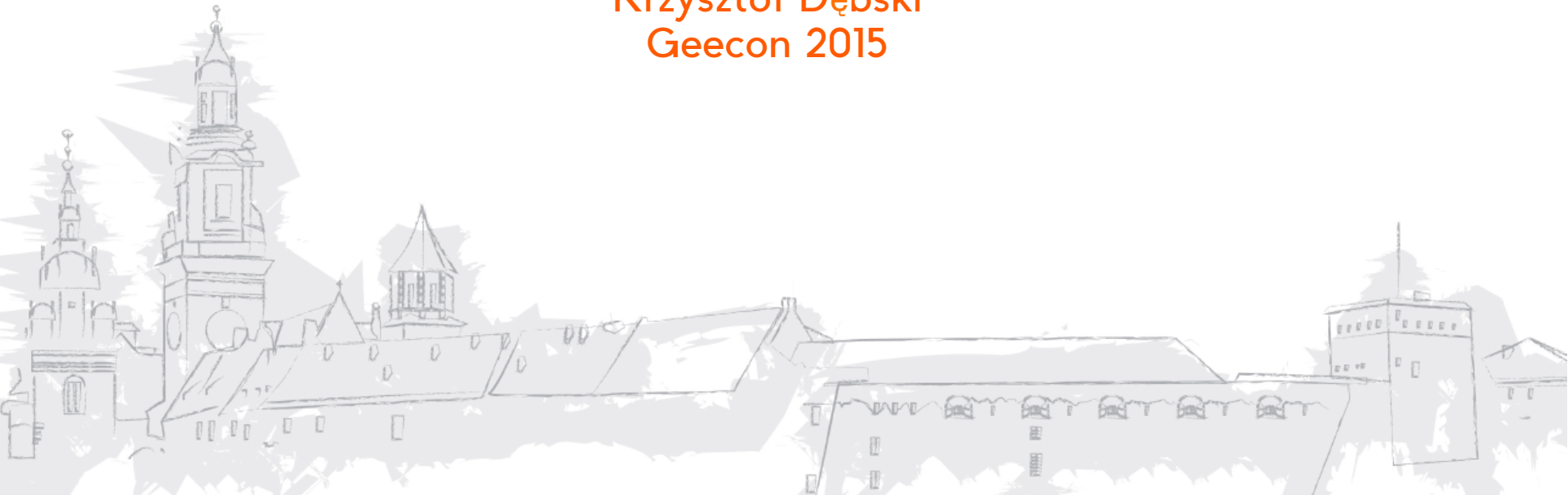
PubSub++

or how to make life with Kafka easier

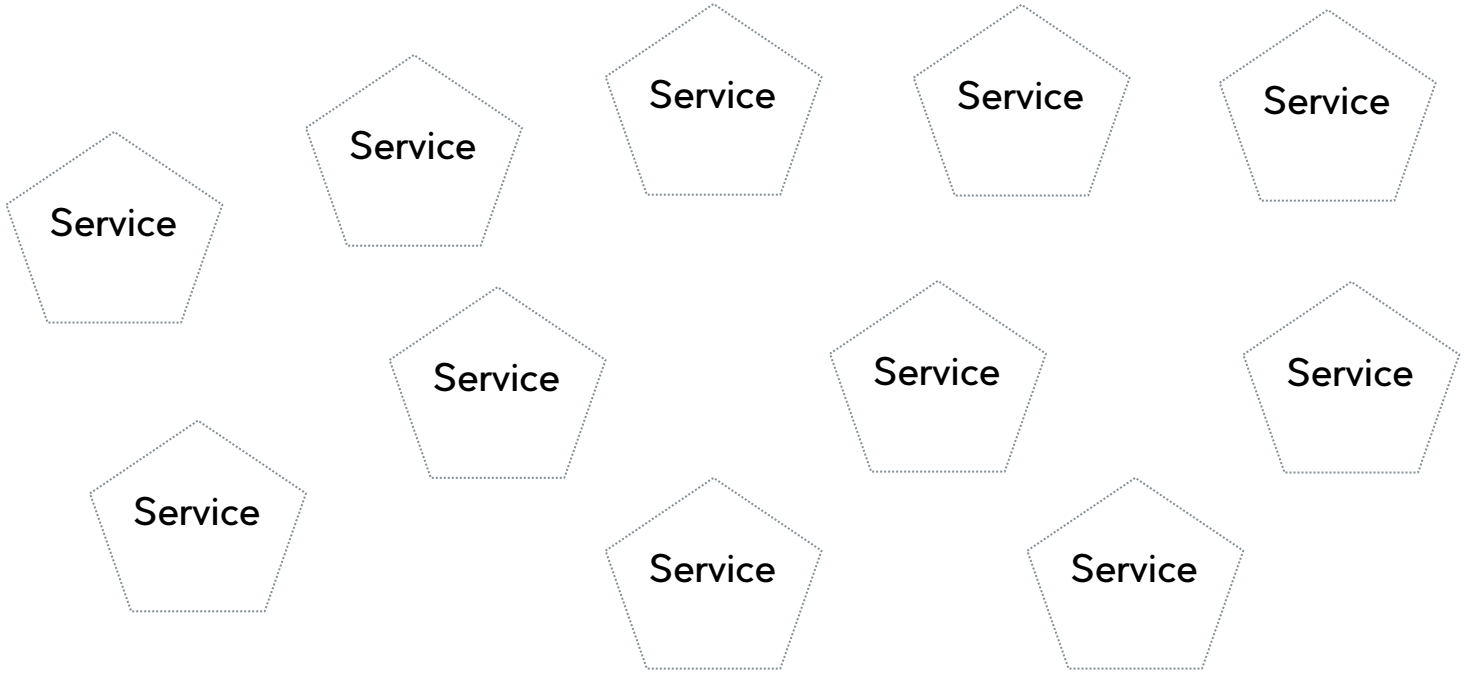
Krzysztof Dębski

Geecon 2015

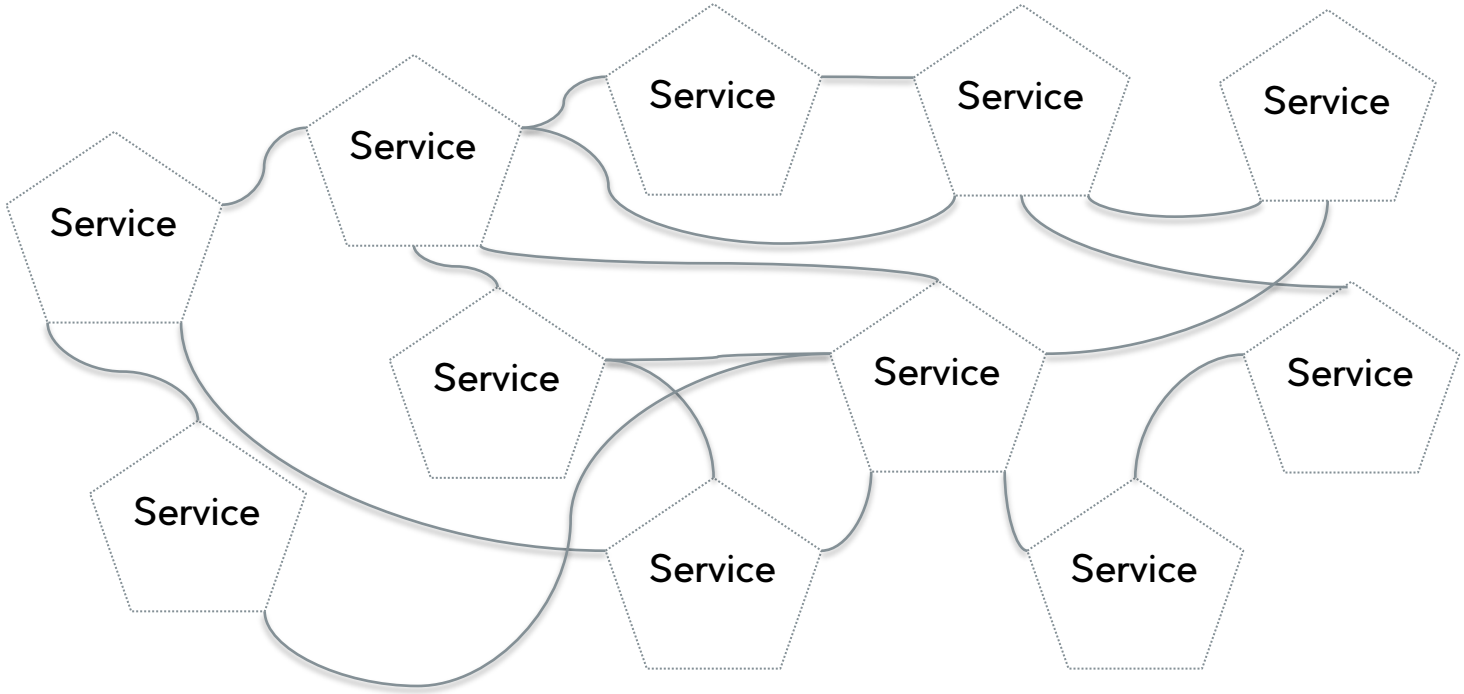


Why do we need a PubSub?

