



Pivotal

FedEx

# 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 already here

Google

NETFLIX

facebook.

amazon

# Fortune 500 Companies are Adopting Cloud Native and Microservice Models

“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

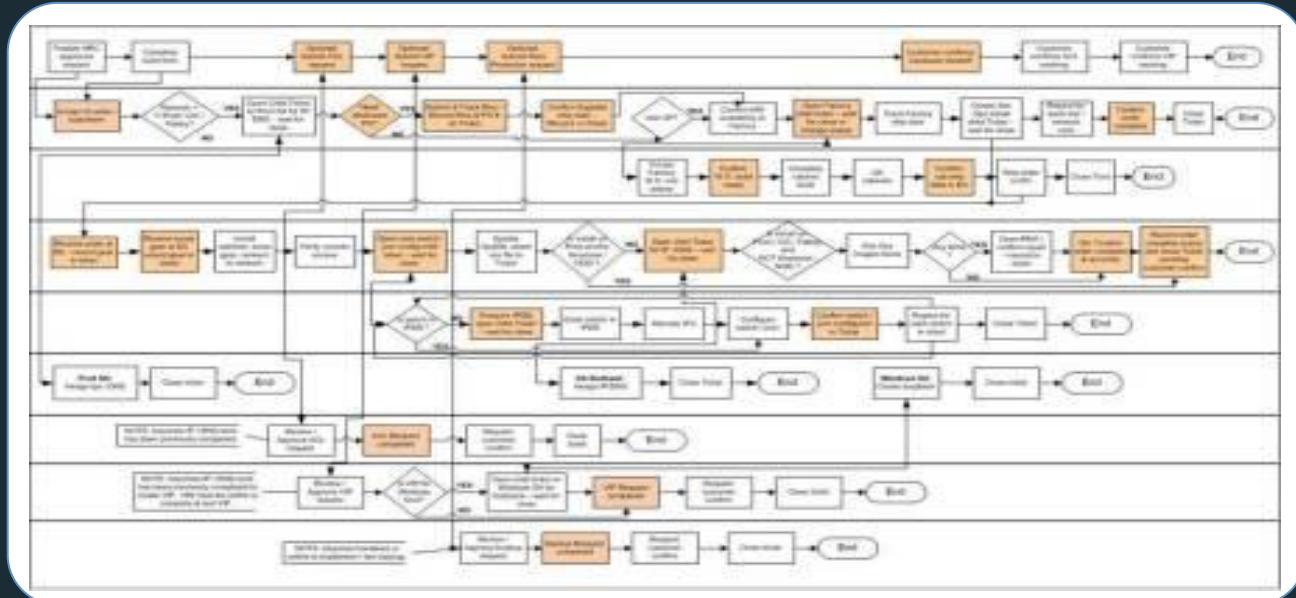
# Monolithic Applications Drive Complex, Manual Deploys & Waterfall Release Cycles



Developer



Operator



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

# The Promise of Cloud Native

- Deploy new **features** \*daily\* to production
- Faster Feedback (experiment)
- Outcome not activity focus (optimize)
- Loose Coupling (upgrades and fixes)
- Self-service, automated middleware
- Free/Cheap horizontal scaling
- No Downtime Upgrades
- Availability and resiliency

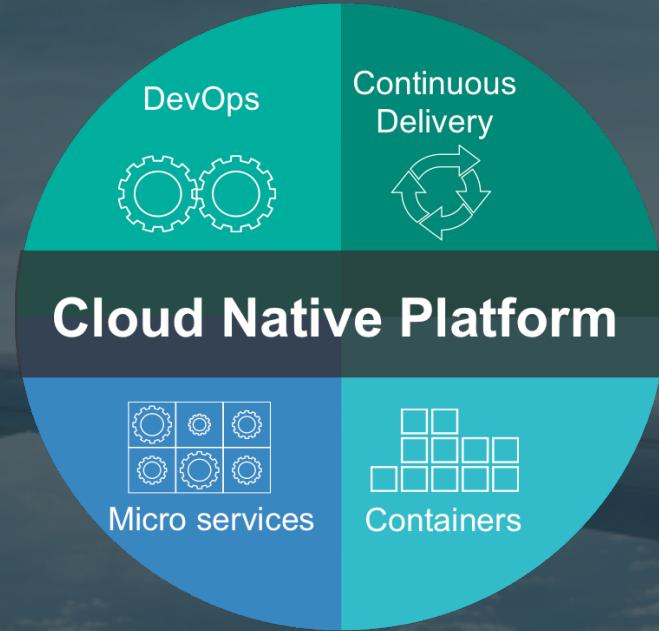
# WHAT

# 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

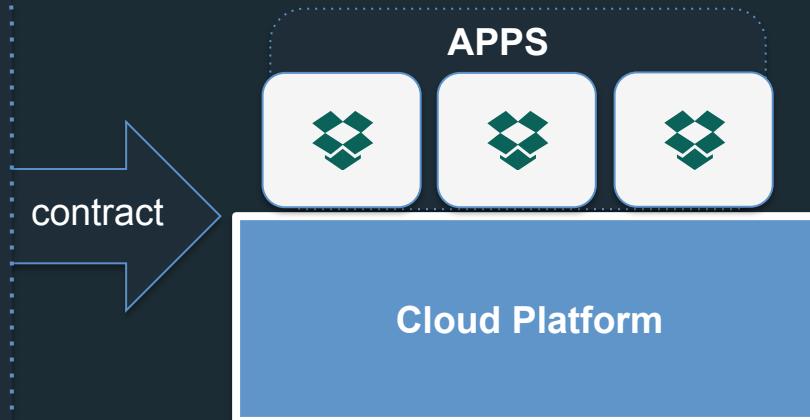
- Twelve Factor Apps (<http://12factor.net/>)
- Continuously Delivered
- Microservices
- DevOps
- Self-Service agile infrastructure
- API-based collaboration
- Anti-fragility

