Deploying a Kafka Cluster

MESSAGE BROKERS

Paweł Kordek
DATA ENGINEER

@pawel_kordek   https://kordek.github.io

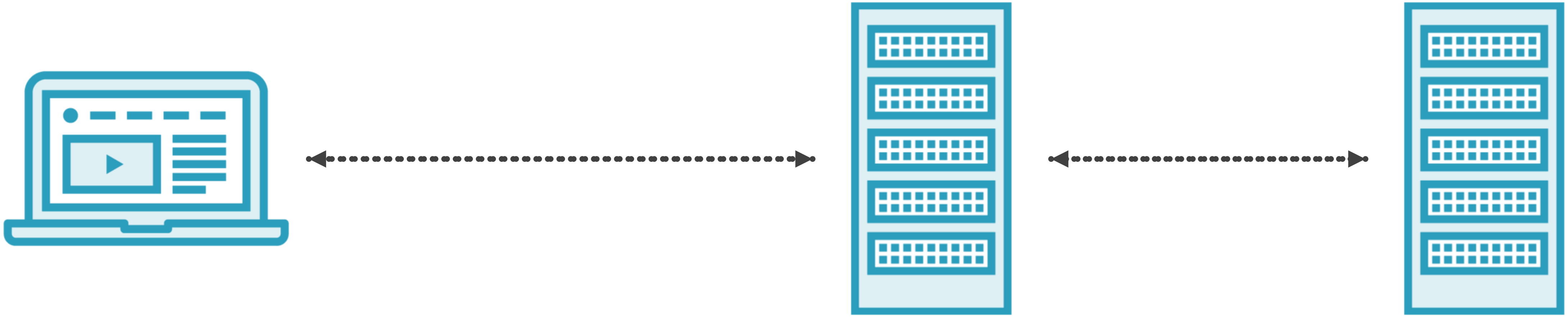
A Broker

Receives messages

Stores them for some time

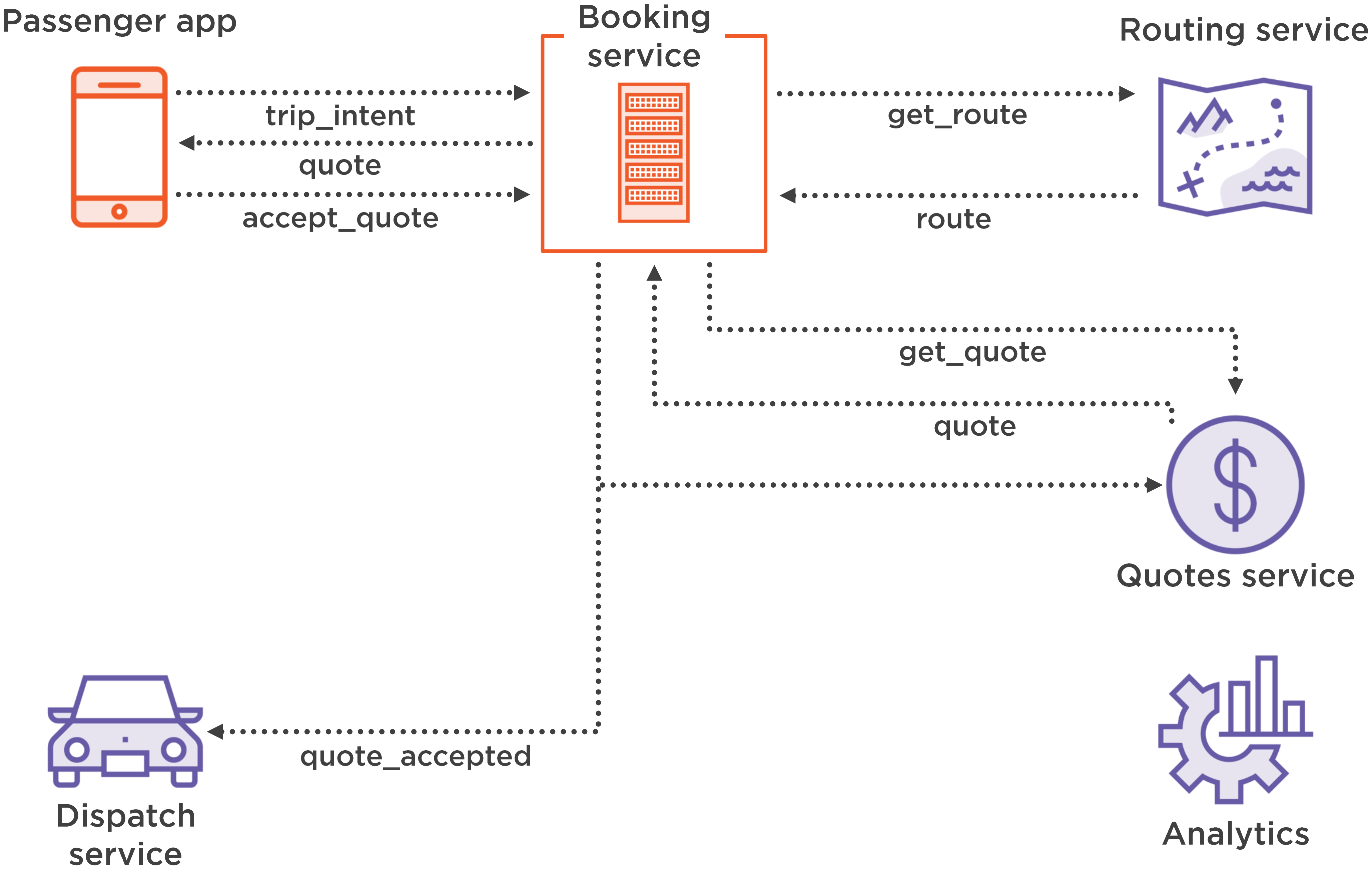
Passes them to other applications

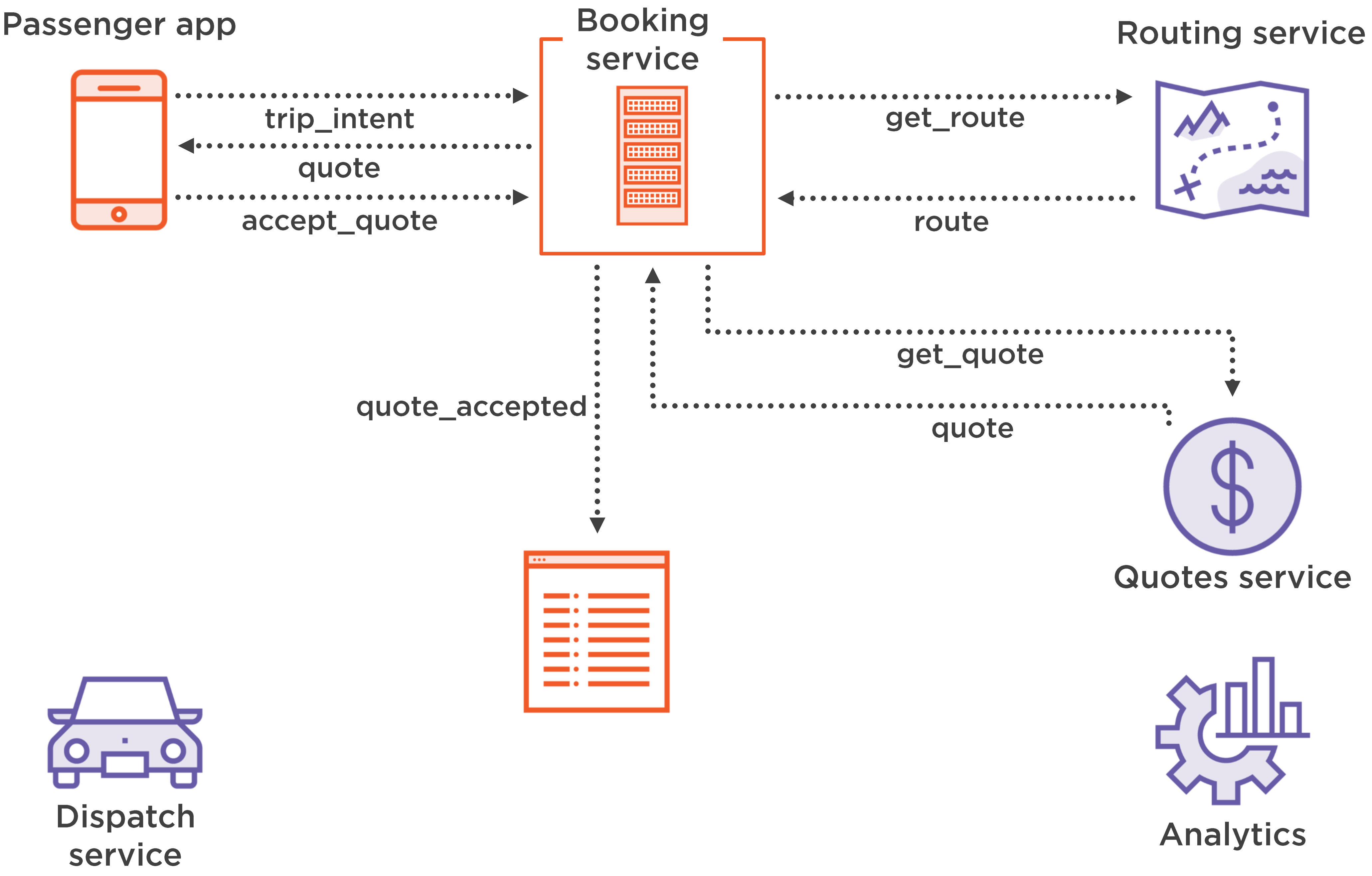
Why would you need one?