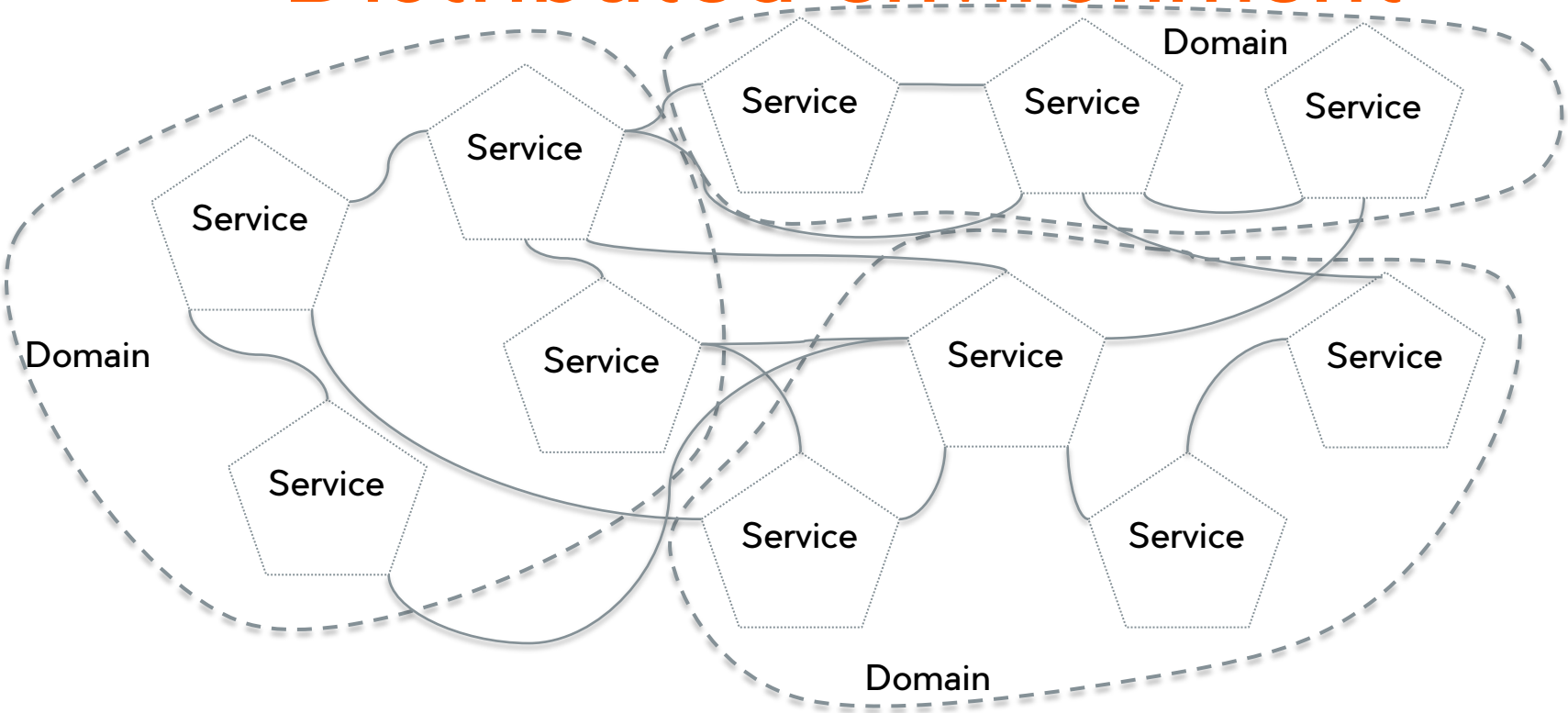
Distributed environment



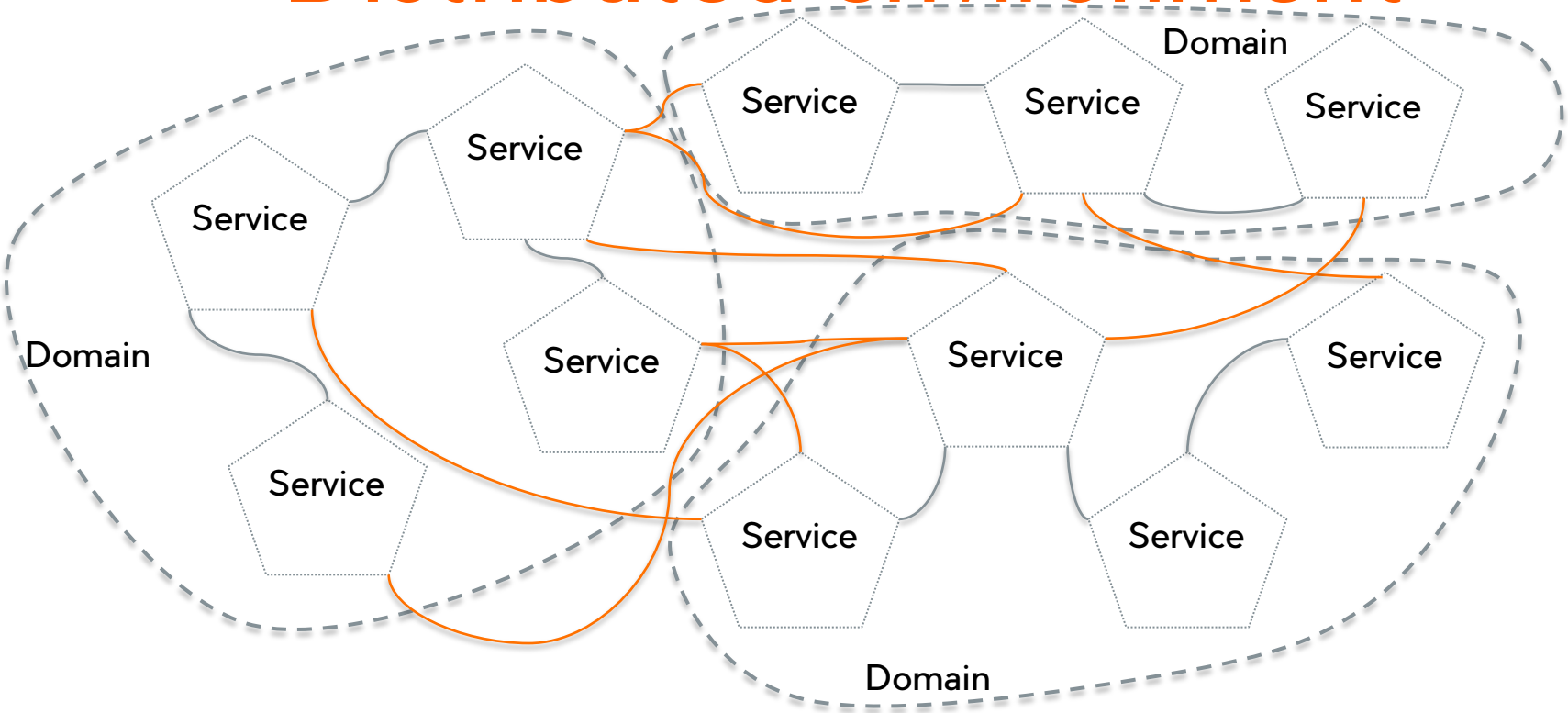
Distributed environment



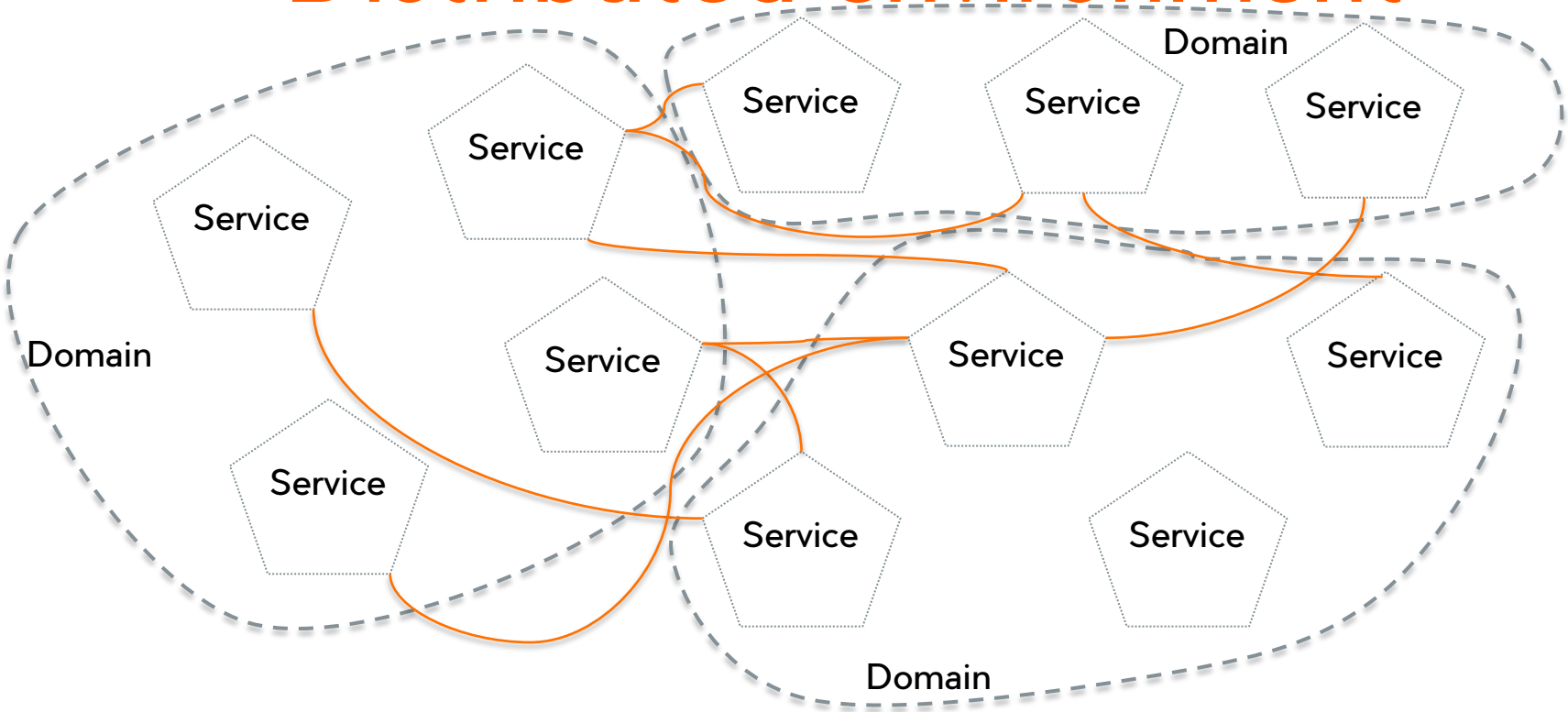
Distributed environment



Distributed environment



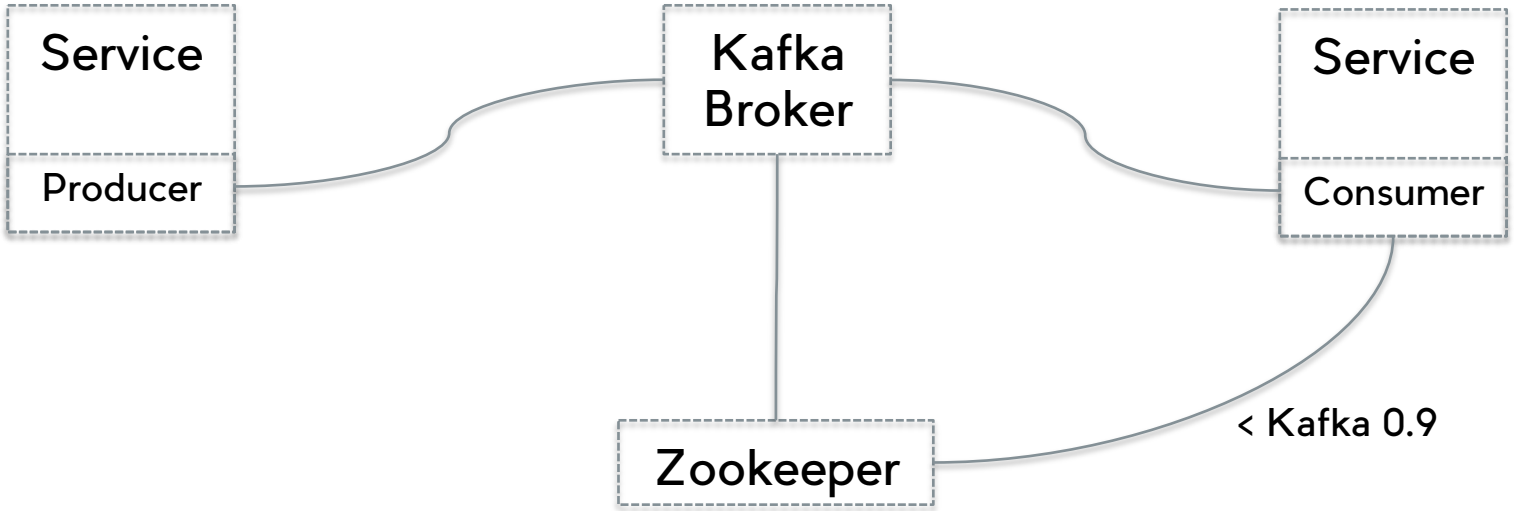
Distributed environment



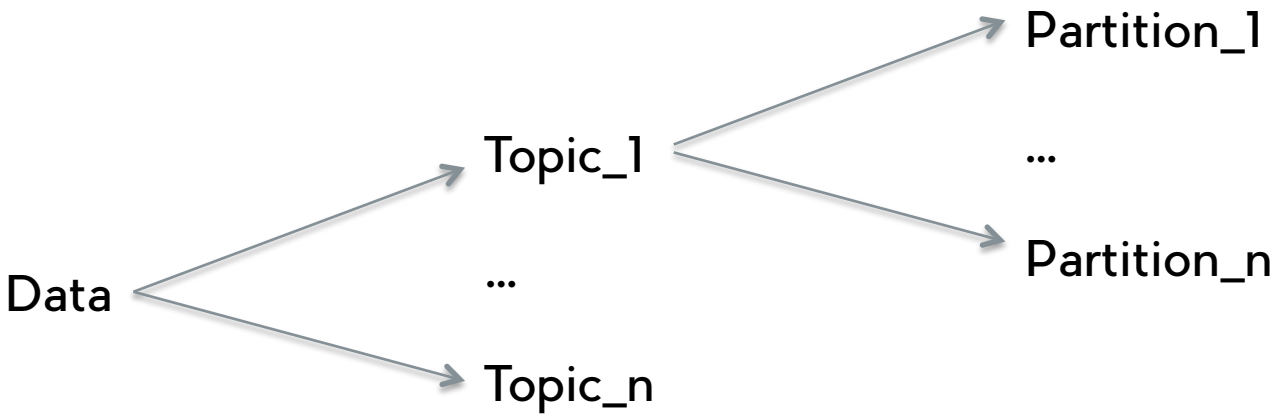
Orchestration vs. Choreography

Kafka

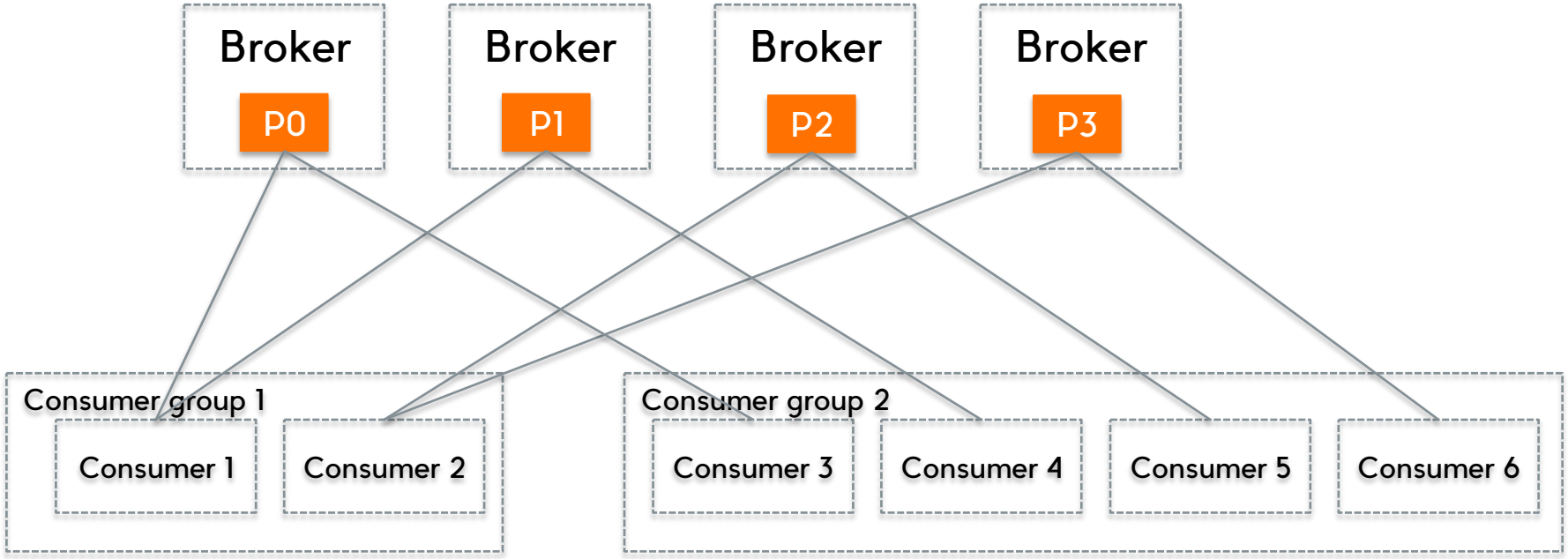
Kafka architecture



