Building Cloud Native Microservices with Spring Cloud

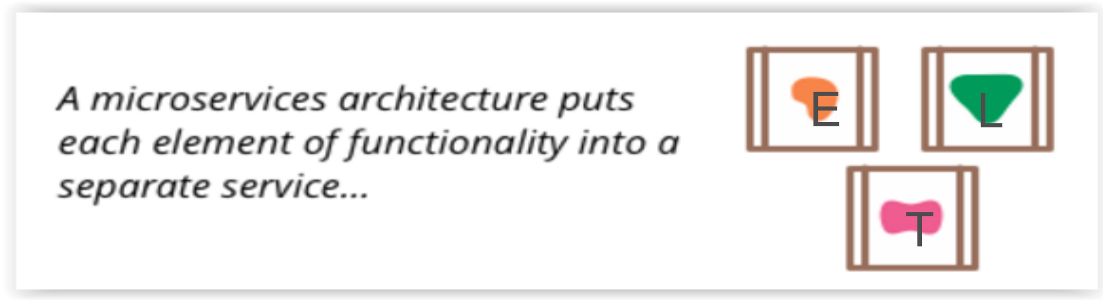
David Turanski, Pivotal Software

Pivotal

Microservices

- …the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and **communicating** with lightweight mechanisms, often an **HTTP** resource API.
  - http://martinfowler.com/articles/microservices.html

Microservices Architecture



A microservices architecture puts each element of functionality into a separate service…

- Services managed as products, with their own lifecycle
- Smart endpoints and dumb pipes
- Decentralized
  - Code
  - Data
  - Deployment
  - Governance

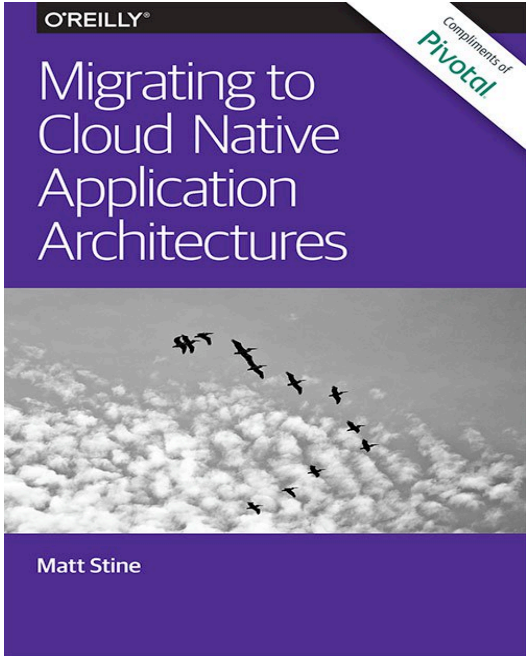
Microservices in the Cloud

- *Cloud Native*: Application architecture suitable for deployment on modern cloud platforms
- http://12factor.net/
- Decompose applications into smaller services that can be independently developed, deployed, and scaled
- Services communicate via remote protocols, e.g. REST, messaging
- Agility: Maximum flexibility, minimum time to market
- Simple yet complex!