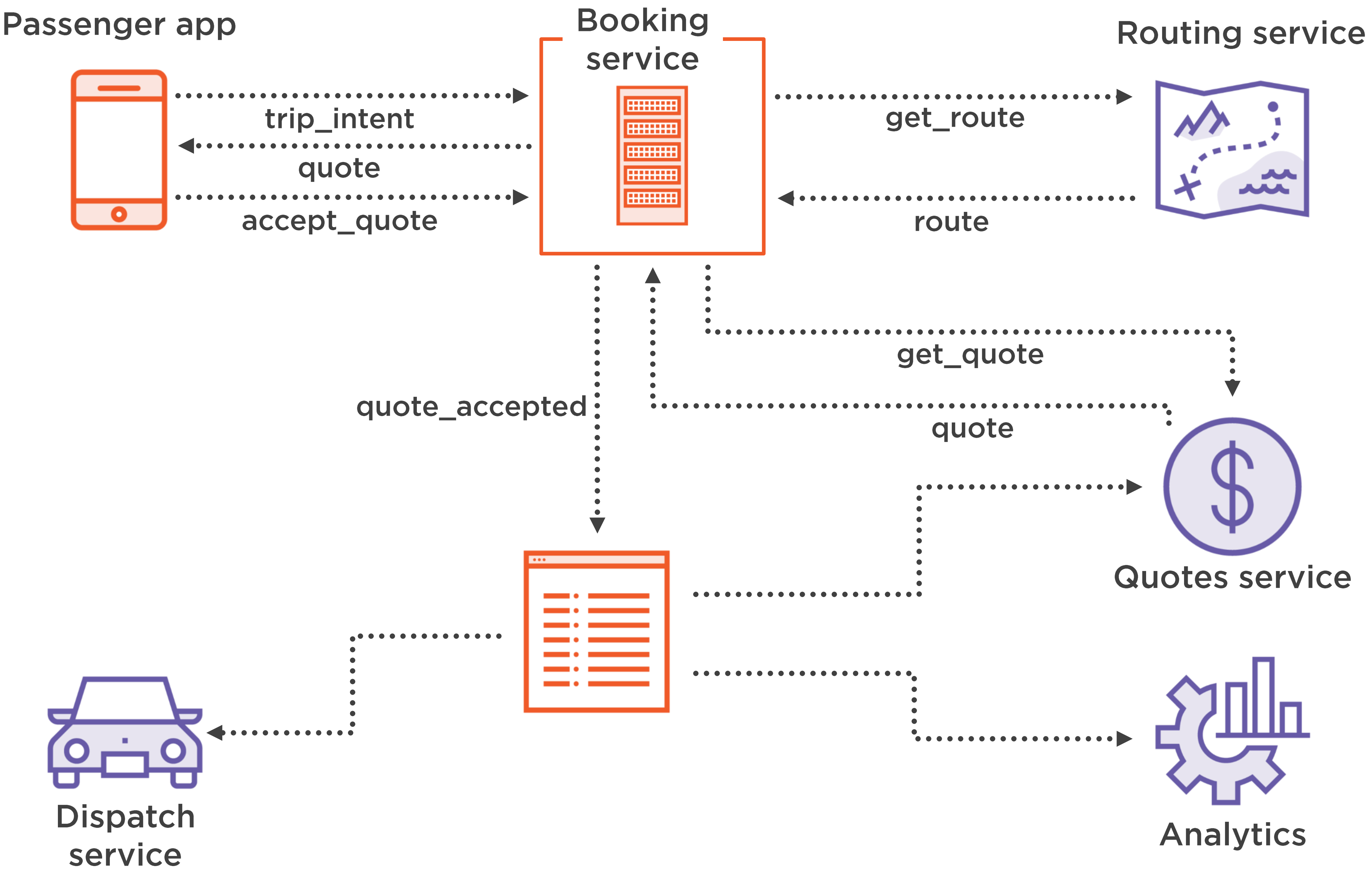


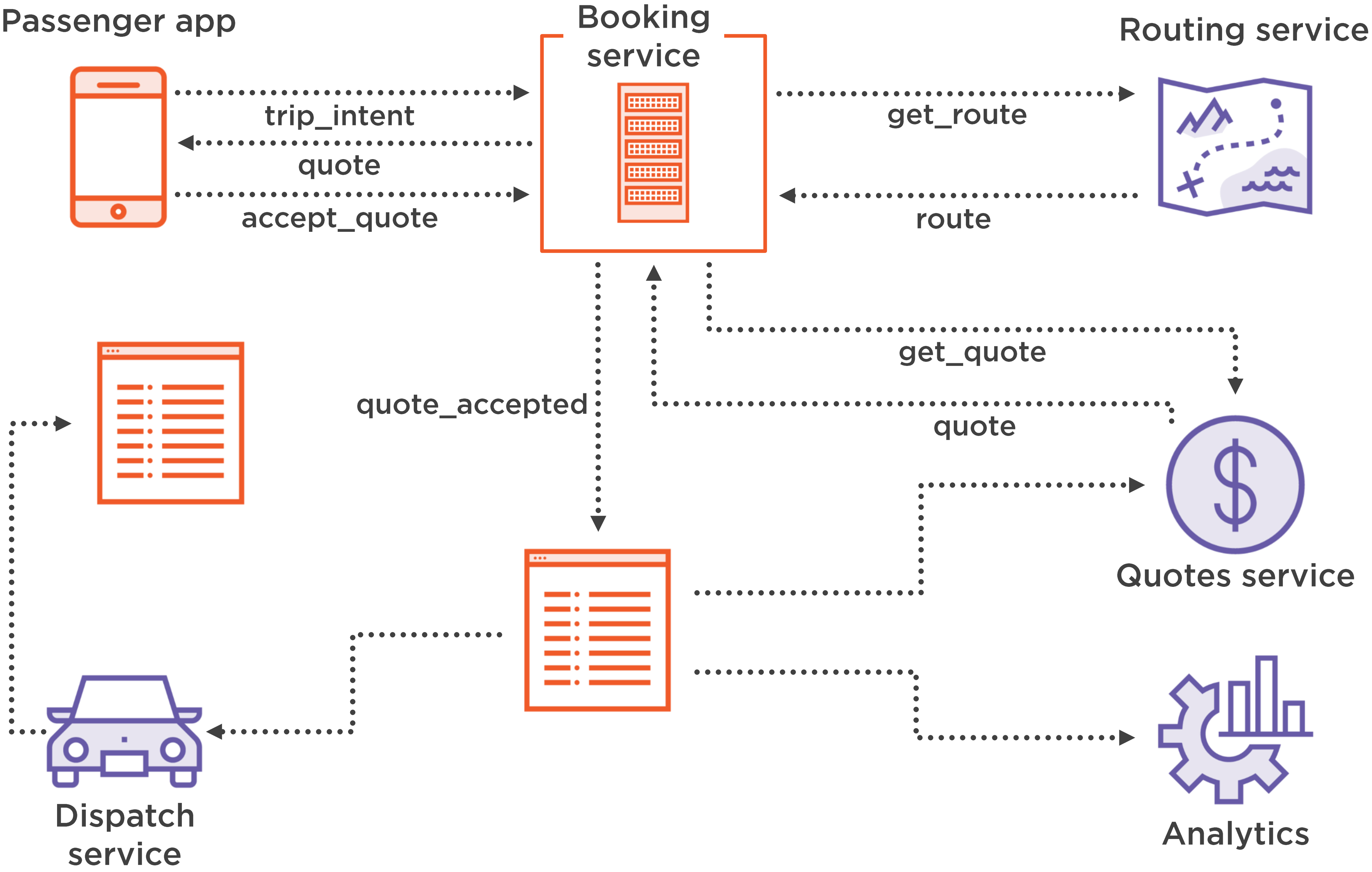
An Example

Hypothetical and simplified architecture of a ride-hailing service.