# Characteristics of Cloud Native Architectures

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

# 12-Factor Applications

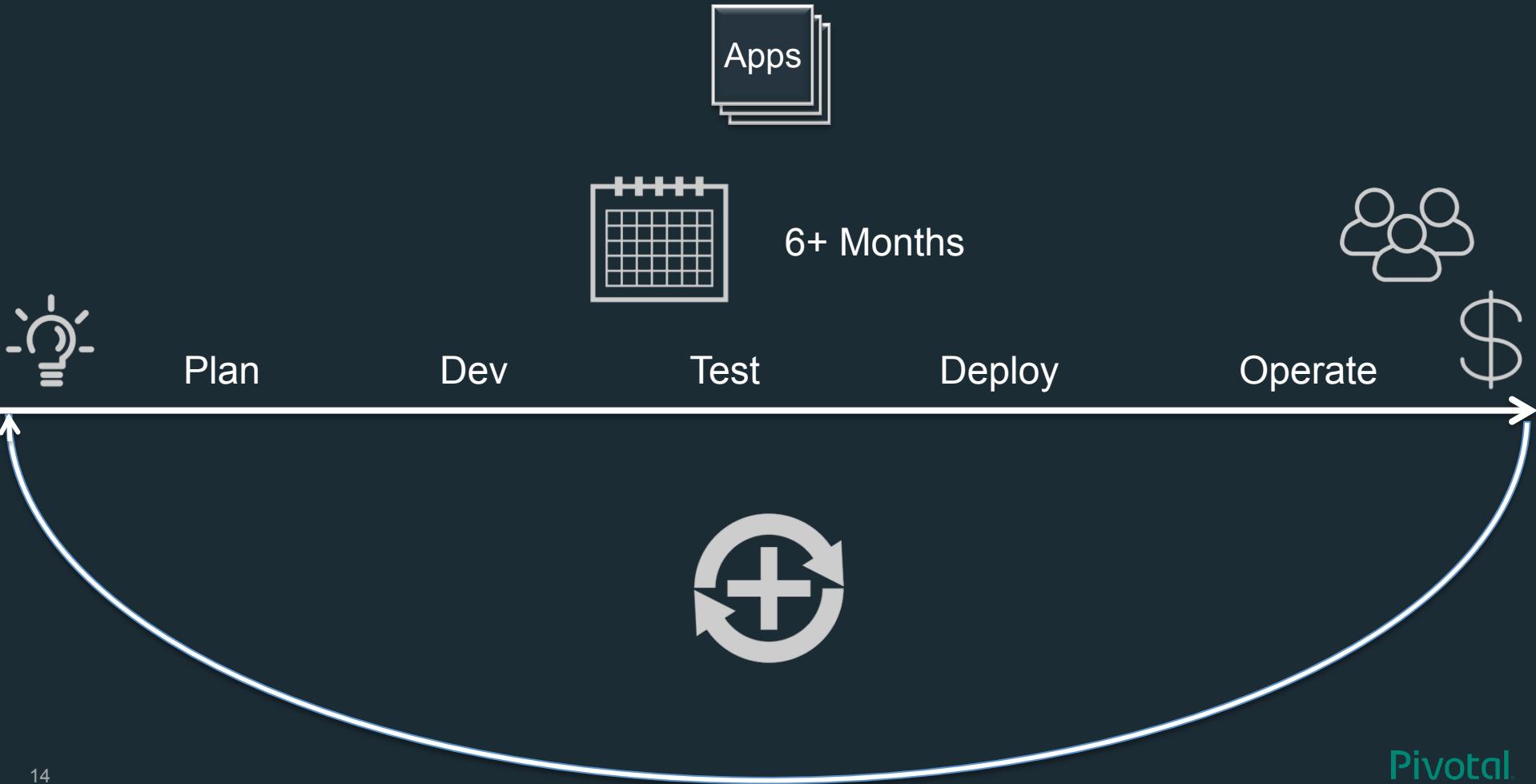
- 1. Codebase
- 2. Dependencies
- 3. Configuration
- 4. Backing Services
- 5. Build, release, run
- 6. Processes
- 7. Port binding
- 8. Concurrency
- 9. Disposability
- 10. Dev/prod parity
- 11. Logs
- 12. Admin processes