Microservices Ecosystem

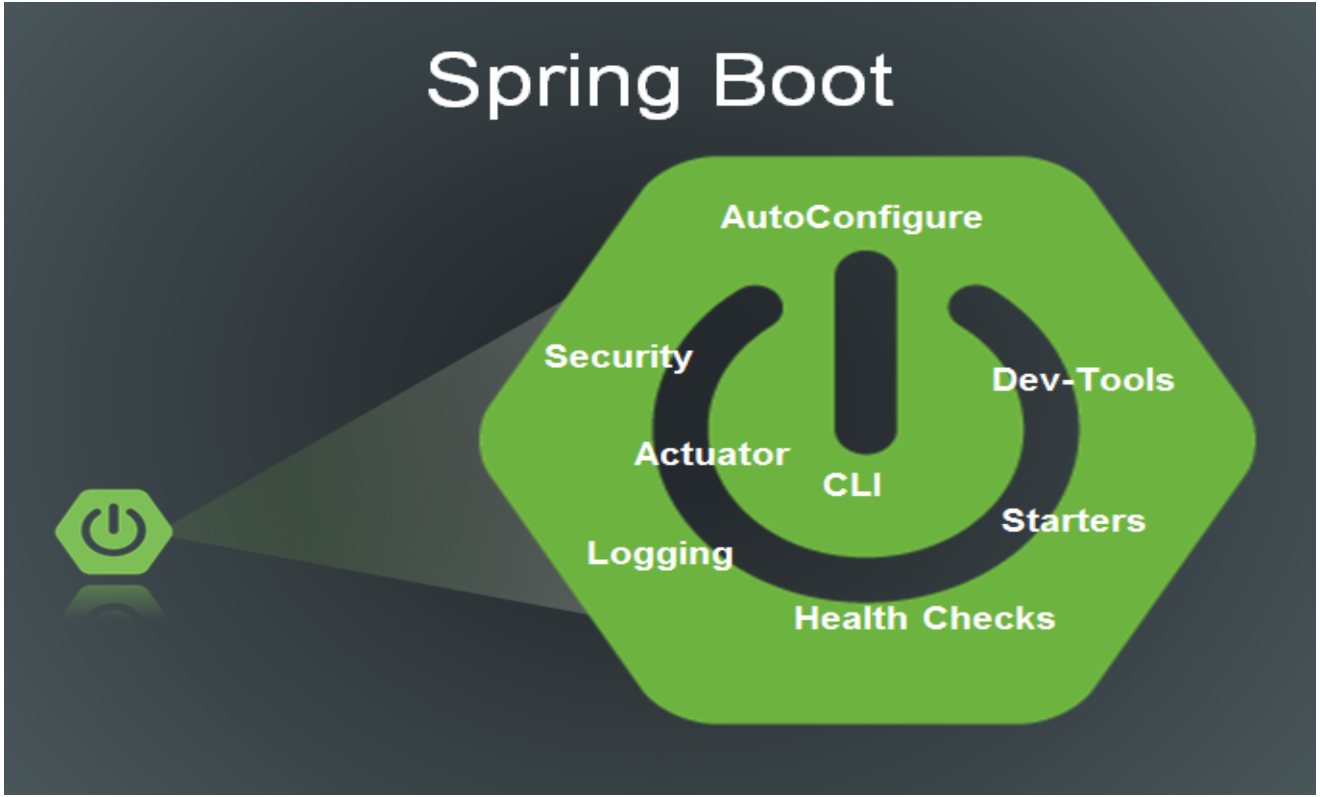
- Make jar not war:
  - Self contained apps, no external containers required
  - DevOps friendly
  - Make docker not jar….
- New patterns and runtime services
  - Service Discovery, Circuit Breaker, Client Side Load Balancing, Dynamic Routing, Leader Election, Canary Deployment, …

Recommended Reading



Spring Ecosystem

- Spring Boot
  - Popular way to quickly build microservice applications
- Spring Cloud
  - Family of projects to support microservice applications


Spring Boot
AutoConfigure
Security
Dev-Tools
Actuator
CLI
Starters
Logging
Health Checks

Spring Cloud

NETFLIX | OSS

Hystrix
Eureka
Ribbon
Zuul
Feign

Service Discovery

Config Server

Leader Election,
Global locks

ZIPKIN

Distributed Tracing

Spring Cloud: There’s a lot of stuff here…

- Distributed/versioned configuration
- Service registration and discovery
- Routing
- Service-to-service calls
- Load balancing
- Circuit Breakers
- Global locks
- Leadership election and cluster state
- Distributed messaging

