In the Eventual Consistency of Succeeding at Microservices

Kenny Bastani
Spring Developer Advocate

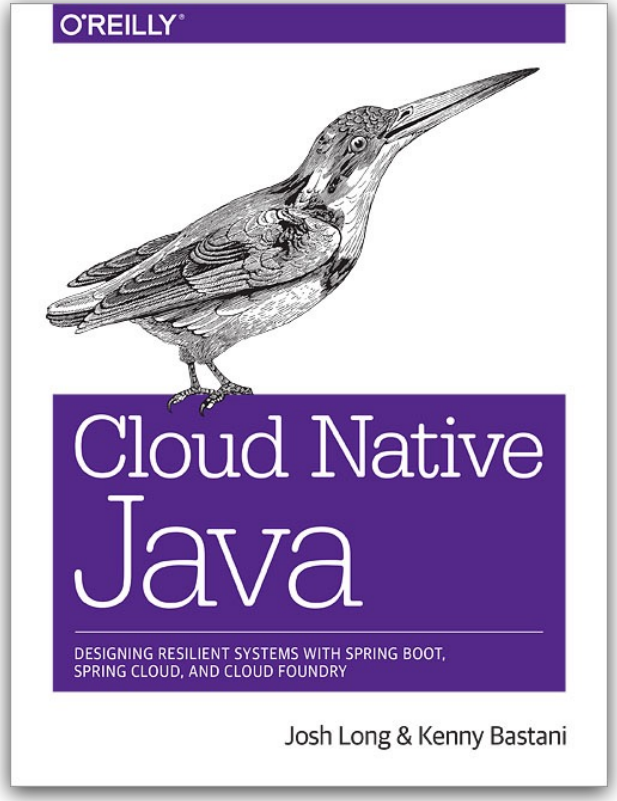
GOTO Chicago 2017

Kenny Bastani

Spring Developer Advocate

spring io

Pivotal



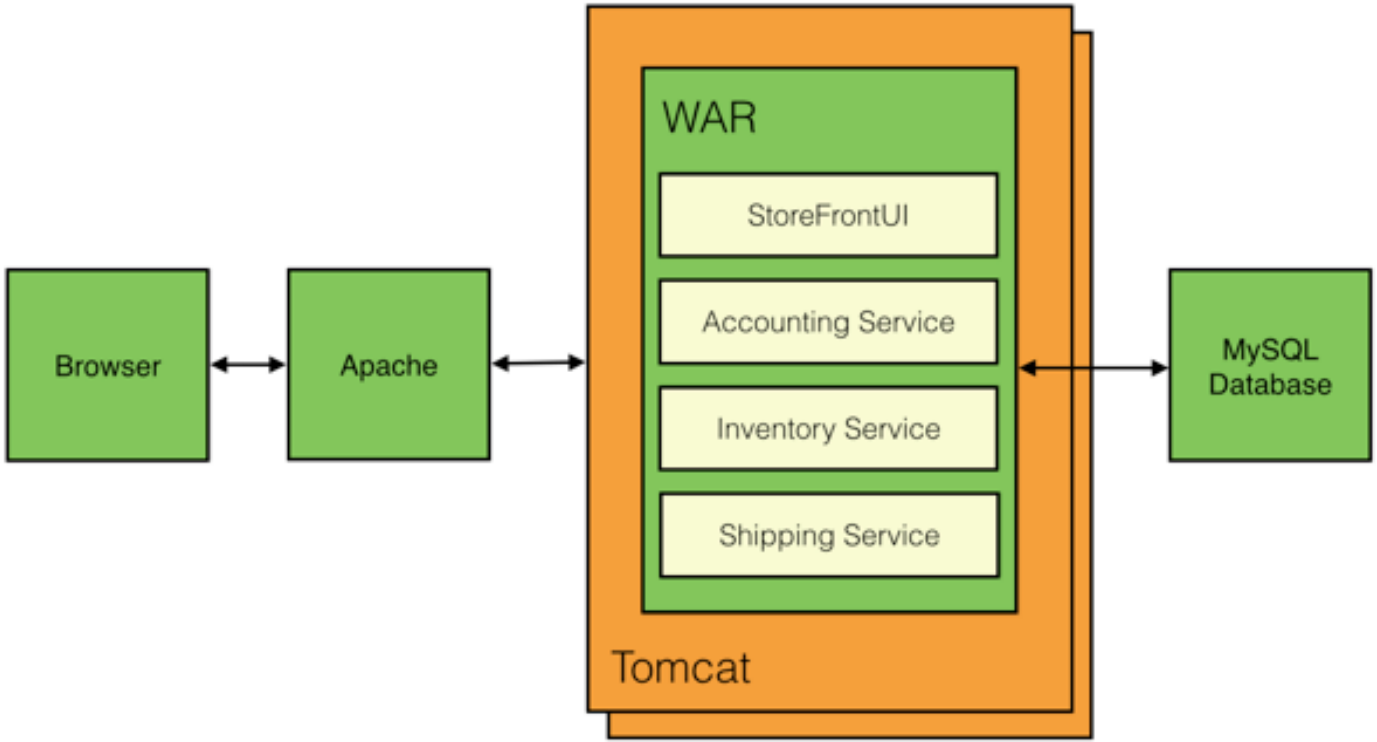
Agenda

	Agenda
1	Monolith to Microservices
2	Online store example
3	Spring Boot
4	Event-driven microservices

Microservices

How did we get there?

We started with the monolith...



Monolith problems...

Slows our velocity getting into production.

Monolith problems...

Engineers take too long to ramp up.

Monolith problems...

The code base becomes too large for any one person to reason about.

Monolith problems...

Development *creates* change.

DBAs *resist* change.

Monolith problems...

Coordinated releases batch many changes together from different teams.

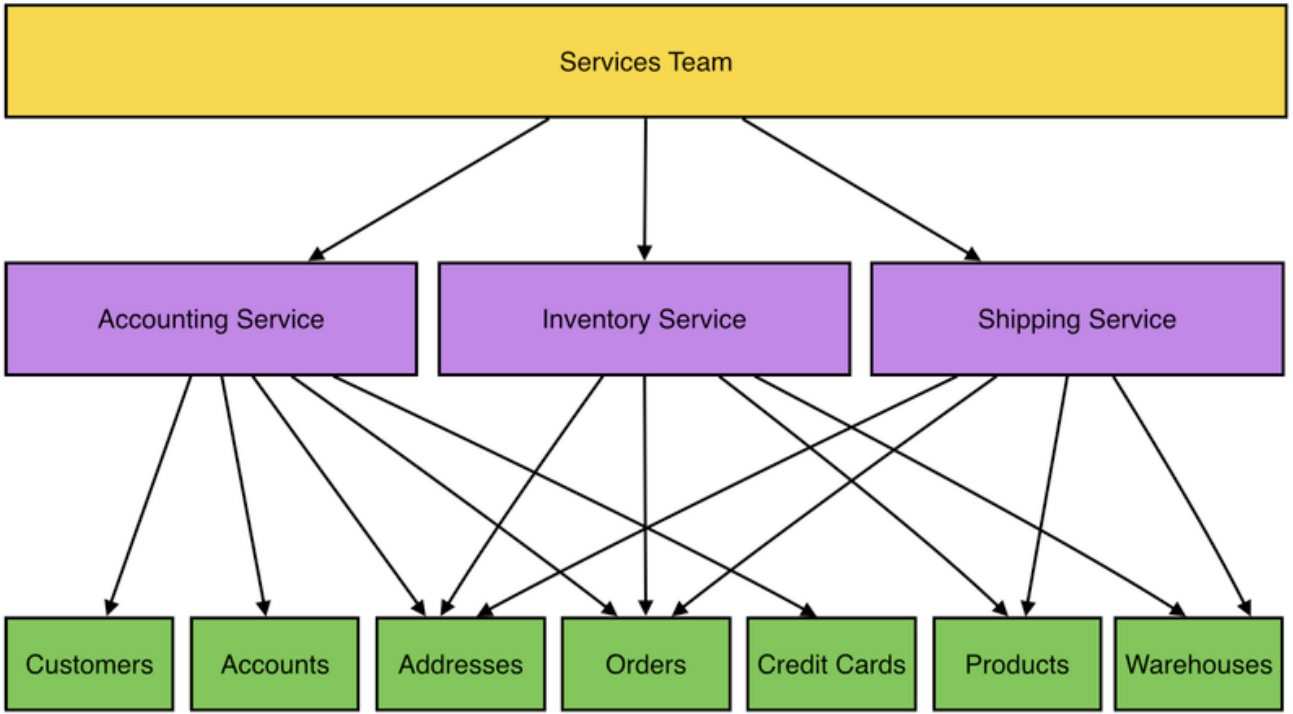
Monolith problems...

