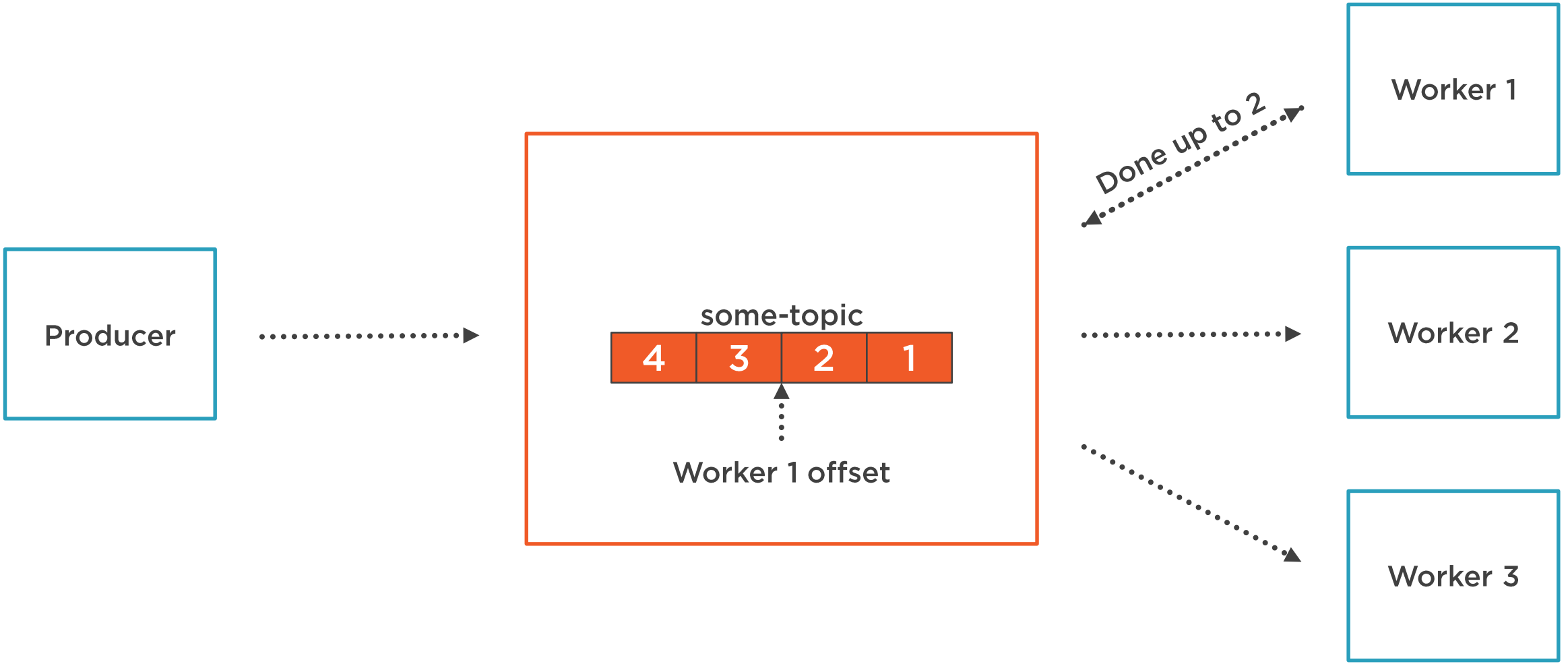

Booking service
20 msg/s
quote-feedback
10 msg/s
Dispatch service


Booking service
20 msg/s
quote-feedback
10 msg/s
Dispatch service

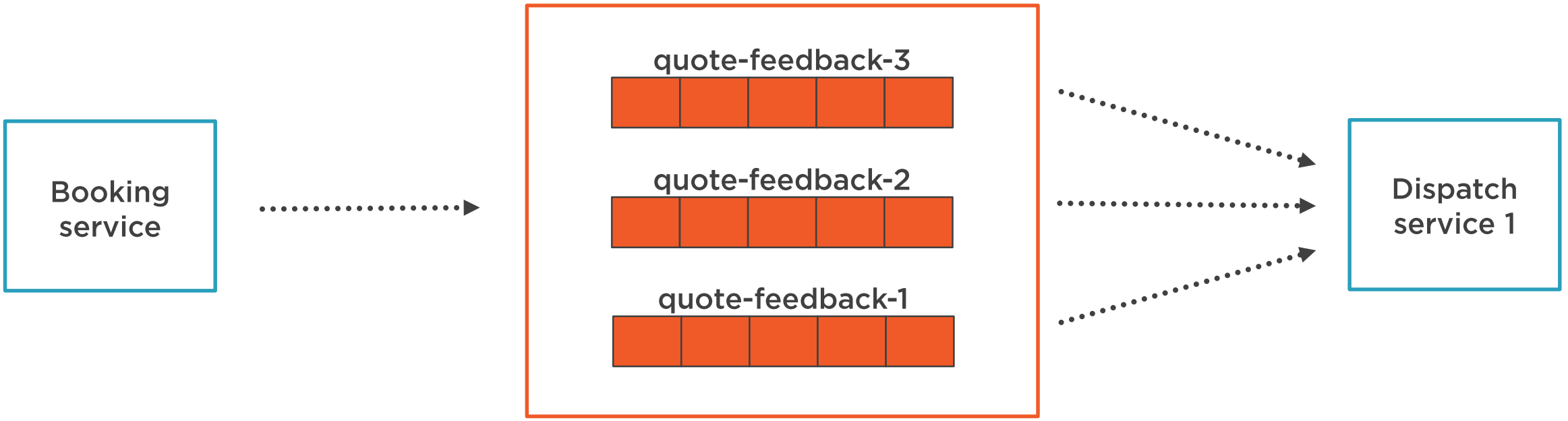

Booking service
20 msg/s
quote-feedback
10 msg/s
Dispatch service


Booking service
20 msg/s
quote-feedback
10 msg/s
Dispatch service
Dispatch service
Dispatch service