Passenger app
Booking service
Routing service
trip_intent
quote
accept_quote
get_route
route
get_quote
quote
Quotes service
quote_accepted
Dispatch service
Analytics


Passenger app
Booking service
Routing service
trip_intent
quote
accept_quote
get_route
route
quote_accepted
get_quote
quote
Quotes service
Dispatch service
Analytics

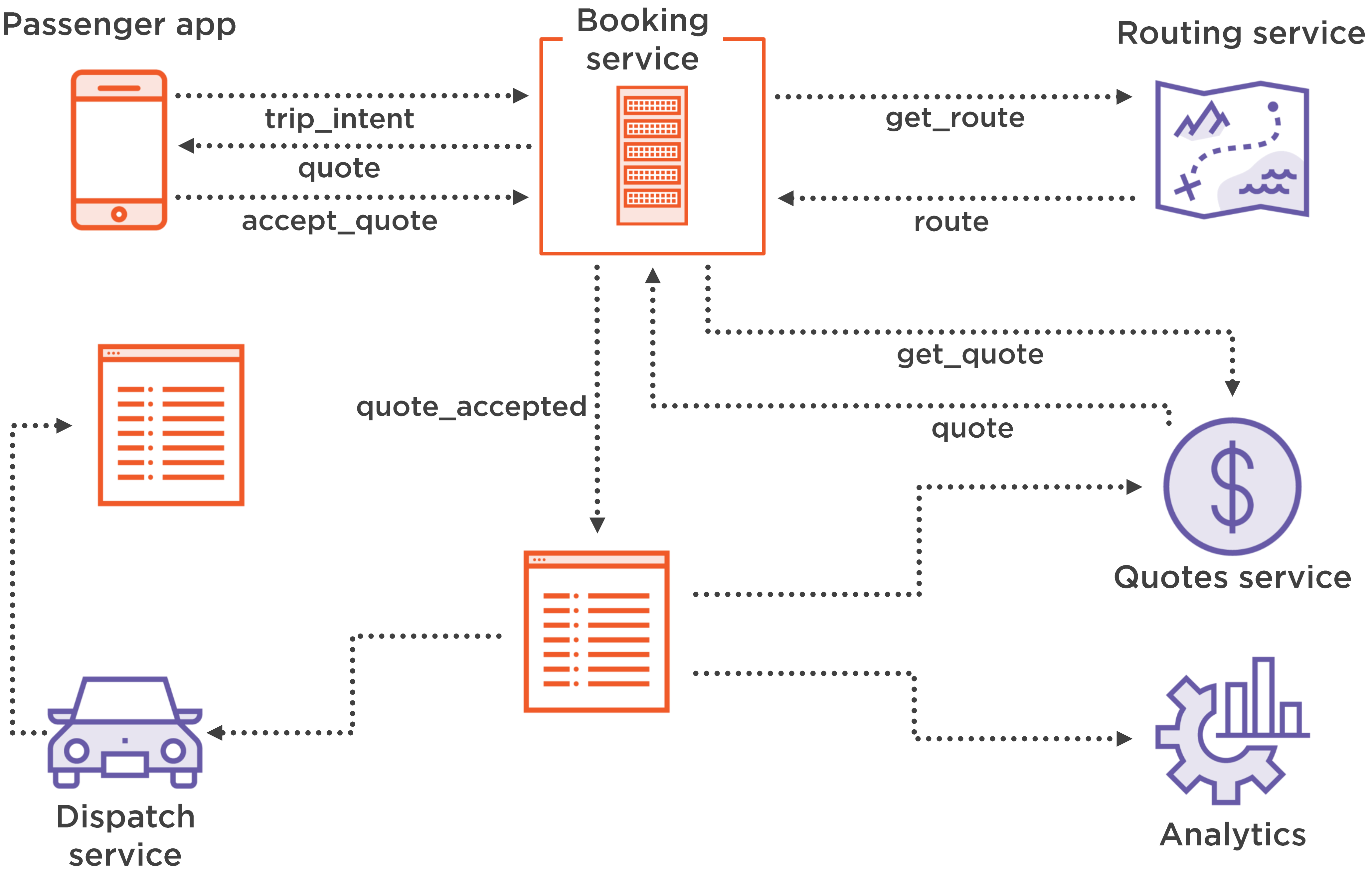

Passenger app
Booking service
Routing service
trip_intent
quote
accept_quote
get_route
route
get_quote
quote_accepted
quote
Quotes service
Dispatch service
Analytics

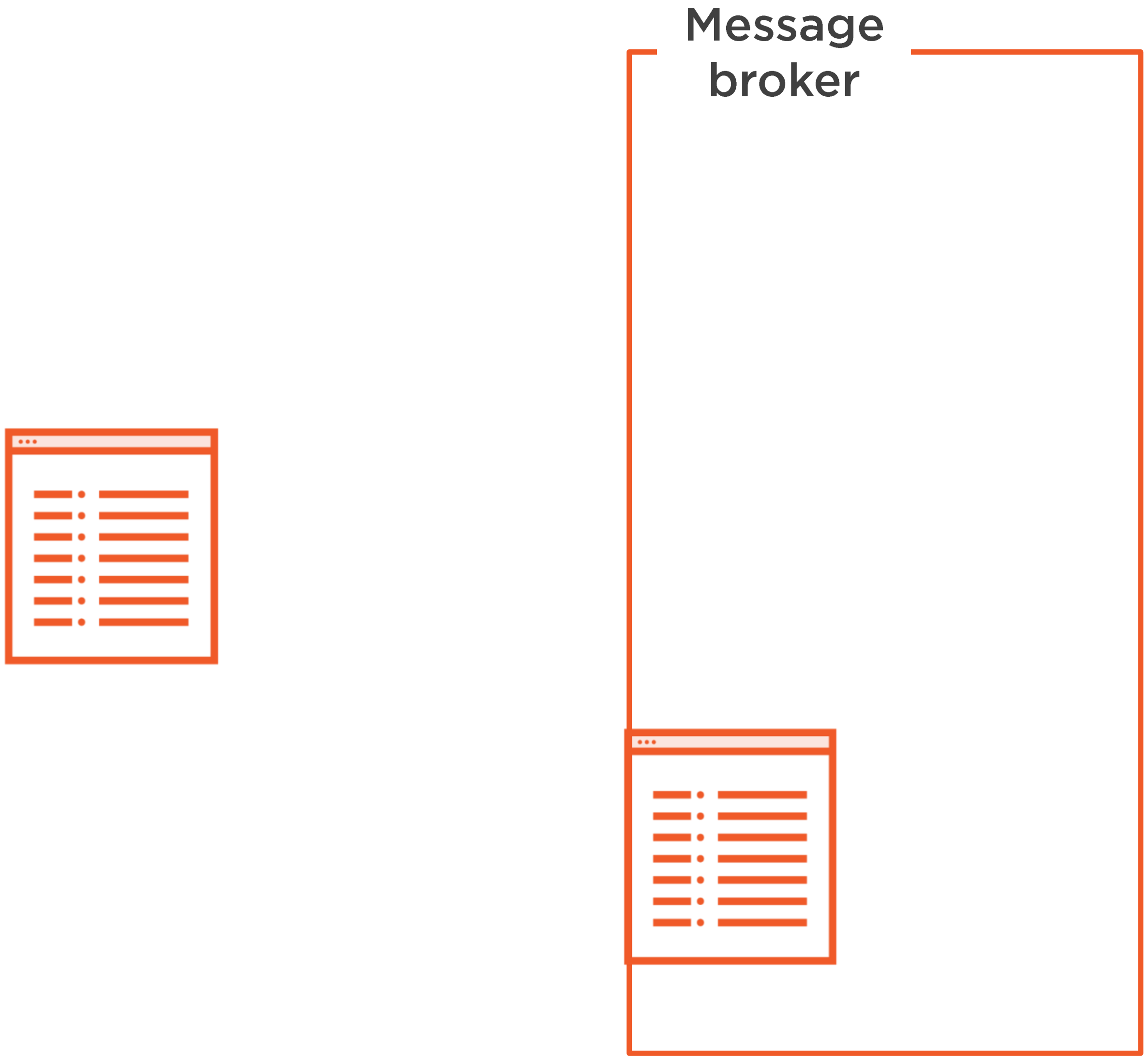

Passenger app
Booking service
Routing service
trip_intent
quote
accept_quote
get_route
route
get_quote
quote
quote_accepted
Quotes service
Dispatch service
Analytics