Operations drives the runtime environment of applications.

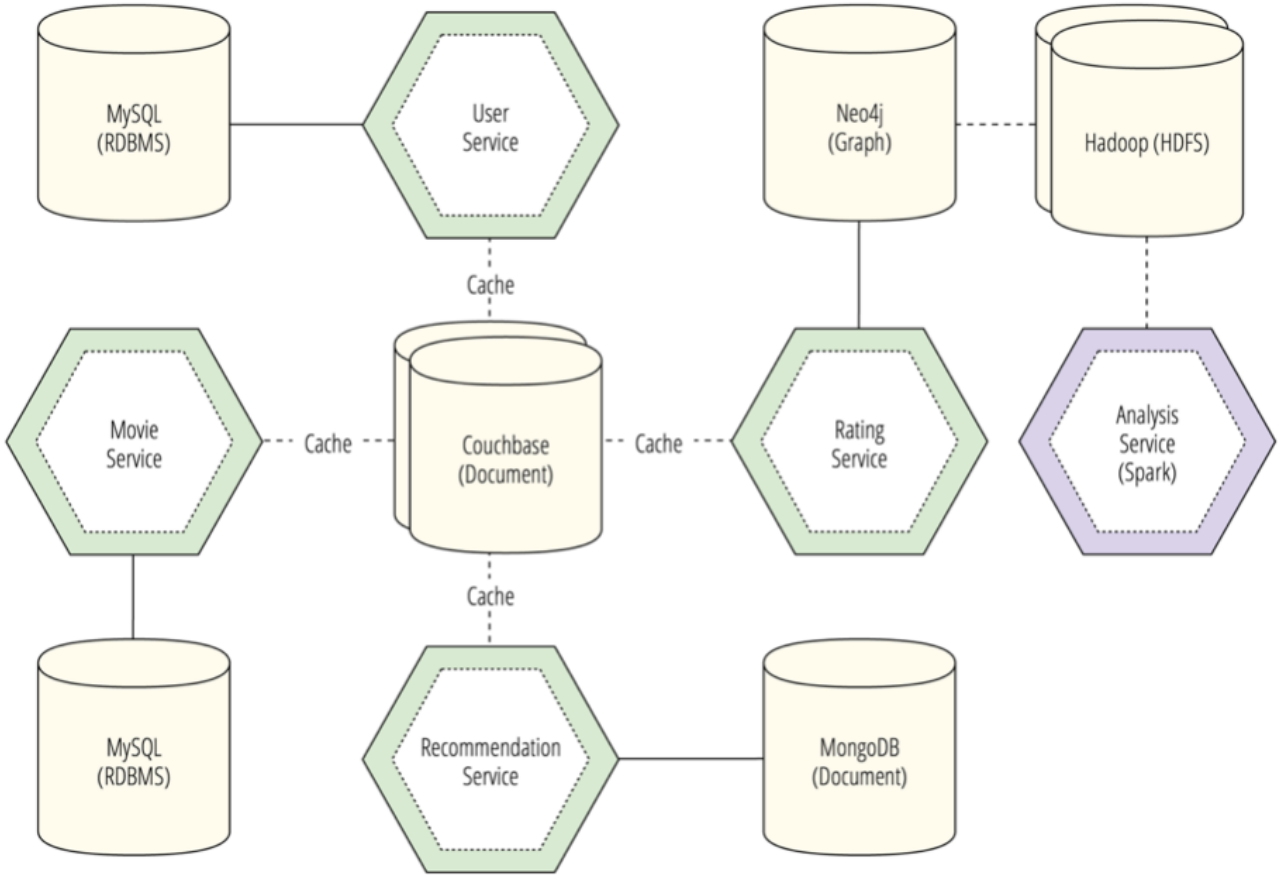
Monolith problems...

Deploy *everything* at once,
or *nothing* at all.

We then moved towards SOA...



We have now arrived at microservices…



Online Store Example

Cloud native application as microservices

Cloud Native Outfitters

online-store-web.cfapps.io/index.html#/products/SKU-34563

Search

Cloud Native Outfitters

Welcome, John Doe! Orders Settings Cart Logout

Home / Products / Like a BOSH (T-Shirt, Women's Medium)

Summary

Product name:

Like a BOSH (T-Shirt, Women's Medium)

Price:

14.99

Product Details

The BOSH Outer Shell (**BOSH**) is an elegant release engineering tool for a more virtualized cloud-native age. The feeling of spinning up a highly available distributed system of VMs is second only to the feeling of frequently pushing code to production. Show the cloud *who's BOSH* with this stylish cloud native ops tee.

Cloud Native Tee Collection
99% YAML, 11% CLI
BOSH CCK *recommended*
4 nines of *re-washability*

