

VIRTUAL BOOK LAUNCH

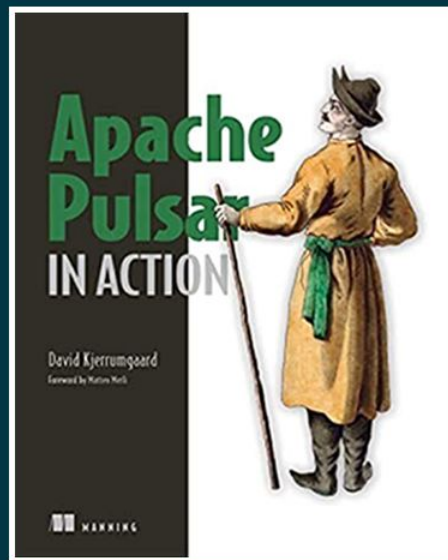
December 9th 12-1 PM PT

Apache Pulsar in Action

Maximize the value of your data with an event-driven architecture.



David Kjerrumgaard
Developer Advocate



Part 1

Getting Started with Apache Pulsar

- Messaging system evolution
- Why Apache Pulsar?

brief contents

PART 1	GETTING STARTED WITH APACHE PULSAR	1
	1 ■ Introduction to Apache Pulsar	3
	2 ■ Pulsar concepts and architecture	38
	3 ■ Interacting with Pulsar	68
PART 2	APACHE PULSAR DEVELOPMENT ESSENTIALS	95
	4 ■ Pulsar functions	97
	5 ■ Pulsar IO connectors	130
	6 ■ Pulsar security	161
	7 ■ Schema registry	191
PART 3	HANDS-ON APPLICATION DEVELOPMENT WITH APACHE PULSAR.....	219
	8 ■ Pulsar Functions patterns	221
	9 ■ Resiliency patterns	241
	10 ■ Data access	271
	11 ■ Machine learning in Pulsar	290
	12 ■ Edge analytics	308