Business cases coming up

Technical debt

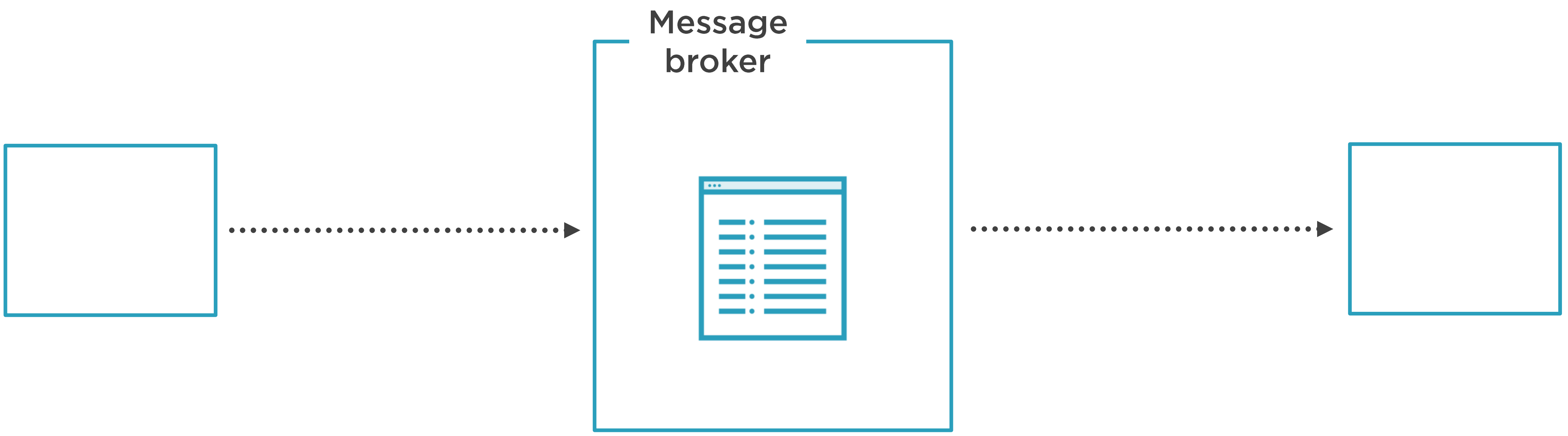
Slower innovation

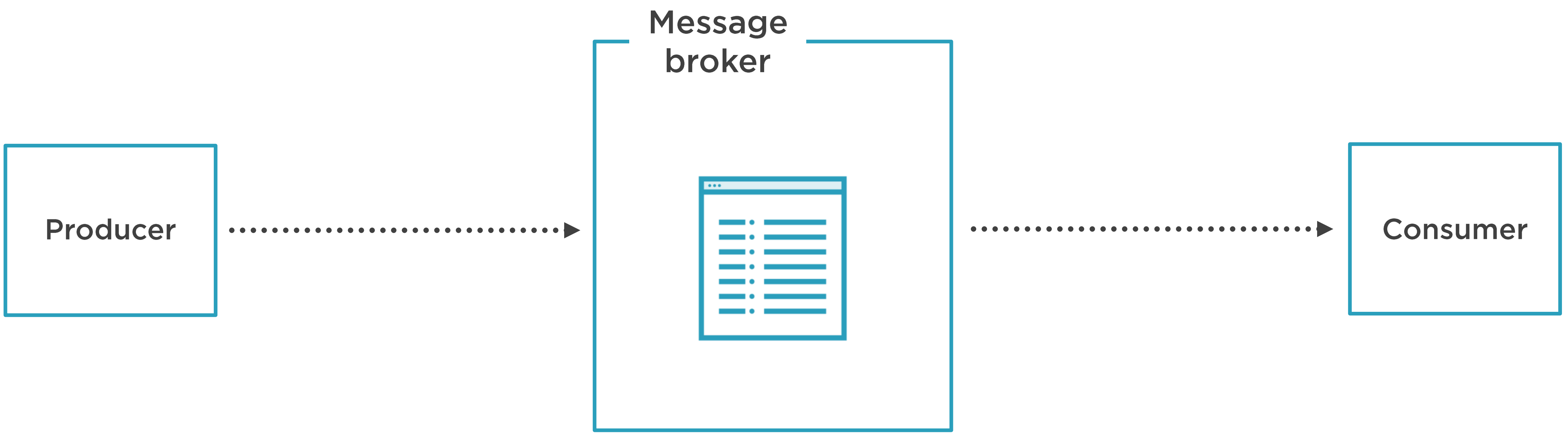

Passenger app
Booking service
Routing service
trip_intent
quote
accept_quote
get_route
route
get_quote
quote
quote_accepted
Quotes service
Dispatch service
Analytics