Price: $14.99

Product number: SKU-34563

0 Add to Cart

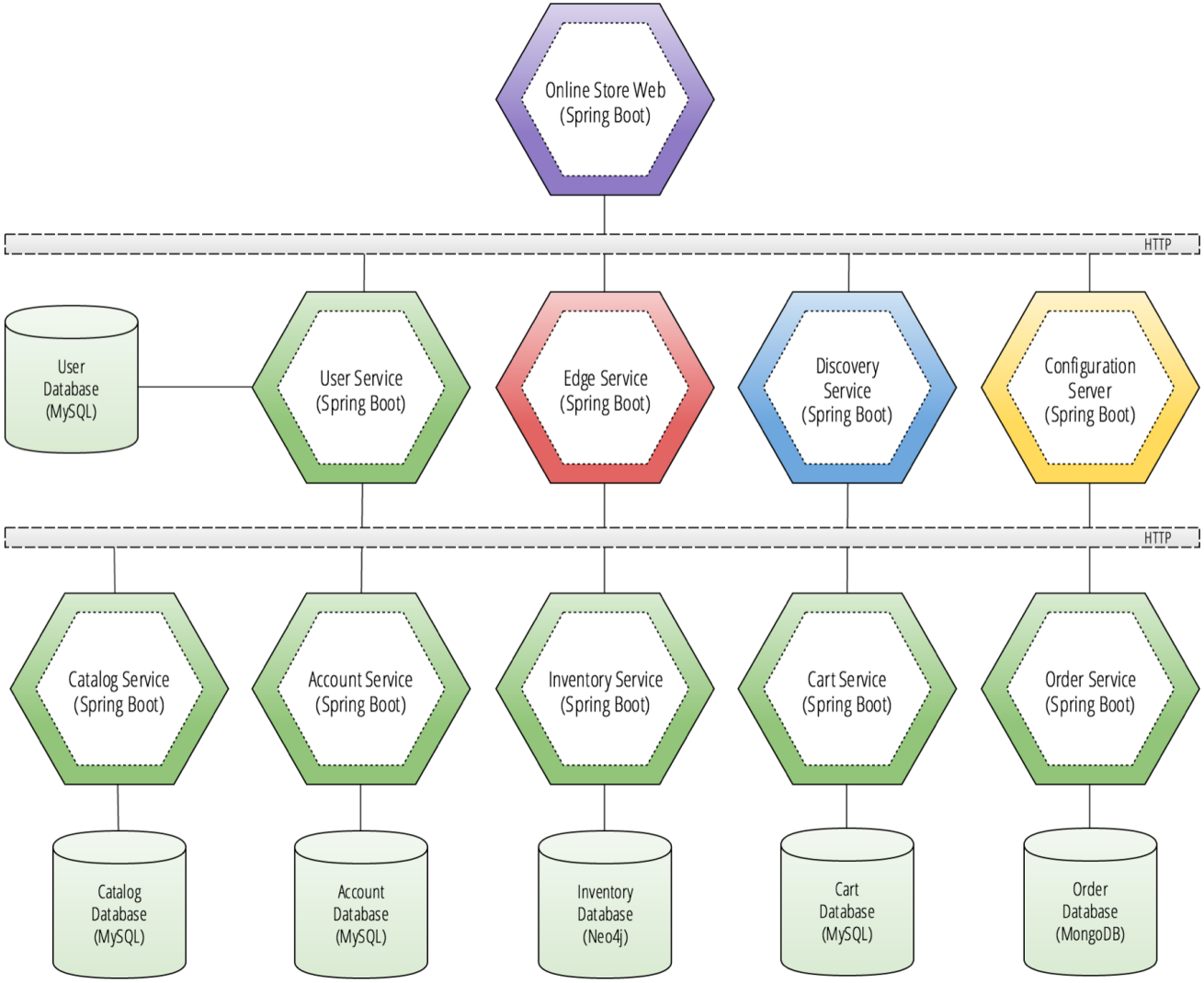
Related

Best. Cloud. Ever. (T-Shirt, Men's

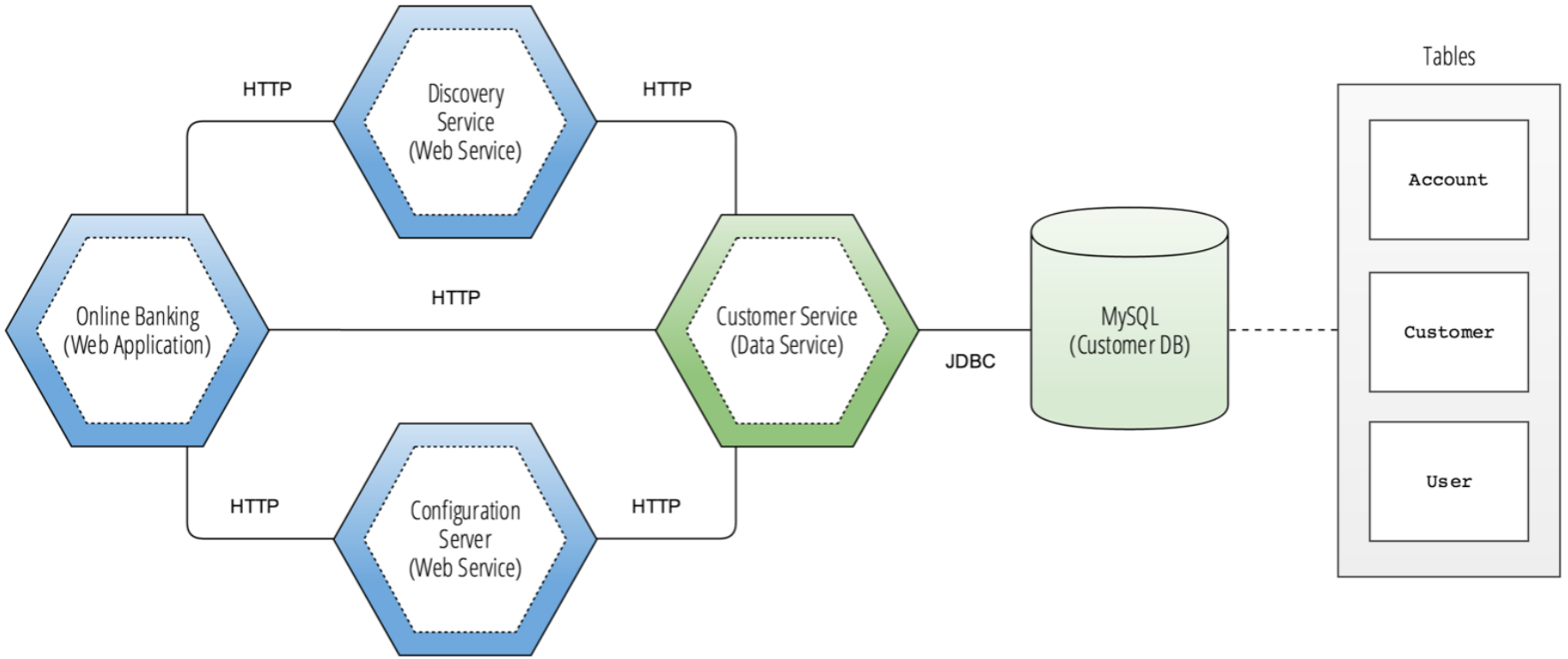
We're gonna need a bigger VM

Like a BOSH (T-Shirt, Women's

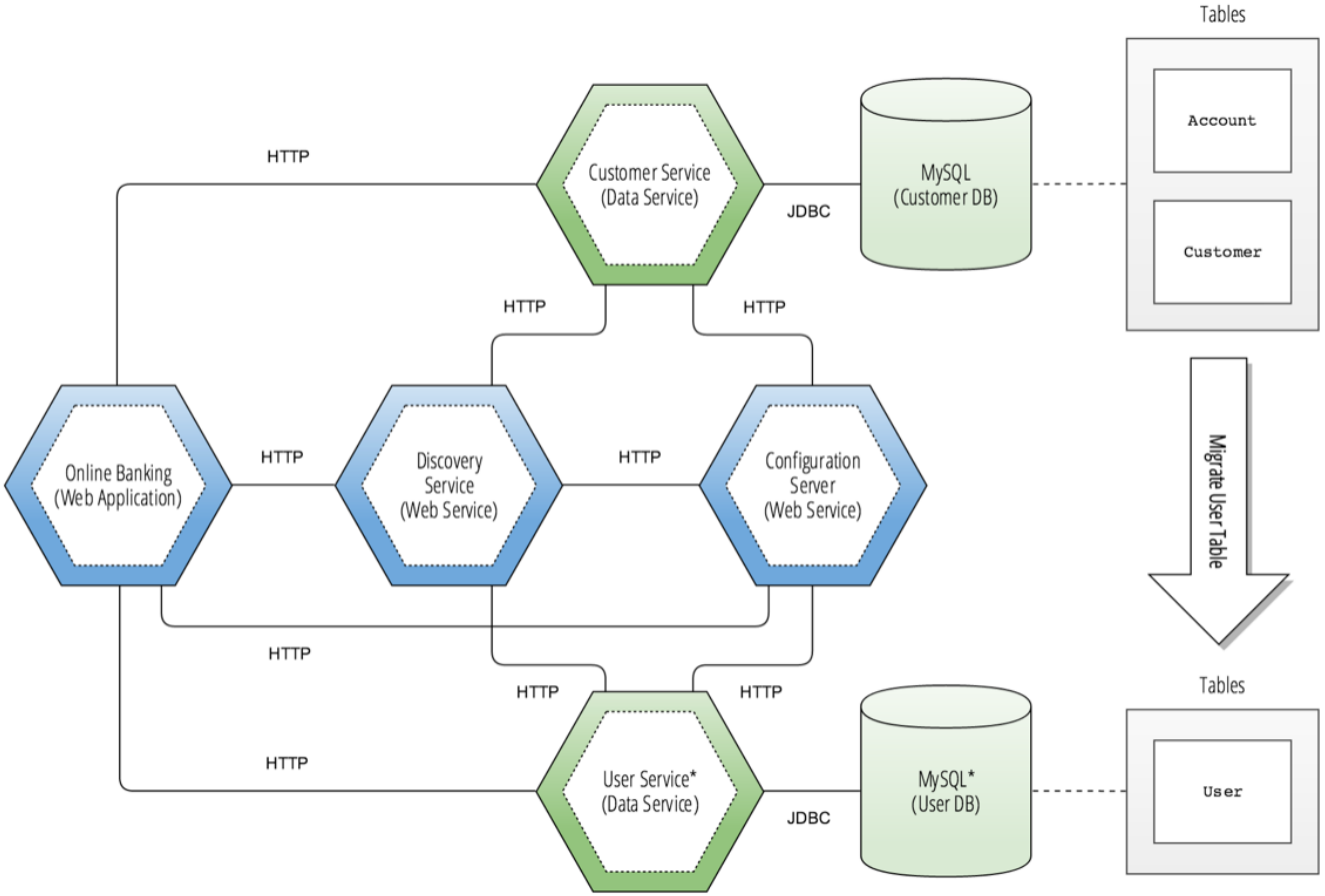
cf push awesome (Hoodie, Men's


Online Store Web
(Spring Boot)
HTTP
User
Database
(MySQL)
User Service
(Spring Boot)
Edge Service
(Spring Boot)
Discovery
Service
(Spring Boot)
Configuration
Server
(Spring Boot)
HTTP
Catalog Service
(Spring Boot)
Account Service
(Spring Boot)
Inventory Service
(Spring Boot)
Cart Service
(Spring Boot)
Order Service
(Spring Boot)
Catalog
Database
(MySQL)
Account
Database
(MySQL)
Inventory
Database
(Neo4j)
Cart
Database
(MySQL)
Order
Database
(MongoDB)

Monolith to Microservice



Splitting the Monolith: User Service Migration



Microservice problems...