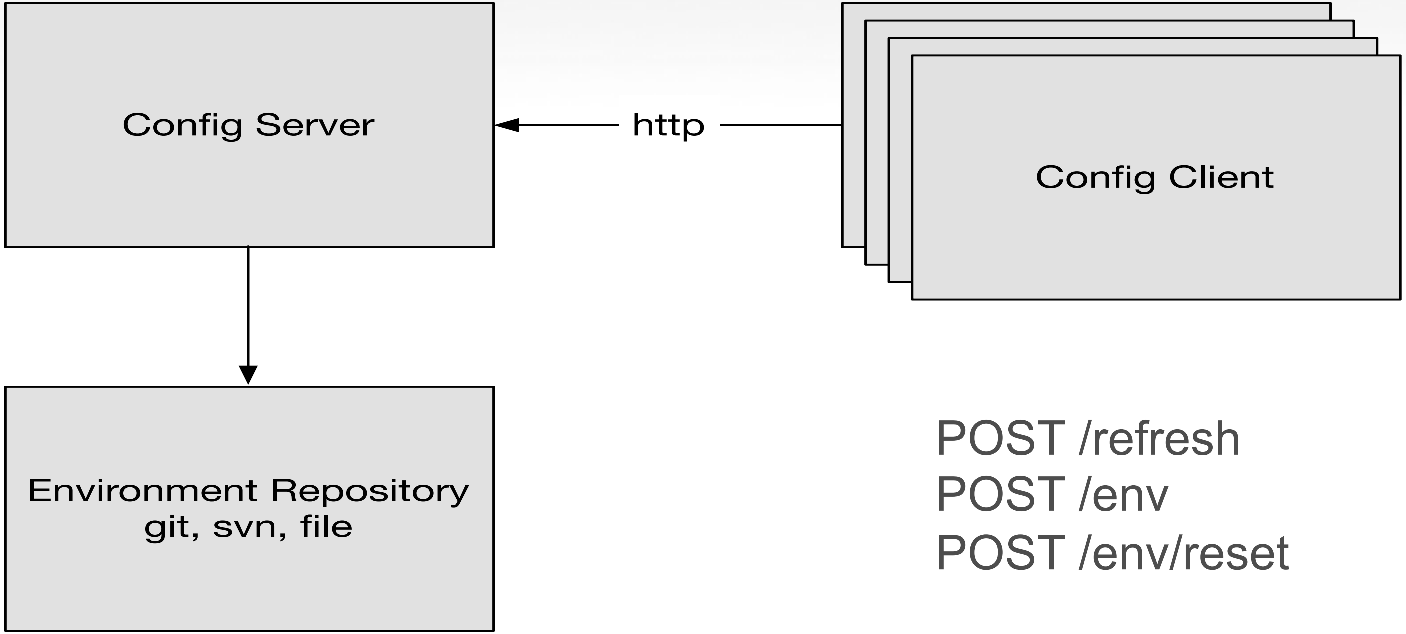
Spring Cloud Config
Spring Cloud Netflix
Spring Cloud Bus
Spring Cloud for Cloud Foundry
Spring Cloud Cluster
Spring Cloud Consul
Spring Cloud Security
Spring Cloud Sleuth
Spring Cloud Data Flow
Spring Cloud Stream
Spring Cloud Stream Starters
Spring Cloud Task
Spring Cloud Zookeeper
Spring Cloud for Amazon Web Services
Spring Cloud Connectors
Spring Cloud Starters
Spring Cloud CLI

Spring Cloud Config

Config Server

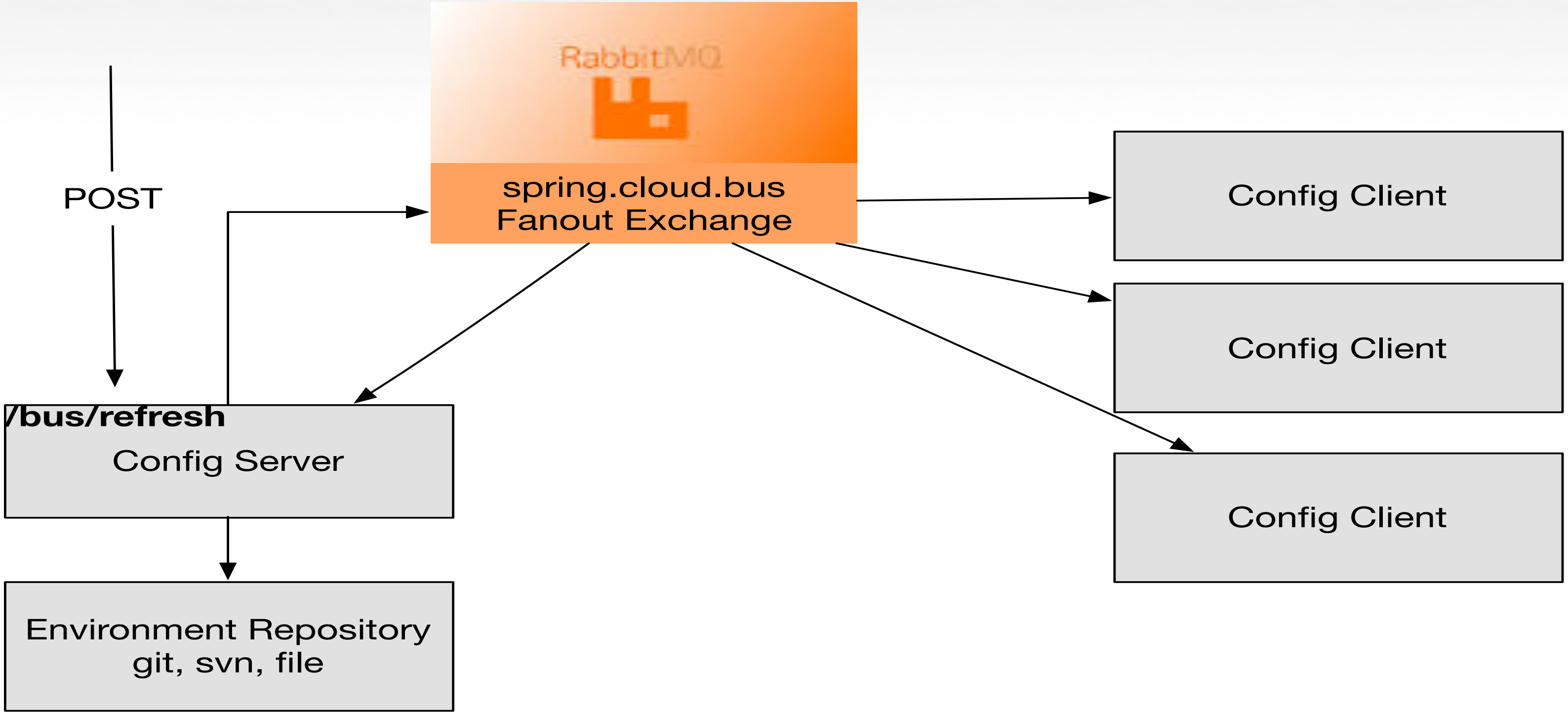
- 12 factor: Strict separation of configuration from code
  - Things are likely to change in different enviroments
  - This does not include internal configuration
- Environment variables are good but hard to centrally manage
- Namespace support for application, Spring profile, and version
  - {application}-{profile}-{label}.properties
  - Or repo per application, or repo per profile
- Encrypted property values
- Spring-cloud-bus: pub-sub refresh to all config clients
- Push notification e.g., via webhooks

Spring Cloud Config

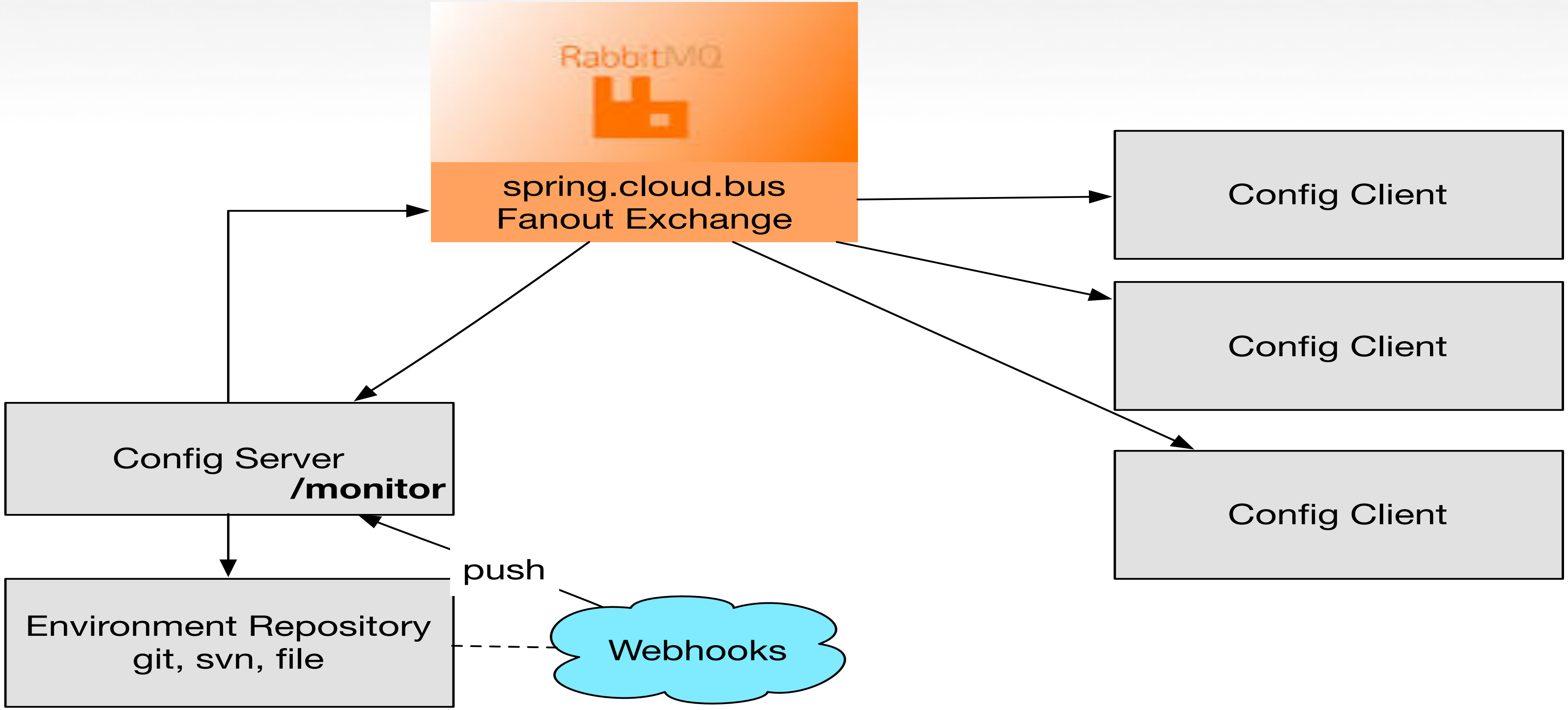


POST /refresh
POST /env
POST /env/reset

Spring Cloud Bus



Push Notification



Demo:
Spring Cloud Config

References

- https://projects.spring.io/spring-cloud/
- http://cloud.spring.io/spring-cloud-static/spring-cloud.html
- https://cloud.spring.io/spring-cloud-config/

Questions?

Thank You.