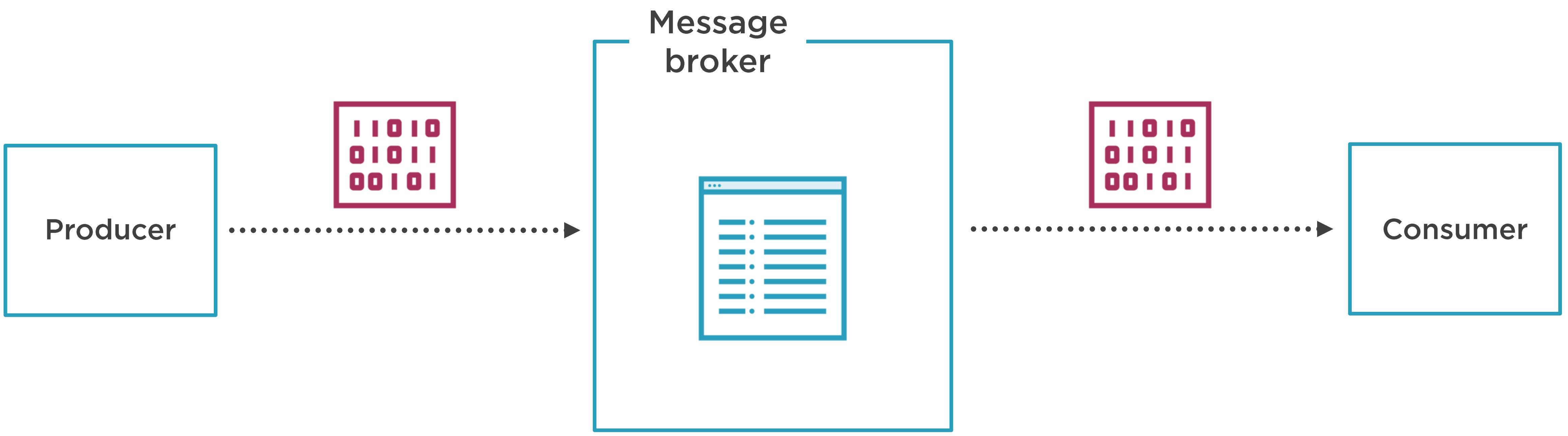


Message broker

Message Broker Patterns


Message
broker

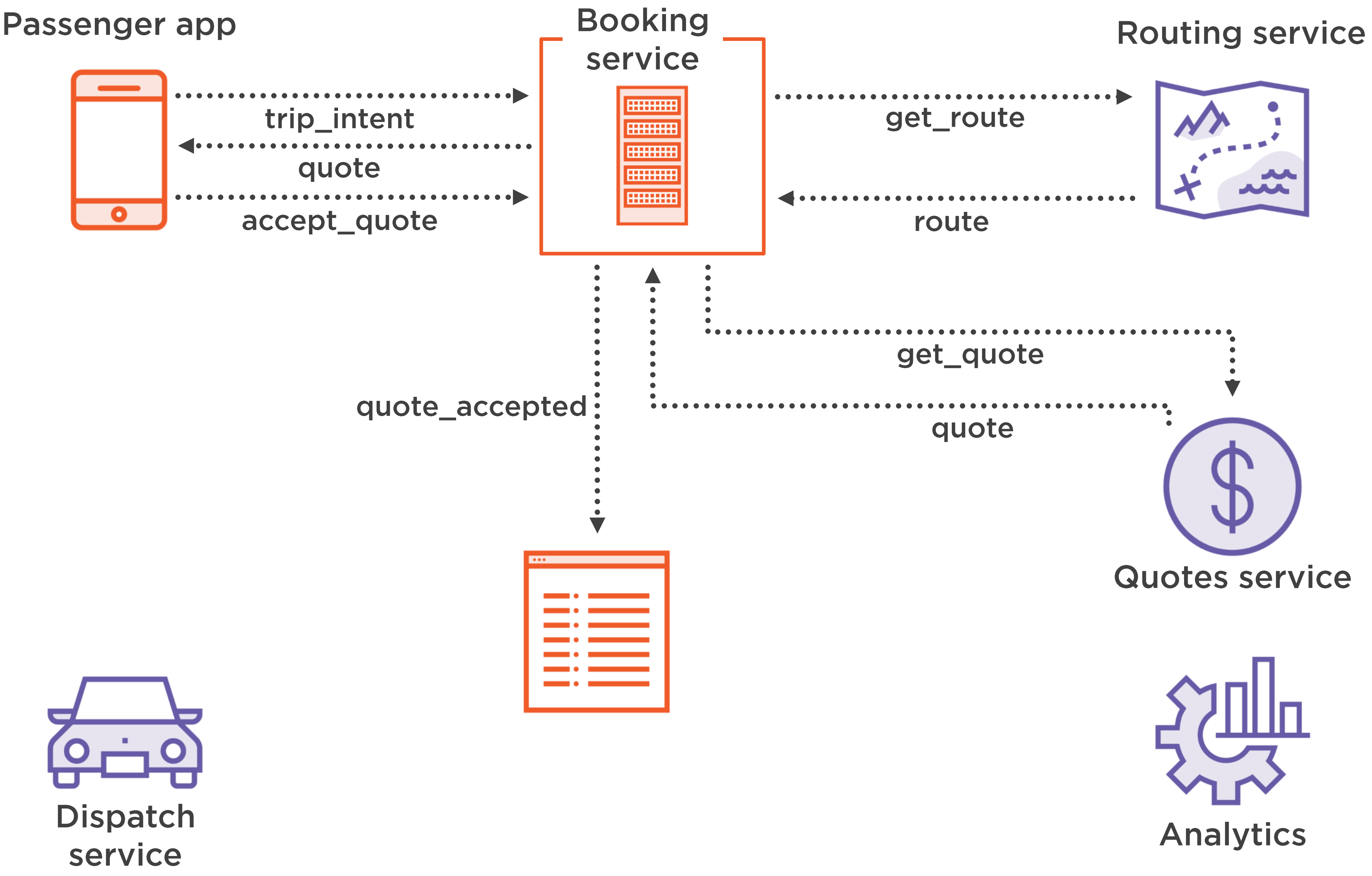

Message broker
Producer
Consumer

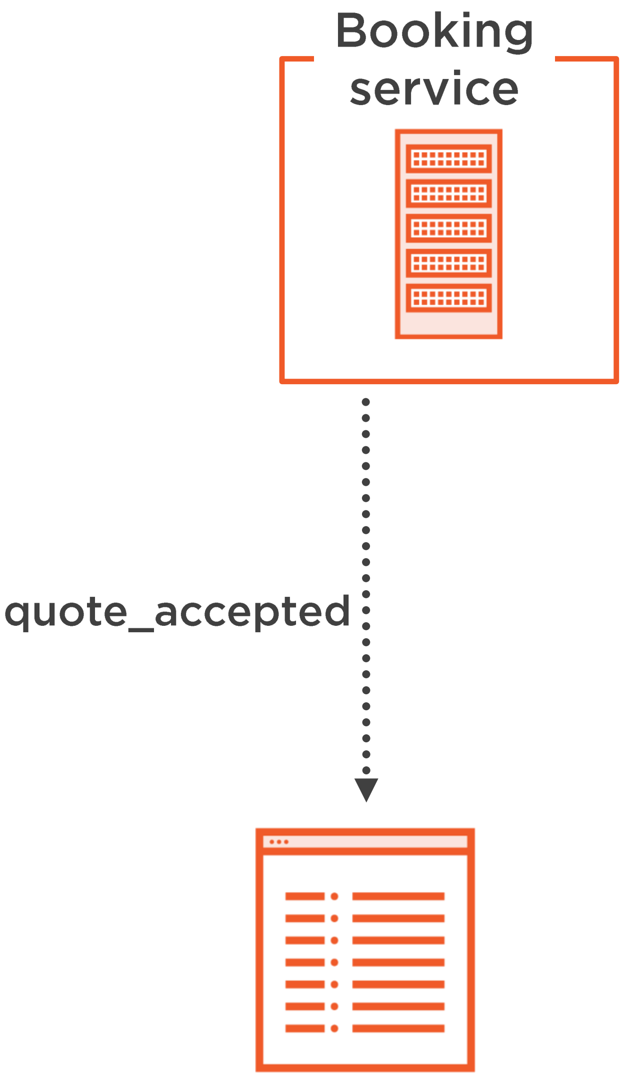

Producer
Message broker
Consumer
11010
01011
00101