No foreign key constraints between services.

Microservice problems…

I’ve got legacy problems.

What about my data warehouse?

Microservice problems...

Teams are creating redundant functionality.

Microservice problems...

Distributed transactions are brittle.

Microservice problems...

Every bug is a murder mystery.

Microservice problems...

Distributed systems are *hard*.

Without an **event-driven** architecture and a **cloud platform**...

Microservices will drown you in hard problems you never knew existed.

Spring Boot

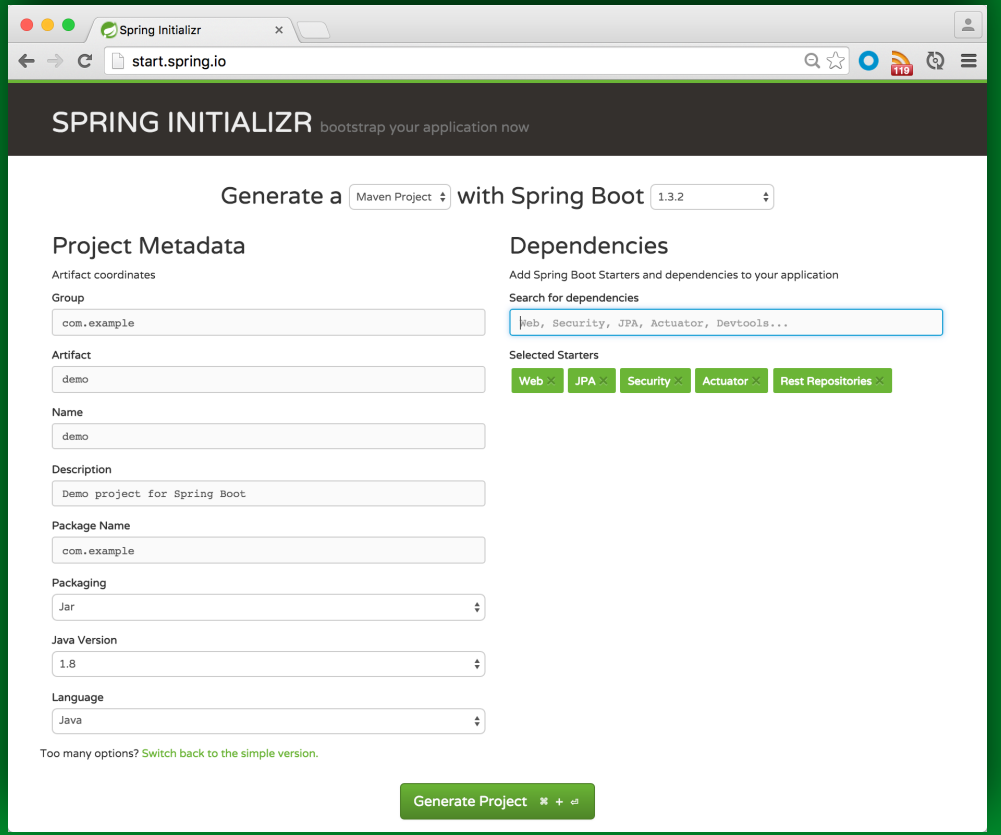
A JVM micro-framework for building microservices

What is Spring Boot?

Spring Initializr for bootstrapping your applications

spring boot

supports rapid development of production-ready applications and services



Event-driven microservices

Implementing event-driven architectures in a distributed system

Event-driven microservices...

Events as a first-class citizen.

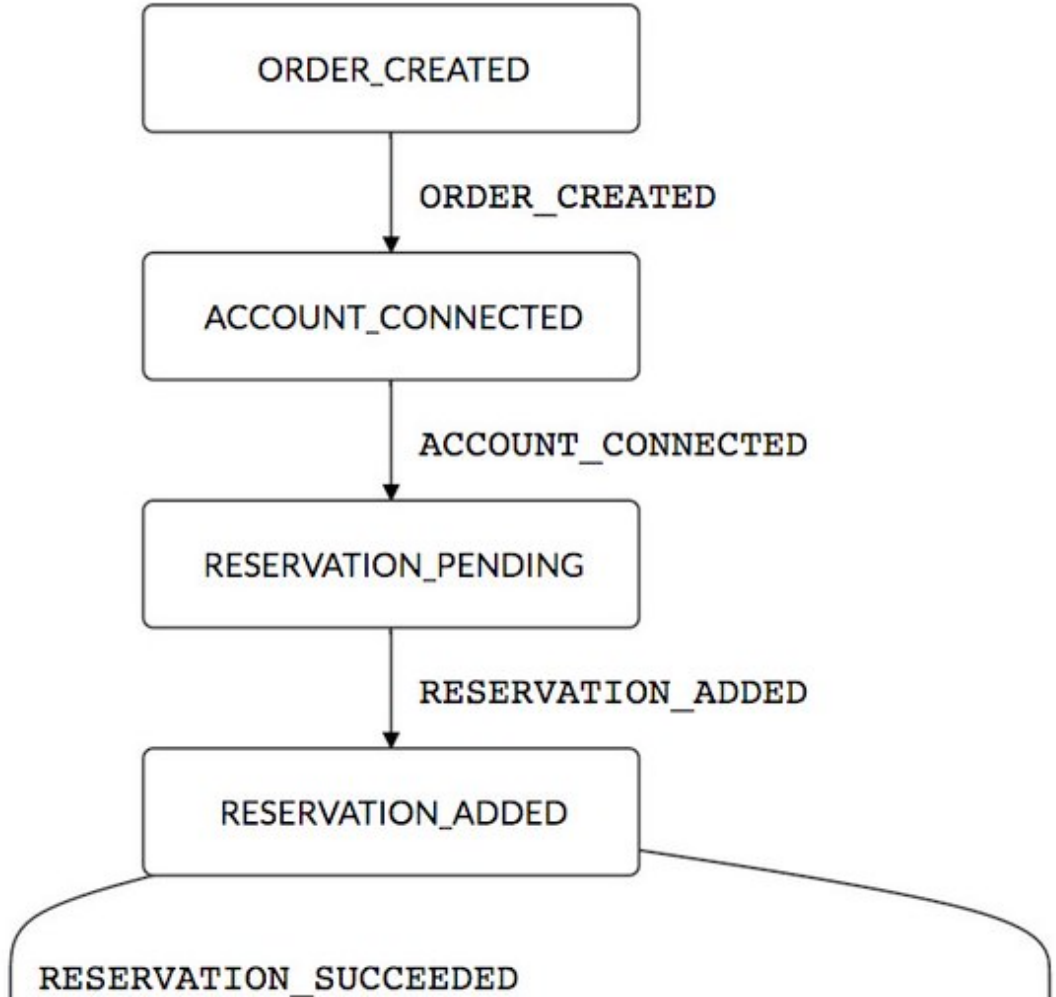
Event-driven microservices...

Domain events have a subject and contain immutable data.

Event-driven microservices...

Every domain event applies a state transition to an aggregate.

Domain events can only be applied if the aggregate is in a valid state.


ORDER_CREATED
ORDER_CREATED
ACCOUNT_CONNECTED
ACCOUNT_CONNECTED
RESERVATION_PENDING
RESERVATION_ADDED
RESERVATION_ADDED
RESERVATION_SUCCEEDED

Create meaningful exceptions if a domain event cannot be applied.

Event sourcing stores every state transition as an event in a log.

addReservation | completeReservation | completeOrder

updateOrderStatus | processPayment | reserveInventory

Event Log

eventId	type	createdAt
404	ORDER_CREATED	2/11/2017, 4:49:40 PM
405	ACCOUNT_CONNECTED	2/11/2017, 4:49:41 PM
406	RESERVATION_PENDING	2/11/2017, 4:49:41 PM

Event-driven microservices...

Build hypermedia APIs that attach event logs as a link on an aggregate.

Commands generate event(s).

Event-driven microservices...

Commands are attached to aggregates.

Status

ORDER_FAILED

▸ SKU-002

▸ SKU-001

+ −

▸ Shipping Address

Update

Commands

connectAccount | connectPayment | createPayment

addReservation | completeReservation | completeOrder

updateOrderStatus | processPayment | reserveInventory

Event-driven microservices...

Event handlers subscribe to an event and apply state changes to an aggregate.