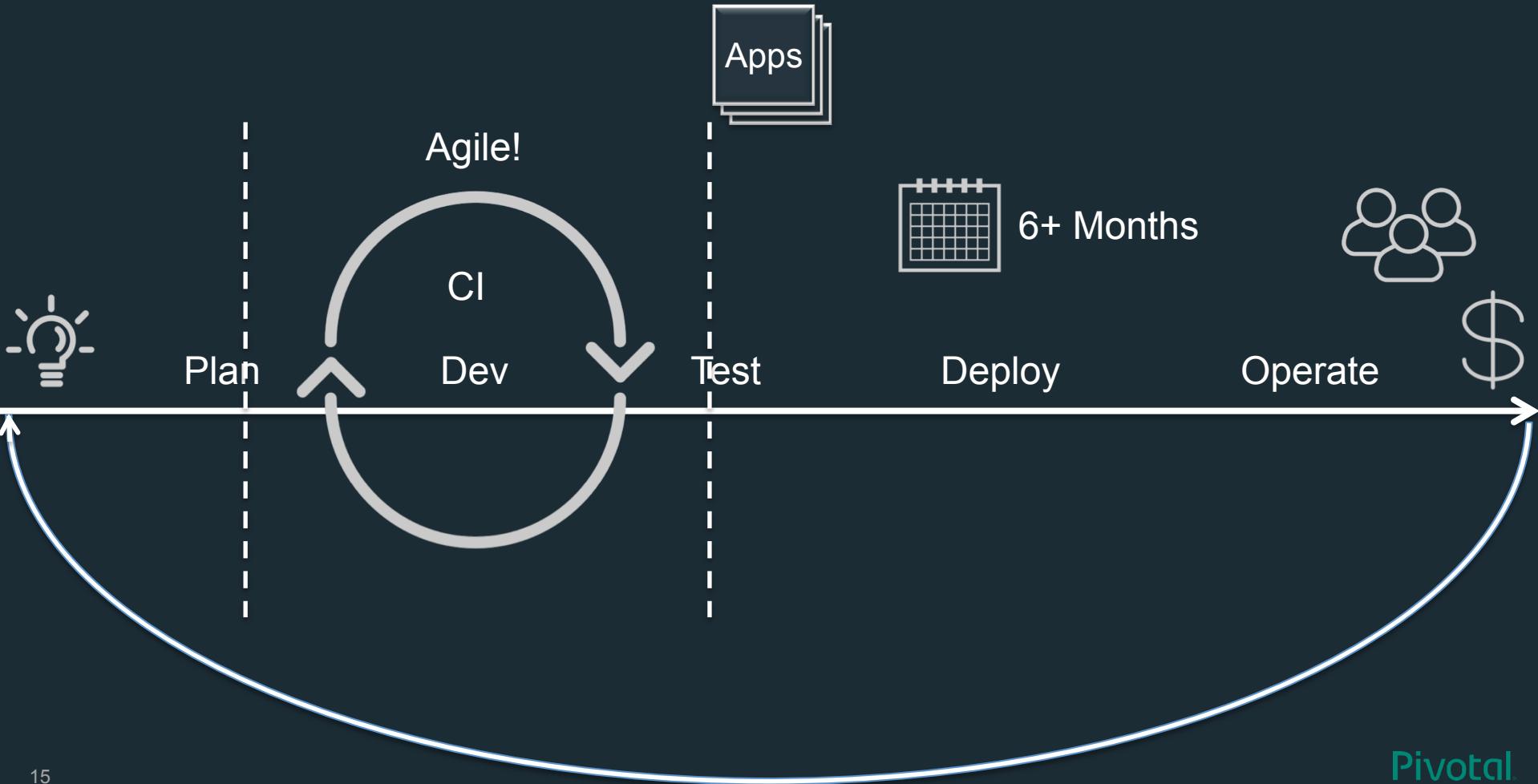


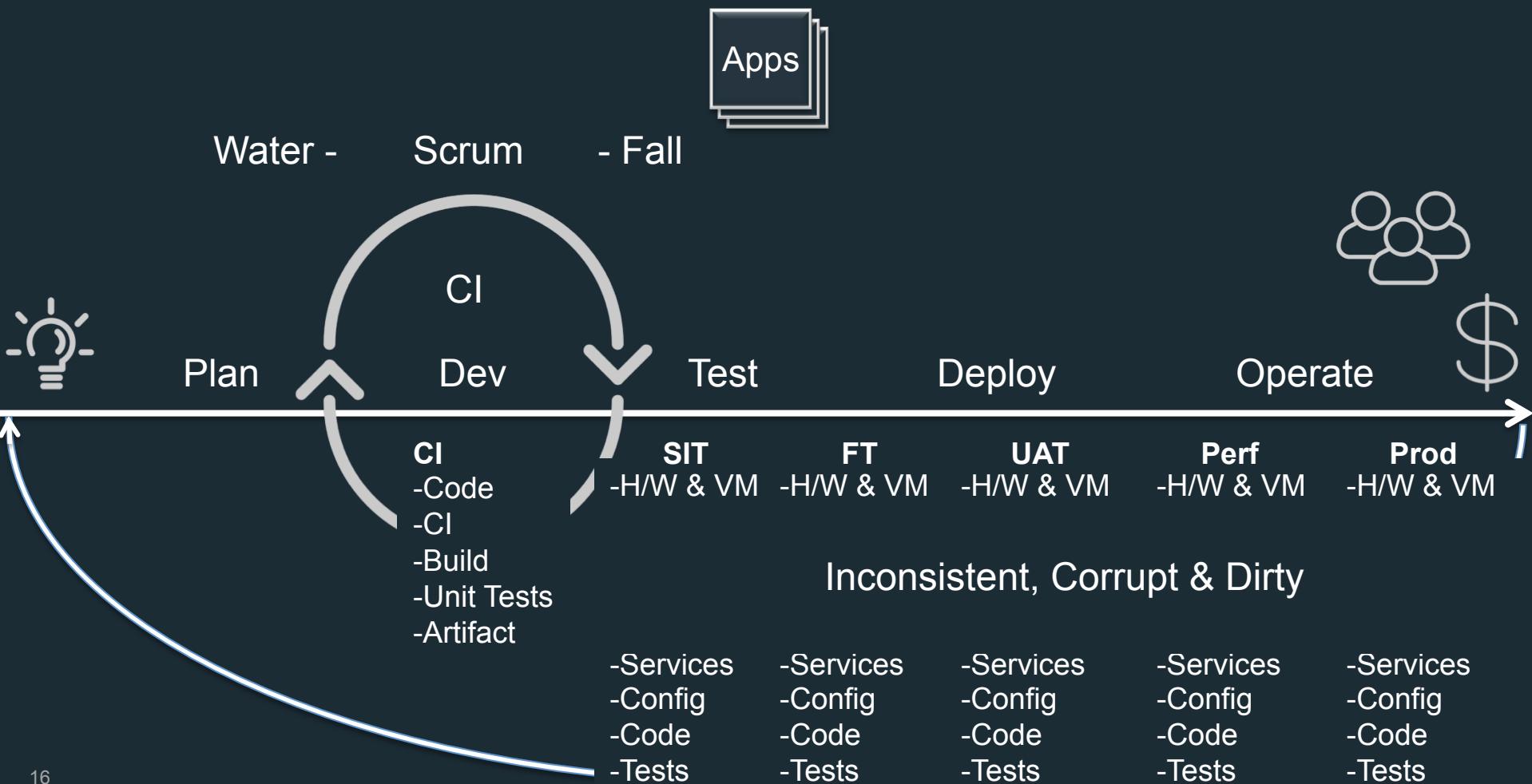
Architectural and development practices – <http://12factor.net>

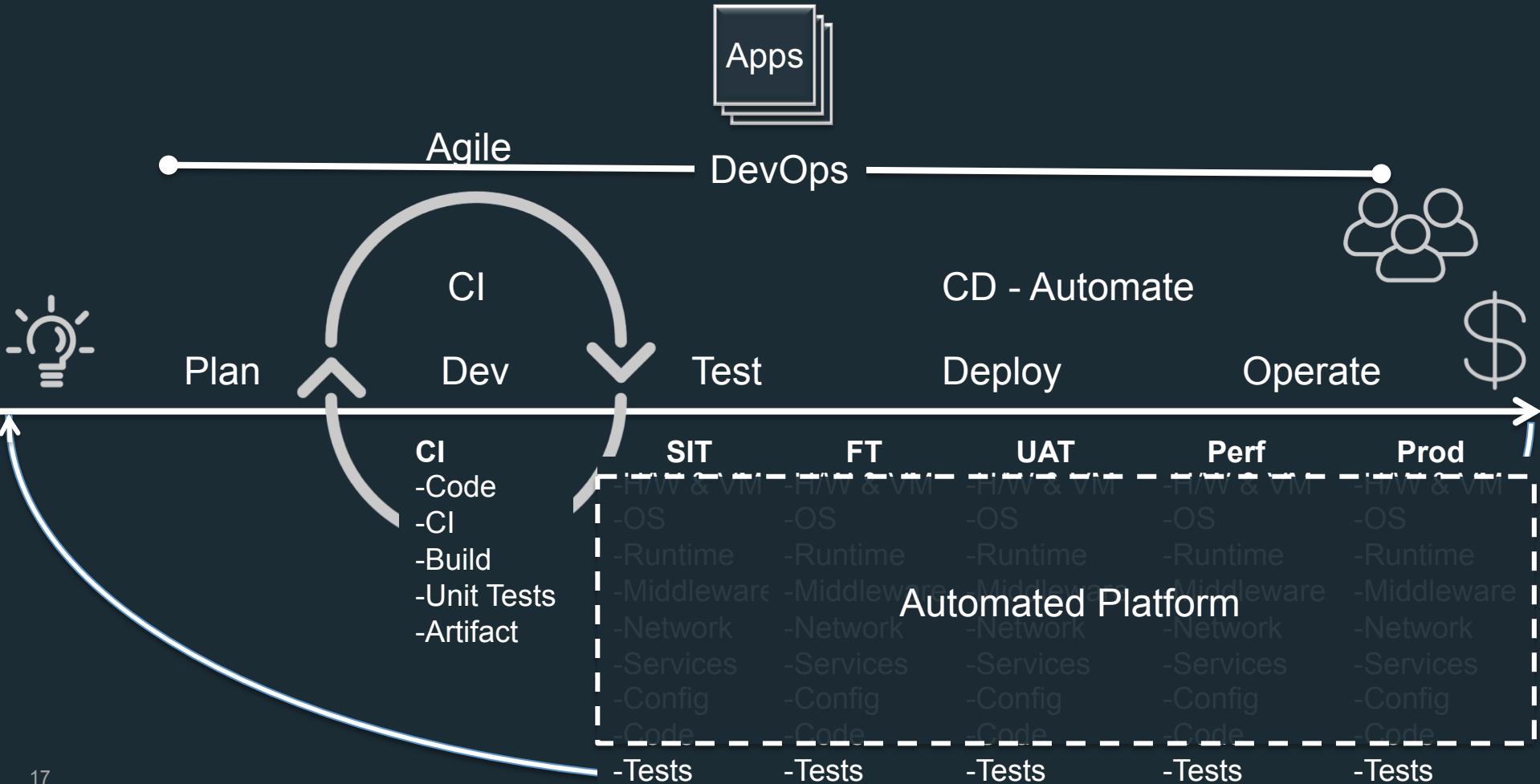
# Characteristics of Cloud Native Architectures