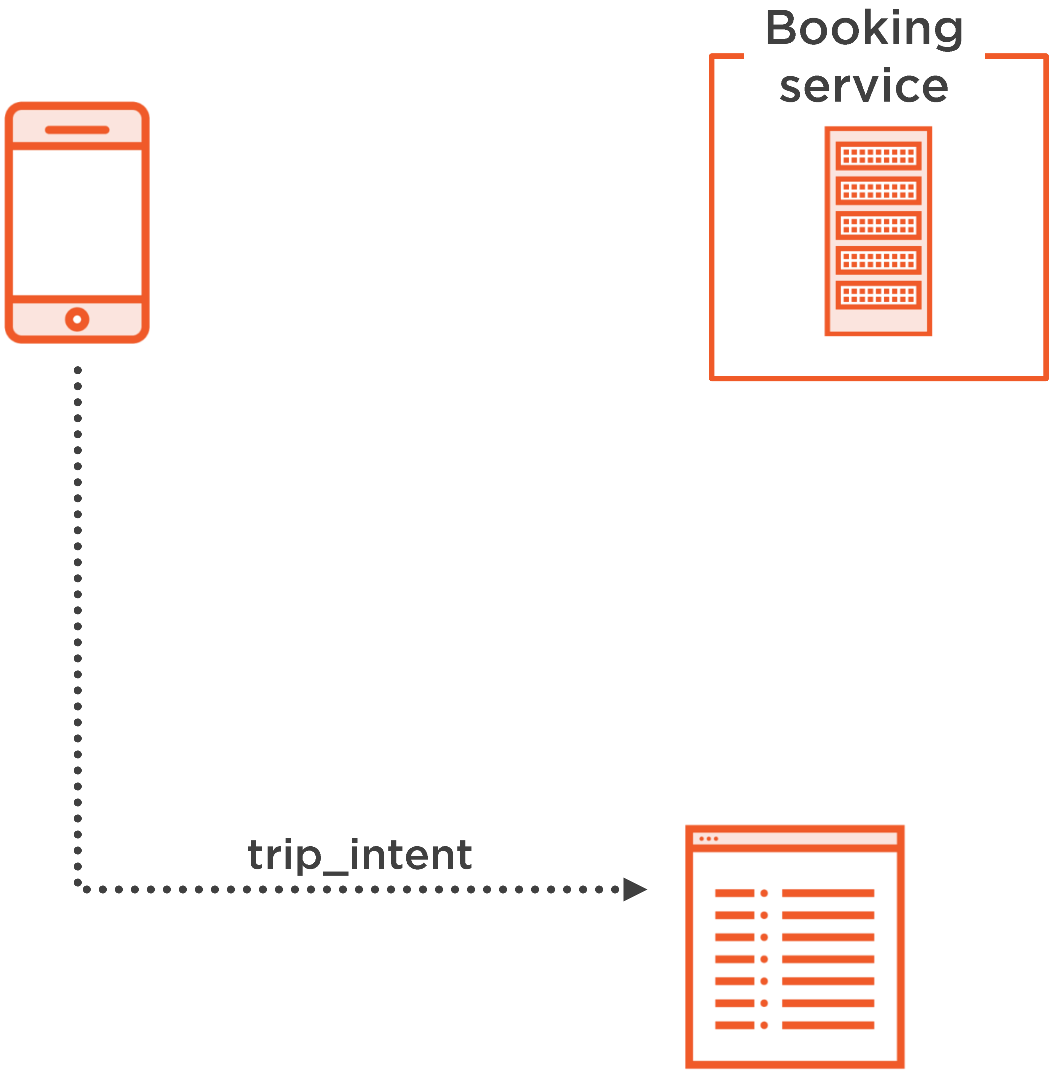
Two Patterns

Publish-subscribe

Message queue

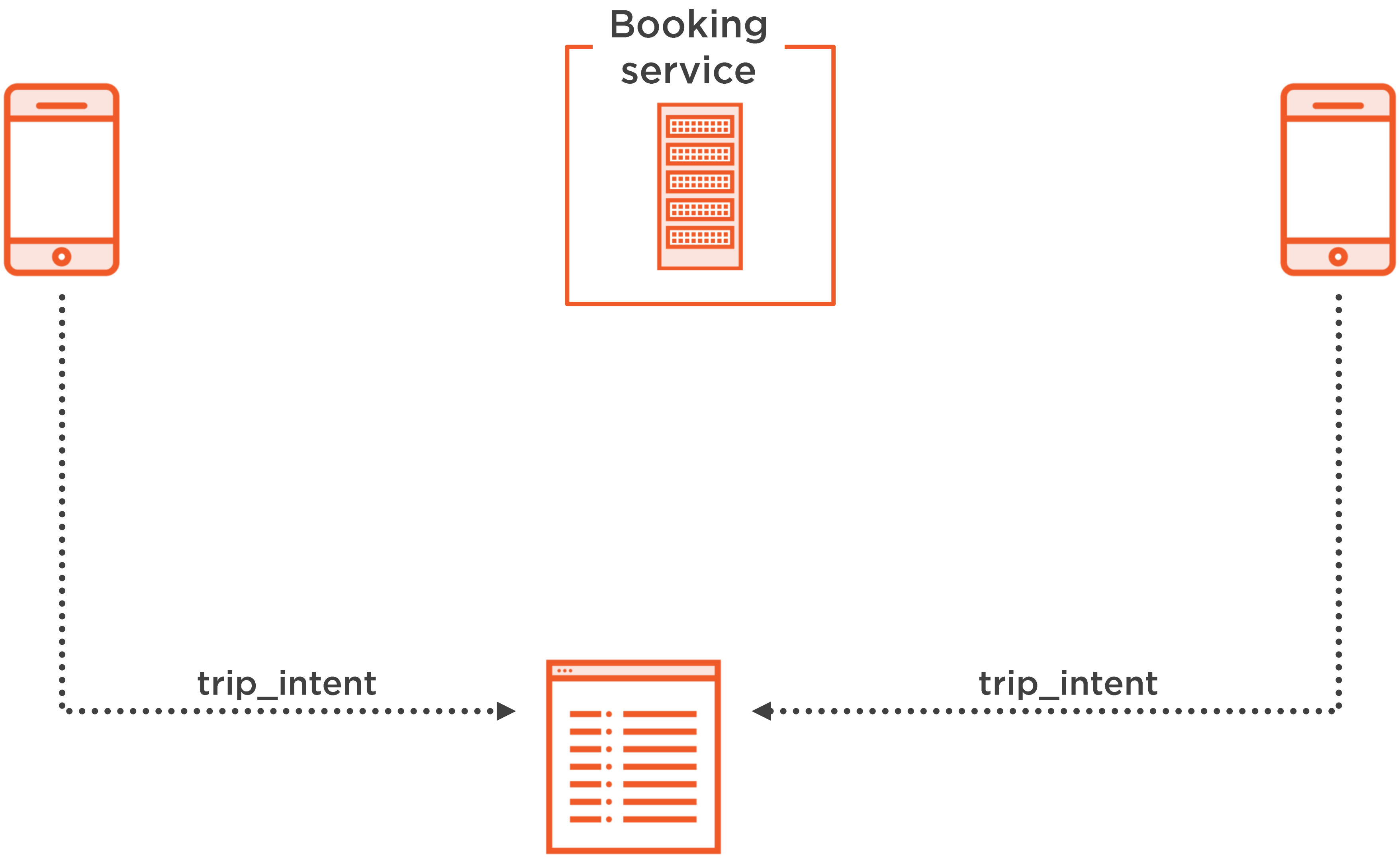

Passenger app
Booking service
Routing service
trip_intent
quote
accept_quote
get_route
route
get_quote
quote
quote_accepted
Quotes service
Dispatch service
Analytics