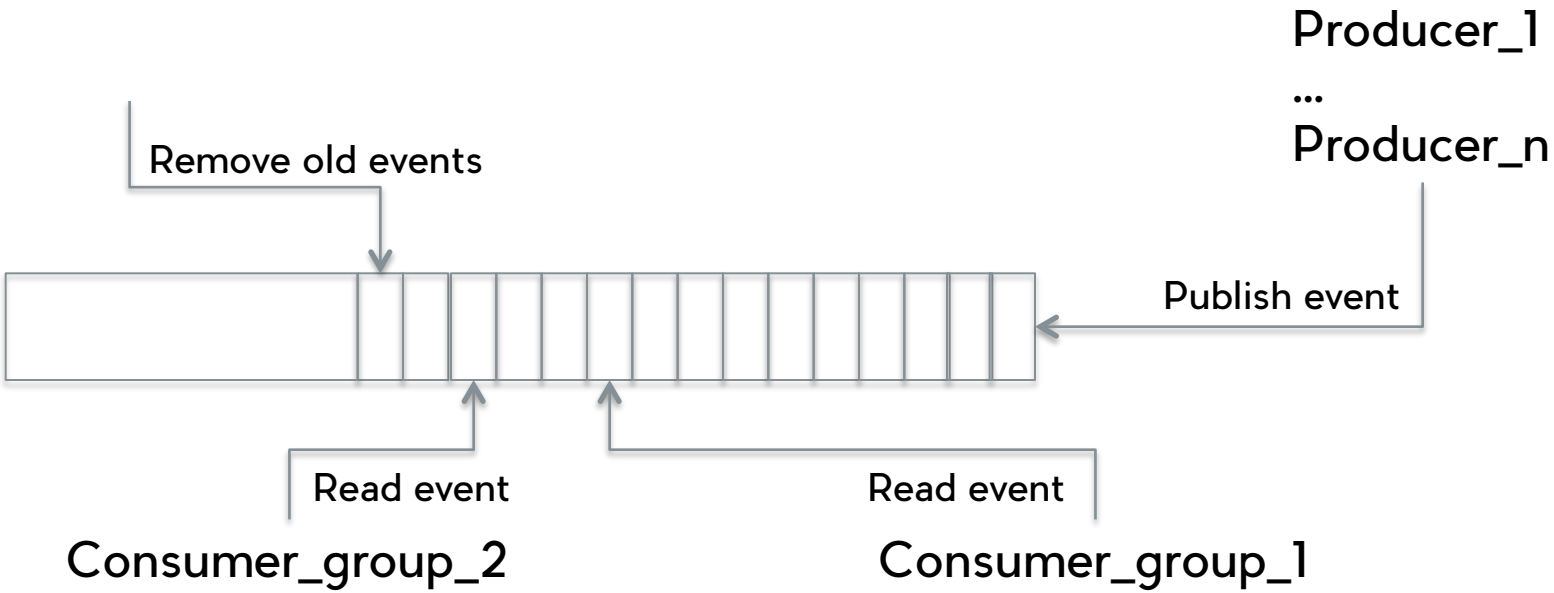
Kafka data



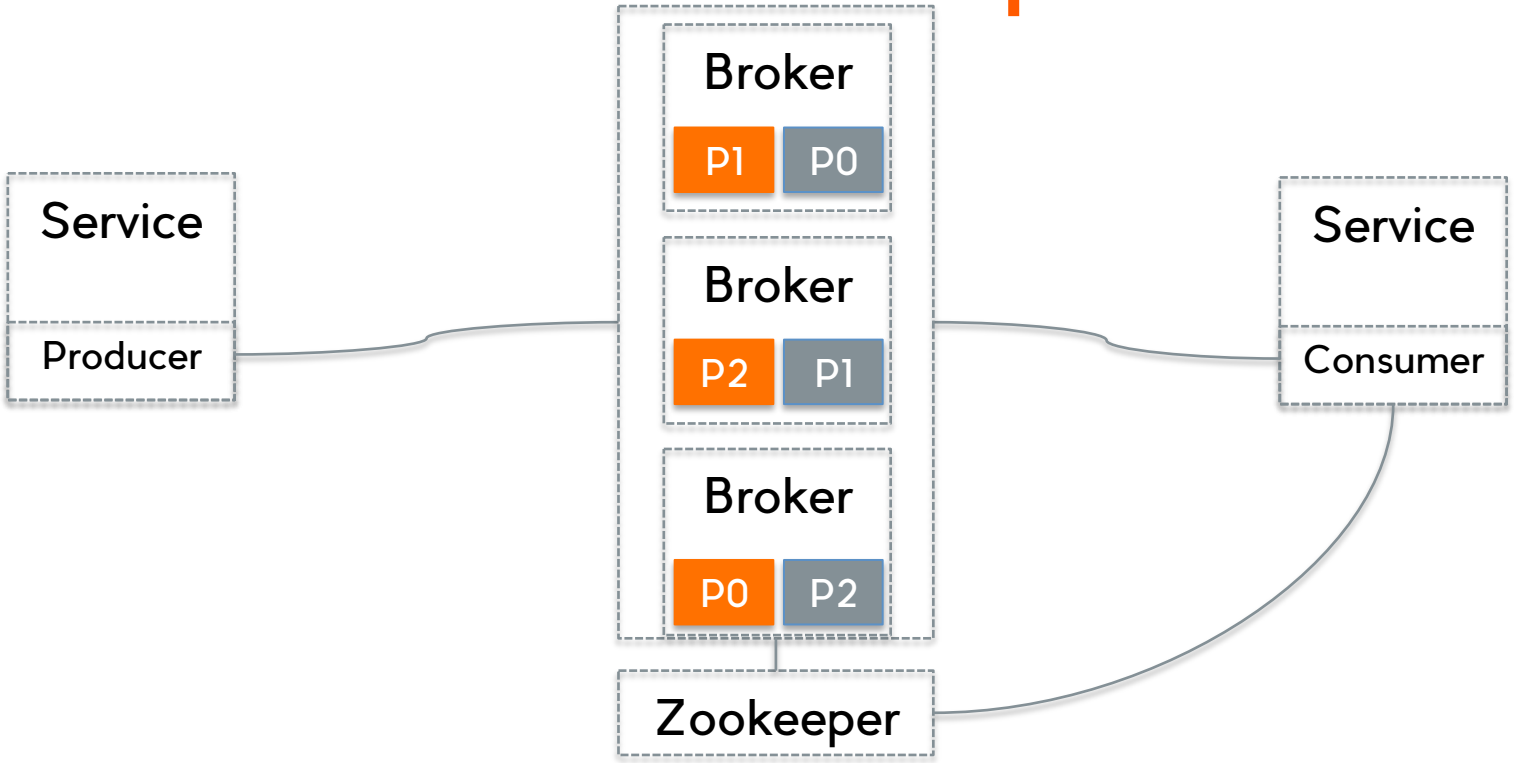
Consumer groups



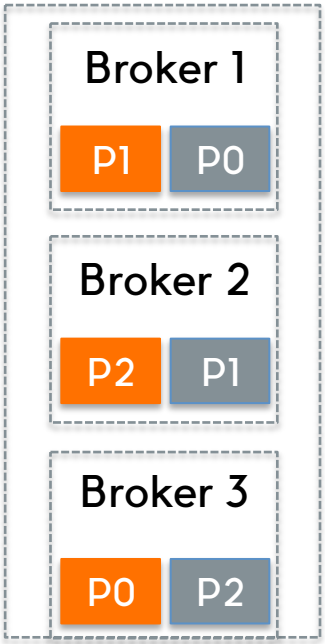
Publish & subscribe



Leader & Replicas



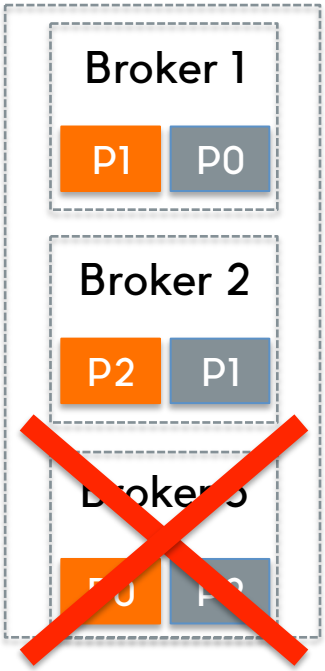
Rebalancing leaders



Topic: test Partition count: 3 Replication factor: 1 Configs: retention.ms=86400000

Topic: test	Partition: 0	Leader: 3	Replicas: 3, 1	ISR: 3, 1