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





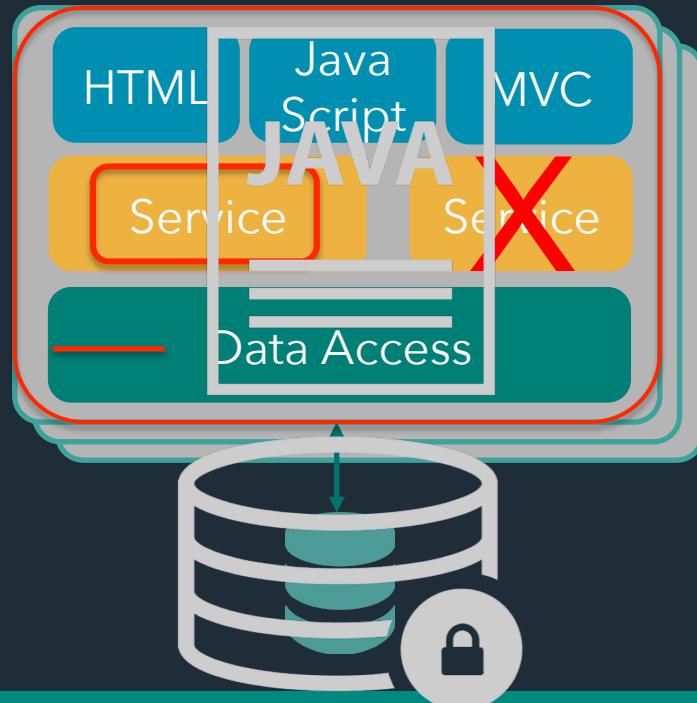
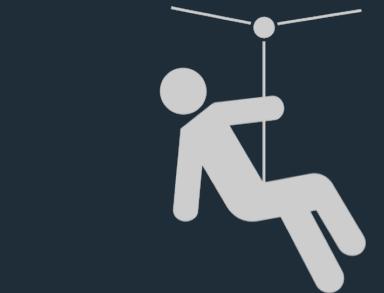




# Characteristics of Cloud Native Architectures

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

# The Monolithic Application



# What are Microservices?

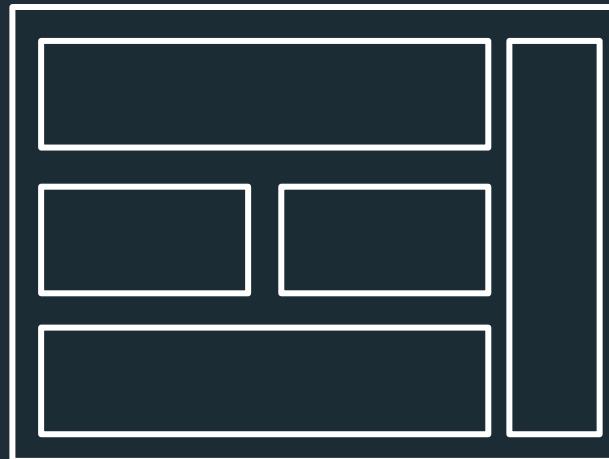
If every service has to be updated in concert,  
it's not loosely coupled!

Loosely coupled service oriented  
architecture with bounded  
contexts

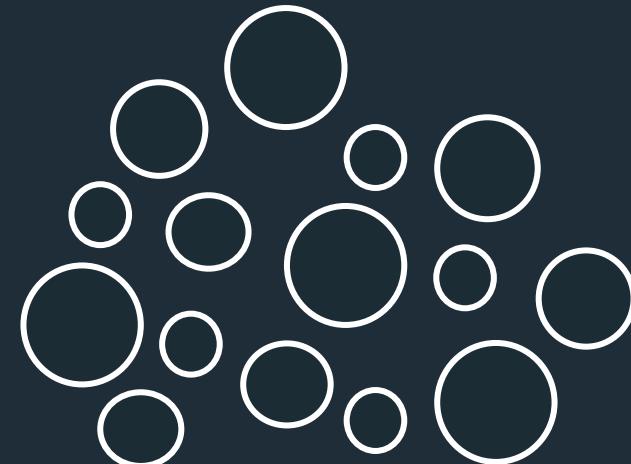
If you have to know about surrounding  
services you don't have a bounded context.