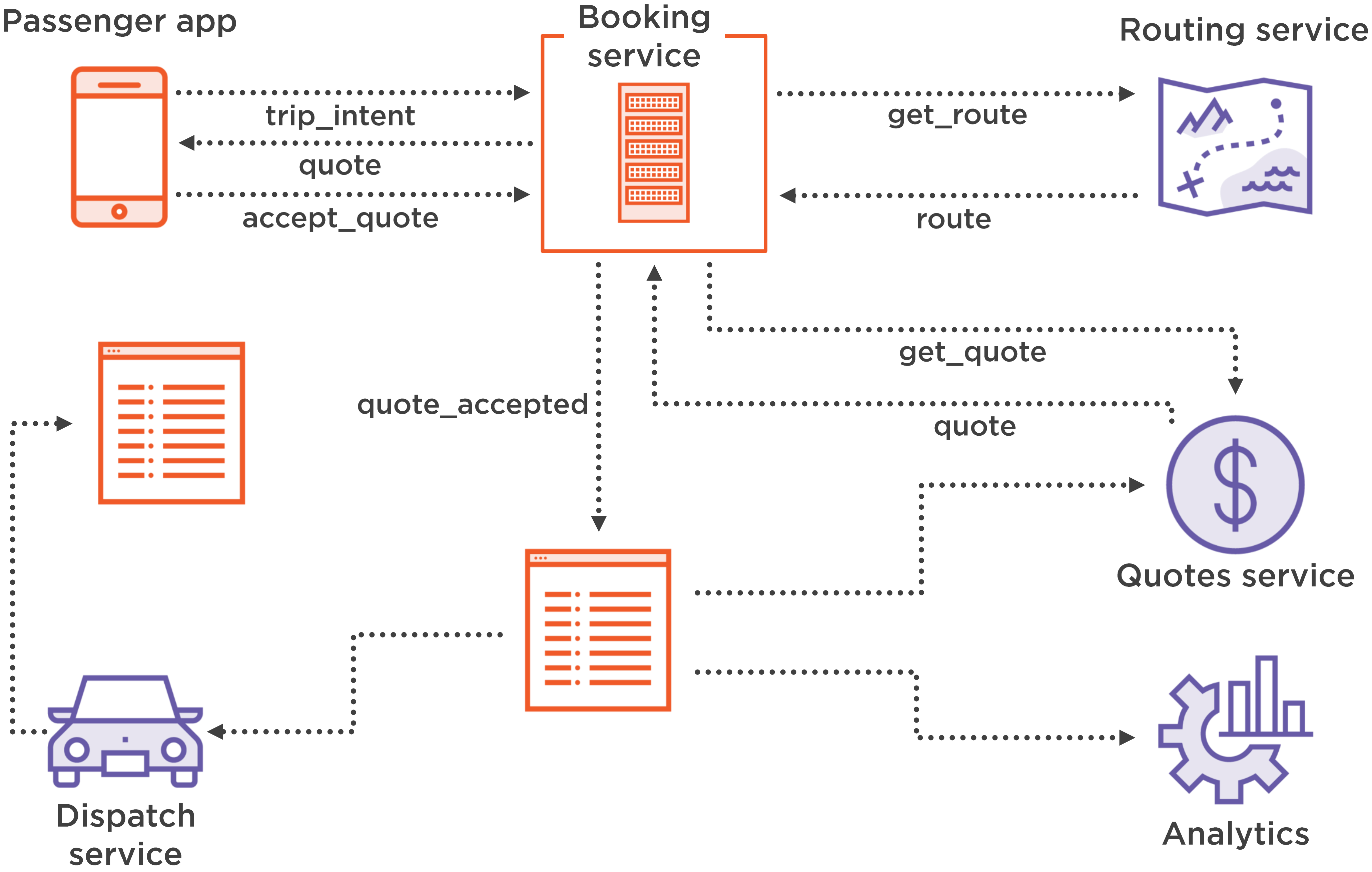

Booking service
quote_accepted

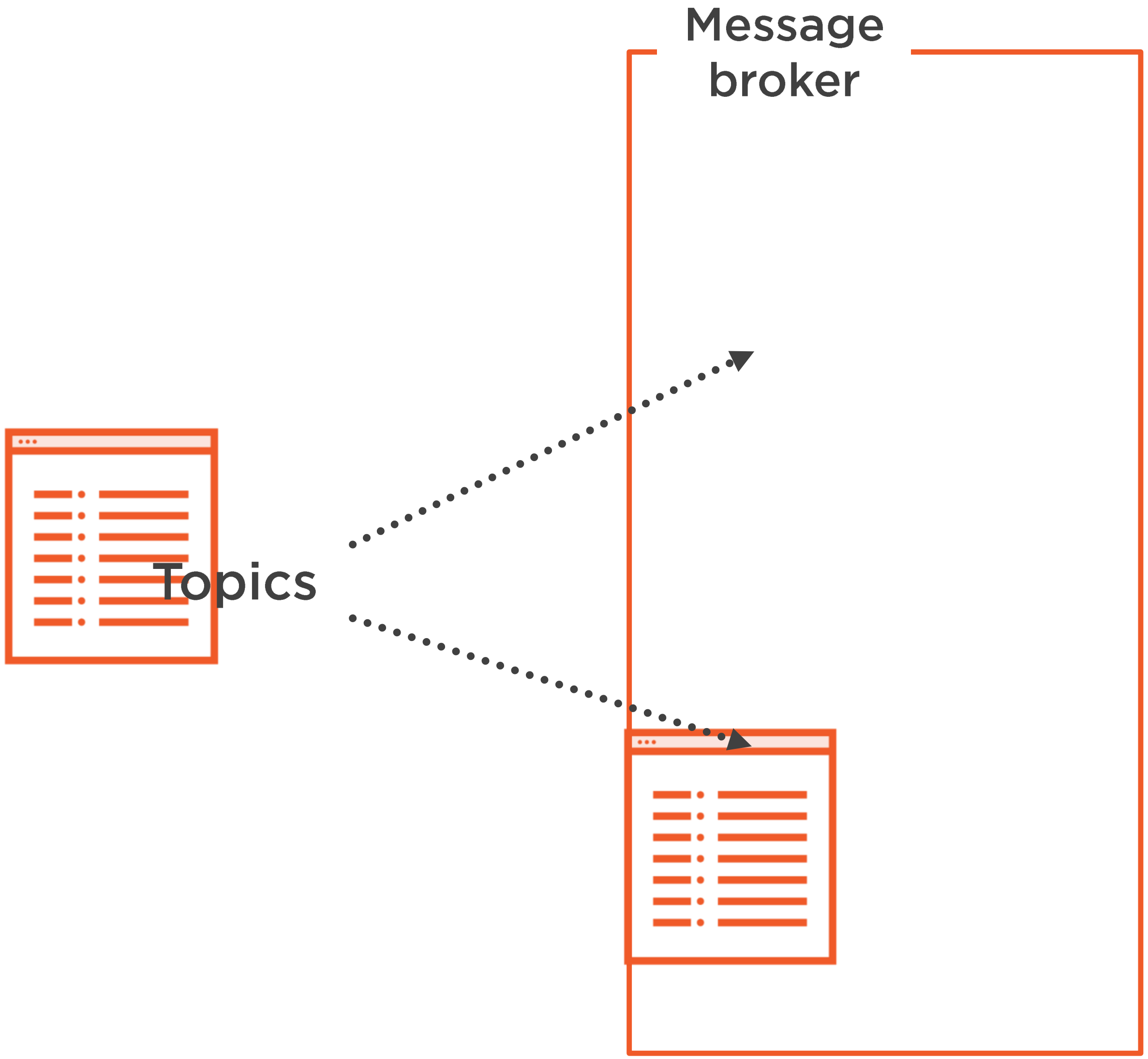


Booking service

trip_intent


Booking
service
trip_intent
trip_intent

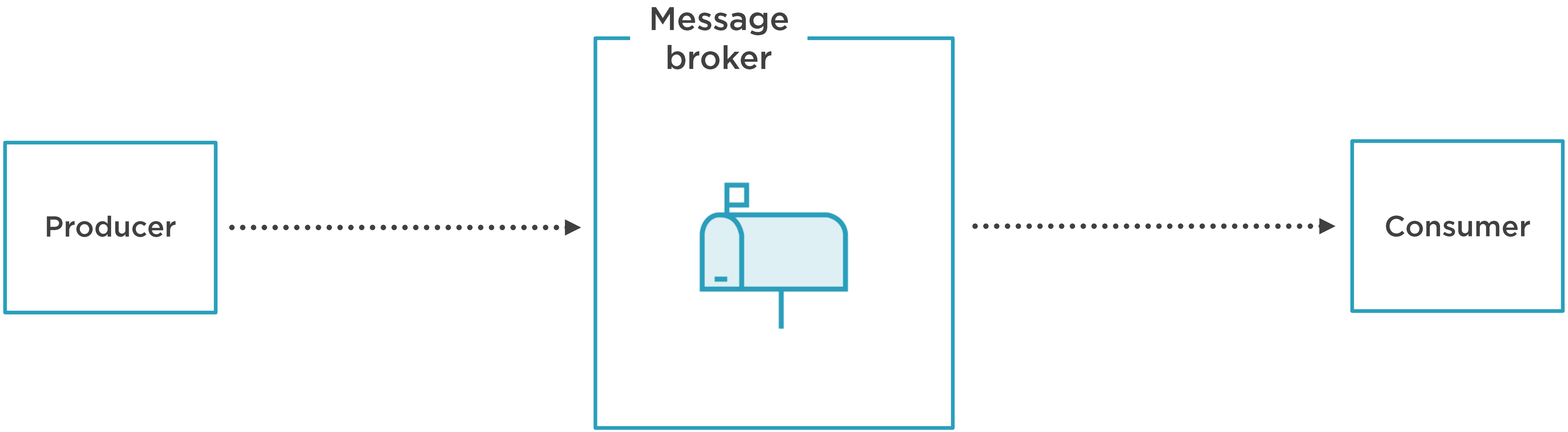

Passenger app
Booking service
Routing service
trip_intent
quote
accept_quote
get_route
route
get_quote
quote
quote_accepted
Quotes service
Dispatch service
Analytics