Messaging System Evolution

Message Broker Era

Enterprise Service Bus Era

Distributed Messaging Era

Engineers at IBM
create IBM MQ for
async distributed
computing

ActiveMQ is
open-sourced

1993

1998

2001

2002

JMS standard
published

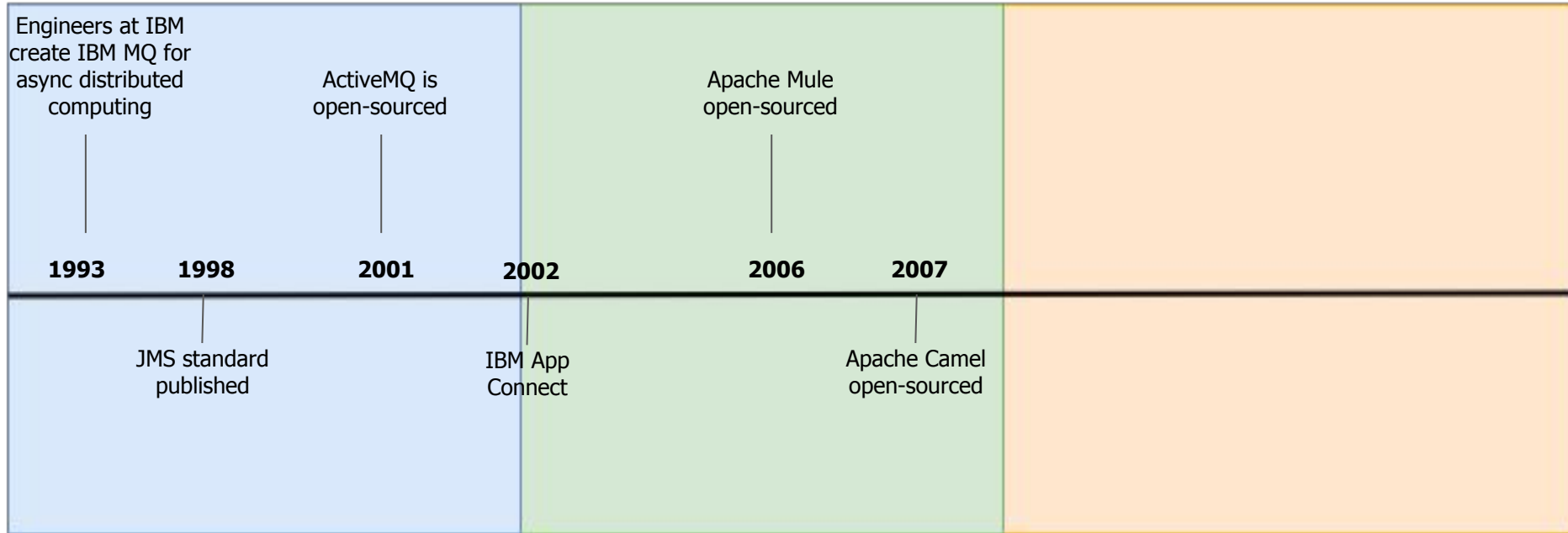
IBM App
Connect

Messaging System Evolution

Message Broker Era

Enterprise Service Bus Era

Distributed Messaging Era

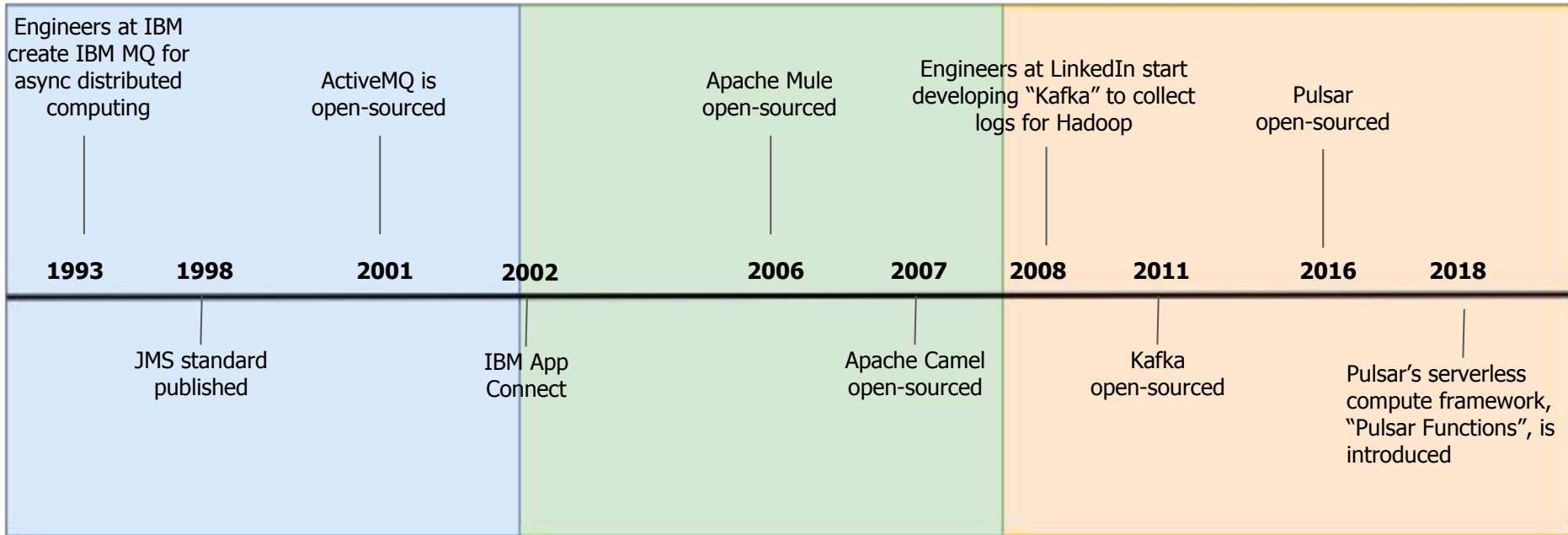


Messaging System Evolution

Message Broker Era

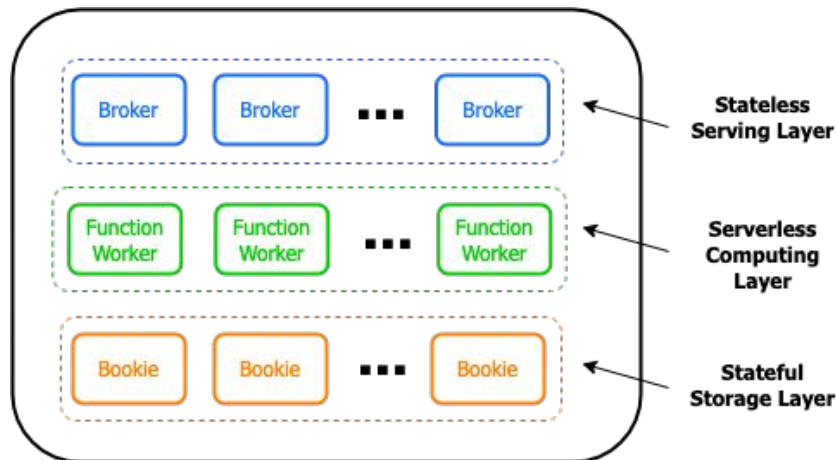
Enterprise Service Bus Era

Distributed Messaging Era



Apache Pulsar Era

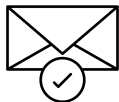
- Serverless computing framework.
- Unbounded storage, multi-tiered architecture, and tiered-storage.
- Streaming & Pub/Sub messaging semantics.