# Trend towards new lightweight architectures

Microservices addressing speed to market and cloud scale

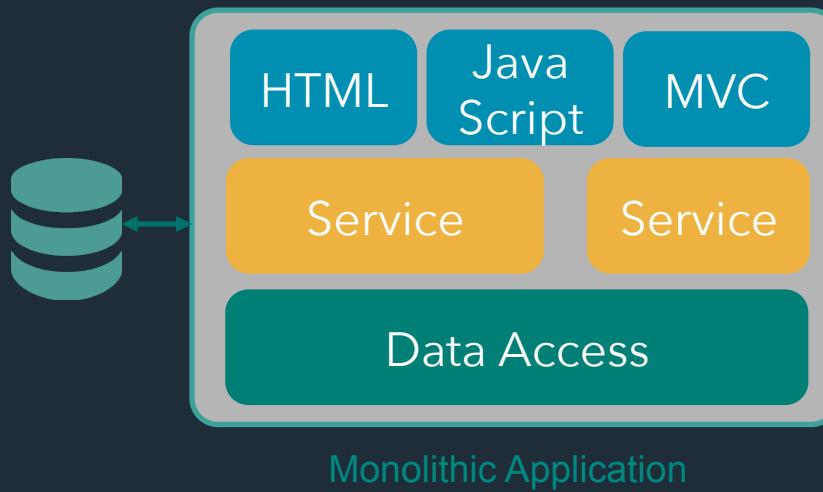


MONOLITHIC/LAYERED



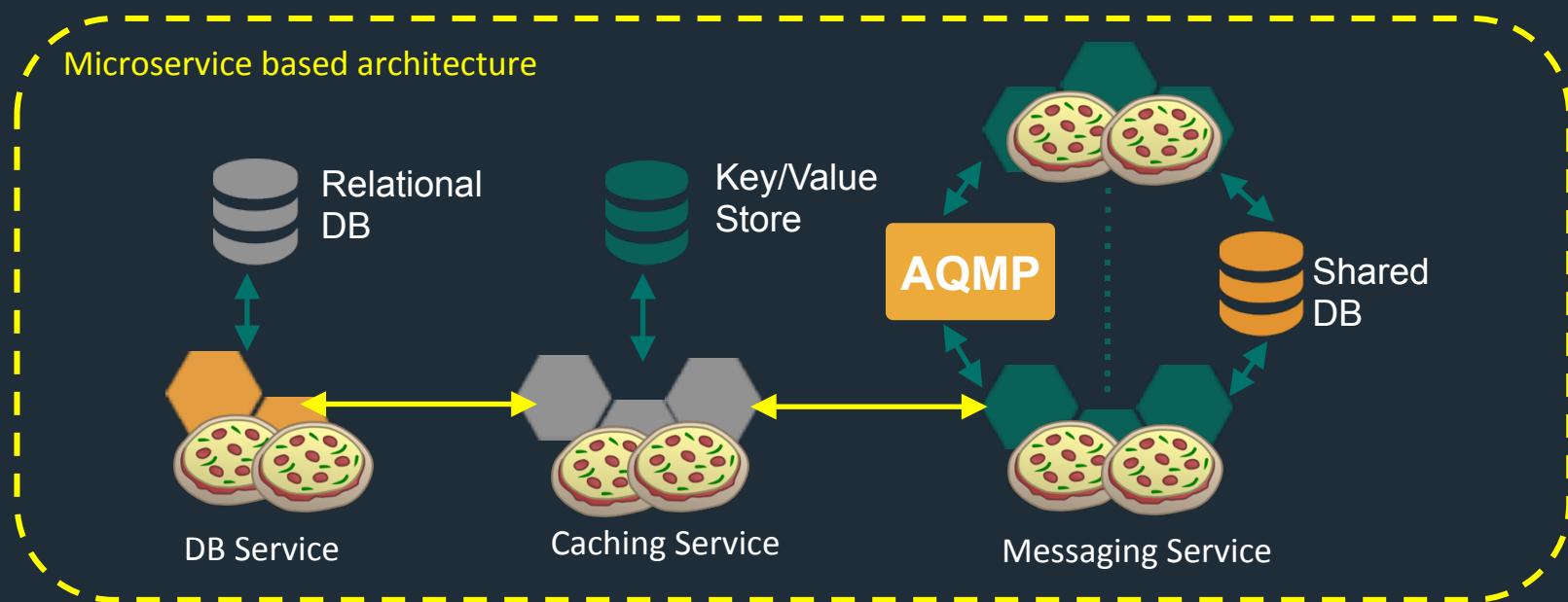
MICROSERVICES

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



> 2 Pizzas Per Project = Too Many Pizzas

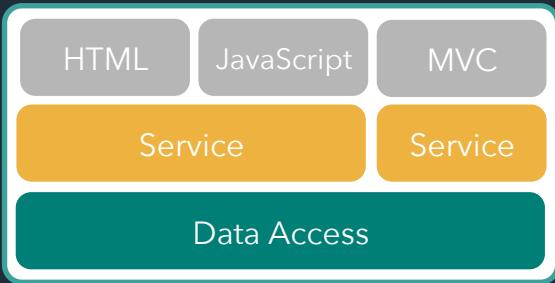
# Agile, Disruptive Companies Use Non-traditional, Modular Approaches to Software Systems



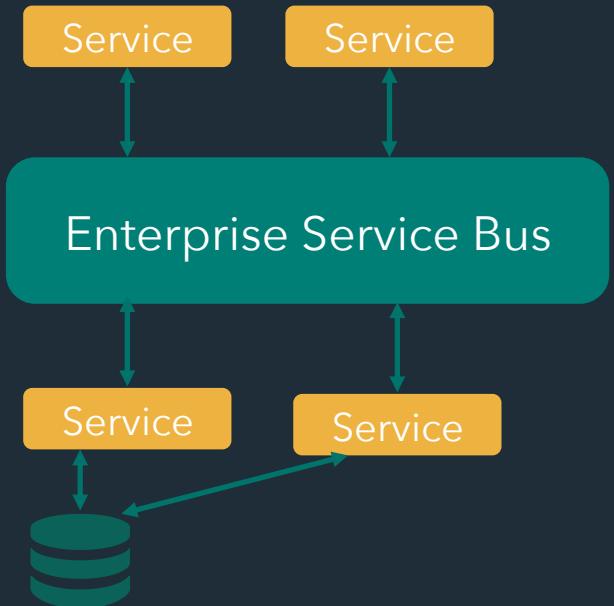
Two Pizzas Per Microservice = Manageable!