Topic: test	Partition: 1	Leader: 1	Replicas: 1, 2	ISR: 1, 2
Topic: test	Partition: 2	Leader: 2	Replicas: 2, 3	ISR: 2, 3

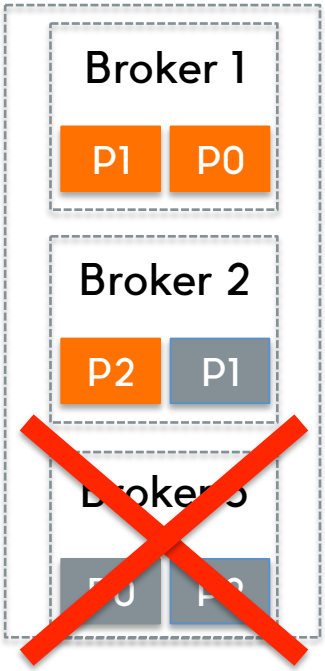
Rebalancing leaders



Topic: test Partition count: 3 Replication factor: 1 Configs: retention.ms=86400000

Topic: test	Partition: 0	Leader: 3	Replicas: 3, 1	ISR: 3, 1
Topic: test	Partition: 1	Leader: 1	Replicas: 1, 2	ISR: 1, 2
Topic: test	Partition: 2	Leader: 2	Replicas: 2, 3	ISR: 2, 3

