Building event-driven Microservices with Kafka Ecosystem

Guido Schmutz
London, 30.5.2018

@gschmutz
guidoschmutz.wordpress.com
Guido Schmutz

Working at Trivadis for more than 21 years
Oracle ACE Director for Fusion Middleware and SOA
Consultant, Trainer Software Architect for Java, Oracle, SOA and Big Data / Fast Data
Head of Trivadis Architecture Board
Technology Manager @ Trivadis

More than 30 years of software development experience

Contact: guido.schmutz@trivadis.com
Blog: http://guidoschmutz.wordpress.com
Slideshare: http://www.slideshare.net/gschmutz
Twitter: gschmutz

Building event-driven Microservices with Kafka Ecosystem
Agenda

1. Where do we come from?
2. What are Microservices?
3. Why not Event Driven?
4. What about streaming sources?
5. What about integrating legacy applications?
6. What about (historical) data analytics?
7. Why Kafka for Event-Driven Microservices?
8. Summary
Where do we come from?
Traditional Approach

Shop Rich UI

Shop Backend Application

Customer Fat Client App

- Customer UI
- Customer BO

- GUI
- Data

Shop UI

Search Facade

Order Facade

- Order DAO
- Order Data

- GUI
- Business

UI Logic

Shared Database

- Sync request/response

Building event-driven Microservices with Kafka Ecosystem
SOA Approach

Contract-first Web Services

Technical layers offer their own interfaces

Reuse on each level

Lower layer often wraps legacy code

Building event-driven Microservices with Kafka Ecosystem
Virtualized SOA Approach

Building event-driven Microservices with Kafka Ecosystem
What are Microservices?
What are Microservices?

- Tightly Scoped behind clear interfaces
- Responsible for managing their own data (not necessarily the infrastructure)
- Should be highly decoupled
- Independently deployable, self-contained and autonomous
- SOA done right ?!
Microservice Approach

Building event-driven Microservices with Kafka Ecosystem
Microservice Approach with API Gateway

Building event-driven Microservices with Kafka Ecosystem
Synchronous World of Request-Response leads to tight, point-to-point couplings

Problem in lower end of chain have a ripple effect on other service
- Crash of service
- Overloaded service / slow response time
- Change of interface

Building event-driven Microservices with Kafka Ecosystem
Why not Event-Driven?
3 mechanisms through which services interact

- **Request-Driven (Imperative)**
  - Command: "Order iPad"
  - Query: "Retrieve my Orders"
  - Service Logic: boolean order(IPad)
  - Service State: List<Orders> getAllOrders(for)

- **Event Driven (Functional)**
  - Event Consume: OrderEvent{iPad}
  - Event Publish: OrderValidatedEvent{iPad}

Building event-driven Microservices with Kafka Ecosystem
Event-Driven (Async) Microservice Approach

- **Shop Web App**
  - Shop UI
  - Shop GUI

- **API Gateway**

- **Customer Microservice**
  - Customer API
  - Customer Logic

- **Order Microservice**
  - Order API
  - Order Logic

- **Product Microservice**
  - Product API
  - Product Logic

- **Stock Microservice**
  - Stock API
  - Stock Logic

- **Event Store**

- **Sync request/response**
- **Async, event pub/sub**
- **Async request/response**

Building event-driven Microservices with Kafka Ecosystem
What about streaming sources?
How to work with streaming data sources

Building event-driven Microservices with Kafka Ecosystem
Streaming Analytics Architecture

Building event-driven Microservices with Kafka Ecosystem
What about integrating legacy applications?
Integrate existing systems through CDC

Capture changes directly on database

Change Data Capture (CDC) => think like a global database trigger

Transform existing systems to event producer

Building event-driven Microservices with Kafka Ecosystem
Integrate existing systems through CDC

- Billing & Ordering
- CRM / Profile
- Marketing Campaigns

- Location
- Mobile Apps
- Social
- Weather Data
- Click stream
- Sensor Data

- Event Store
- Change Data Capture
- Event Stream

- Stream Processing Cluster
  - Stream Processor
  - Reference / Models
  - Dashboard
  - Results