# Microservices are NOT

Monolithic Application



OR

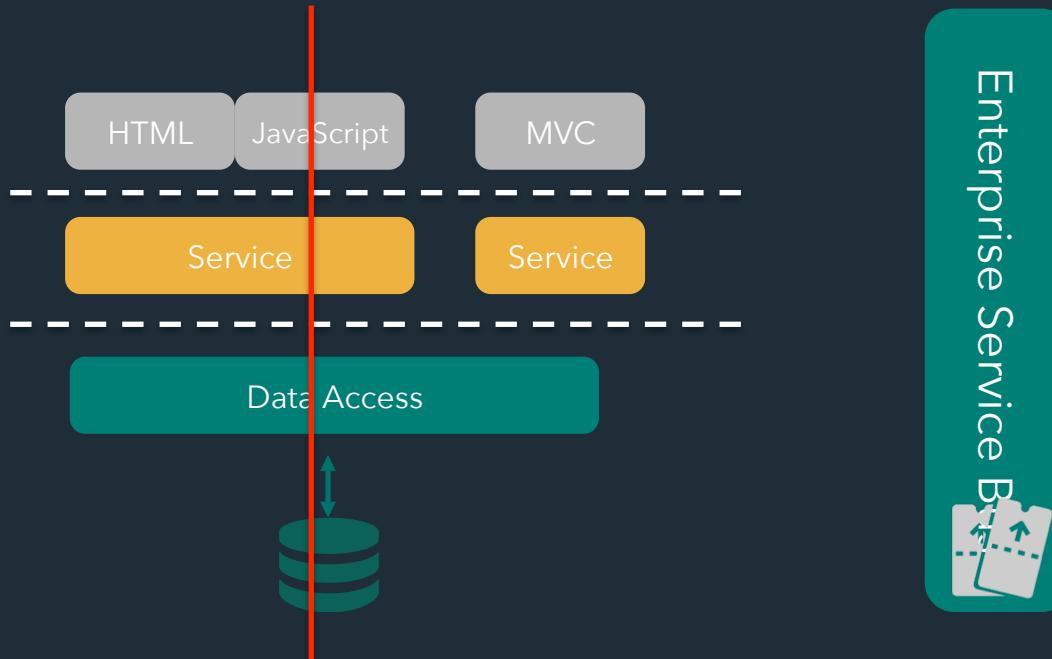


Tightly Coupled

Centralized

# SOA Architecture

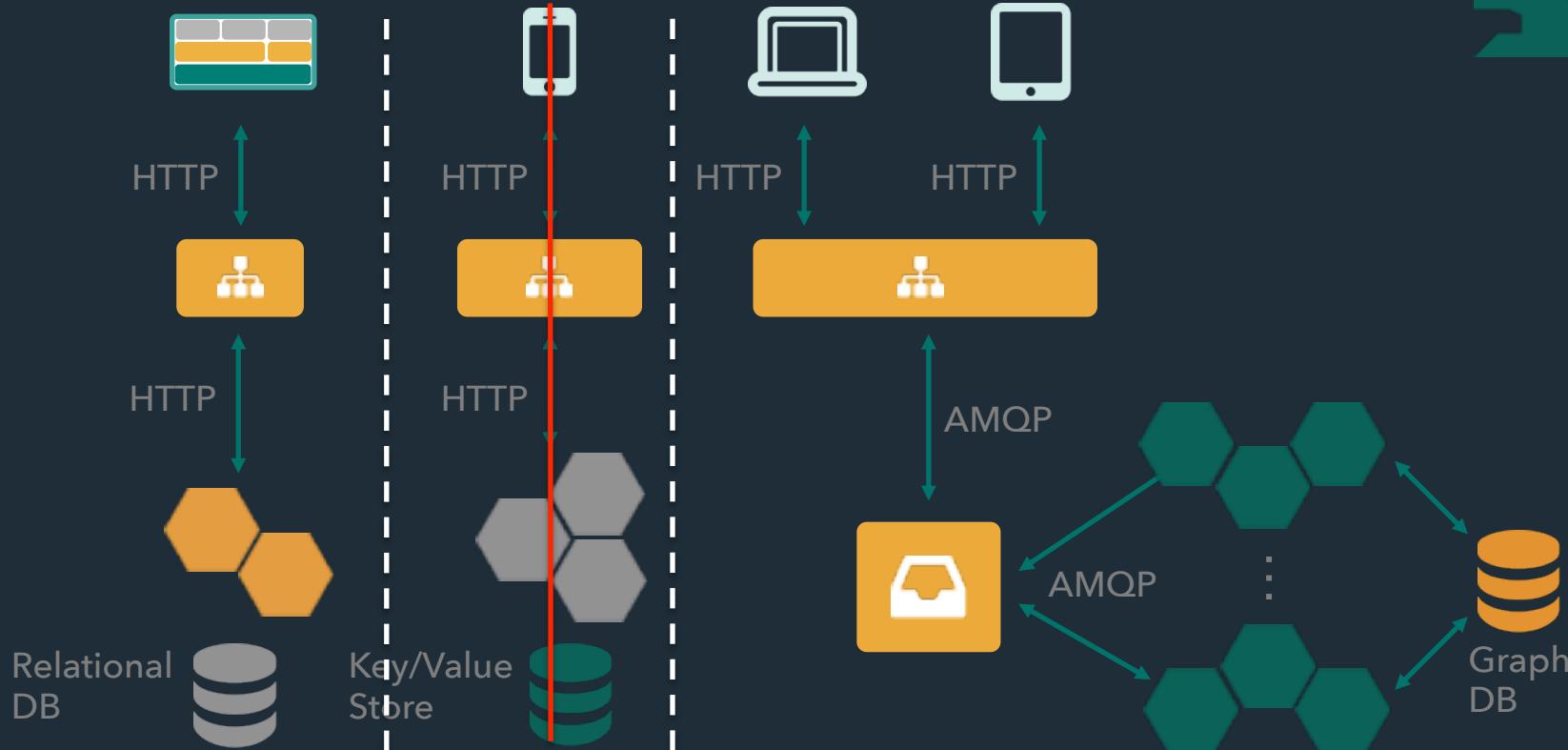
Business Change



# Microservice Architecture

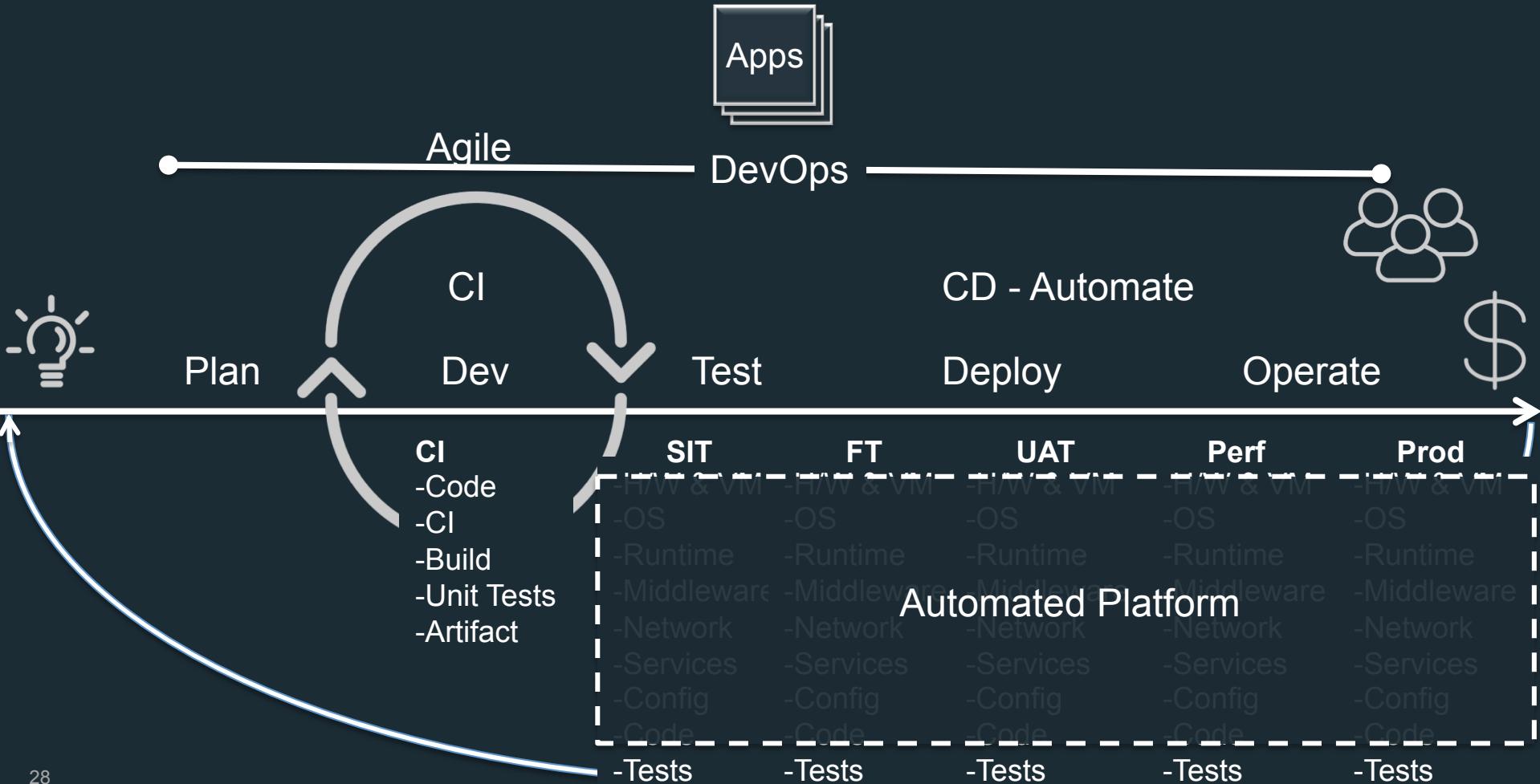


Business Change



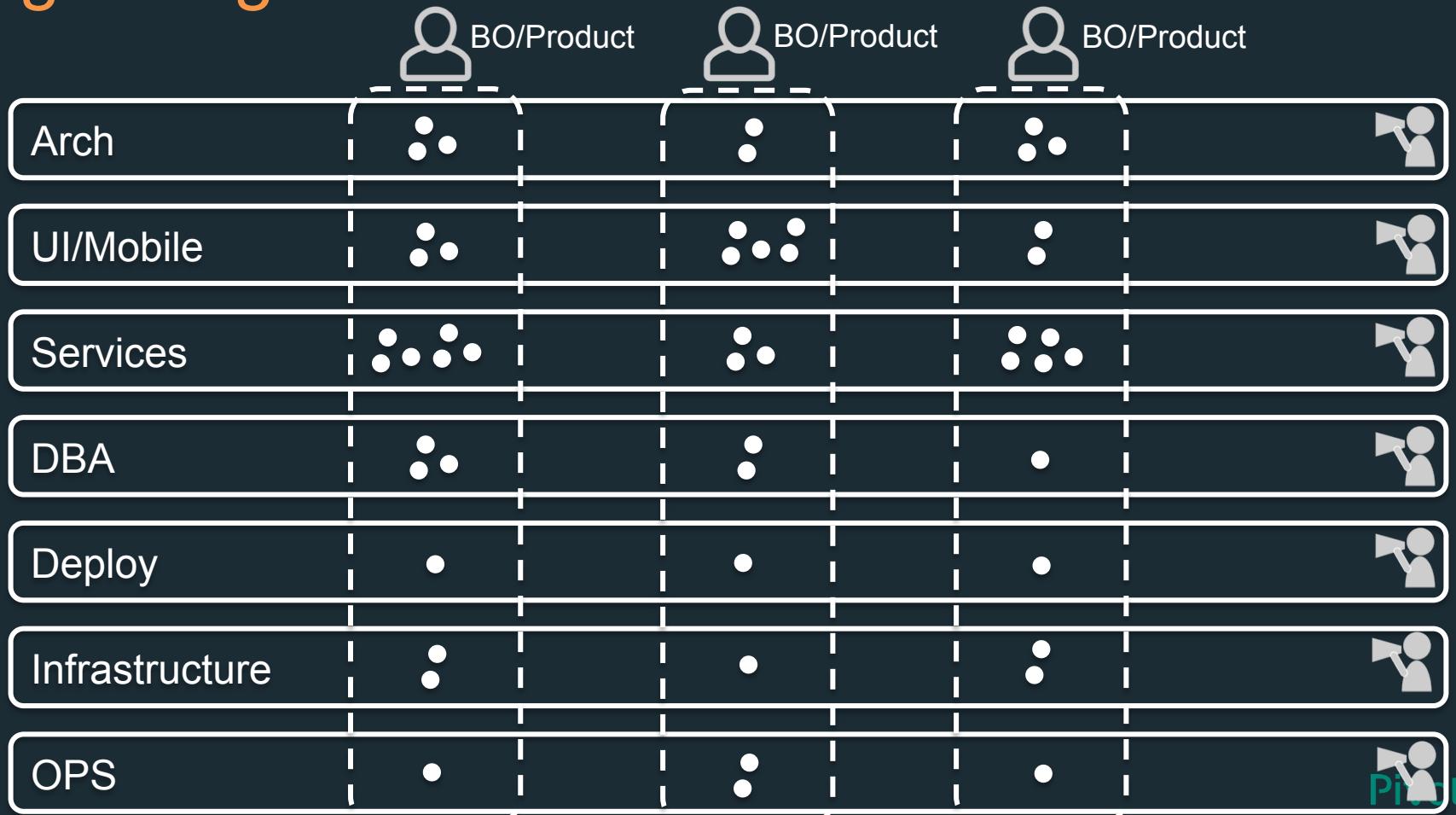
# Characteristics of Cloud Native Architectures

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



DevOps is a **professional movement** to adopt modern **roles**, cultural **behaviors**, **practices**, **design patterns** and **technology** for acquiring, developing and operating software services with increased **agility** and **reliability**.