Event handlers use event sourcing to generate the current state of an aggregate.

ss

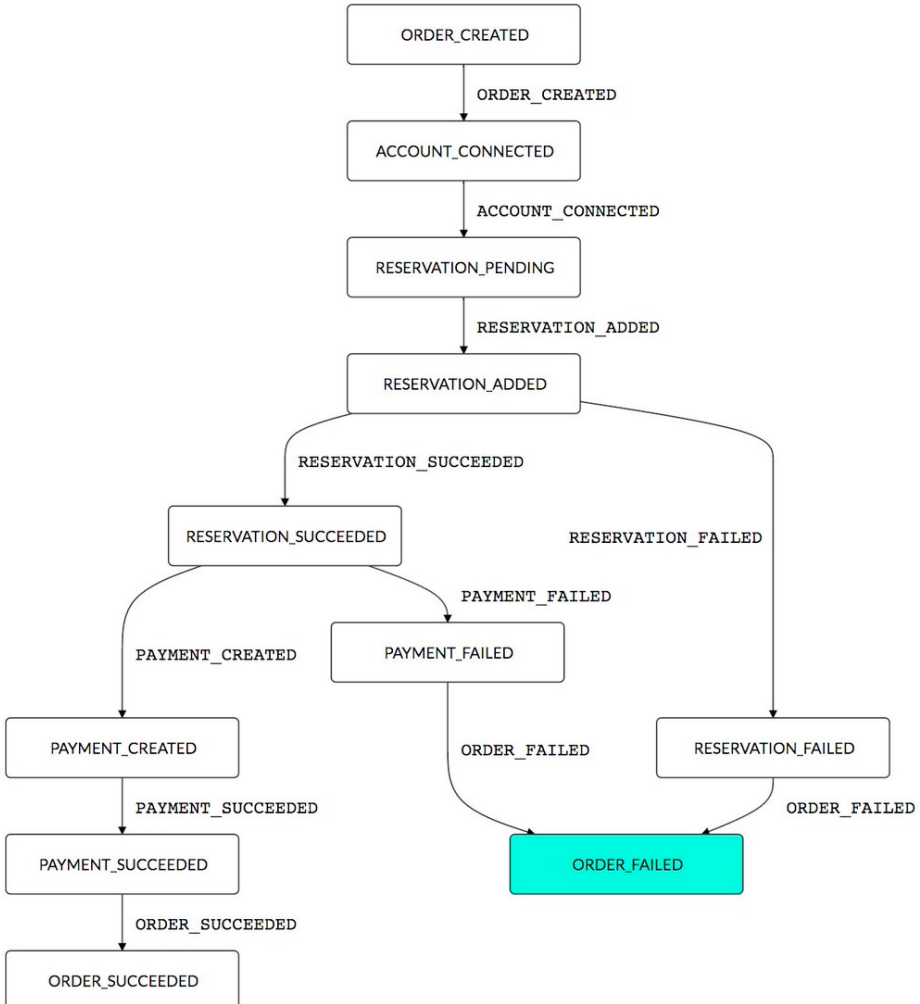
ectPayment | createPayment

leteReservation | completeOrder

ocessPayment | reserveInventory

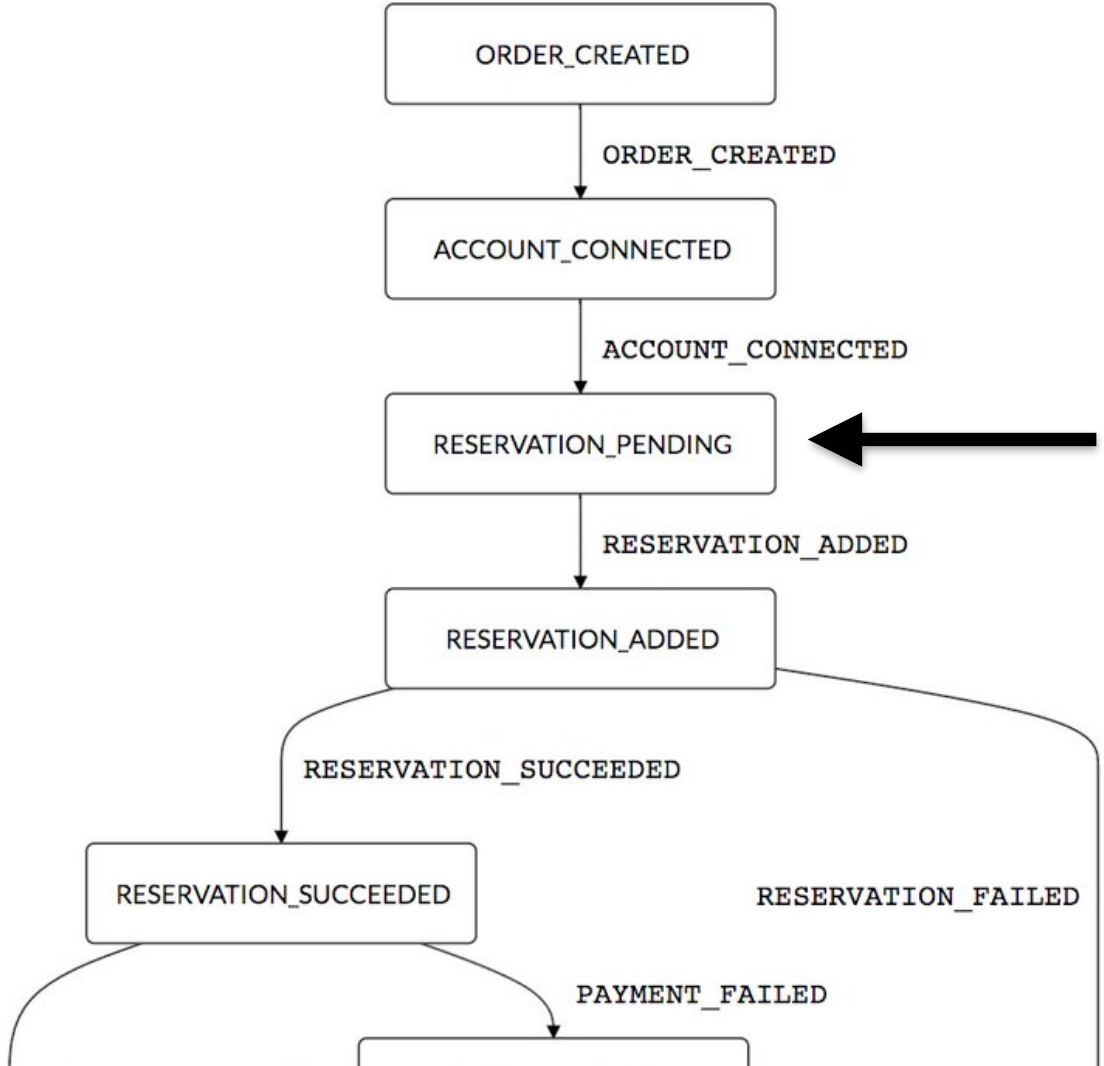
	createdAt
	2/11/2017, 4:49:40 PM
ECTED	2/11/2017, 4:49:41 PM
NDING	2/11/2017, 4:49:41 PM

