Modern Microservices with Scala and Spring

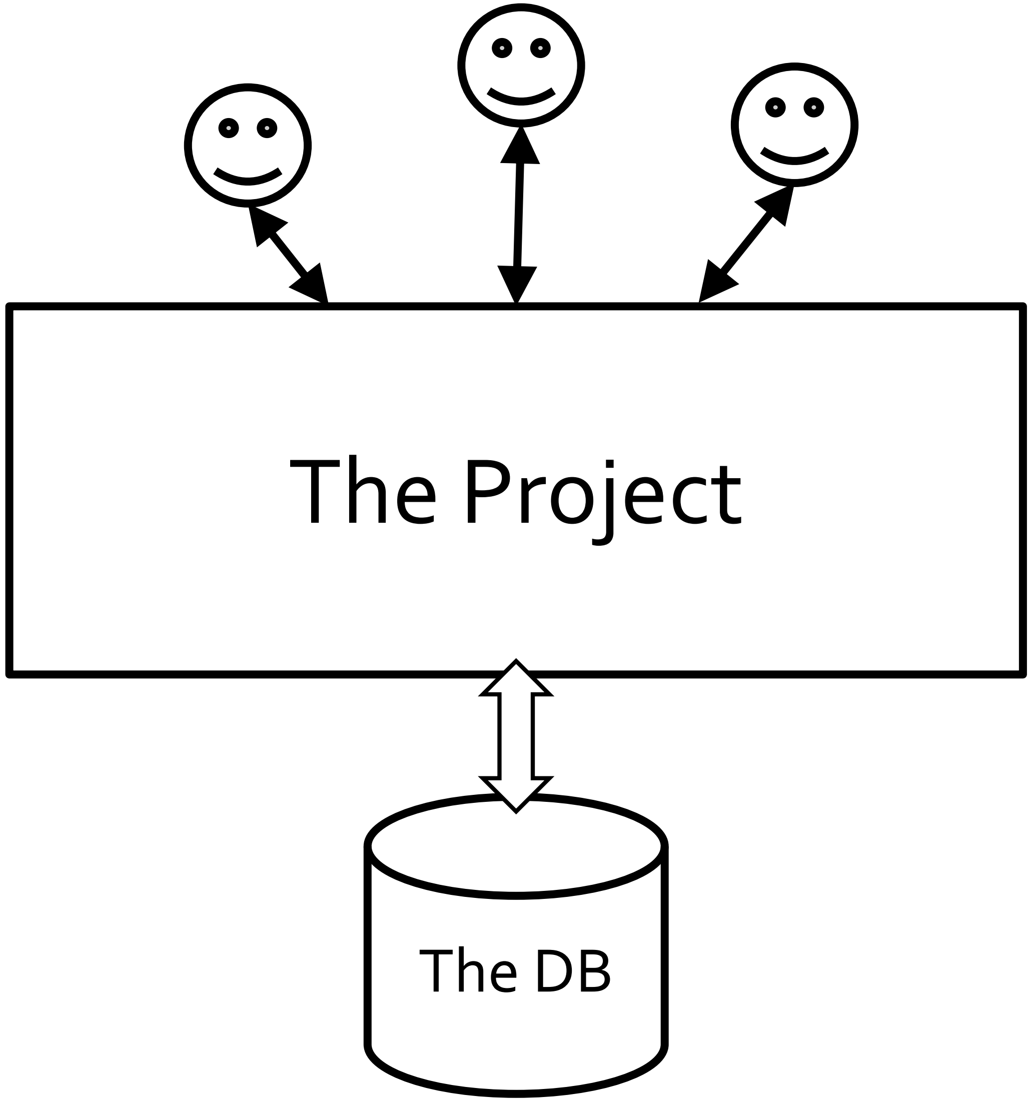
Jesper de Jong

@jesperdj

Modern Microservices with Scala and Spring

1. What are microservices?
   - A very short introduction
2. What is Spring Cloud?
   - How is it useful for building microservices?
3. Demo
   - Building a microservice using Spring Cloud and Scala

Architecture: The Monolith



Architecture: The Monolith

Advantages

- Single codebase
- In-process calls between modules
- Simple to deploy
- Simple to scale