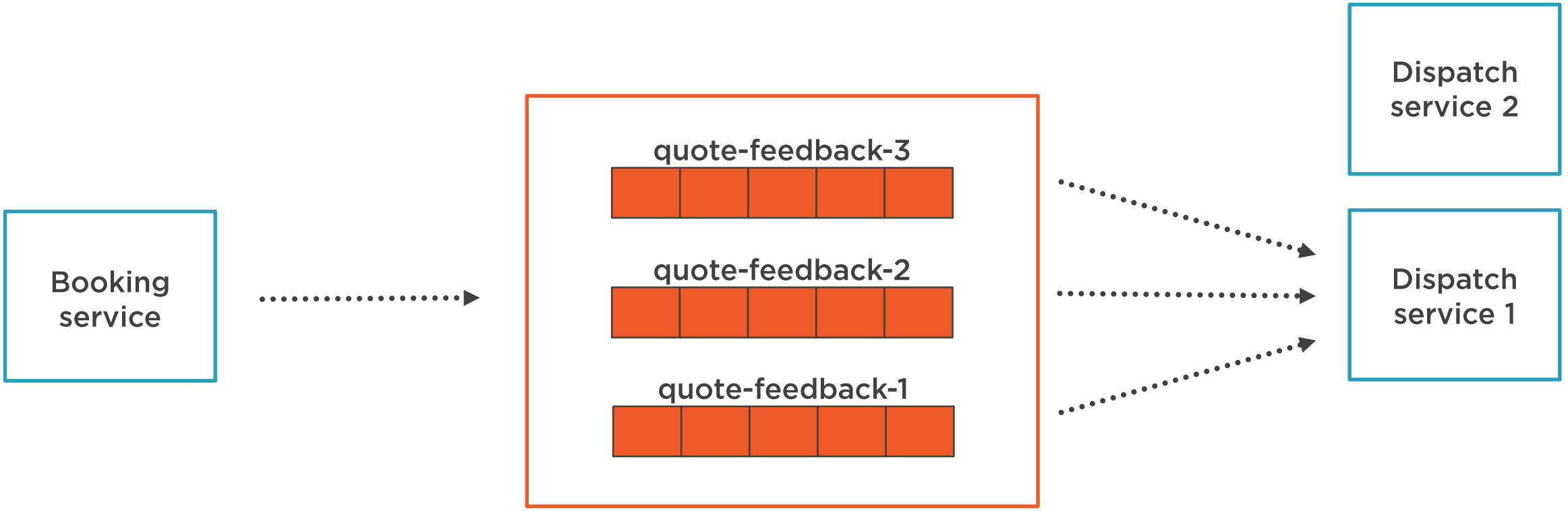

Producer
some-queue
D
C
B
A
ACK: A
Worker 1
Worker 2
Worker 3


Producer
some-topic
4
3
2
1
Worker 1 offset
Done up to 2