- Multi-protocol support



Why Apache Pulsar?



**Unified
Messaging
Platform**



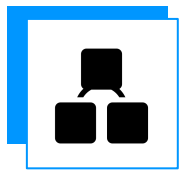
**Guaranteed
Message
Delivery**



Resiliency



**Infinite
Scalability**



Use cases

- Unified Messaging Platform
- AdTech
- Fraud Detection
- Connected Car
- IoT Analytics
- Microservices Development

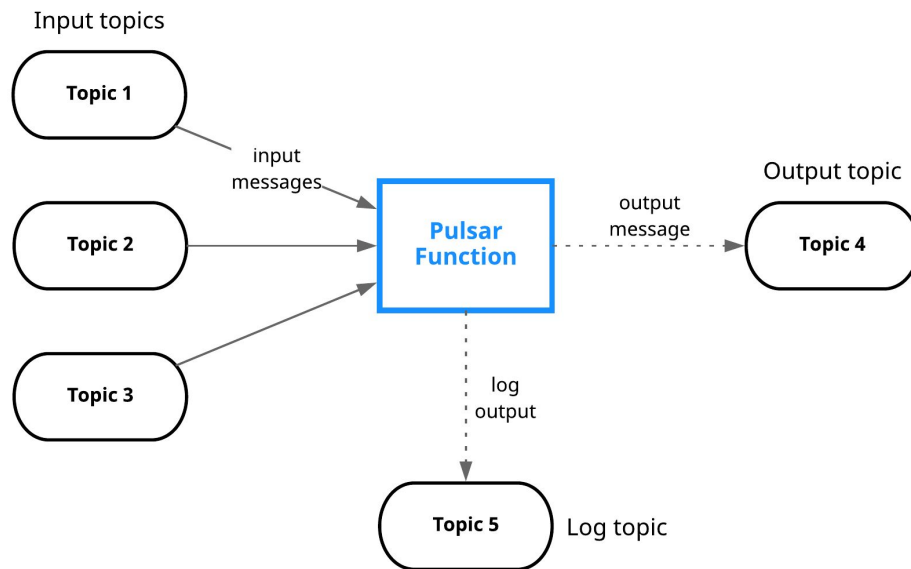
Part 2

Apache Pulsar Development

- Pulsar Functions
- <https://github.com/david-streamlio/pulsar-in-action>
- <https://github.com/david-streamlio/pulsar-in-action-python>
- <https://github.com/david-streamlio/pulsar-in-action-go>

What are Pulsar Functions?

- Lambda-style functions that use Pulsar as the message bus.
- Handles producer/consumer setup
- Applies user supplied business logic against consumed message.



Why Pulsar Functions?