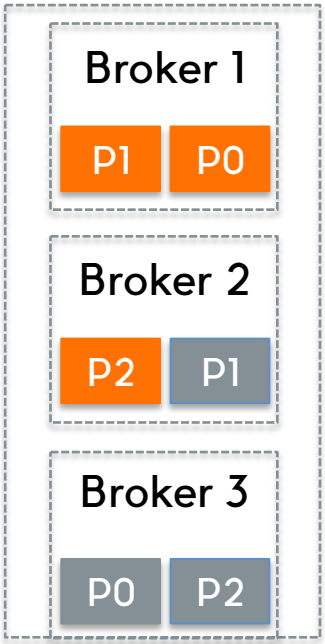
Rebalancing leaders



Topic: test Partition count: 3 Replication factor: 1 Configs: retention.ms=86400000

Topic: test	Partition: 0	Leader: 1	Replicas: 3, 1	ISR: 1
Topic: test	Partition: 1	Leader: 1	Replicas: 1, 2	ISR: 1, 2
Topic: test	Partition: 2	Leader: 2	Replicas: 2, 3	ISR: 2

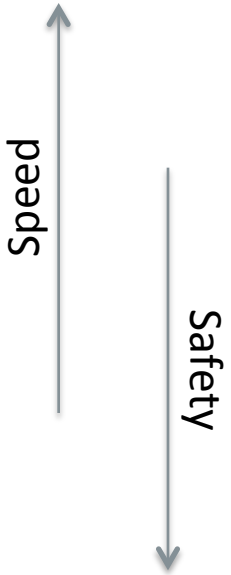
Rebalancing leaders



Topic: test Partition count: 3 Replication factor: 1 Configs: retention.ms=86400000

Topic: test	Partition: 0	Leader: 1	Replicas: 3, 1	ISR: 1, 3
Topic: test	Partition: 1	Leader: 1	Replicas: 1, 2	ISR: 1, 2
Topic: test	Partition: 2	Leader: 2	Replicas: 2, 3	ISR: 2, 3