Disadvantages

- Complex when app gets large
- Modules not encapsulated
- Updates are hard
- Scalability limitations
- Hard to manage when dev team grows

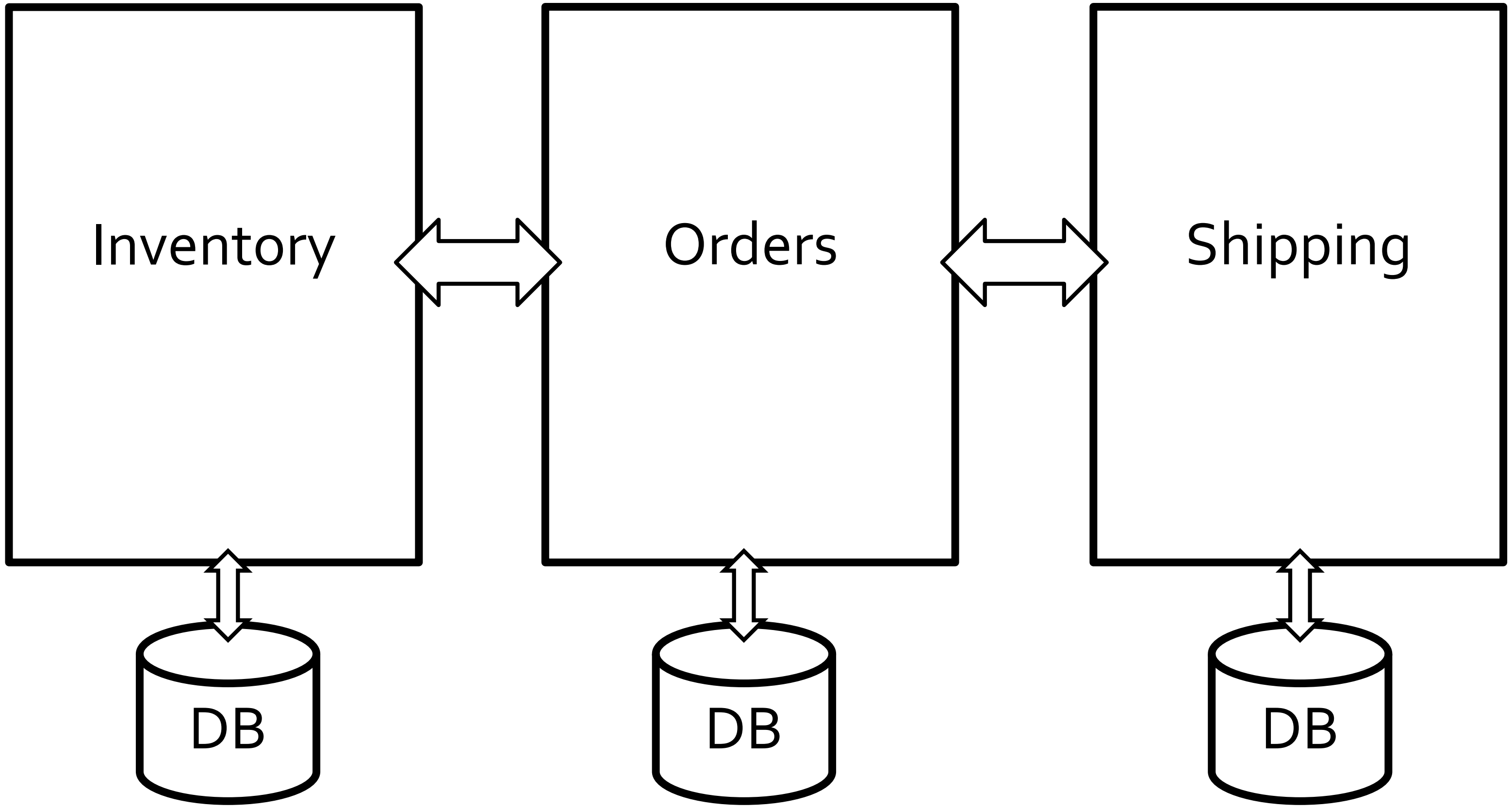
Architecture: Microservices

Microservices

- Split up the system into small, independent services
- Services communicate over the network
- Single business purpose
- Well-defined API
- Independently developed, deployed, operated
- Independently scalable