All of this 

```
PulsarClient client = PulsarClient.builder()
    .serviceUrl("pulsar://broker1:6650")
    .build(); // Client discovers all brokers

consumer = client.newConsumer(Schema.STRING)
    .topic("persistent://public/default/test_topic")
    .subscriptionName("my-subscription")
    .subscribe();

Producer<String> producer = client.newProducer(Schema.STRING)
    .topic("persistent://public/default/test_topic")
    .create();

while (true) {
    Message<String> message = consumer.receive();
    String result = doBusinessLogic(message);
    producer.newMessage()
        .value(result)
        .send();
    consumer.acknowledge(message);
}
```

Why Pulsar Functions?

Simplifies to this

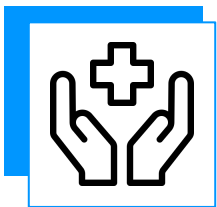


```
import java.util.function.Function;

public class MyFunction implements Function<String, String> {
    public String apply(String input) {
        return doBusinessLogic(input);
    }
}
```

The incoming messages are passed into the function one-by-one

The returned value is automatically published to the output topic



Benefits of Pulsar Functions

- Allow you to focus on the business logic.
- Eliminates boilerplate code.
- Handles message consumption and publication
- No need for another processing framework.
- Can be scaled up independently