Worker 1
Worker 2
Worker 3

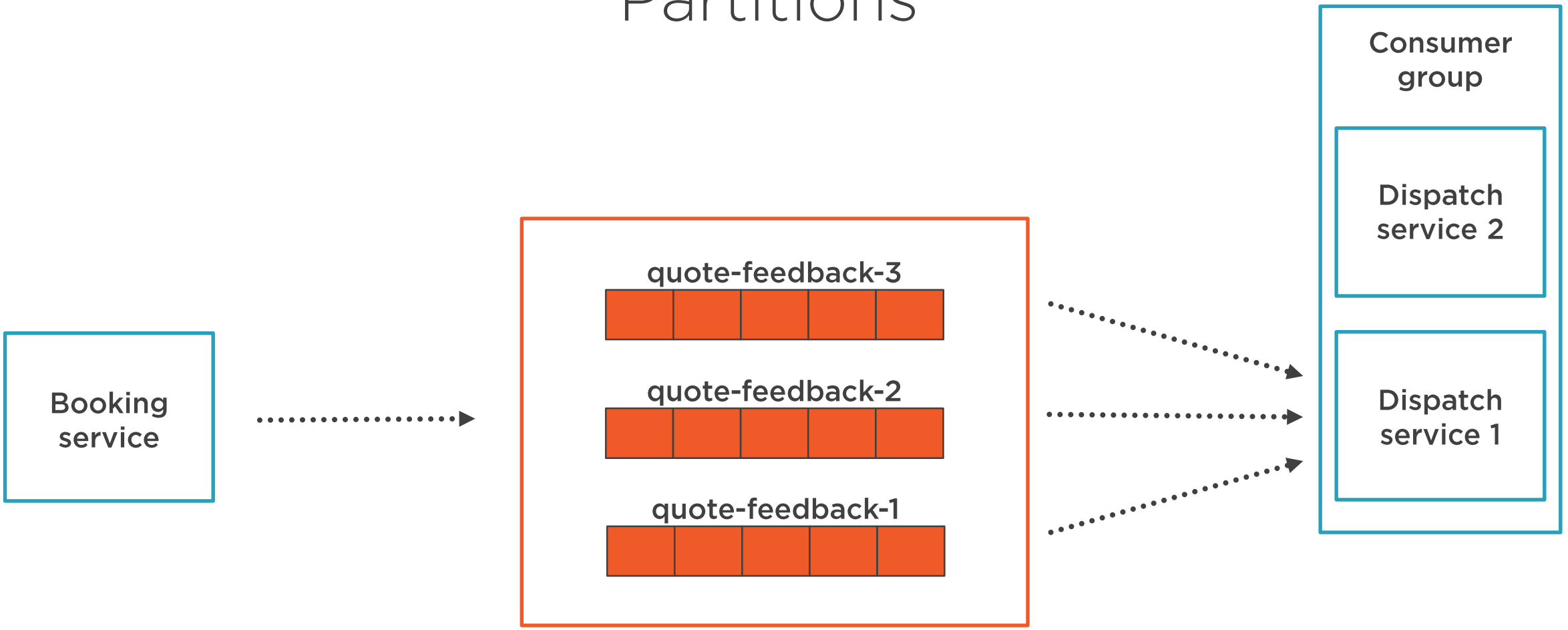
Partitions


Booking service
quote-feedback-3
quote-feedback-2
quote-feedback-1
Dispatch service 1