State Flow



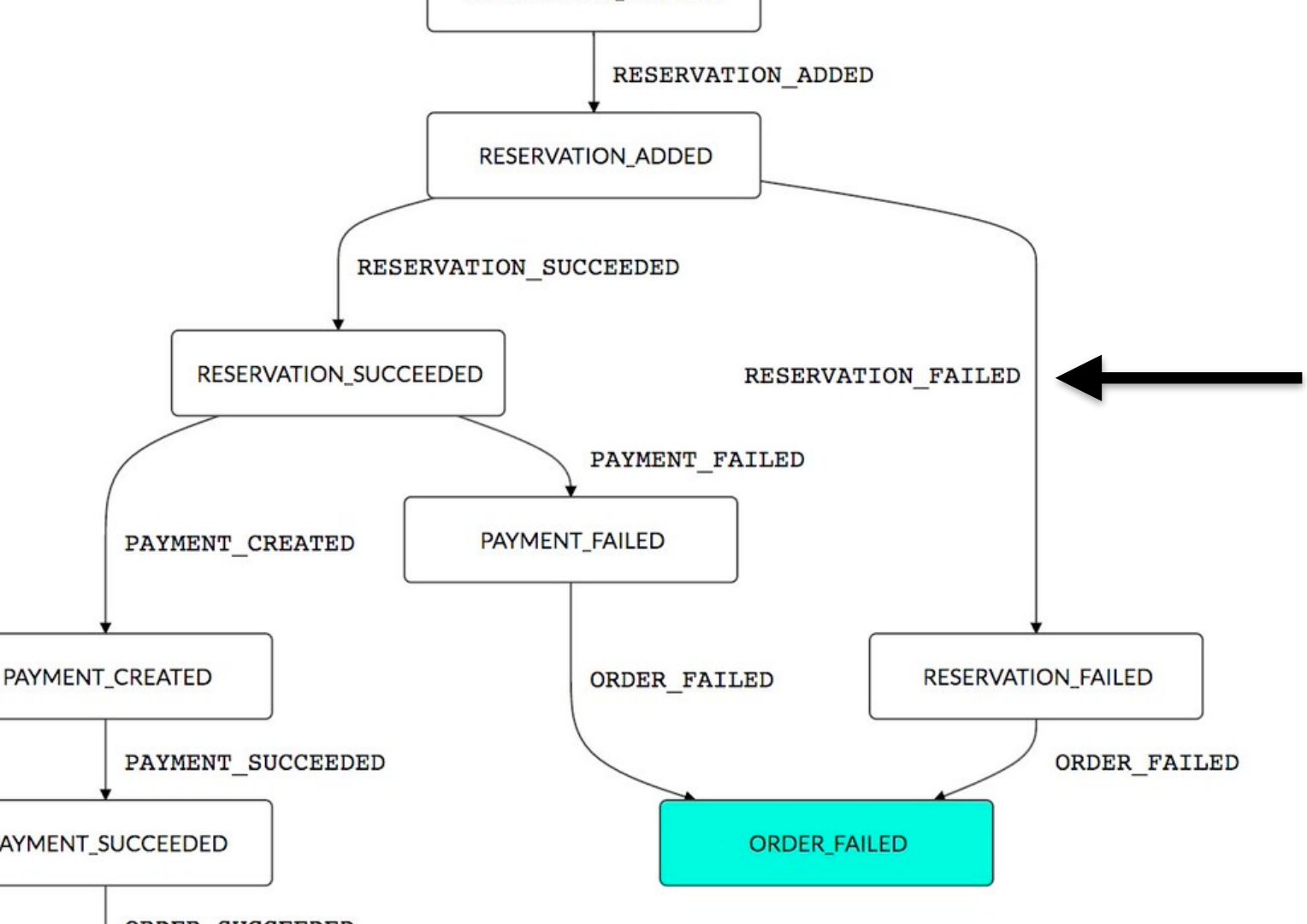
Event-driven microservices...

Event handlers are nodes on a directed graph.


ORDER_CREATED
ORDER_CREATED
ACCOUNT_CONNECTED
ACCOUNT_CONNECTED
RESERVATION_PENDING
RESERVATION_ADDED
RESERVATION_ADDED
RESERVATION_SUCCEEDED
RESERVATION_SUCCEEDED
RESERVATION_FAILED
PAYMENT_FAILED

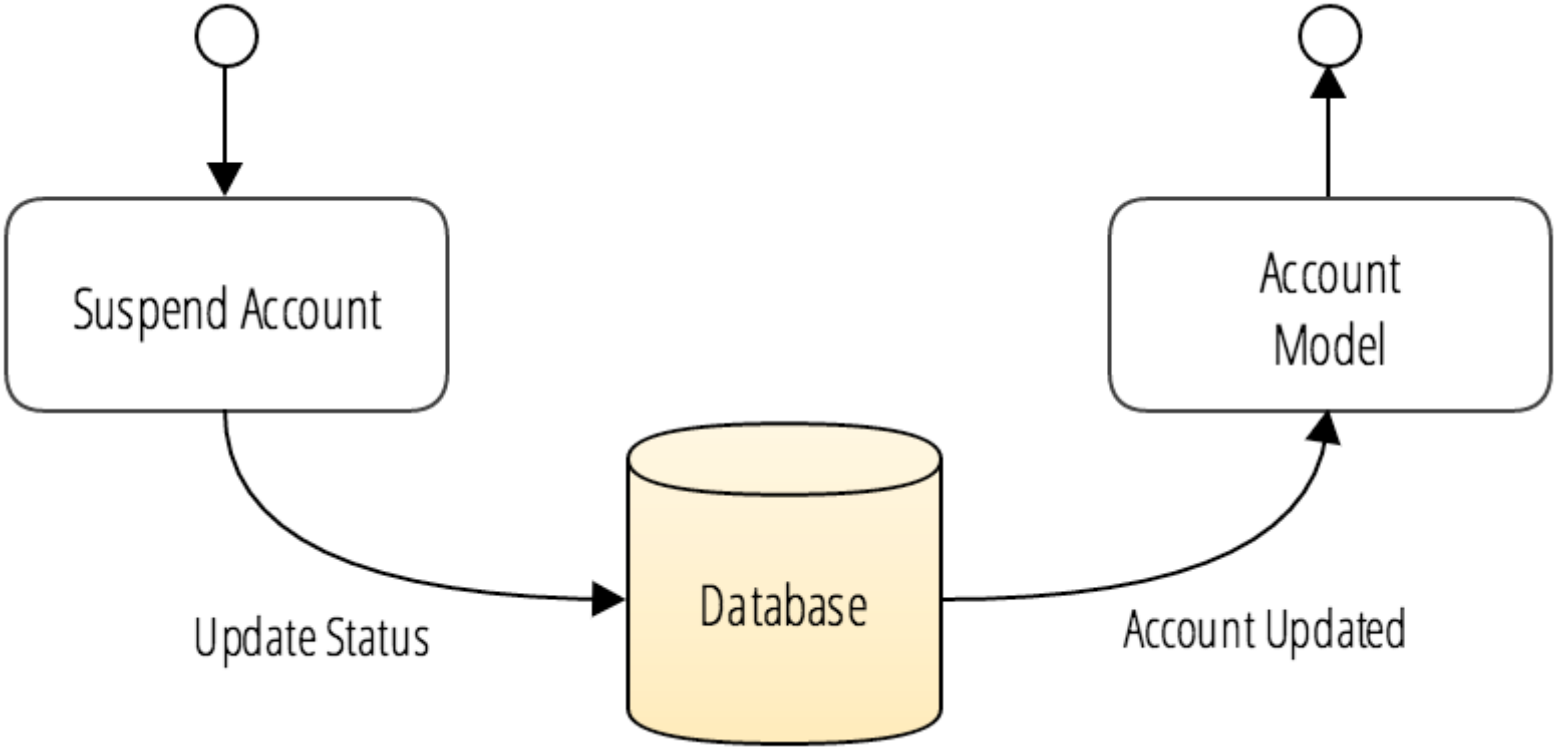
Event-driven microservices...

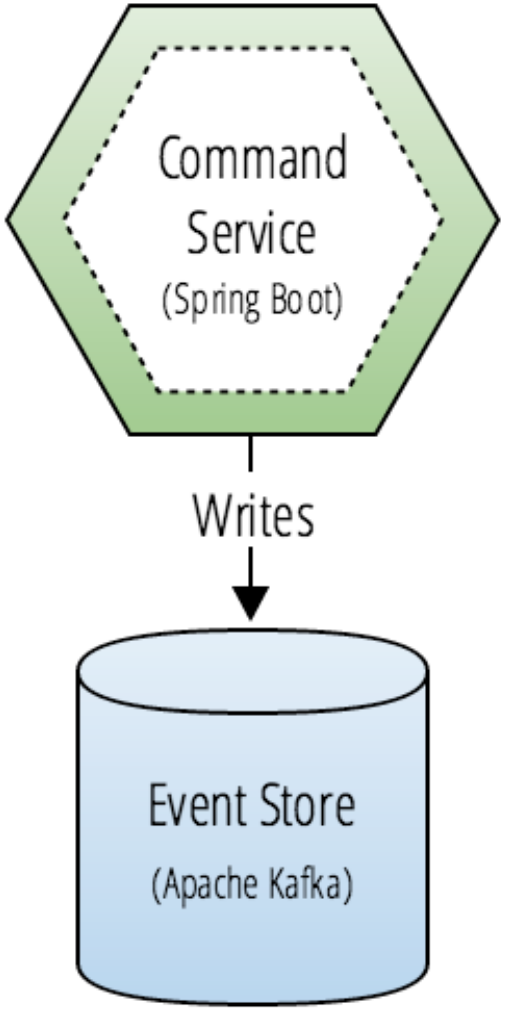
Events are edges on a directed graph.