Part 3

Hands-On Application
Development with Apache
Pulsar

- Microservices Development
- Function Design Patterns

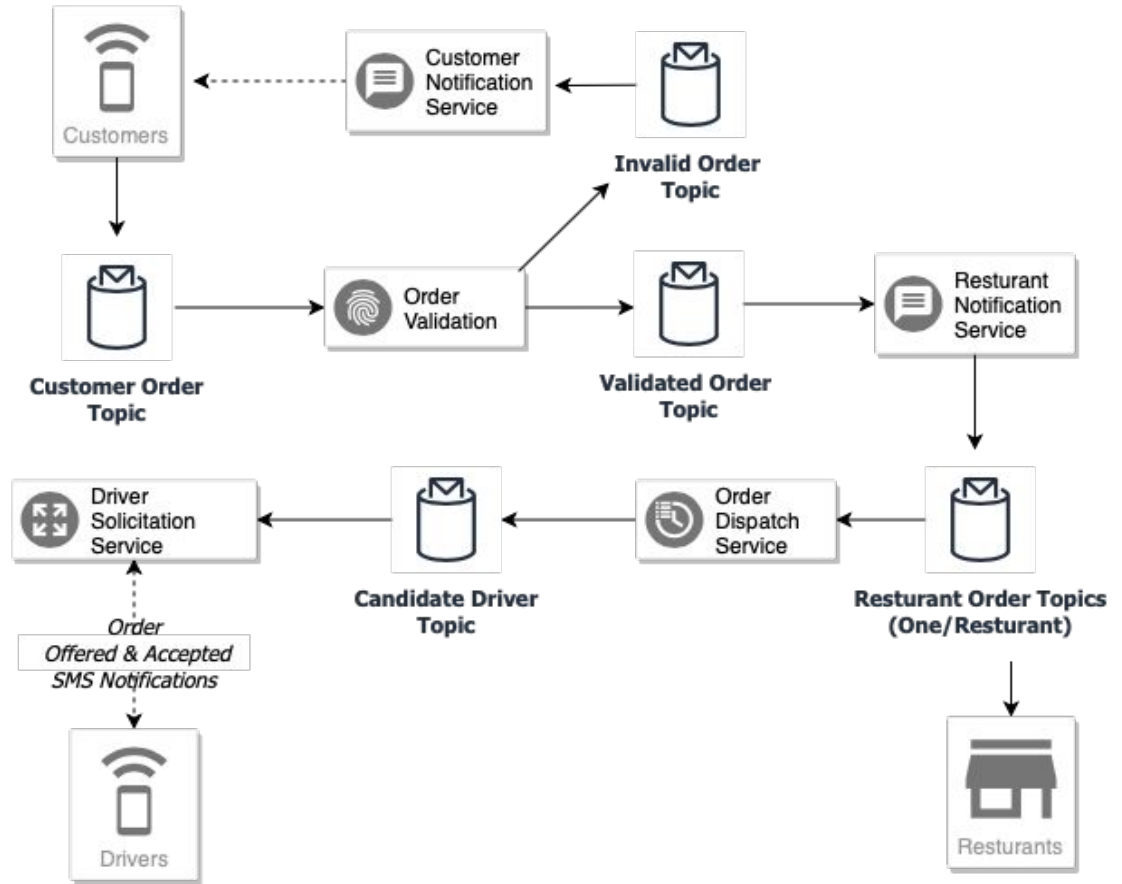
Microservices Goal & Characteristics

- ❑ Highly maintainable and testable
- ❑ Loosely coupled with other services
- ❑ Independently deployable
- ❑ Capable of being developed by a small team. (*Two pizza rule*)

Why Pulsar Functions for Microservices?

Desired Characteristic	Pulsar Functions...
Highly maintainable and testable	Are small pieces of code written in popular languages such as Java, Python, or Go. They can be easily maintained in source control repositories and tested with existing frameworks automatically.
Loosely coupled with other services	Are not directly linked to one another. They communicate via messages.
Independently deployable	Are designed to be deployed independently.
Can be developed by a small team	Are often developed by a single developer.
Inter-service Communication	Support all message patterns, using Pulsar as the underlying message bus.
Deployment & Composition	Can run as individual threads, processes, or K8s pods. You can also deploy multiple Pulsar Functions as a single unit using the Function Mesh.