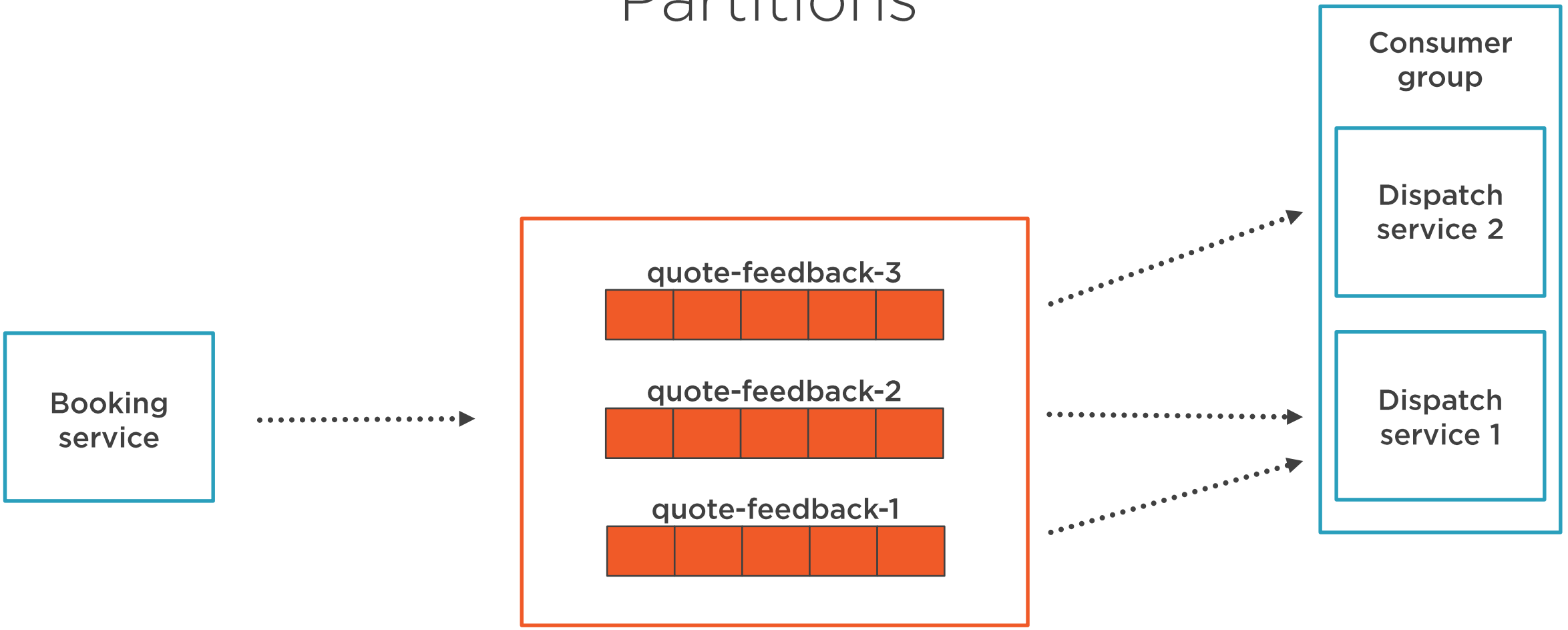
Partitions


Booking service
quote-feedback-3
quote-feedback-2
quote-feedback-1
Dispatch service 2
Dispatch service 1