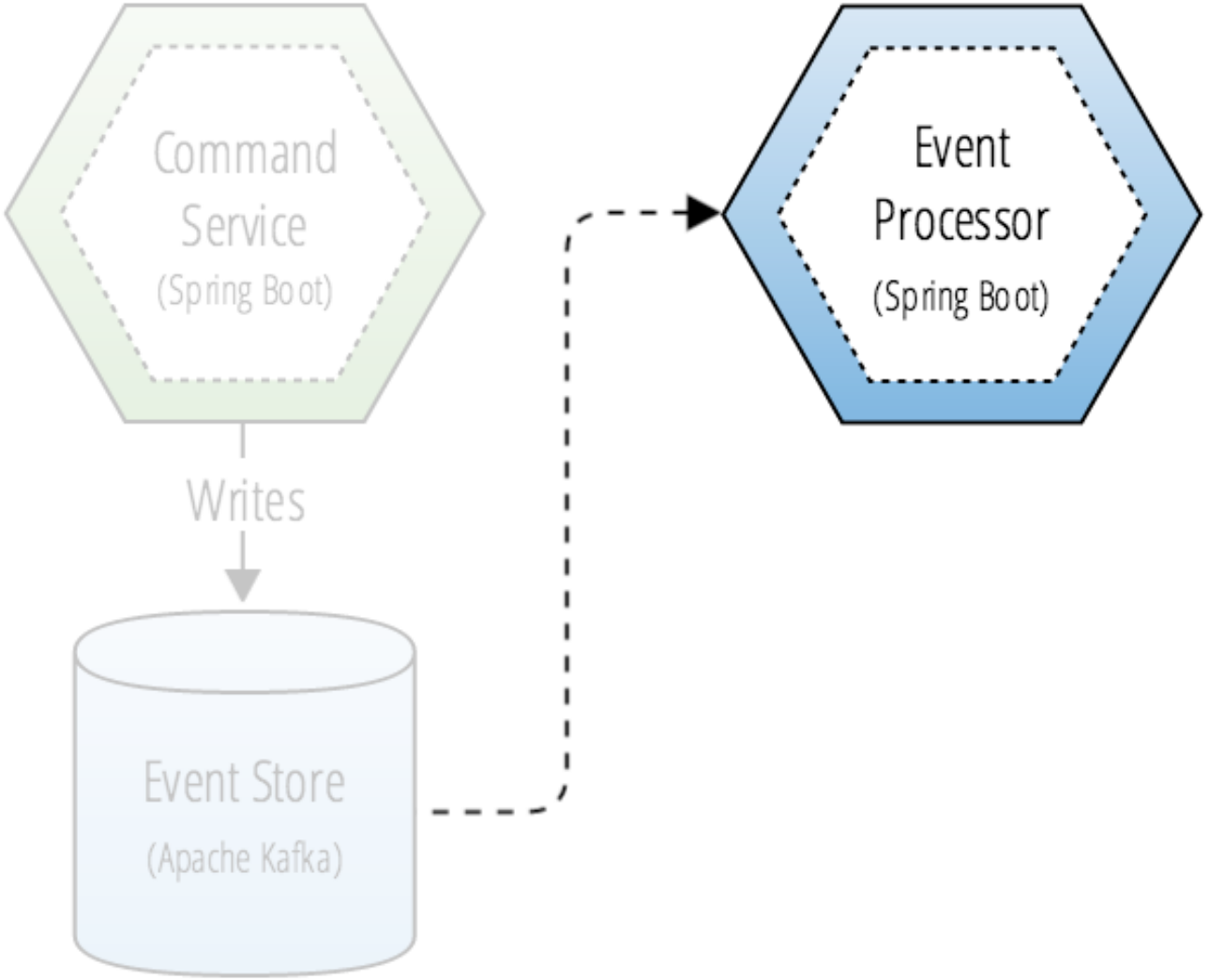

RESERVATION_ADDED
RESERVATION_ADDED
RESERVATION_SUCCEEDED
RESERVATION_SUCCEEDED
RESERVATION_FAILED
PAYMENT_FAILED
PAYMENT_CREATED
PAYMENT_FAILED
PAYMENT_CREATED
ORDER_FAILED
RESERVATION_FAILED
PAYMENT_SUCCEEDED
ORDER_FAILED
AYMENT_SUCCEEDED
ORDER_FAILED

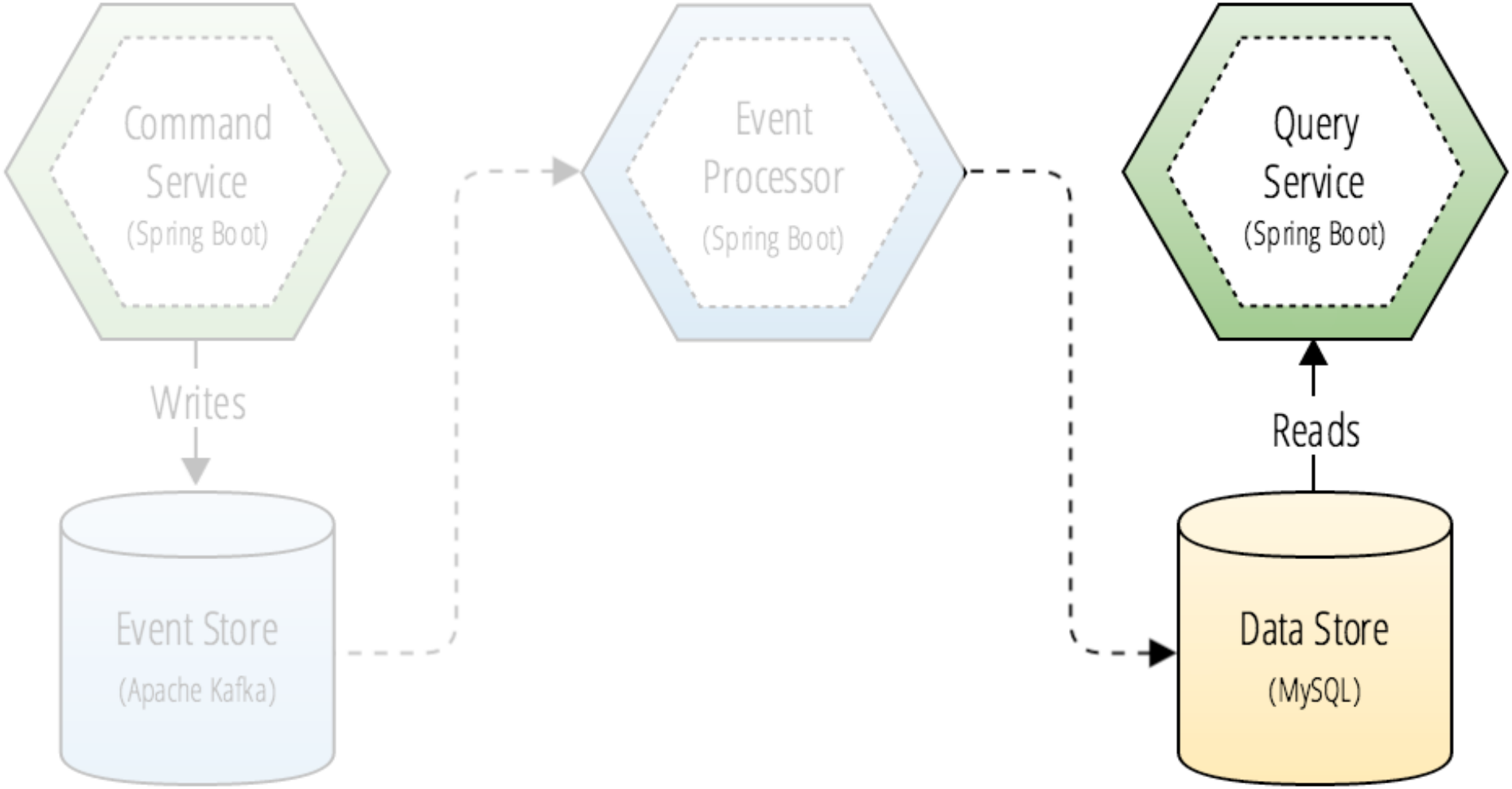
CQRS is used to create materialized views from streams of events.