Architecture: Microservices

Independent services



Architecture: Microservices

Advantages

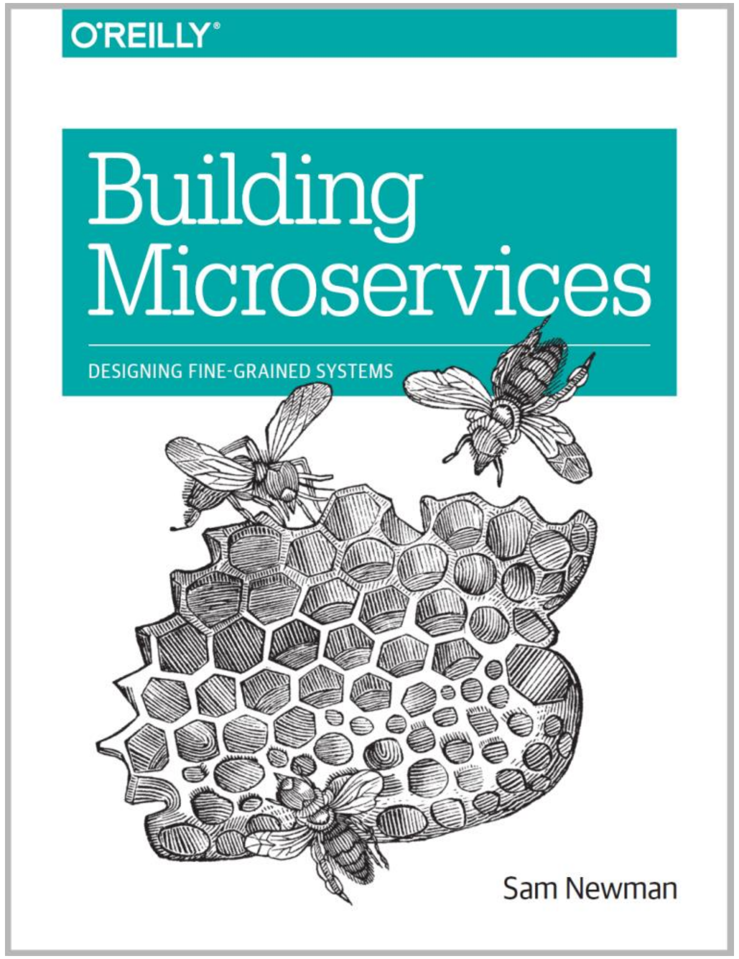
- Each service is simple
- Strong encapsulation
- Better fault tolerance
- Updates are easy
- Flexible scalability
- Easier to scale to more developers

Architecture: Microservices

Challenges

- Complexity of a distributed system
  - Distributed transactions are hard (avoid if you can)
  - Calls across the network vs. local method calls
- More complex deployment, management, monitoring
  - Need appropriate tools for management and monitoring

Architecture: Microservices


O'REILLY

Building Microservices

DESIGNING FINE-GRAINED SYSTEMS

Sam Newman

http://microservices.io

Spring Boot
Spring Cloud

Spring Cloud

spring
by Pivotal™

- Tools for building common patterns in distributed systems
    - Configuration management, service discovery, routing, load balancing, circuit breaker, distributed messaging, ...
- Built upon Spring Boot

Demo

Building microservices in Scala with Spring Cloud

Demo

Recap

1. Whiteboard Service
2. Configuration Service
3. Discovery Service
4. Whiteboard Client
5. Client-side load balancing with Ribbon
6. Declarative REST service with Feign
7. Circuit breaker with Hystrix

Conclusion

Scala and Spring Cloud

- Scala interoperability with Java is good
  - Some ugly parts, for example with annotations
- JPA is really made for Java
  - You end up writing Java-like code in Scala
- Alternatives for a more Scala-native approach
  - Lightbend Lagom, built on Akka and Play

Conclusion

More on the web

- Demo code
  - https://github.com/jesperdj/spring-cloud-demo
- Spring Initializr
  - http://start.spring.io
- Spring Cloud
  - http://cloud.spring.io
- Scala
  - http://scala-lang.org
- Lightbend
  - http://www.lightbend.com