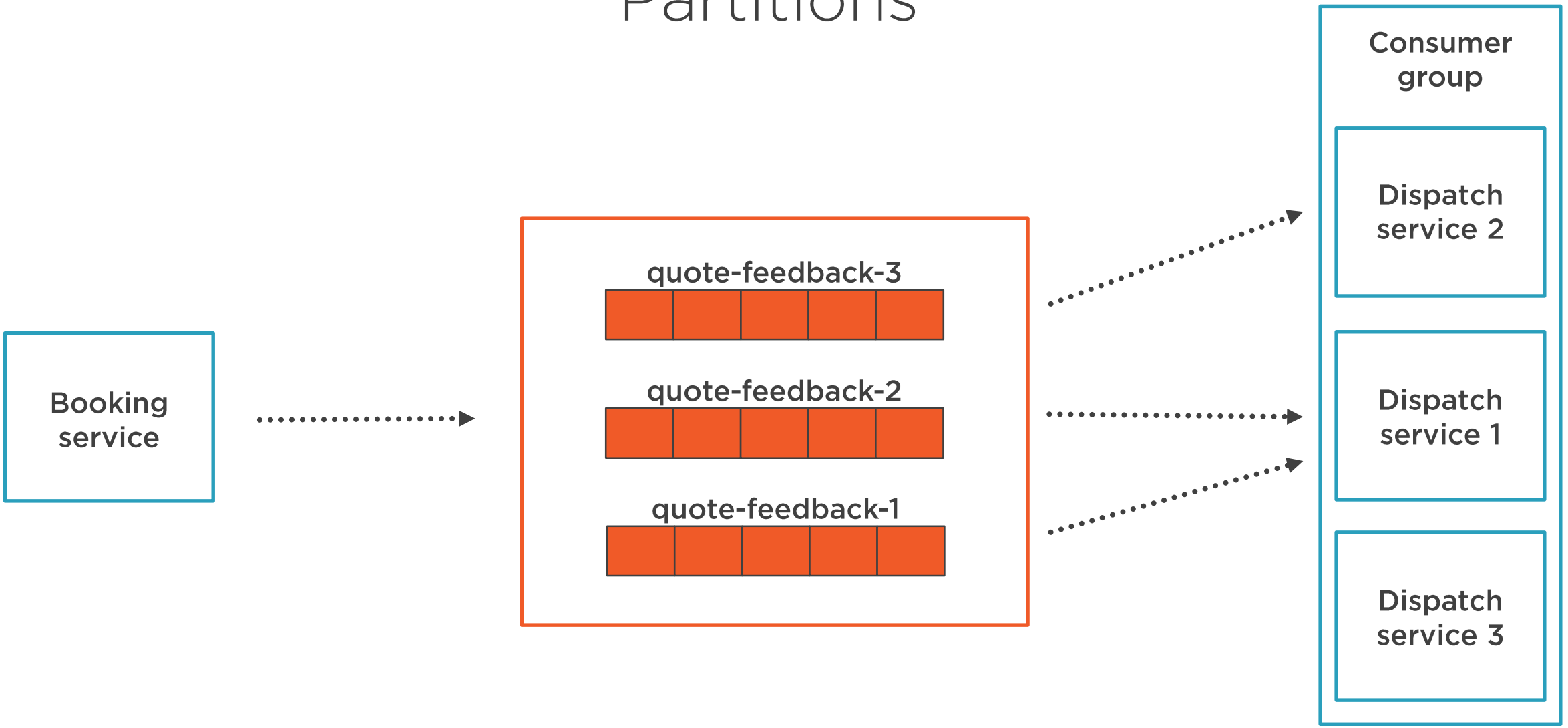
Partitions


Booking service
quote-feedback-3
quote-feedback-2
quote-feedback-1
Consumer group
Dispatch service 2
Dispatch service 1