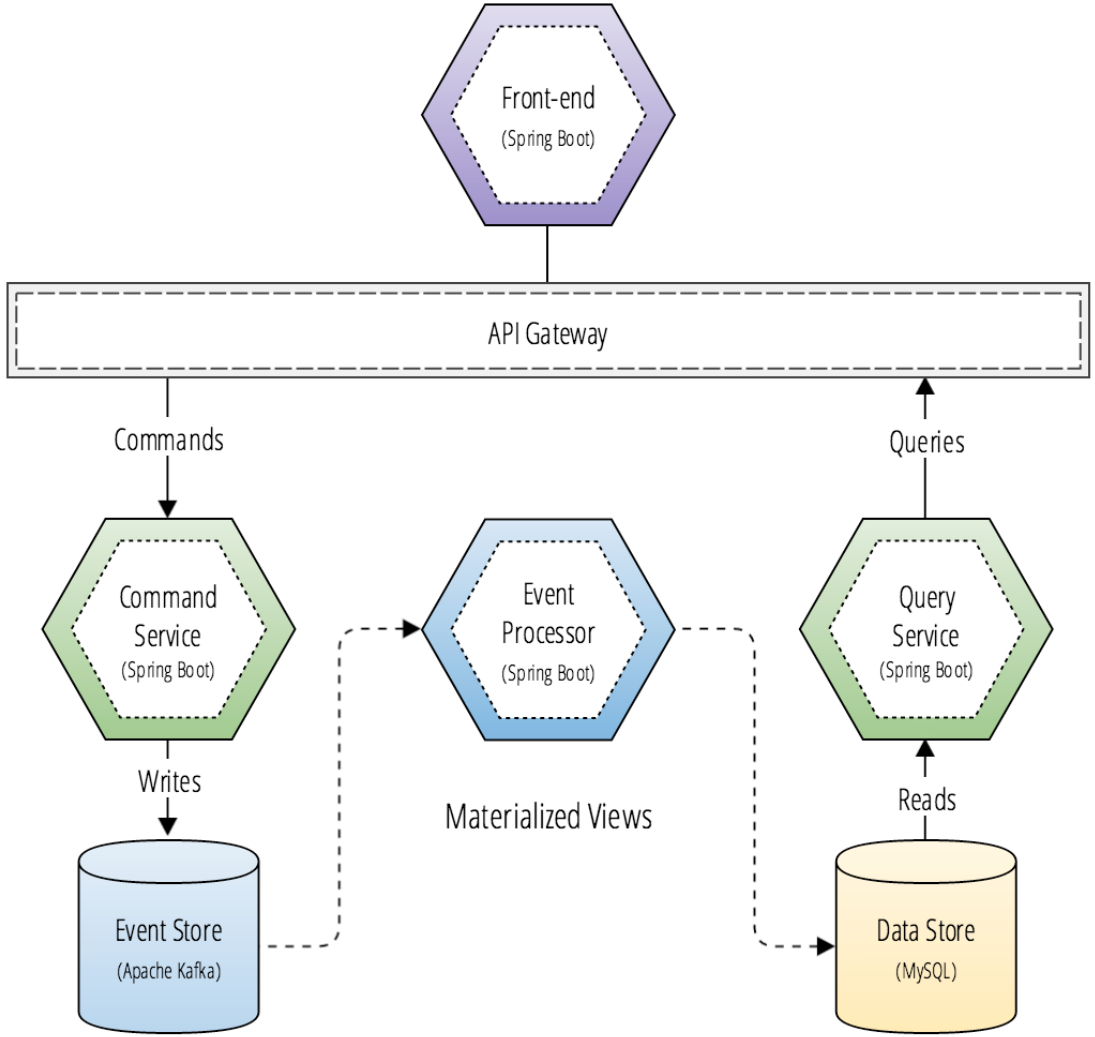
Command Query model




Command
Service
(Spring Boot)
Writes
Event Store
(Apache Kafka)

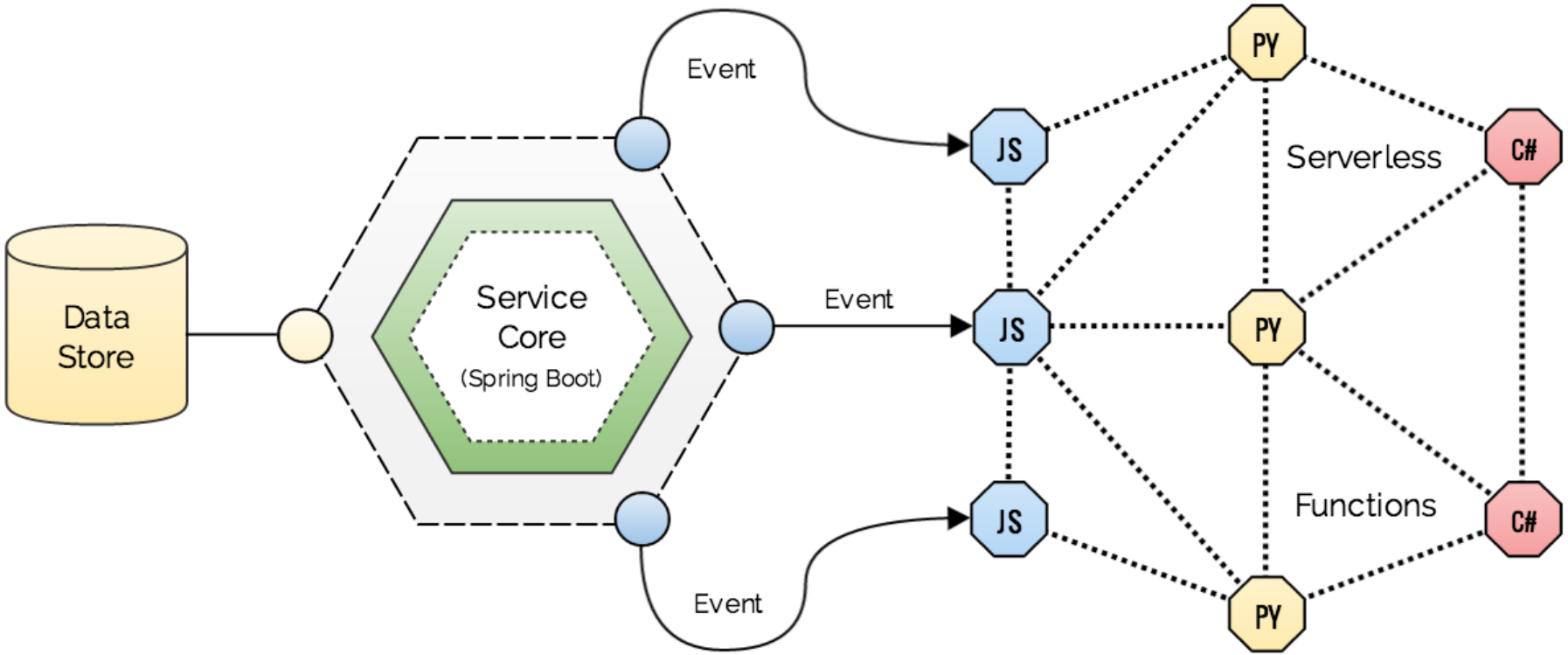

Command
Service
(Spring Boot)
Event
Processor
(Spring Boot)
Writes
Event Store
(Apache Kafka)


Command
Service
(Spring Boot)
Writes
Event Store
(Apache Kafka)
Event
Processor
(Spring Boot)
Query
Service
(Spring Boot)
Reads
Data Store
(MySQL)


Front-end
(Spring Boot)
API Gateway
Commands
Queries
Command
Service
(Spring Boot)
Event
Processor
(Spring Boot)
Query
Service
(Spring Boot)
Writes
Materialized Views
Reads
Event Store
(Apache Kafka)
Data Store
(MySQL)

Serverless event handlers

Microservices can be event sources


Data Store
Service Core
(Spring Boot)
Event
Event
Event
JS
JS
JS
PY
PY
PY
Serverless
Functions
C#
C#

Do what makes sense

But don’t build microservices without events

Chicago: May 30, 31

Spring Days 2017

$200.00
2 Days, 10 sessions
2 Spring team talks
Live Open Space discussions

Please provide feedback using the GOTO app

Thank you!

Thank you!

© 2016 Pivotal Software, Inc. All rights reserved.