Acknowledges



0 - don't wait for response from broker

1 - only leader needs to respond

-1 - all replicas in sync

Events security

Basically, there is no authentication

You can create any topic

You can publish everywhere

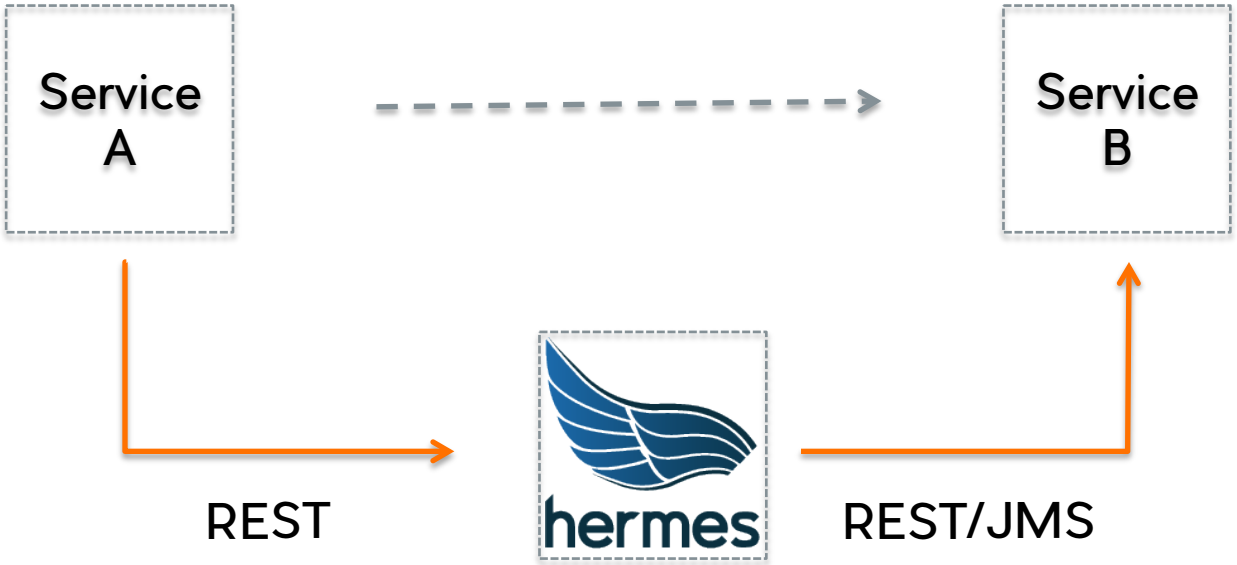
You can subscribe to any topic


hermes

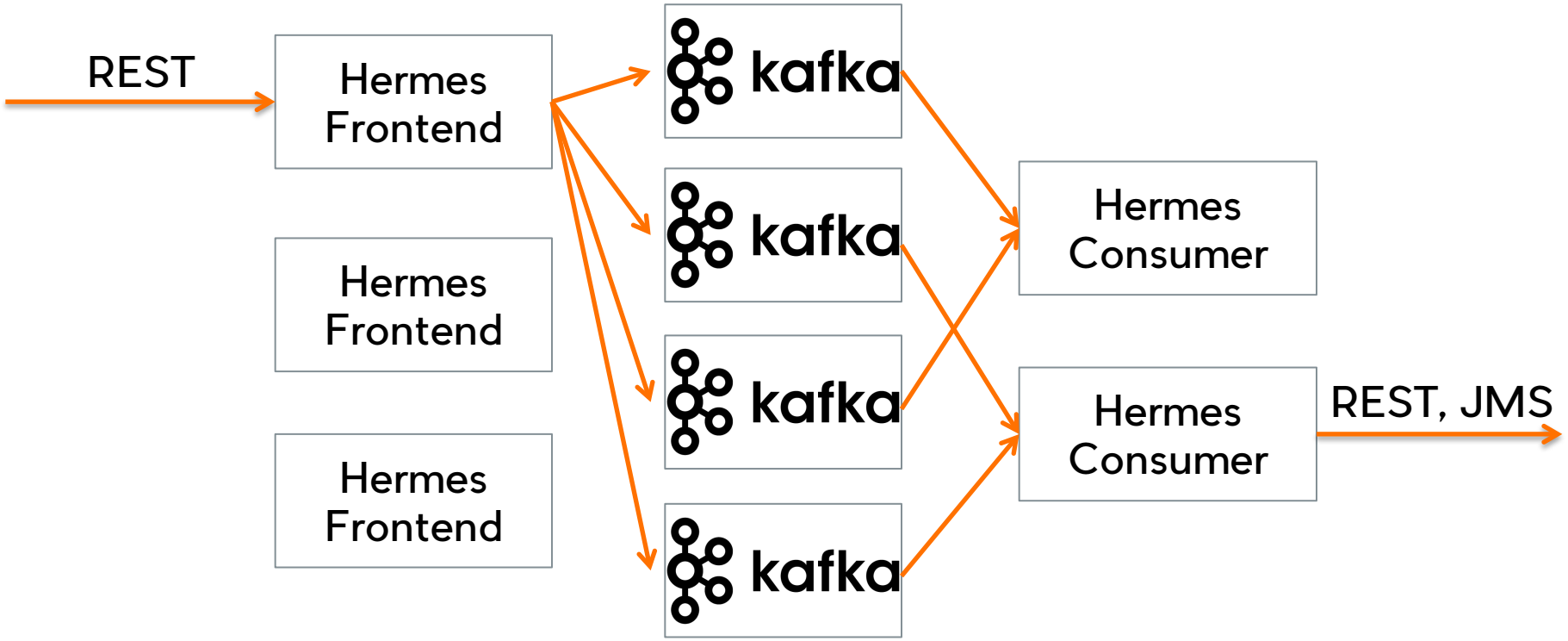
Push vs. Pull

Synchronous world

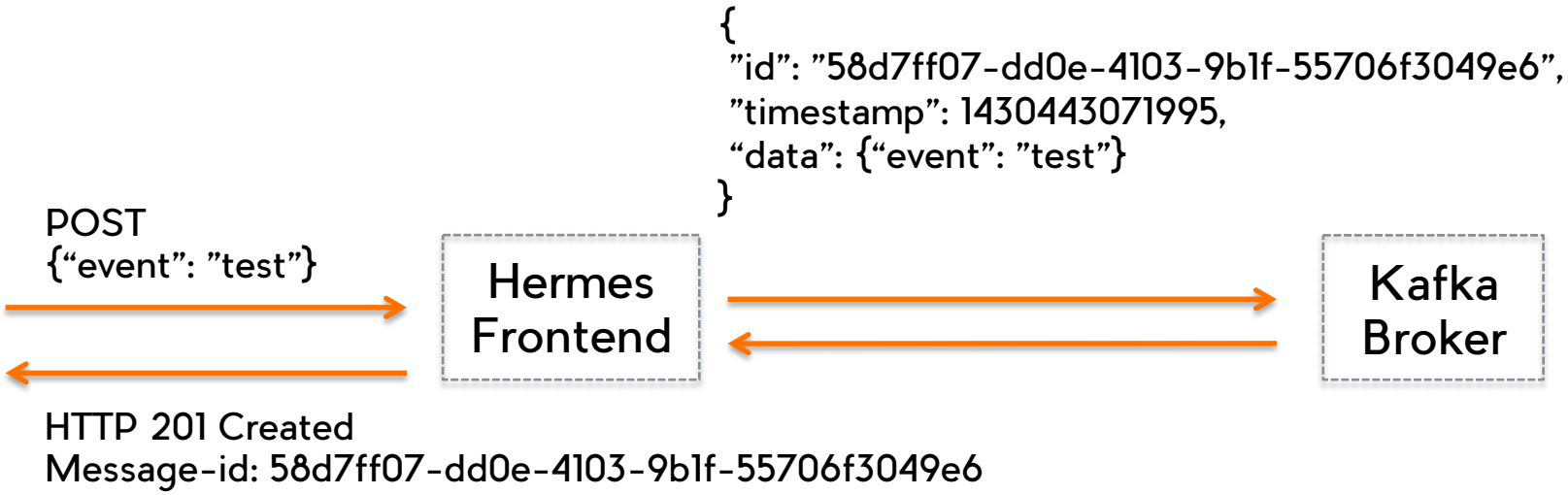
Asynchronous made simple



Hermes

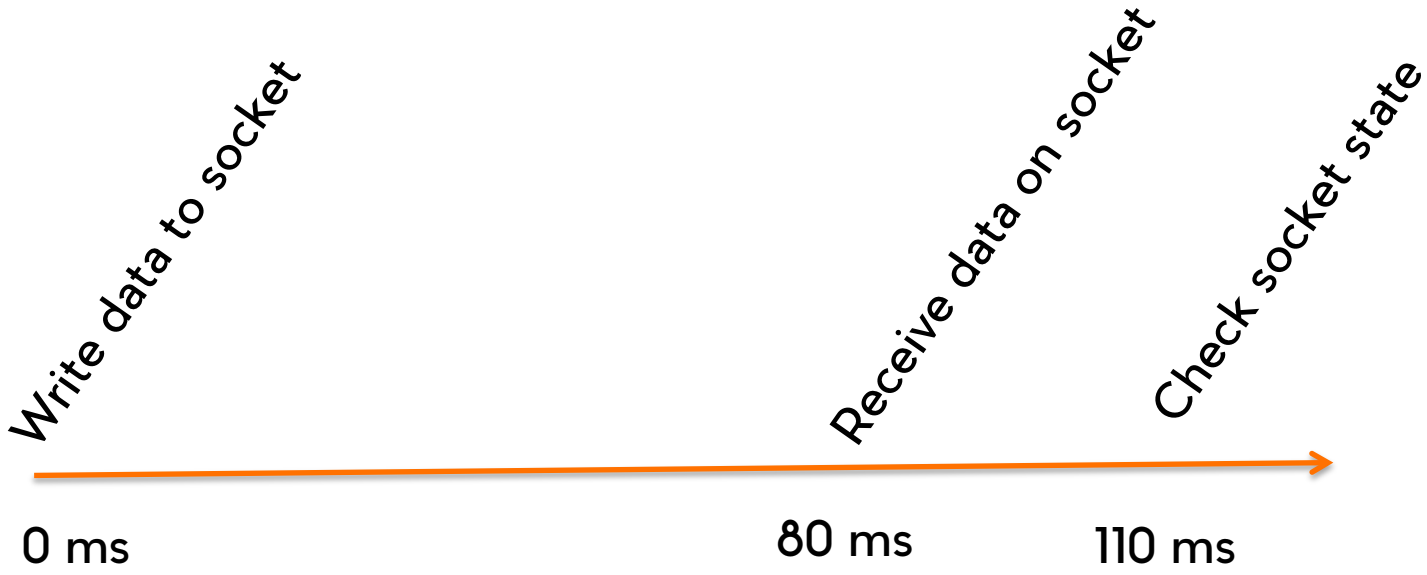

REST
Hermes Frontend
Hermes Frontend
Hermes Frontend
kafka
kafka
kafka
kafka
Hermes Consumer
Hermes Consumer
REST, JMS

Event identification



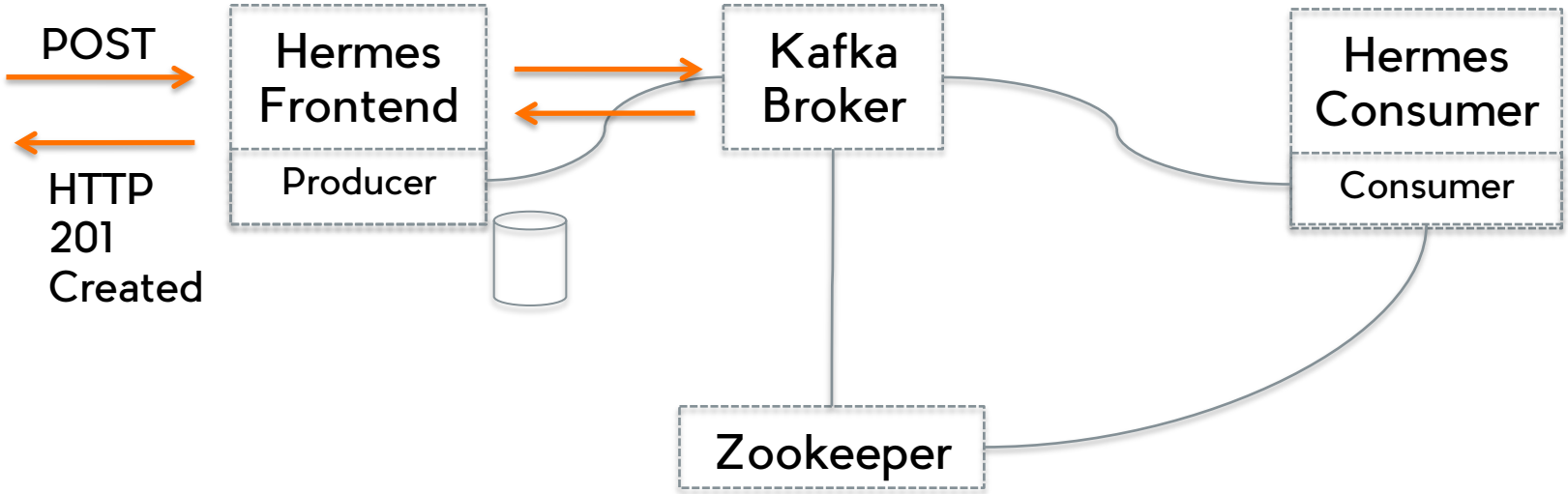
100 ms timeout

Timeout measurement

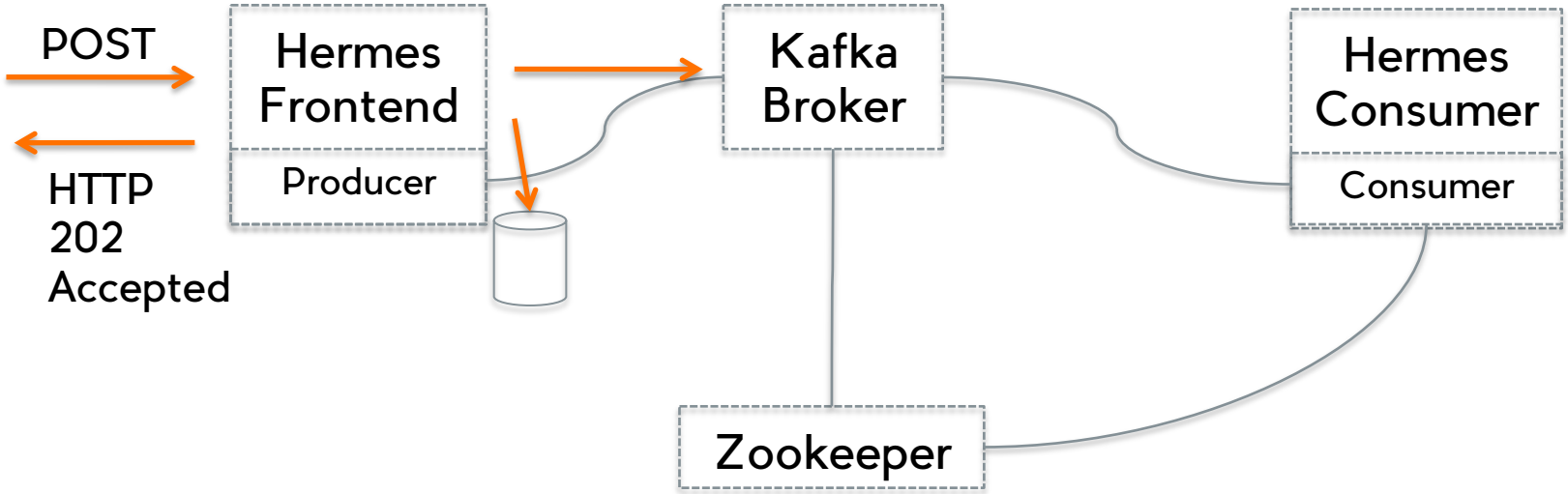


65 ms timeout
(without network)