# Agile Org Structures



# 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

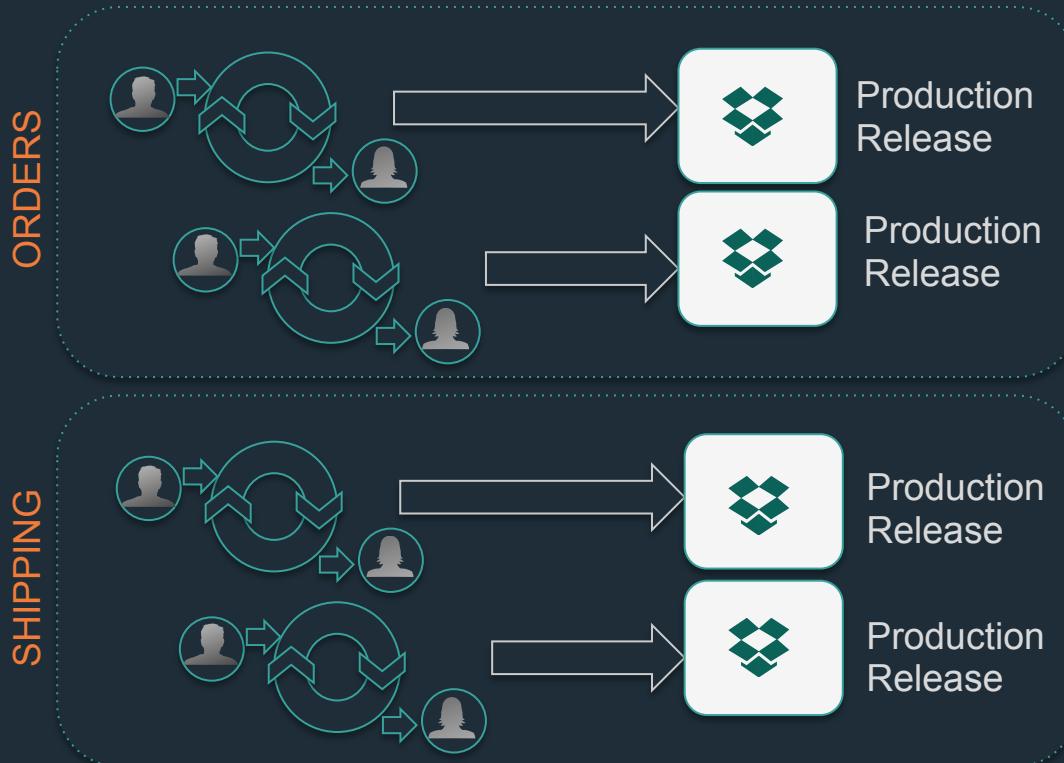
Enable Continuous Delivery



<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



# Characteristics of Cloud Native Architectures

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

# It Takes a Platform

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

# Characteristics of Cloud Native Architectures

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

My Queue

Pending All

New

Tickets showing 1 - 4 of 4



## Change Status

Change the ticket status.

NEW

OPEN

AWAITING

IN PROGRESS

CLOSED

 -- Actions --

Categories 1 - 4 SEARCH

#SP00000285

NEW

## Need help with deactivating staff (1)

Hi there, We are new to Happyfox and would appreciate some help with deactivating staff who has just quit...

last updated

9 minutes ago

assigned to  
johnraised by  
Blake Callaghanpriority  
VERY HIGHcategory  
Supportdue date:  
Not Set

#SP00000290

NEW

## Do not find Submit button in ticket details (1)

Hi, We do not find the submit button when creating ticket or updating the comments. Please help.

last updated

17 minutes ago

assigned to  
jamesraised by  
Melissa Hudgenspriority  
VERY HIGHcategory  
Supportdue date:  
Not Set

#SP00000292

NEW

## Canned Response - Provide us some assistance (1)

Hi support, we are looking for setting up canned response templates. Need your assistance with the stand...

last updated

29 minutes ago

assigned to  
jenniferraised by  
Justin Vensfieldpriority  
Mediumcategory  
Billingdue date:  
Not Set

#SP00000293

NEW

## How to we pull data from High Rise CRM? (1)

Hello Support, We would like to have the capability to pull customer information from HighRise CRM. Is it...

last updated

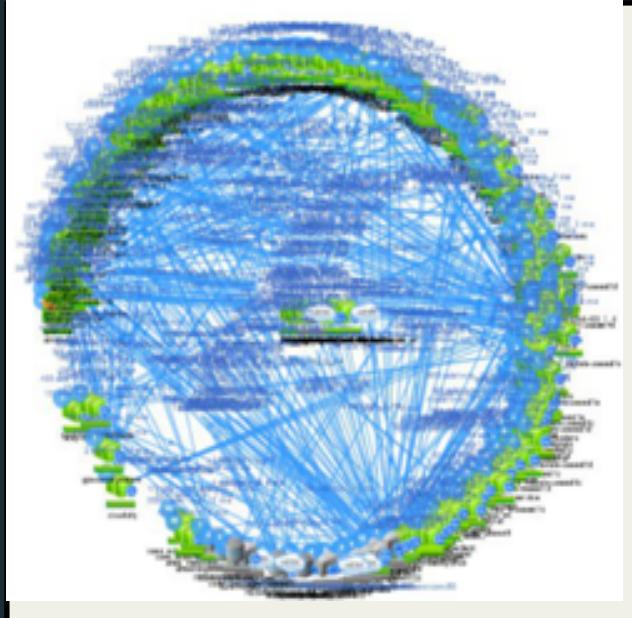
30 minutes ago

assigned to  
scottraised by  
Bryan Normanpriority  
Mediumcategory  
General Inquirydue date:  
Not Set

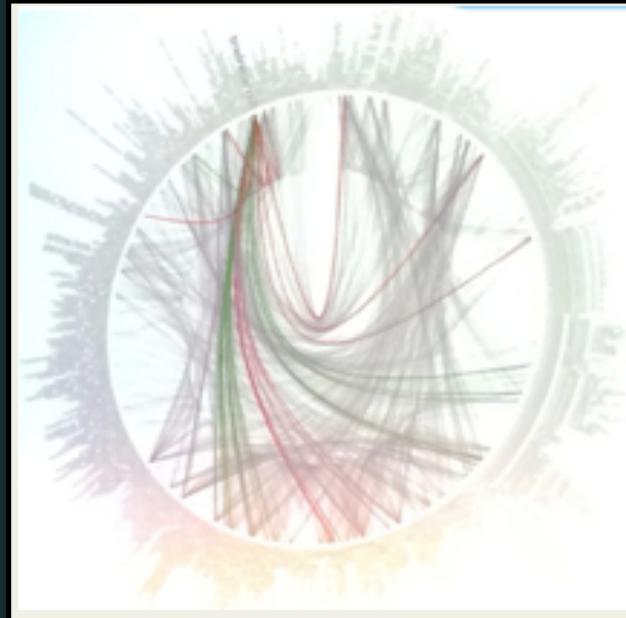
# Characteristics of Cloud Native Architectures

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

# However... Microservices are Hard



NETFLIX



twitter

# Challenges of Distributed Systems

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

# HOW

# Cloud Native Platform and Contracts

## Culture



## