



Pivotal



Cloud Native Enterprise

Continuous Delivery & Microservice Architectures in
the Enterprise with Spring and Cloud Foundry

Agenda

1. Why?
2. What?
3. How?

Why

Disruption to Enterprise IT is here

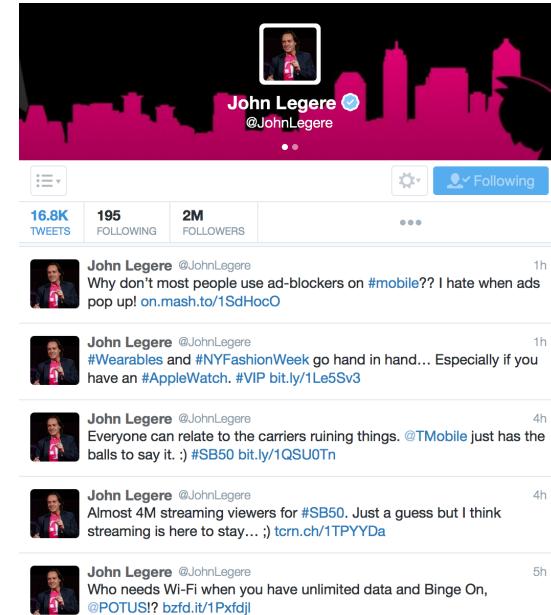
Google

NETFLIX

facebook.

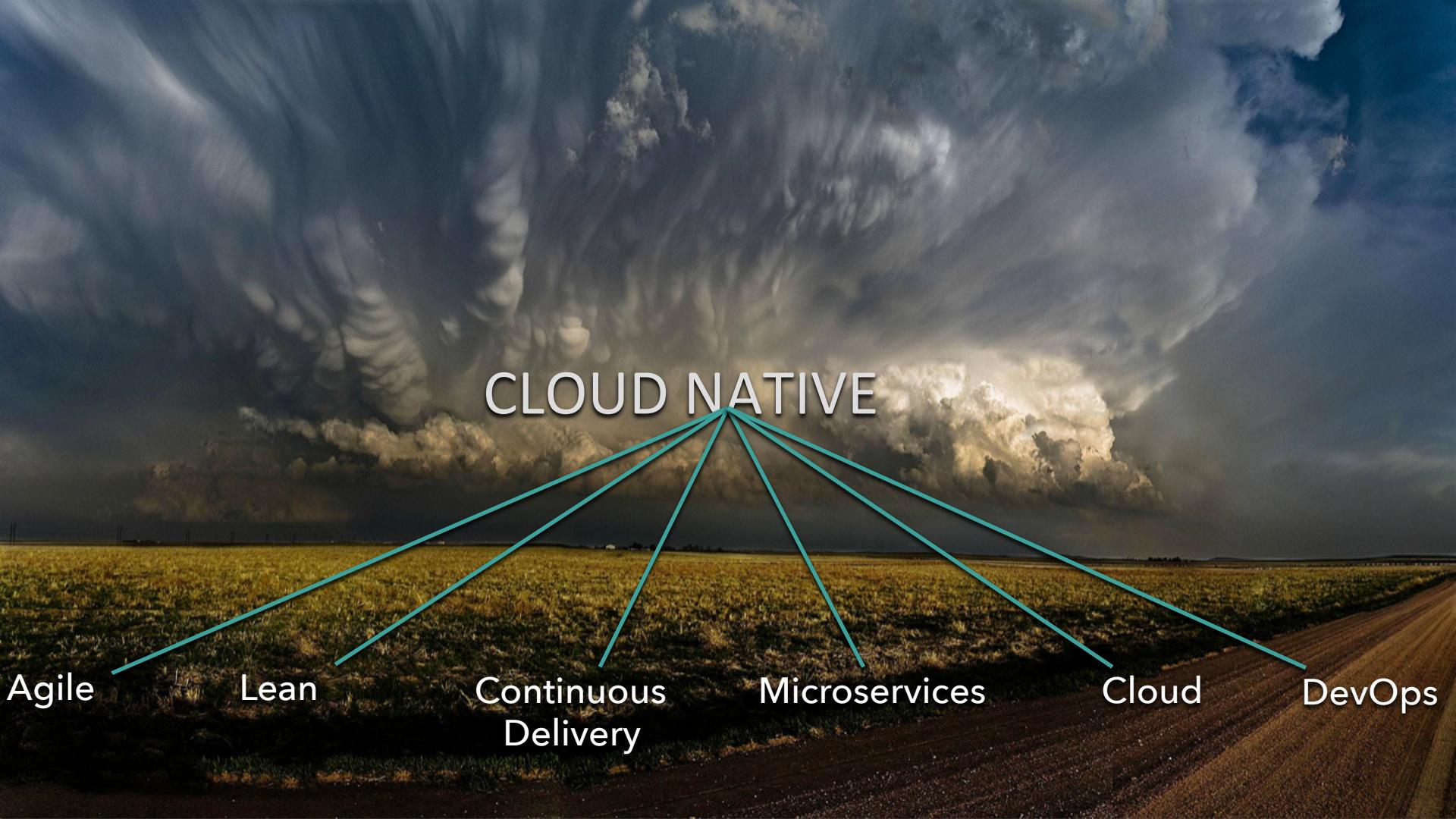
amazon

From the outsiders



From the uber insider

Pivotal



CLOUD NATIVE

Agile

Lean

Continuous
Delivery

Microservices

Cloud

DevOps

The Promise of Cloud Native

- Deploy new **features** daily to production
- Automate middleware
- Free/Cheap horizontal scaling
- No Downtime Upgrades
- Contract between App & Platform

Cloud native and microservice models are moving into the Fortune 500

“I said to my vendors, I don’t want five years ago. I want five years from now.



“Two people built an app and got it into the App Store in five weeks”



“The adopters we speak to today, like GE, HP, Equinix, PayPal, Capital One, Goldman Sachs, Airbnb, Medallia, Square, and Xoom say that microservices are well worth the tradeoffs.”

WALL STREET JOURNAL

What

What is Cloud Native?

A term that recognizes that getting software to work in the cloud requires a broad set of components that work together.

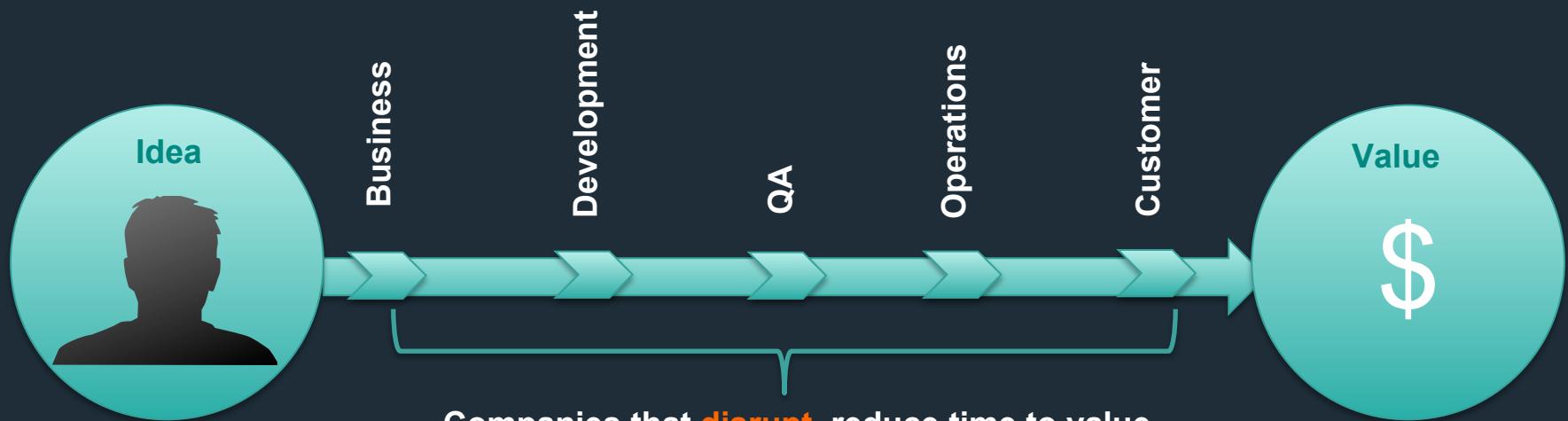
It requires an architecture that departs from traditional enterprise application design, such as 'Microservices'

It requires a Cloud Native platform

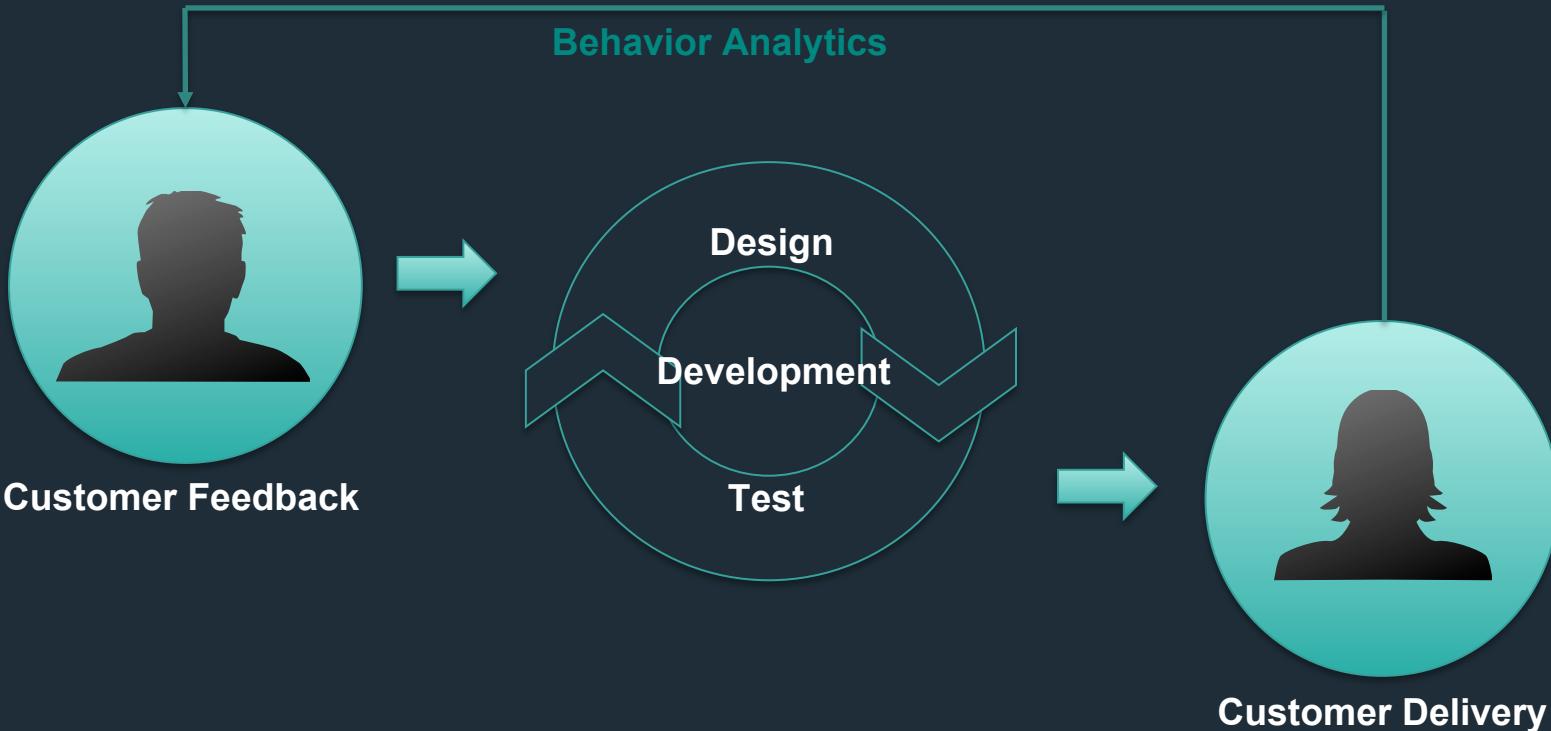
Characteristics of Cloud Native Architectures

- Continuously Delivered
- Twelve Factor Apps (<http://12factor.net/>)
- Microservices
- Self-Service agile infrastructure
- API-based collaboration
- Anti-fragility

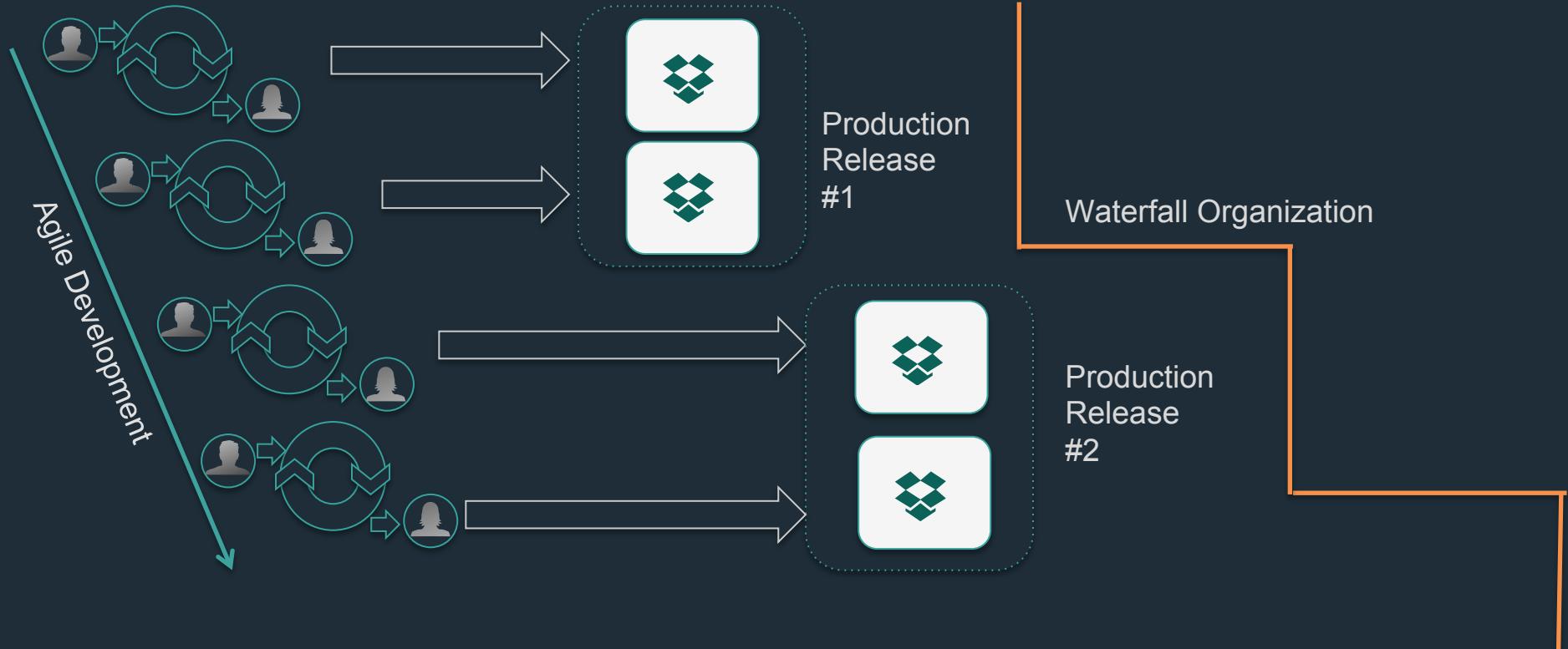
Enabling Continuous Delivery



Enabling Agile Delivery



WaterScrumFall



Conway's Law

"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure."

- Melvin Conway, 1967

Inverse Conway Maneuver

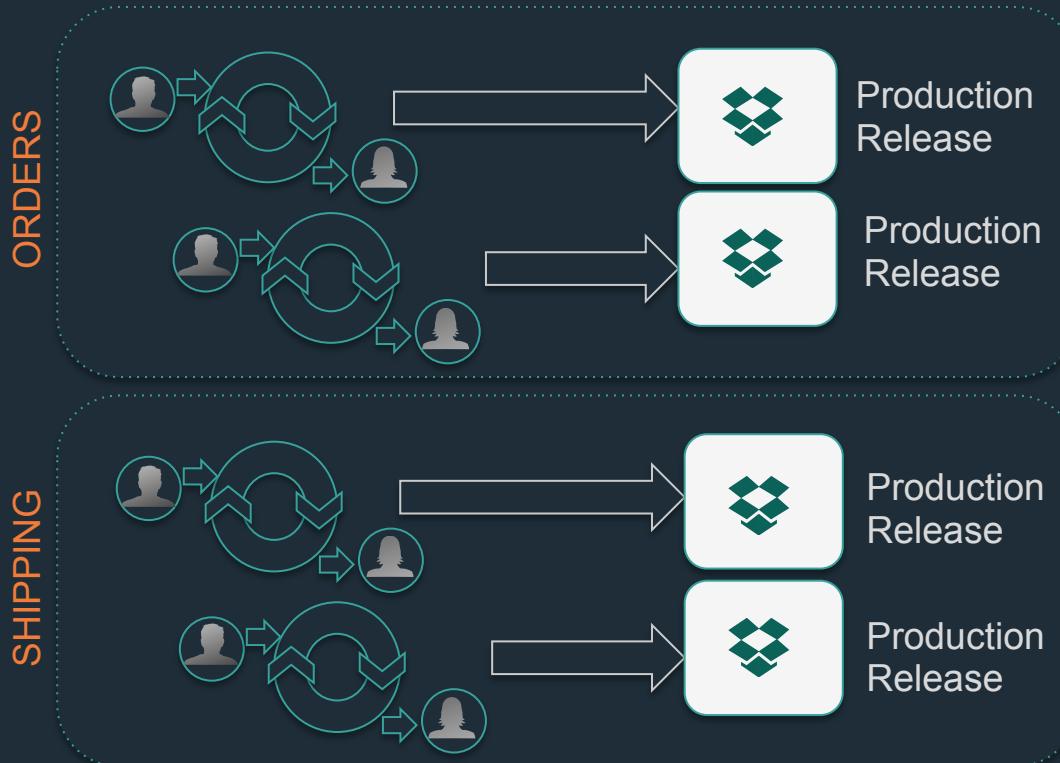
Enable Continuous Delivery



<http://jonnyleroy.com/2011/02/03/dealing-with-creaky-legacy-platforms/>

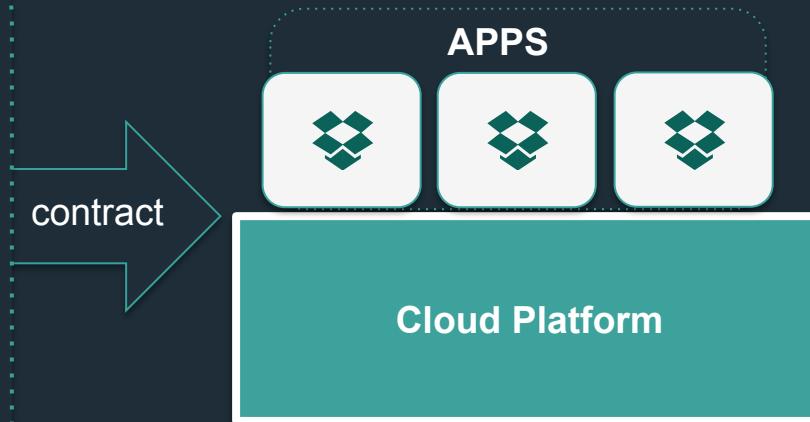
<http://www.slideshare.net/adriancockcroft/microservices-the-good-bad-and-the-ugly>

Enabling Continuous Delivery



12-Factor Applications

- 1. Codebase
- 2. Dependencies
- 3. Configuration
- 4. Backing Services
- 5. Build, release, run
- 6. Processes
- 7. Port binding
- 8. Concurrency
- 9. Disposability
- 10. Dev/prod parity
- 11. Logs
- 12. Admin processes



Architectural and development practices – <http://12factor.net>

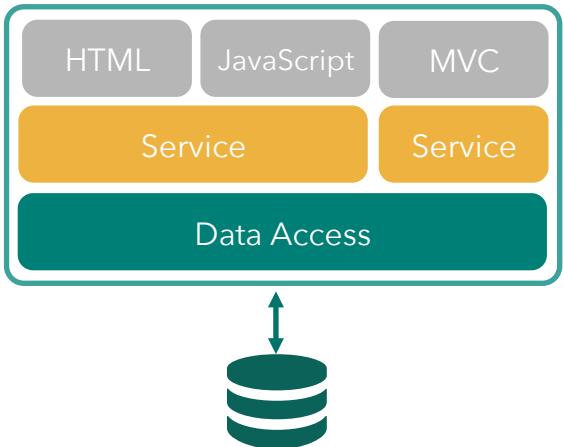
What are Microservices?

If every service has to be updated in concert,
it's not loosely coupled!

**Loosely coupled service oriented
architecture with bounded contexts**

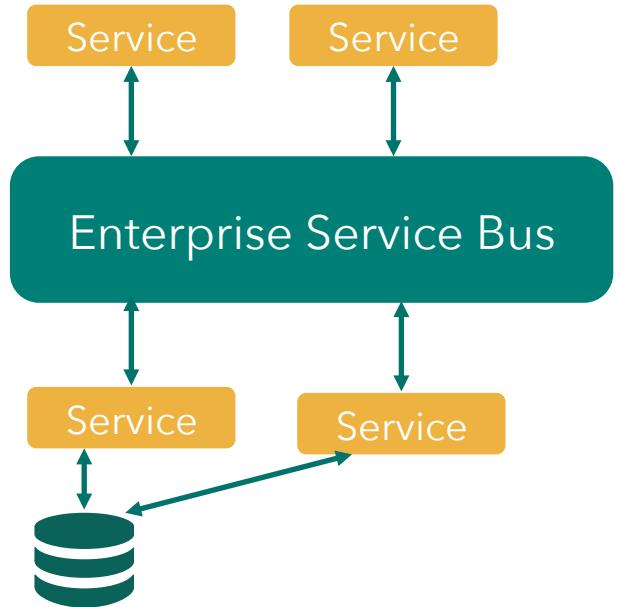
If you have to know about surrounding
services you don't have a bounded context.

Microservices are NOT



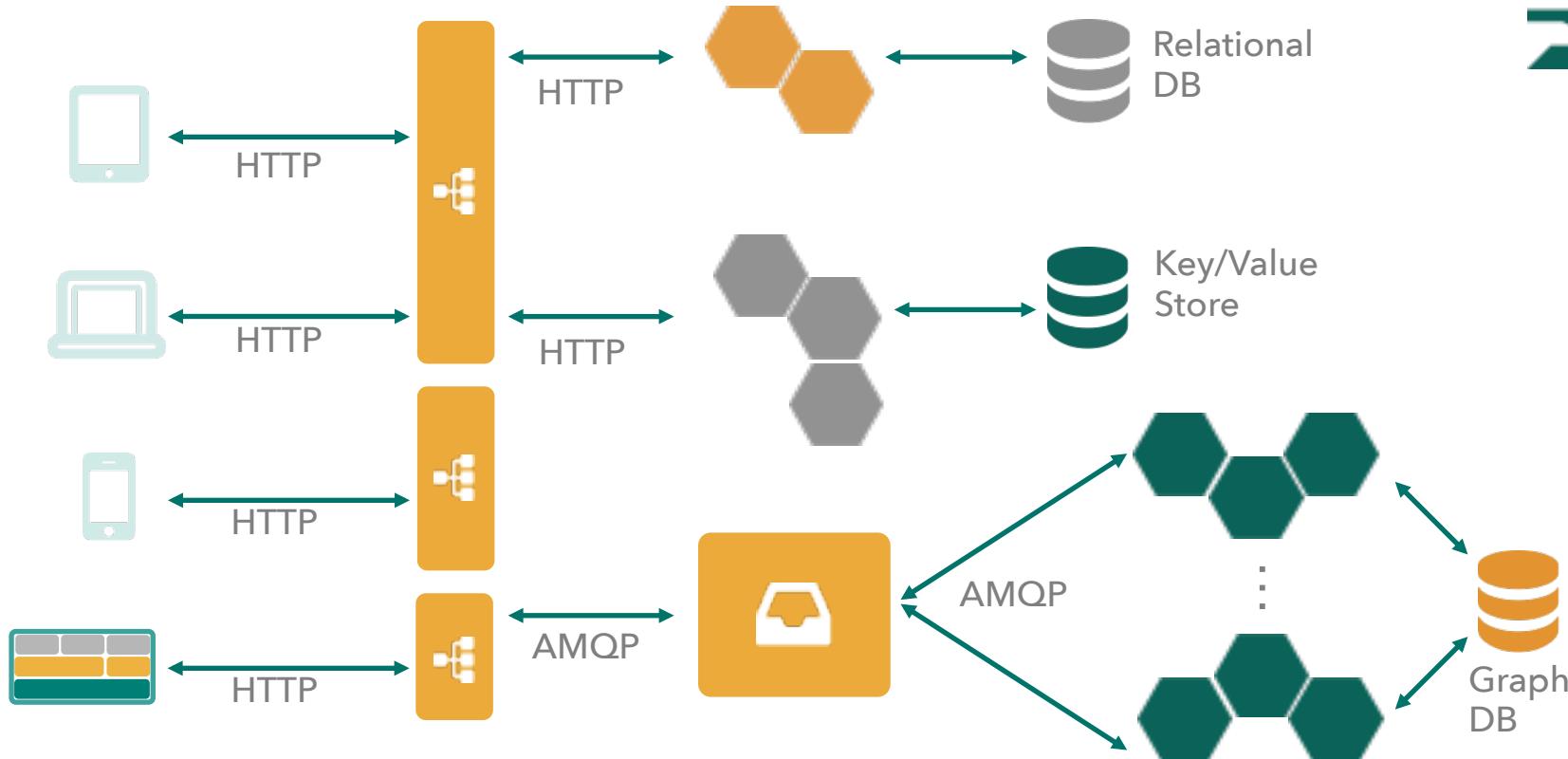
Tightly Coupled

OR

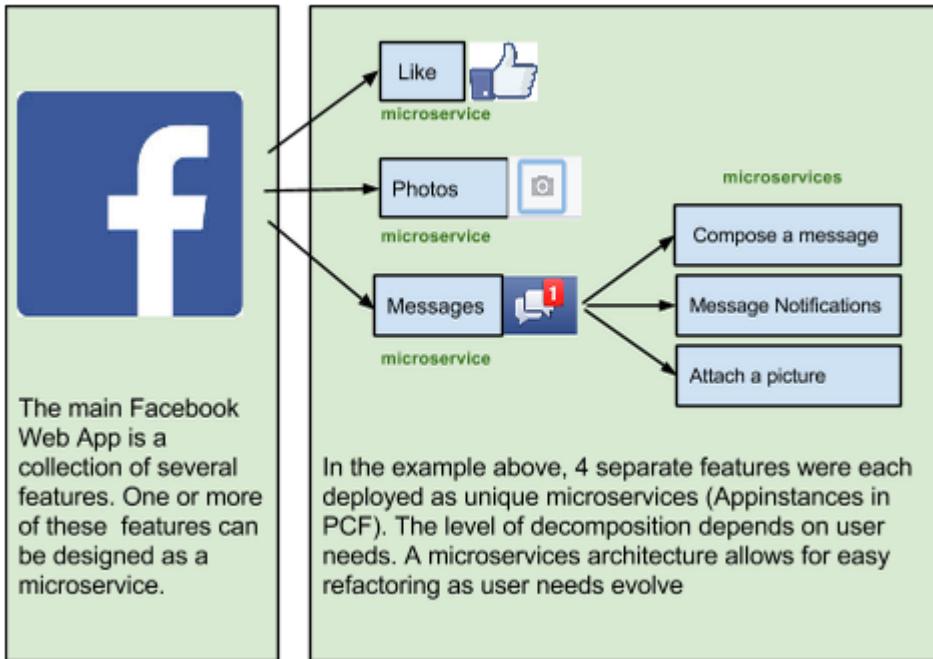


Centralized

Microservice Architecture



A Familiar Microservice Example



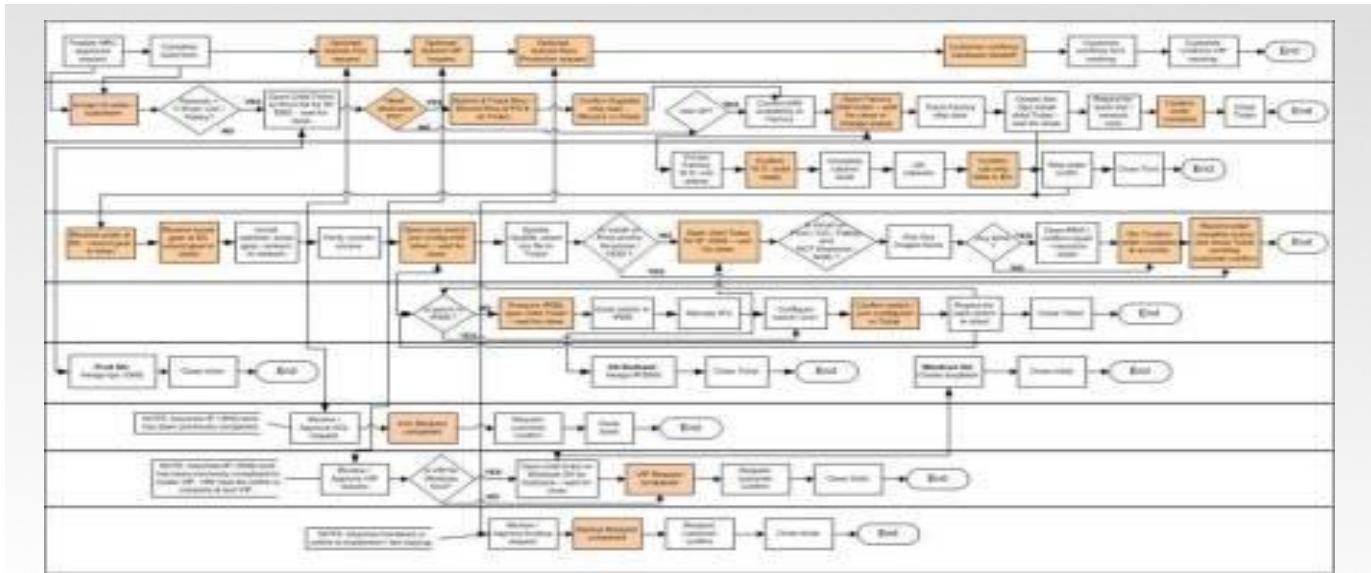
Monolithic Applications Drive Complex, Manual Deploys & Waterfall Release Cycles



Developer



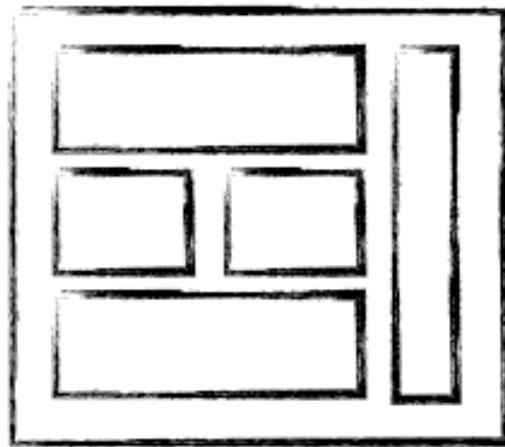
Operator



Can you deliver full CI/CD for every major app in your portfolio today, or are you doing 75+ step manual deployments?

Trend towards new lightweight architectures

Microservices addressing speed to market and cloud scale

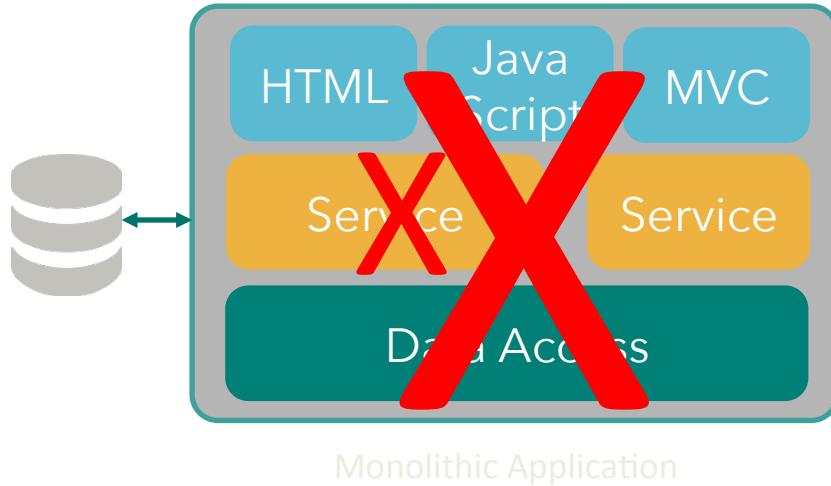


MONOLITHIC/LAYERED



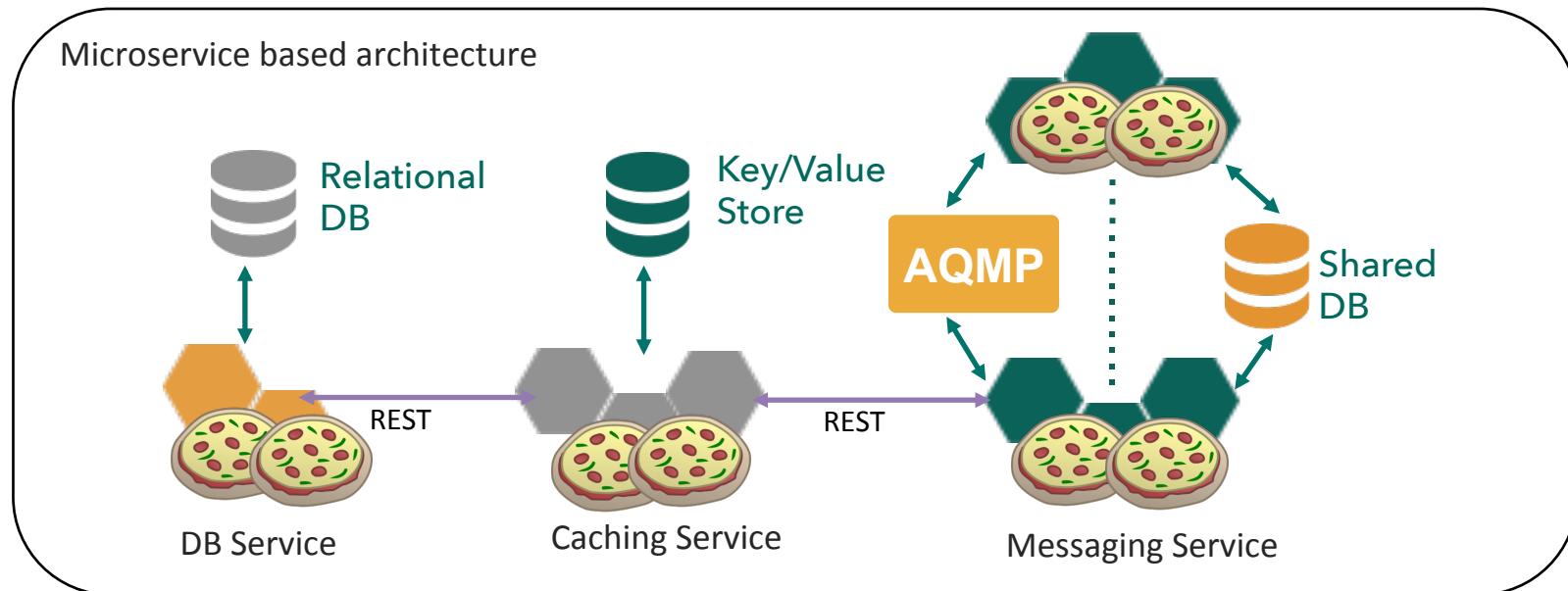
MICRO SERVICES

One-Size-Fits-All Methodologies have become an Anti-pattern to the Business



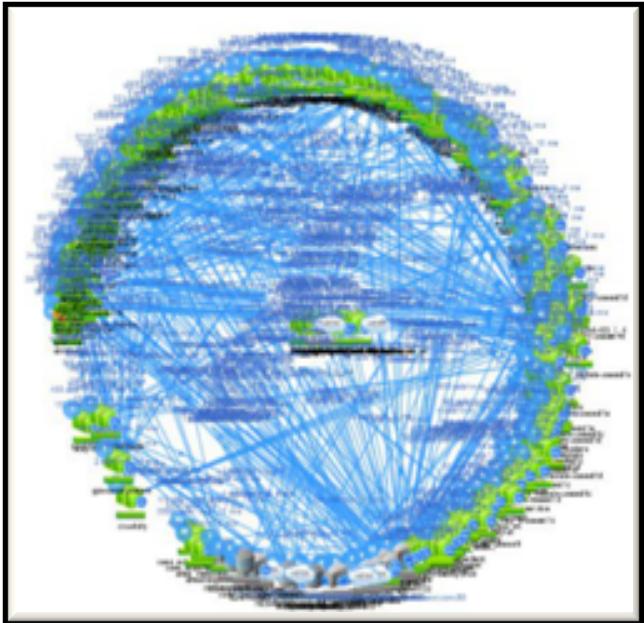
> 2 Pizzas Per Project = Too Many Pizzas

Agile, Disruptive Companies Use Non-traditional, Modular Approaches to Software Systems

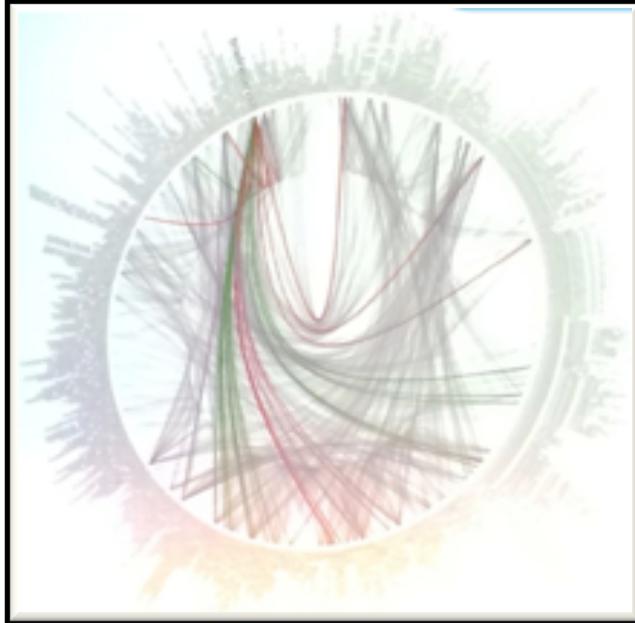


Two Pizzas Per Microservice = Manageable!

But.... microservices are hard



NETFLIX



twitter

Challenges of Distributed Systems

- Configuration Management
- Service Registration & Discovery
- Routing & Load Balancing
- Fault Tolerance (Circuit Breakers)
- Monitoring
- Concurrent API Aggregation & Transformation

How

It Takes a Platform

An end-to-end platform that makes implementing distributed application best practices, a **turn-key** and **first** practice

Cloud Native Platform and Contracts

Culture



Dev



Dev



IT Ops

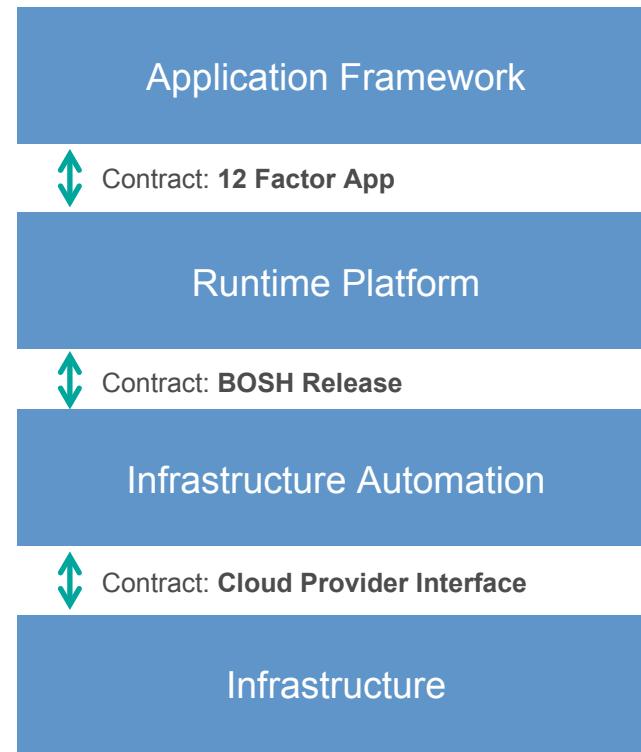


IT Ops



IT Ops

Framework



Tools



Spring Cloud



Spring Boot



Pivotal
Cloud Foundry



BOSH



AWS



Azure



VMWare



OpenStack

Pivotal



Application Framework



Spring Cloud



Spring Boot

Spring Boot



From 0 to app in < 5 min

Spring Boot provides

- A single point of focus (as opposed to large collection of spring-* projects)
- Prebuild “starters”
- Common non-functional requirements for a “real” application
- Exposes a lot of useful features by default
- Gets out of the way quickly if you want to change defaults



Config Server



Service Registry



Circuit Breaker

- Eureka
- Hystrix + Turbine
- Ribbon
- Feign
- Zuul
- Archaius



NETFLIX

OSS

D E M O



CF

Companies want to be fast like Netflix

- Netflix needed to be faster to win / disrupt
- Pioneer & vocal proponent of microservices - the key to their speed and success
- Netflix OSS supplies parts, but it's not a solution
- Difficult for enterprises to build it themselves
- Pivotal offers the closest thing to “Netflix in a box” today



“

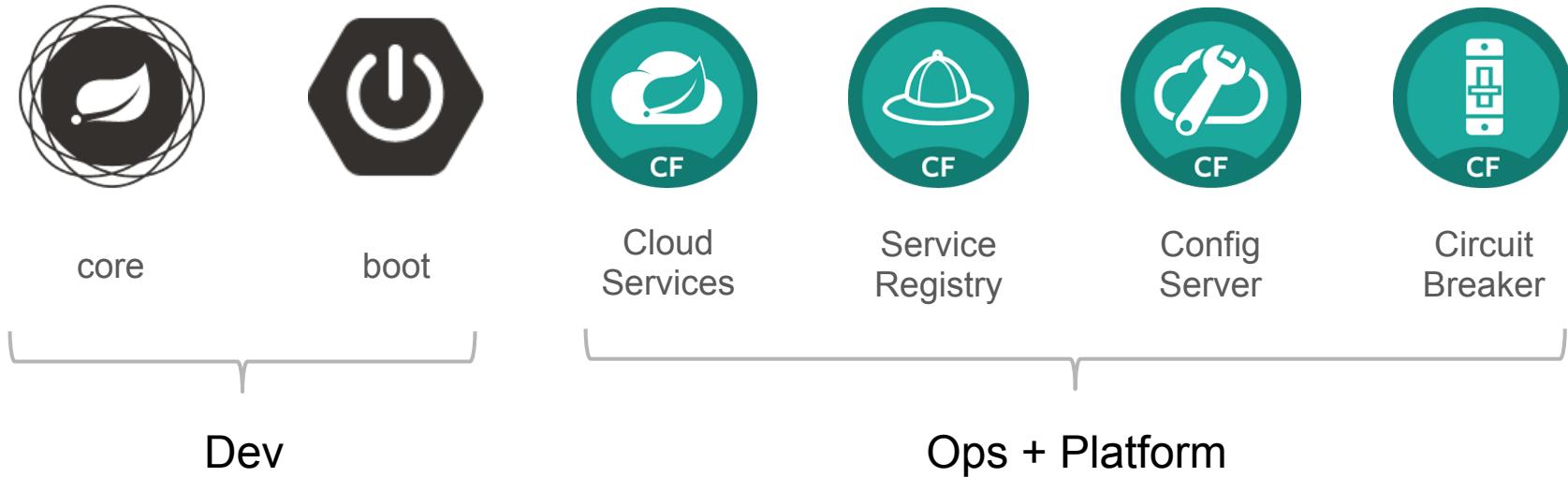
“Velocity on the JVM is the Killer App”

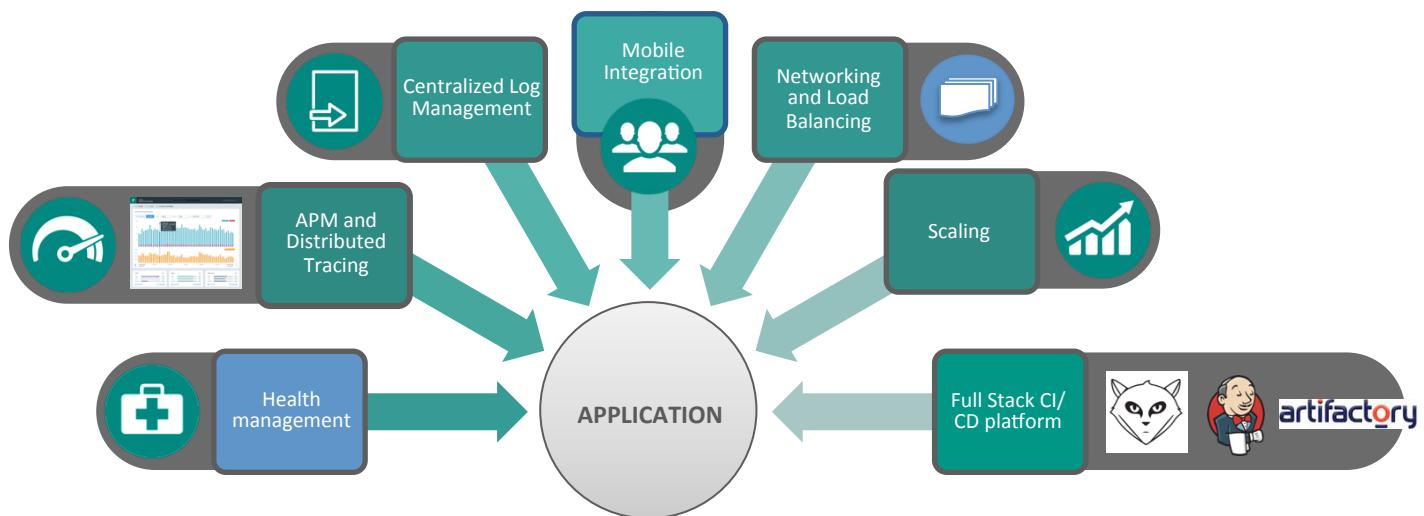
- Andy Glover (Netflix Eng) @ SpringOne2GX 2014 Keynote

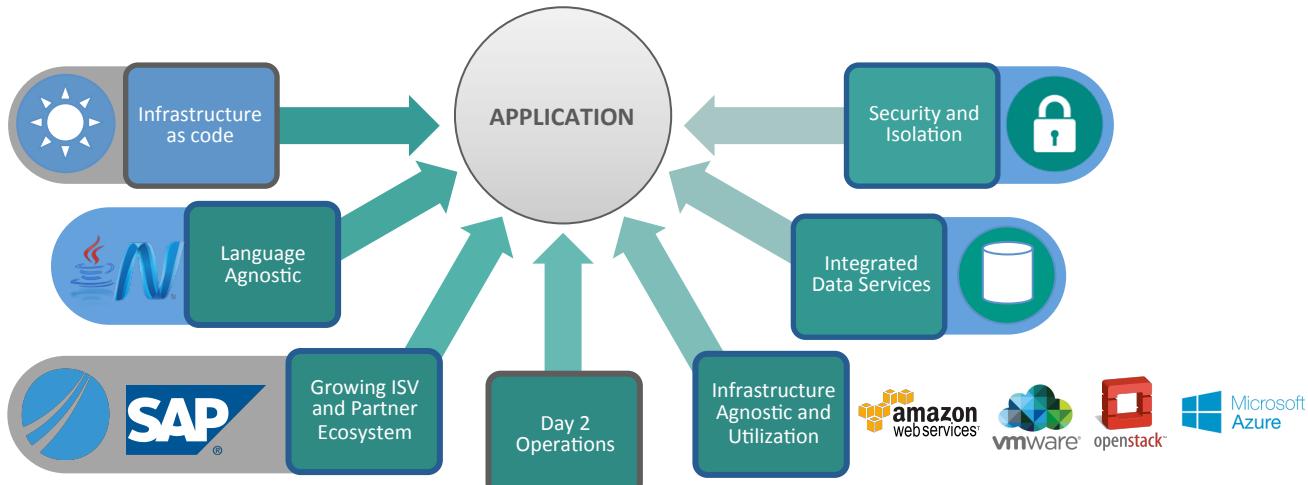
<https://youtu.be/xU267-YHN5c?t=1938>

Spring Cloud Services

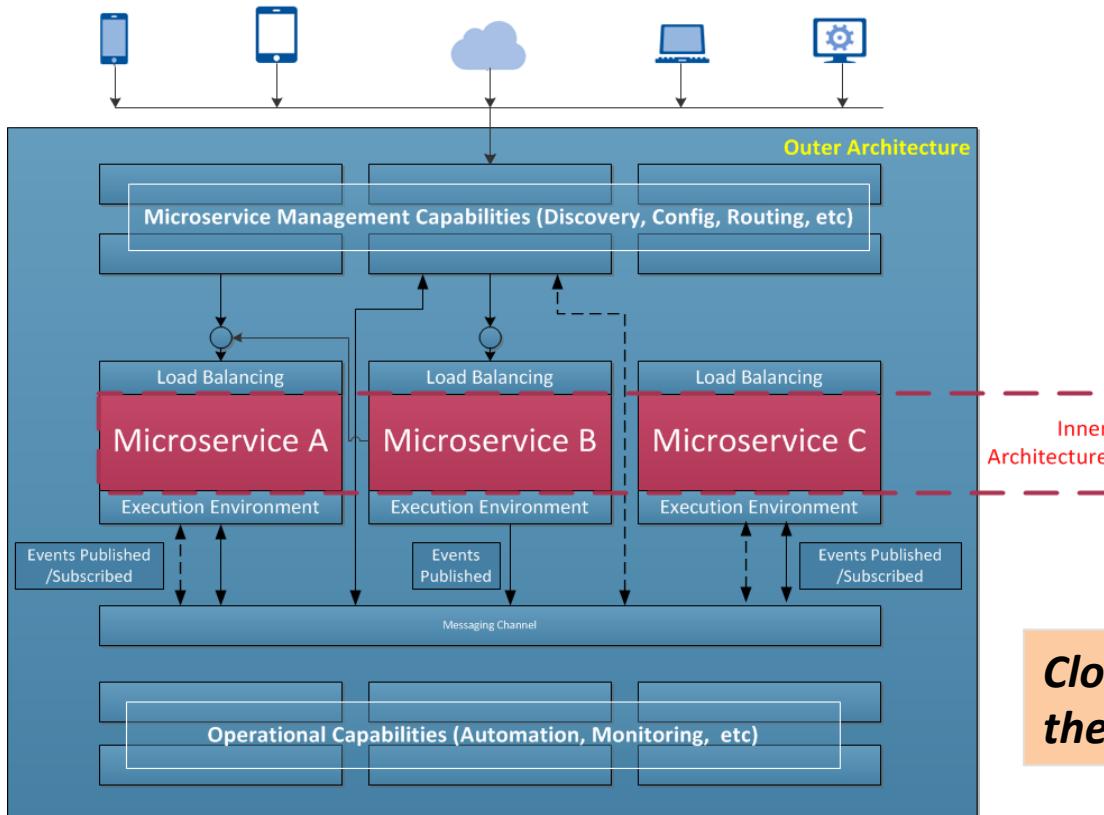
Rich, production ready library based on Netflix OSS for cloud native components, security and management.







The Role of Cloud Native Platforms in Microservices

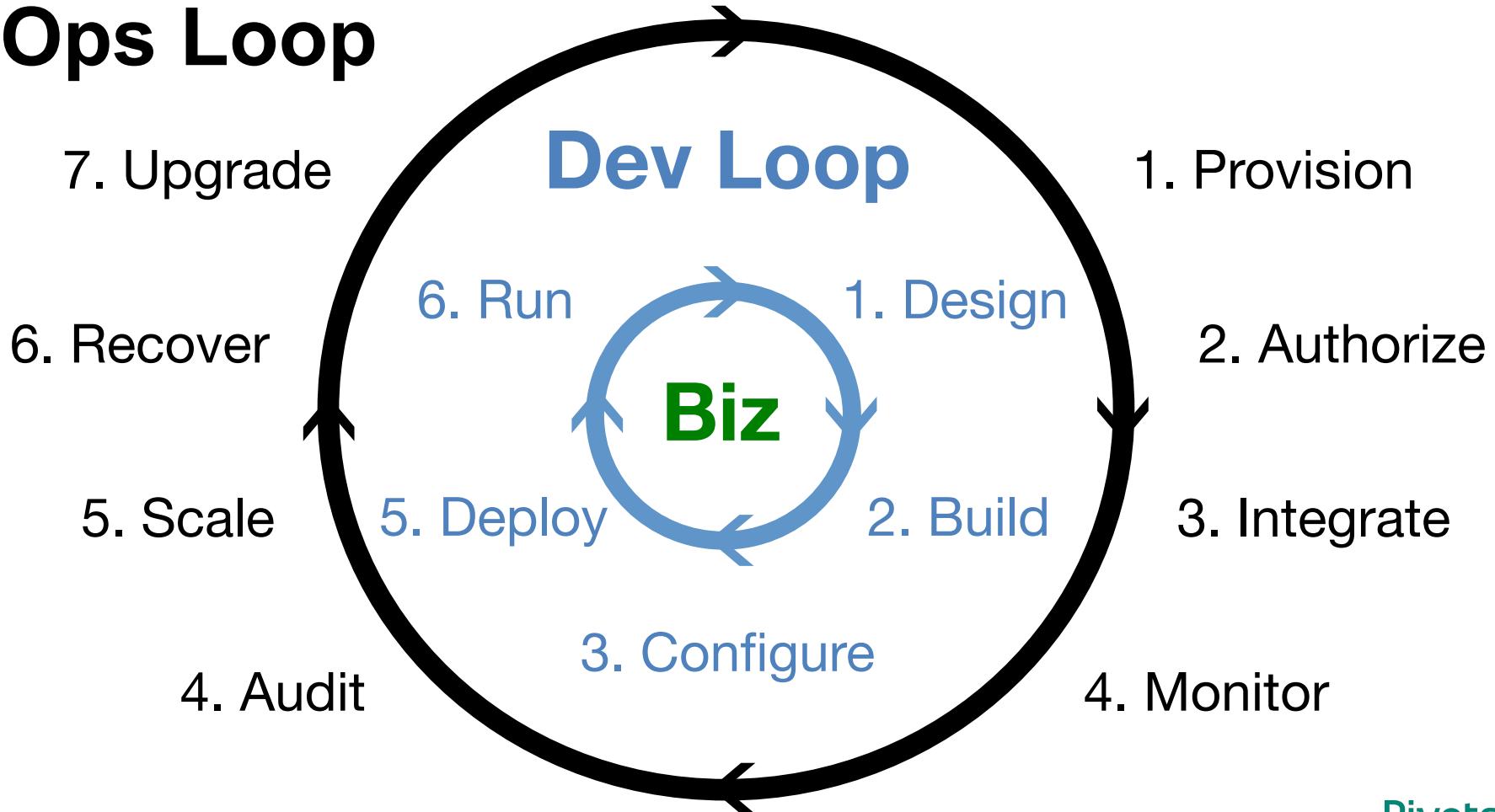


"The hardest part of building microservices is in the outer architecture (All of the blue boxed components)."

"Easiest part of the microservices is in the inner architecture. The architecture of an individual microservice."

Cloud Native platforms provide the outer architecture

Ops Loop



Application Lifecycle Management: CI/CD



AUTOMATION.

Integrate tools and automate processes from testing to builds and deployment.

SPEED.

Releasing more frequently with fewer bits will reduce complexity and improve time-to-market.

QUALITY.

Shorten feedback loop using test-driven development to surface problems sooner.

AGILITY.

Push updates on regular basis with no downtime to improve customer experience and time to market.

Build Pipeline Operations

Commit Code Change



Distributed revision control and source code management. Collaborative software development.

Automate Build & Test



Build, test and deploy software projects continuously and incrementally. Thousands of compatible plugins.

Manage Binaries & Build Artifacts



Share binaries and manage distributions. Manage artifact lifecycle.

Pivotal Cloud Foundry (Elastic Runtime)

Pivotal®

Develop, Test, QA and Production on the same platform. Horizontal scaling, high availability, security, logging, update management and other operational benefits for every application. Built-in ecosystem of services. Deploy, operate and scale on IAAS of choice. Simple, developer friendly commands and APIs.

Pivotal

Cloud Native Maturity Model

Cloud Native

- Microservices architecture
- API-first design

Cloud Resilient

- Fault-tolerant and resilient design
- Cloud-agnostic runtime implementation
- Bundled metrics and monitoring
- Proactive failure testing

Cloud Friendly

- 12 Factor App methodology
- Horizontally scalable
- Leverages platform for high availability

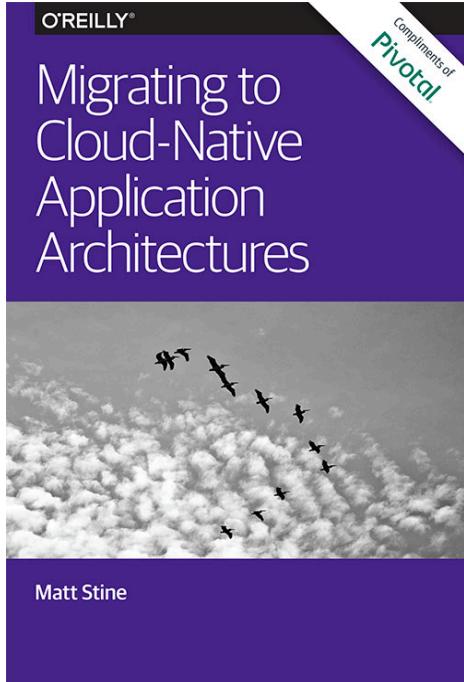
Cloud Ready

- No permanent disk access
- Self-contained application
- Platform-managed ports and networking
- Consumes platform-managed backing services

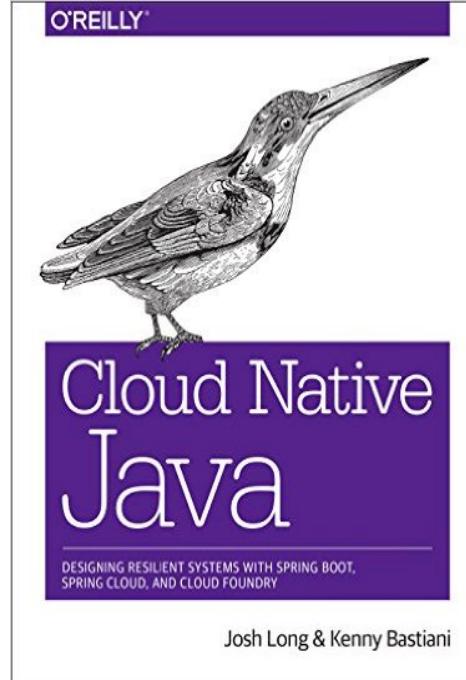
How

Do we get started

Read



By Matt Stine (@mstine)



By Josh Long (@starbuxman)
and Kenny Bastani

Bootstrap your Application Now: <http://start.spring.io>

SPRING INITIALIZR bootstrap your application now

Generate a Maven Project with Spring Boot 1.3.2

Project Metadata

Artifact coordinates

Group copy

Artifact

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

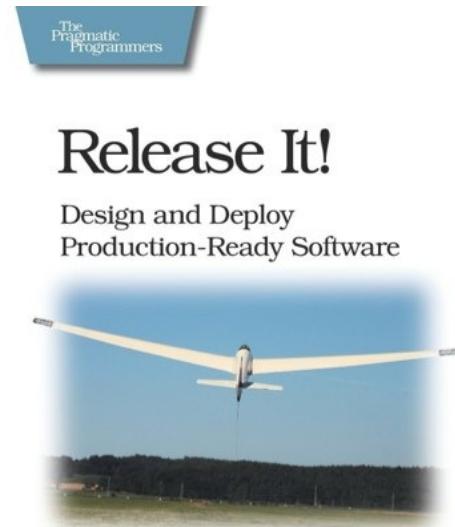
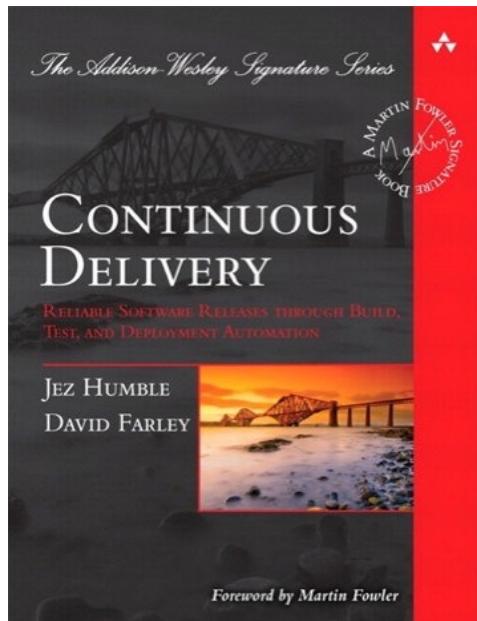
Selected Starters

Generate Project ⌘ + ↵

Don't know what to look for? Want more options? [Switch to the full version.](#)

start.spring.io is powered by [Spring Initializr](#) and [Pivotal Web Services](#)

Foundations



Michael T. Nygard

From the authors of *The Visible Ops Handbook*

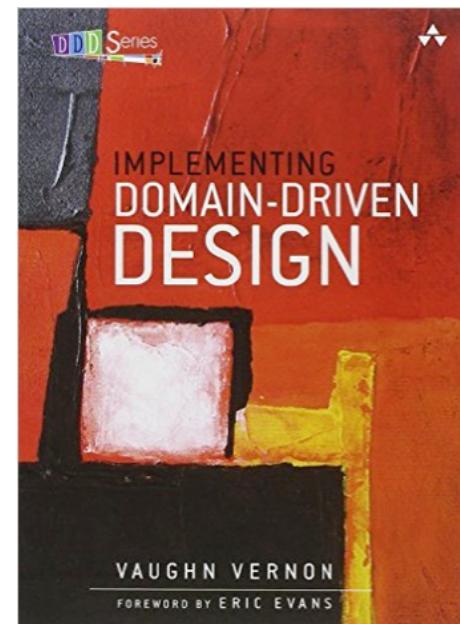
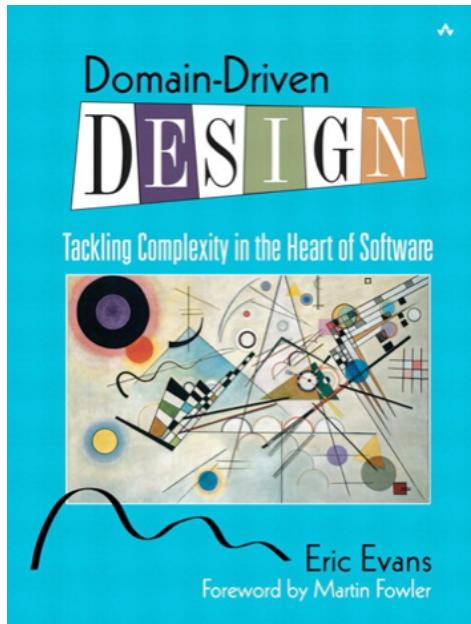
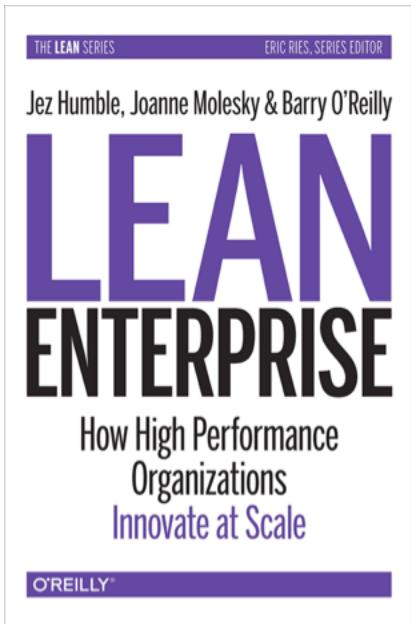


The Phoenix Project

A Novel About IT, DevOps,
and Helping Your Business Win

Gene Kim, Kevin Behr, and George Spafford

Foundations



BREAK