Normal operation



Slow kafka response

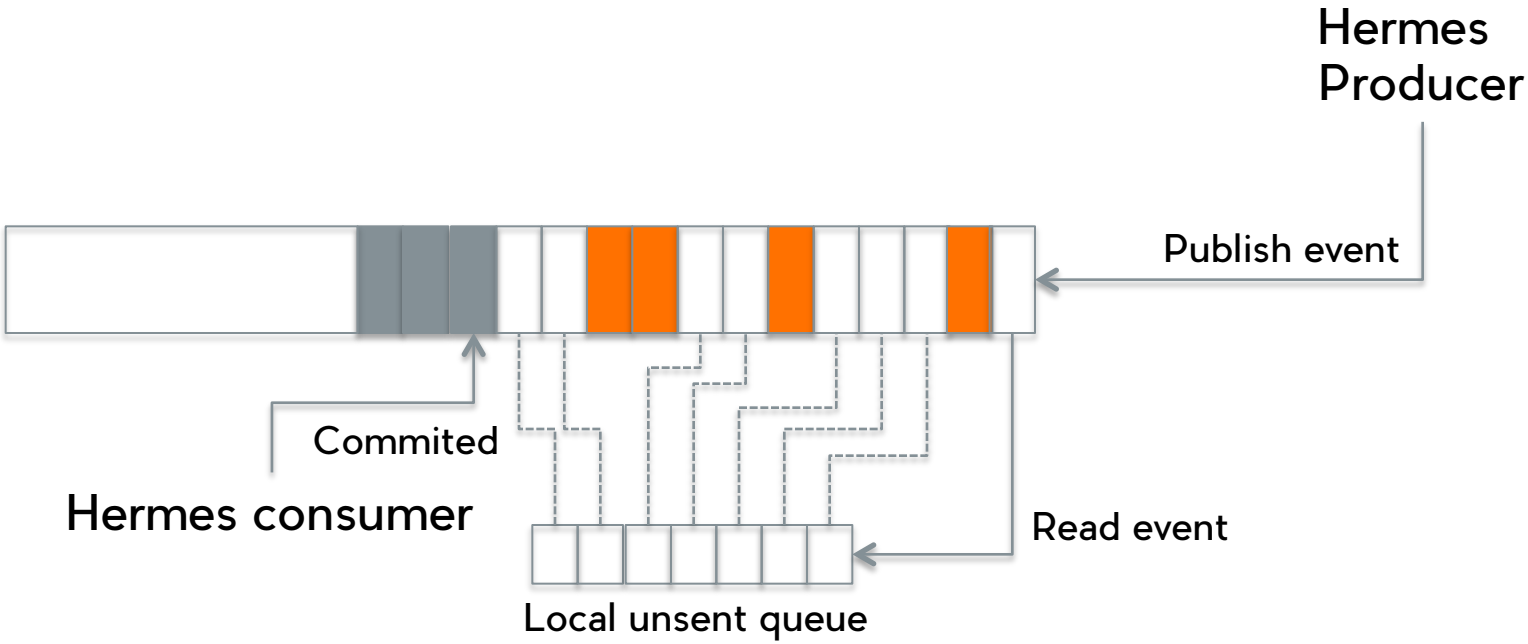


Nagle's algorithm

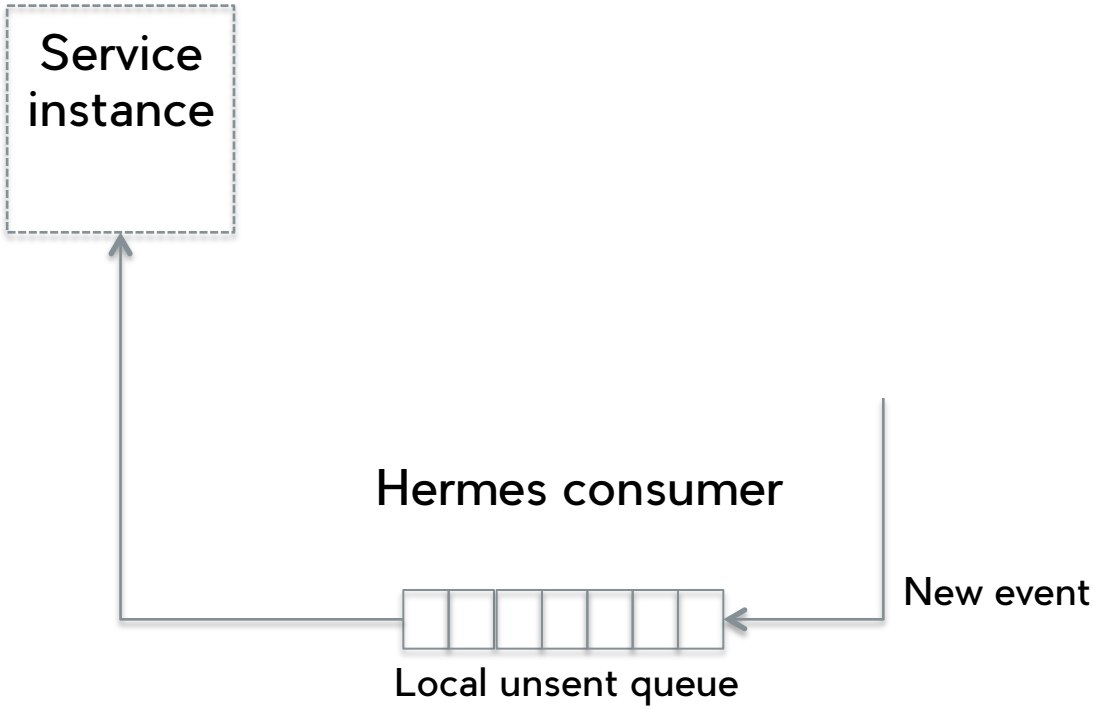
42 ms -> 6 ms

Concatenated events

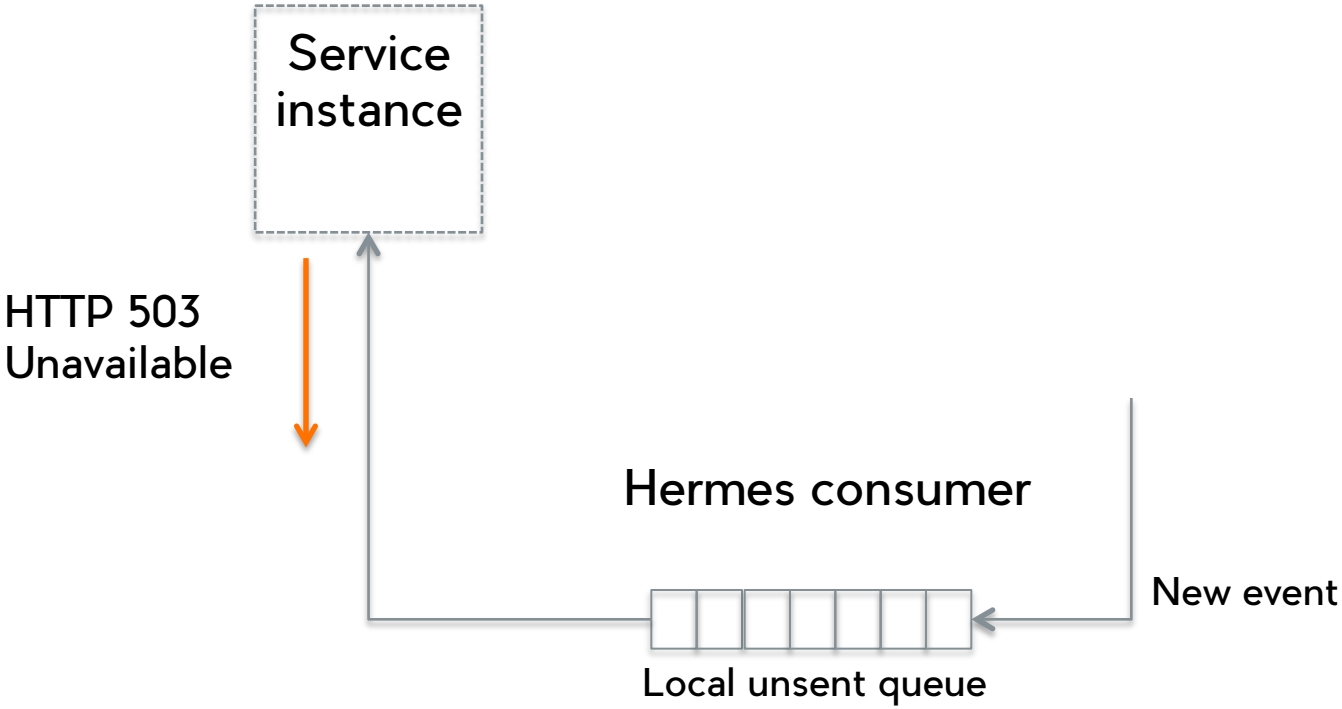
Improved offset management



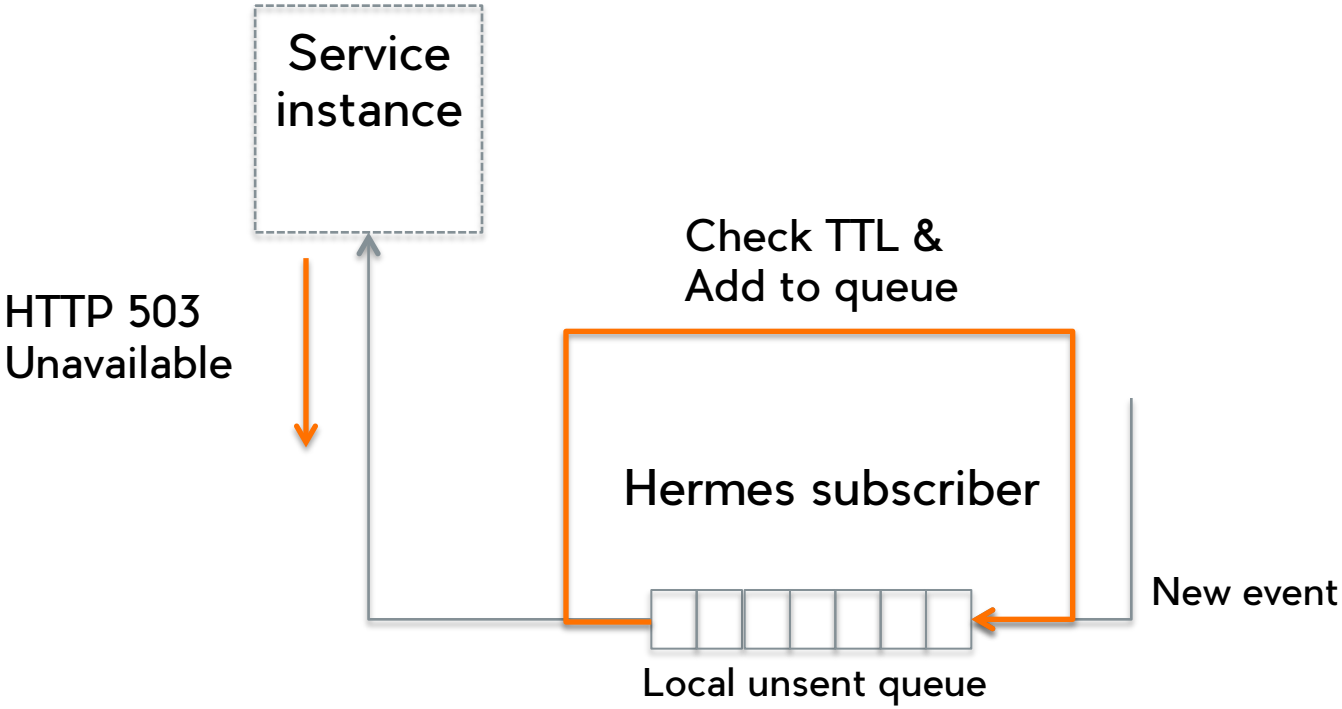
Resend messages



Resend messages

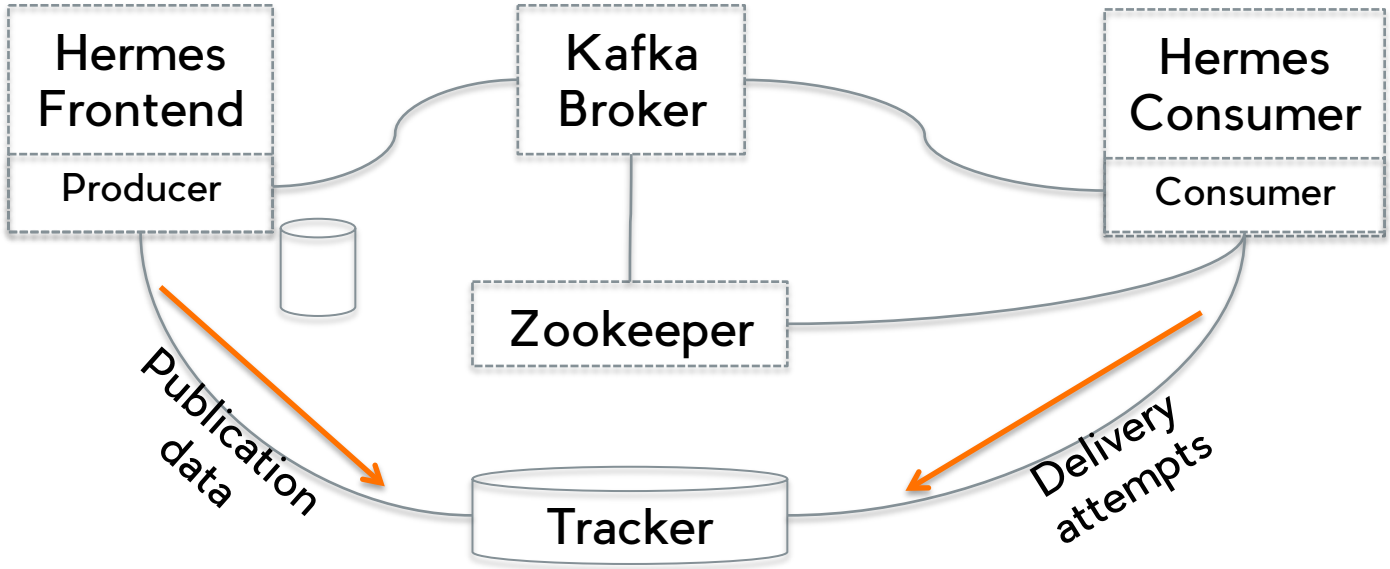


Resend messages



Resend speed

Lost events


Hermes Frontend
Producer
Kafka Broker
Hermes Consumer
Consumer
Zookeeper
Publication data
Tracker
Delivery attempts

Groups and Topics

pl.allegro.hermes.demo.basic

Group

Topic

Security

Authentication and authorization interfaces

You can create any topic in your group