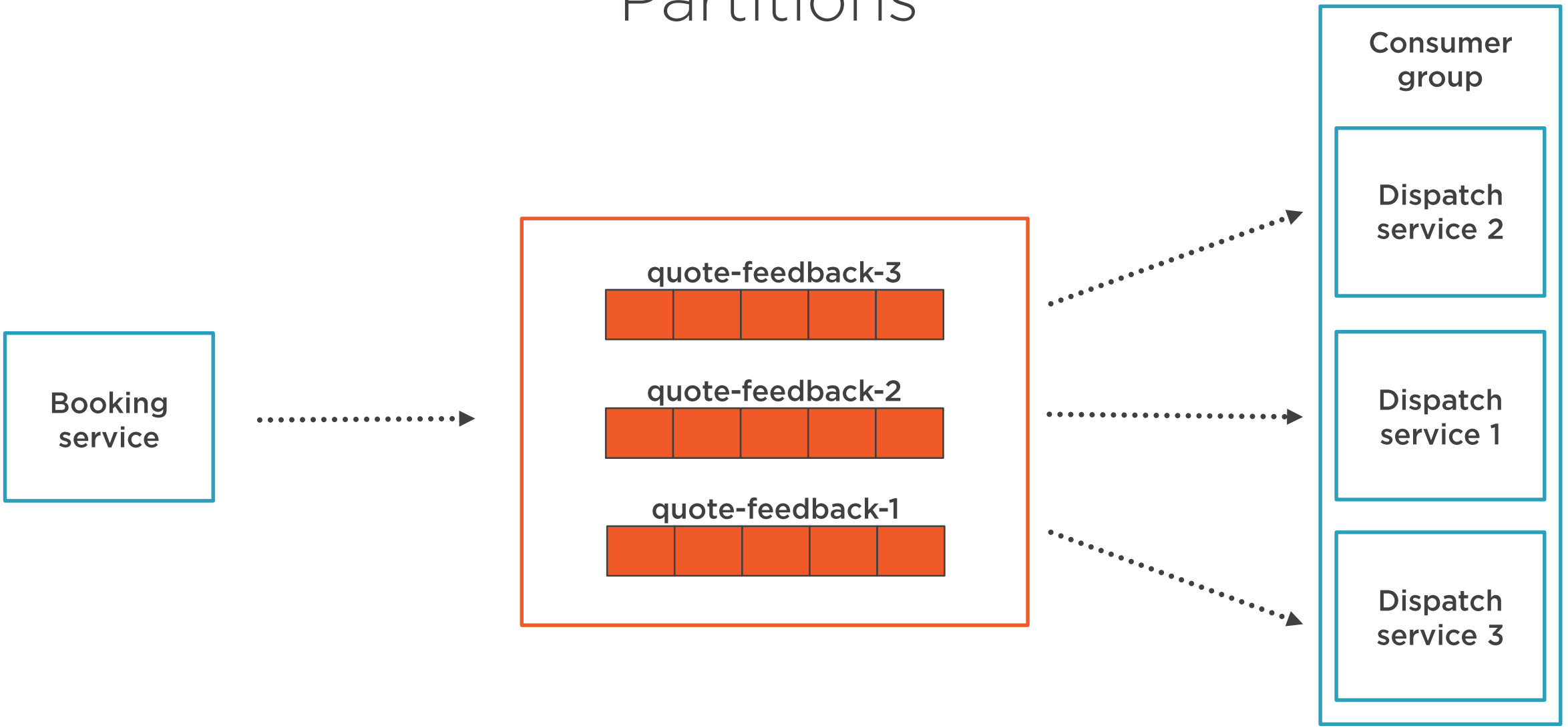
Partitions


Booking service
quote-feedback-3
quote-feedback-2
quote-feedback-1
Consumer group
Dispatch service 2
Dispatch service 1