- Microservice Cluster
  - Microservice
  - State
  - API

- Building event-driven Microservices with Kafka Ecosystem
What about (historical) data analytics?
Building event-driven Microservices with Kafka Ecosystem
Why Kafka for Event-Driven Microservices?
Apache Kafka – A Streaming Platform

High-Level Architecture

- Producer
- Producer
- Producer

Zookeeper Ensemble

Kafka Cluster

- Broker 1
- Broker 2
- Broker 3

Consumer
Consumer
Consumer

Distributed Log at the Core

Logs do not (necessarily) forget

Scale-Out Architecture

- Kafka Cluster
- Consumer Group 1
- Consumer Group 2

Producer 1
Producer 2
Producer 3

Broker 1
Broker 2
Broker 3

Rewind
Replay

Service

Logic
State

Building event-driven Microservices with Kafka Ecosystem
Apache Kafka – Schema Registry
Apache Kafka – scalable message processing and more!

Building event-driven Microservices with Kafka Ecosystem
Change Data Capture (CDC)

Building event-driven Microservices with Kafka Ecosystem
Building event-driven Microservices with Kafka Ecosystem
Hold Data for Long-Term – Data Retention

1. Never

2. Time based (TTL)
   log.retention.{ms | minutes | hours}

3. Size based
   log.retention.bytes

4. Log compaction based
   (entries with same key are removed):

   kafka-topics.sh --zookeeper zk:2181 \
   --create --topic customers \
   --replication-factor 1 \
   --partitions 1 \
   --config cleanup.policy=compact
# Topic Viewed as Event Stream or State Stream (Change Log)

<table>
<thead>
<tr>
<th>Event Stream</th>
<th>State Stream (Change Log Stream)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-10-02 11, Peter, Muster, 3010, Berne</td>
<td>2015-10-02 11, Peter, Muster, 3010, Berne</td>
</tr>
<tr>
<td>2016-10-04 12, Paul, Steffen, 8001, Zurich</td>
<td>2016-10-04 12, Paul, Steffen, 8001, Zurich</td>
</tr>
<tr>
<td>2016-12-02 21, Lisa, Meier, 3043, Ittigen</td>
<td>2016-12-02 21, Lisa, Meier, 3043, Ittigen</td>
</tr>
<tr>
<td>2017-05-03 11, Peter, Muster, 3015, Berne</td>
<td>2017-05-03 11, Peter, Muster, 3015, Berne</td>
</tr>
<tr>
<td>2017-05-03 21, Lisa, Steffen, 8001, Zurich</td>
<td>2017-05-03 21, Lisa, Steffen, 8001, Zurich</td>
</tr>
<tr>
<td>2017-07-03 11, Peter, Muster, 3052, Zollikofen</td>
<td>2017-07-03 11, Peter, Muster, 3052, Zollikofen</td>
</tr>
</tbody>
</table>
Keep Topics in Compacted Form

Building event-driven Microservices with Kafka Ecosystem
Keep Topics both in Original and Compacted Form

Building event-driven Microservices with Kafka Ecosystem
Enrich Stream with Static Data with Kafka Streams

Building event-driven Microservices with Kafka Ecosystem
Summary

Building event-driven Microservices with Kafka Ecosystem
Summary

- service autonomy is key in a Microservices Architecture!
- not all communication need to be synchronous => separate into
  - commands
  - events
  - queries
- Kafka is well suited as an event broker / event store
  - brings many more interesting features beyond just “message passing”
References

Microservices Blog Series, Ben Stopford, Confluent:
• [https://www.confluent.io/blog/tag/microservices](https://www.confluent.io/blog/tag/microservices)

Apache Kafka for Microservices: A Confluent Online Talk Series:
• [https://www.confluent.io/landing-page/microservices-online-talk-series/](https://www.confluent.io/landing-page/microservices-online-talk-series/)

Turning the database inside-out with Apache Samza, Martin Kleppmann, Con

Event sourcing, CQRS, stream processing and Apache Kafka: What’s the connection?, Neha Narkhede, Confluent:

Immutability Changes Everything, Pat Helland, Salesforce:

Commander: Better Distributed Applications through CQRS and Event Sourcing, Bobby Calderwood:
• [https://www.youtube.com/watch?v=B1-gS0oEtYc](https://www.youtube.com/watch?v=B1-gS0oEtYc)
Technology on its own won't help you. You need to know how to use it properly.