You can publish everywhere

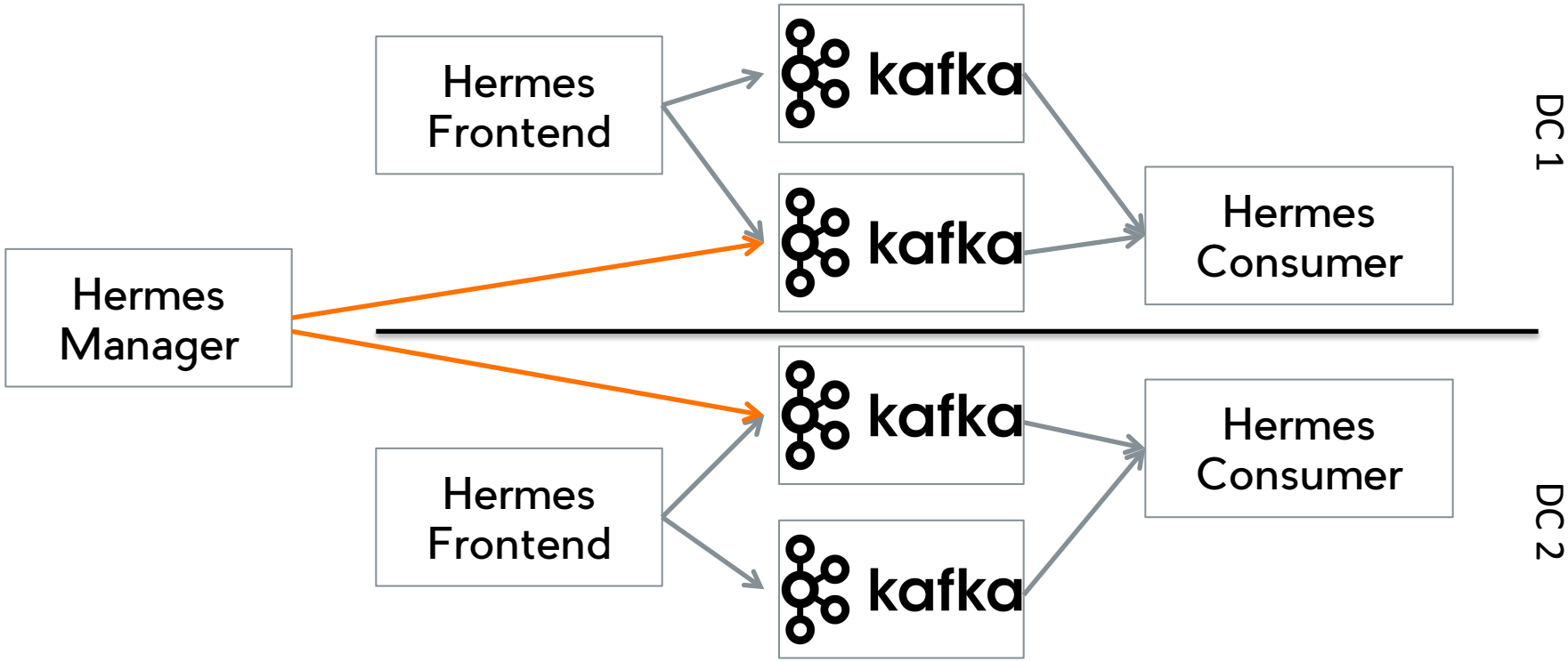
Group owner defines subscriptions

Schema

JSON Schema are defined per topic

Validation on Hermes frontend

Multi datacenter



What we plan to do?

Lower overhead

Avro

RFC 7540

Lower overhead

Avro

RFC 7540 => HTTP/2

Improve security

Service authentication

What's in it for me?

PubSub++ quest

CDz2dj

What’ve we learned?

Tradeoffs

You cannot have exactly one delivery

http://bravenewgeek.com/you-cannot-have-exactly-once-delivery/

Clients

Not all clients are the same.

Early adopters are your friends

You know exactly what features are needed.

One more thing...

allegrotech.io

@debskichris