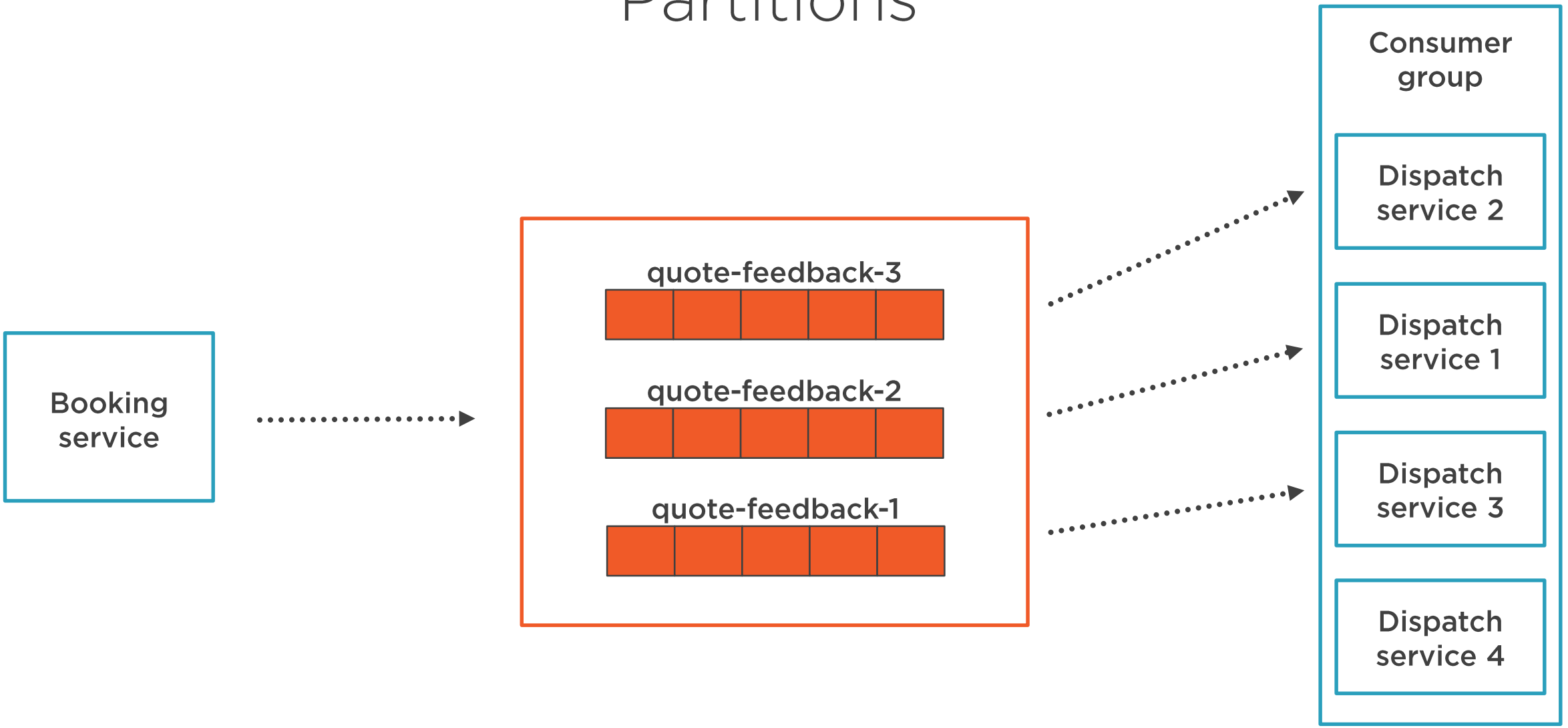
Partitions


Booking service
quote-feedback-3
quote-feedback-2
quote-feedback-1
Consumer group
Dispatch service 2
Dispatch service 1
Dispatch service 3

Partitions


Booking service
quote-feedback-3
quote-feedback-2
quote-feedback-1
Consumer group
Dispatch service 2
Dispatch service 1
Dispatch service 3

Partitions


Booking service
quote-feedback-3
quote-feedback-2
quote-feedback-1
Consumer group
Dispatch service 2
Dispatch service 1
Dispatch service 3
Dispatch service 4

Number of partitions should be chosen up front

End-to-end Demo

Consuming Messages

```
Properties props = new Properties();
props.setProperty("bootstrap.servers", "localhost:9092");
props.setProperty("group.id", "consumer-group-name")
props.setProperty("value.deserializer", ...);

KafkaConsumer<String, String> = new KafkaConsumer<>(props);
consumer.subscribe(List.of("quote-feedback"));

while(true) {
    var records = consumer.poll(...);
    for(var record : records) {
        System.out.println(record.value());
    }
}
```

Producing Messages

```
Properties props = new Properties();
props.setProperty("bootstrap.servers", "localhost:9092");
props.setProperty("value.serializer", ...);

KafkaProducer<String, String> = new KafkaProducer<>(props);

ProducerRecord<String, String> r = new ProducerRecord("topic", "value");

producer.send();
```

Summary

- Running Kafka
- Creating topics
- Understanding partitions
- Running end-to-end scenario