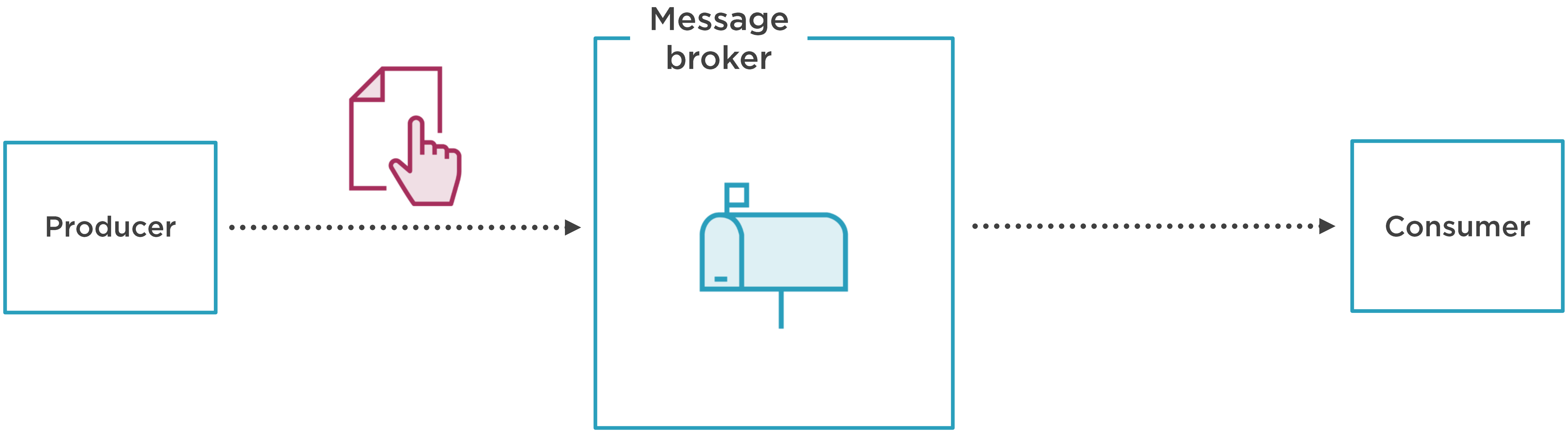

Message broker
Topics

Two Patterns

Publish-subscribe

Message queue

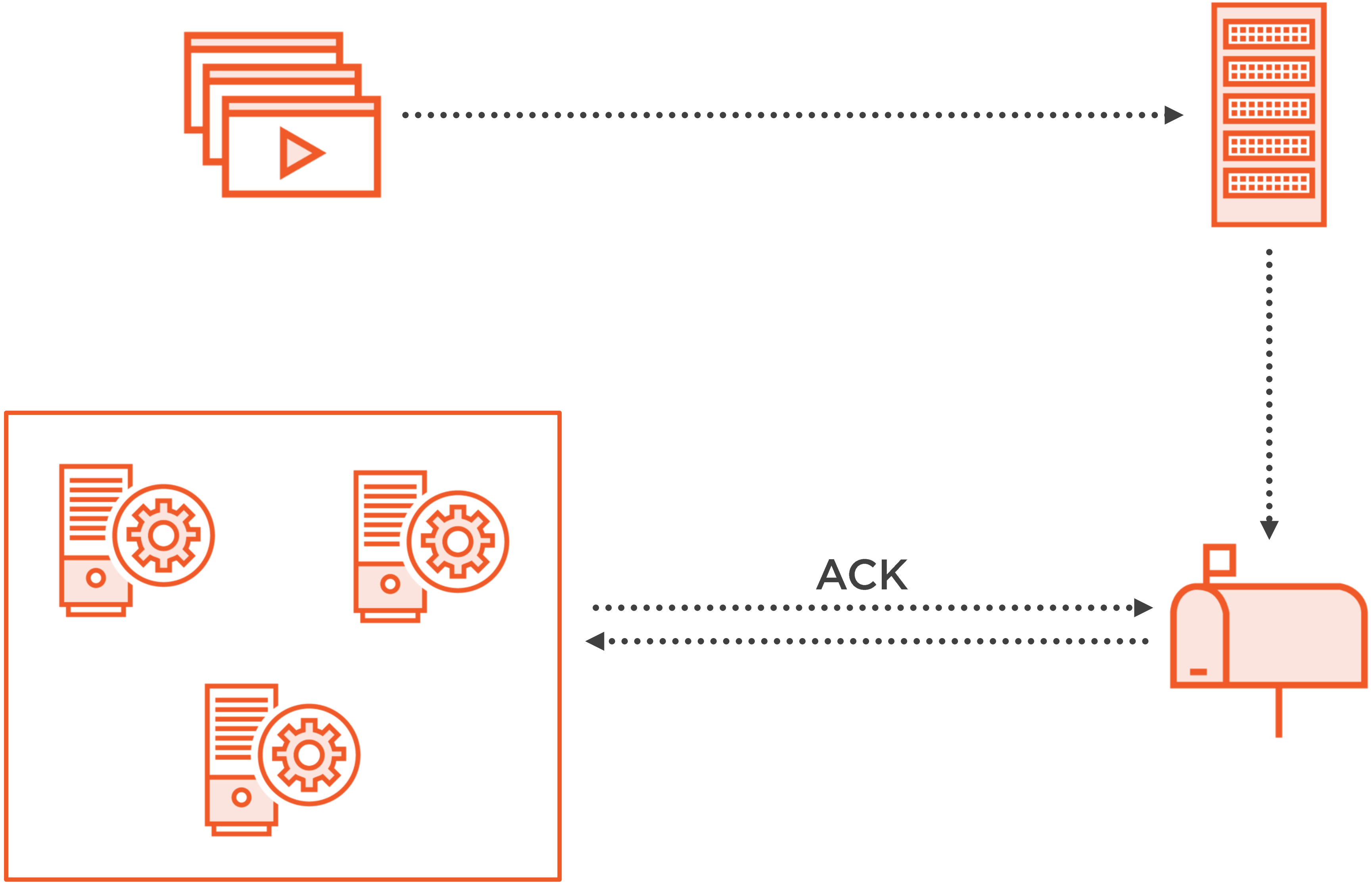

Message broker
Producer
Consumer



Message broker

Producer

Consumer


ACK

Main Difference

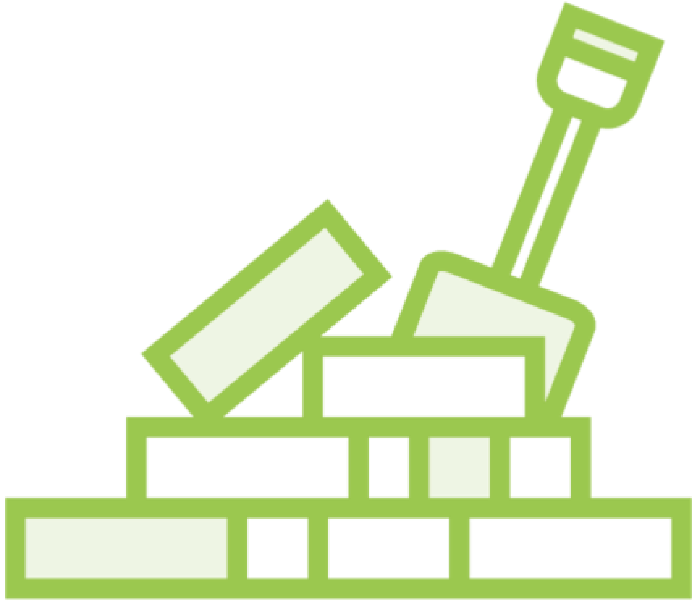
Publish-subscribe

No assumptions are made about the number and activity of consumers

Message queue

Recipient of the message is clearly defined and is expected to process it only once

Kafka



Created at LinkedIn

Processing of user activity was the principal use-case

Other brokers were evaluated – each had shortcomings

Key Features

Scalability and high availability

Compact message format

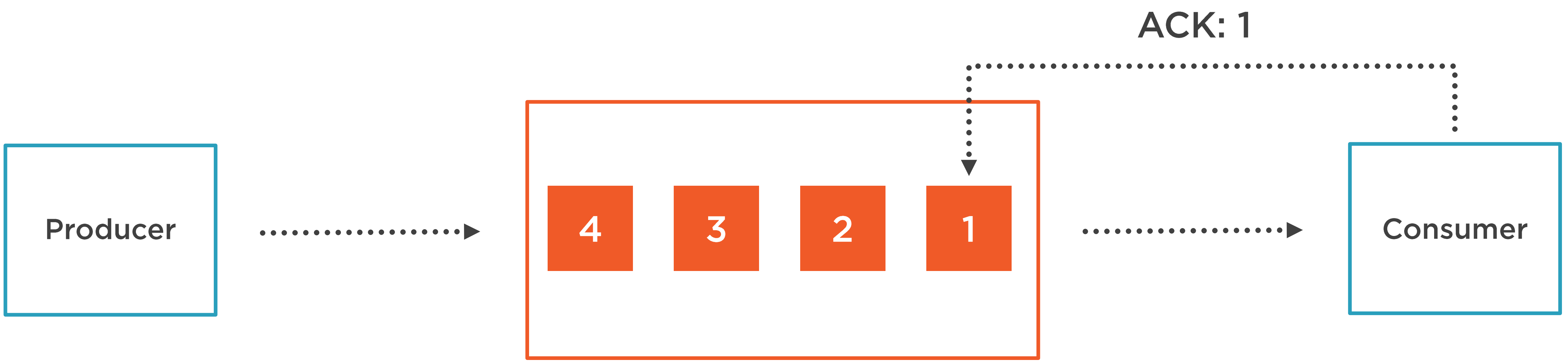
On-disk storage

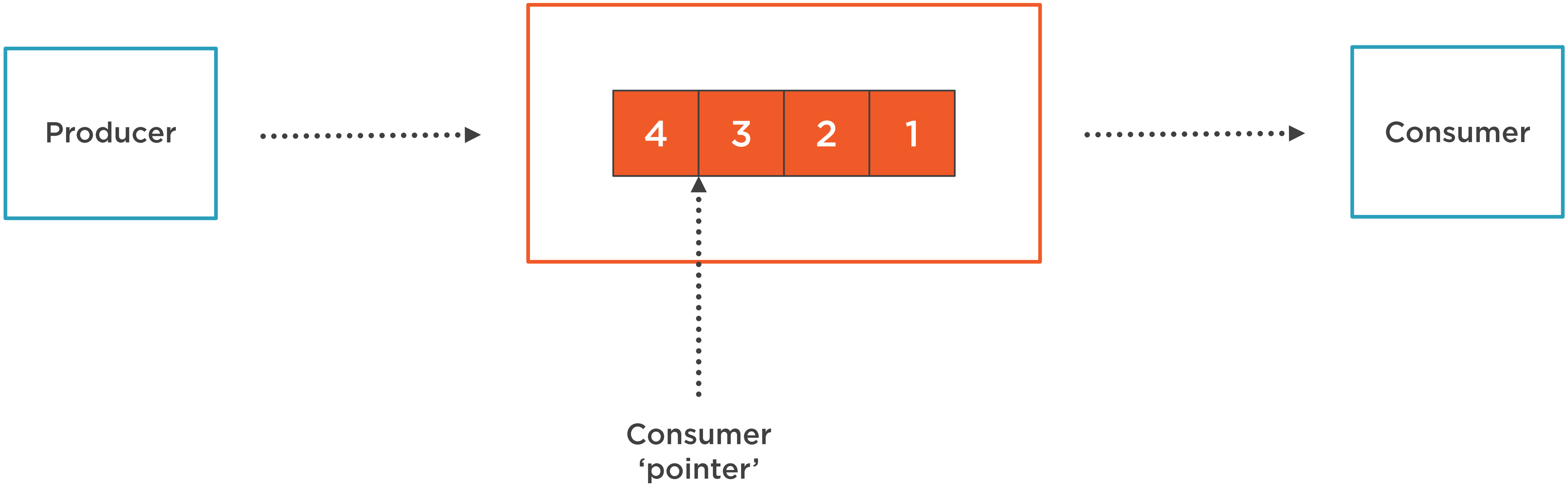
Kafka: Pub/Sub or Message Queue?

They are just generic patterns …

… not implementation details

Kafka is more of a Pub/Sub system


ACK: 1
Producer
4
3
2
1
Consumer


Producer
4
3
2
1
Consumer
Consumer ‘pointer’

Messages are always kept on disk

This allows ‘going back’ in time

Final Thoughts

It is crucial to understand particular technology, not only the patterns

Kafka is a very solid and flexible piece of technology

Summary

Benefits of a message broker

Patterns: Pub/Sub and queue

Kafka and other technologies