Event-Based Microservices Application

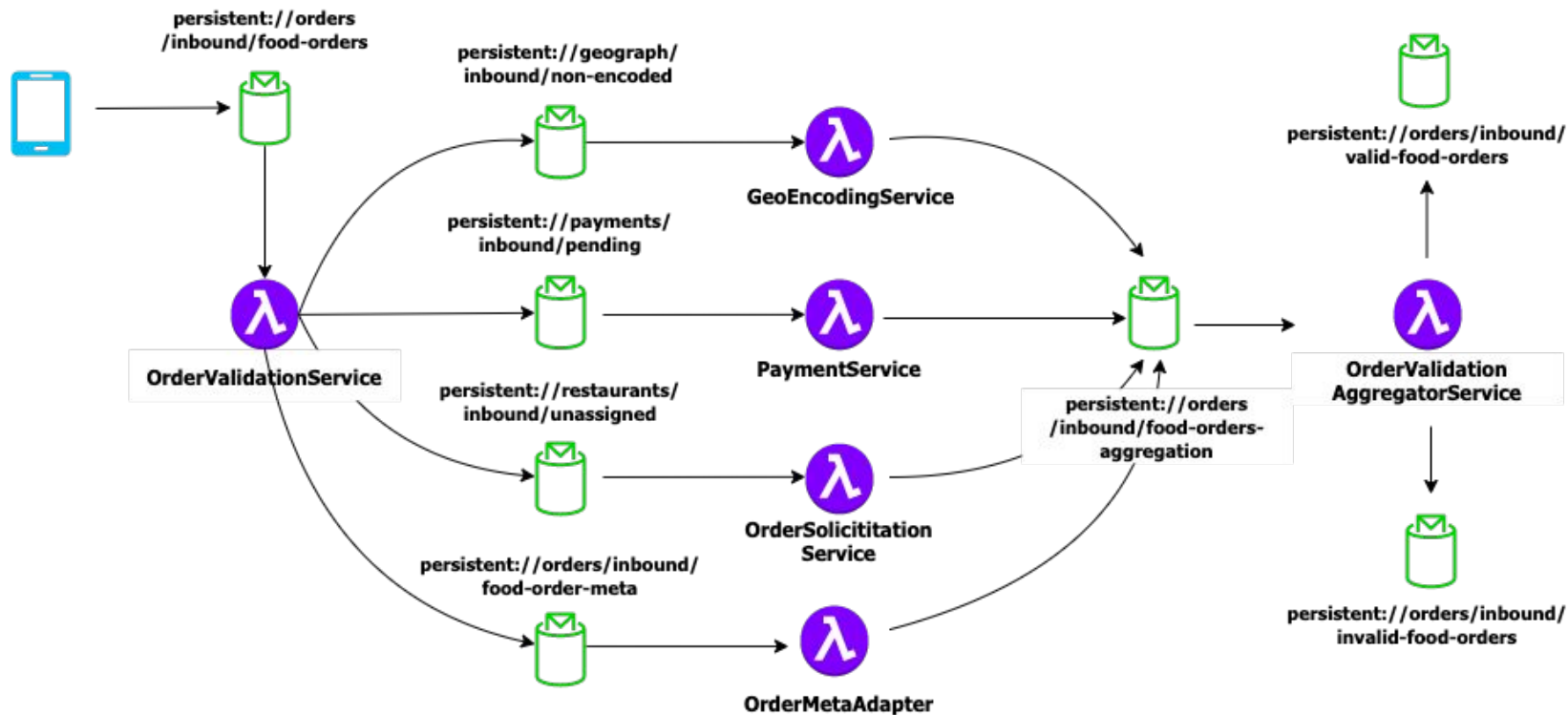


Exercises

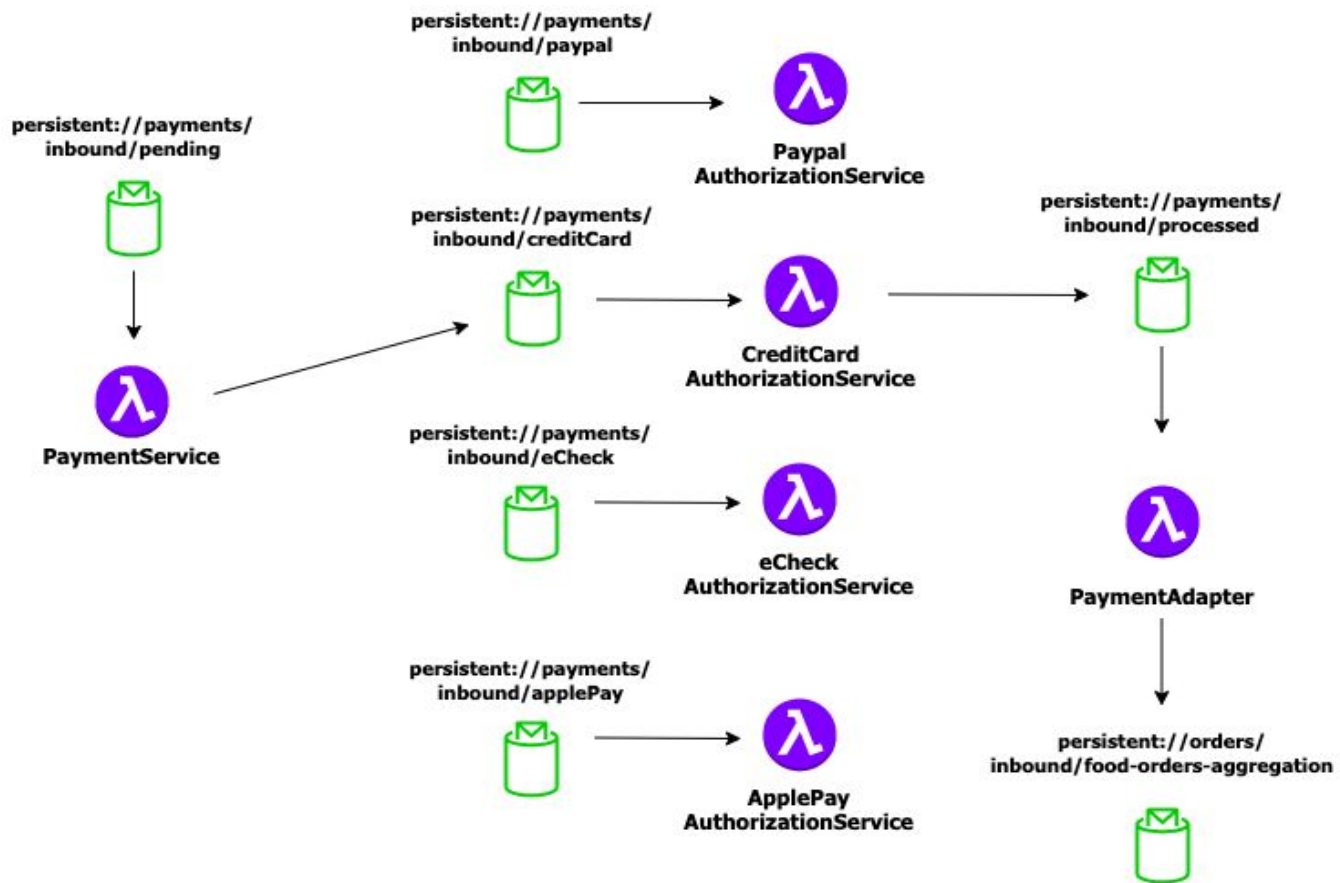
Code walk through of Pulsar Function-based microservices from the book.

- Order Validation Service
- Payment Service

Order Validation



Payment Validation



Live Demo

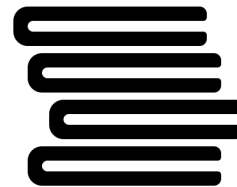
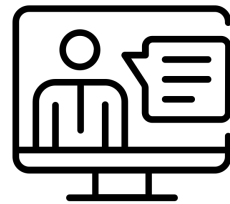
This section focuses on:

- ✓ the development of the order validation
- ✓ use case for the food delivery application featured in the book
- ✓ addresses more-complex use cases including design patterns and resiliency.

<https://github.com/david-streamlio/GottaEat>

What's Next?

Here are resources to continue your journey
with Apache Pulsar





Watch the full series.

- **Part One:** *Developing Event-Driven Microservices with Pulsar*
- **Part Two:** *Stateful Microservices*
- **Part Three:** *Streaming Analytics Using FlinkSQL*

Hosts: David Kjerrumgaard, Ioannis Polyzos, Addison Higham, and Tim Spann

Microservices Webinar Series

[Watch the Series](#)



StreamNative Ambassador Program 2022

[Learn More](#)



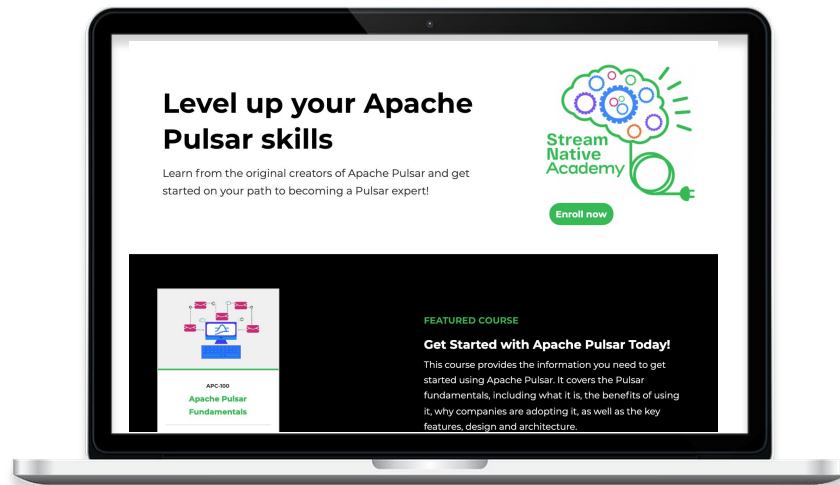
Pulsar vs. Kafka:

A More Accurate Perspective from Use Cases and Community to Features and Performance

[Download the Whitepaper](#)

Now Available On-Demand Pulsar Training

[Start Today](#)



Let's Keep in Touch!



David Kjerrumgaard

Developer Advocate



@davidkjerrumga1



<https://www.linkedin.com/in/davidkjerrumgaard>



<https://github.com/davidkjerrumgaard>



Timothy Spann

Developer Advocate



<https://www.linkedin.com/in/timothyspann>



<https://github.com/tspannhw>

Questions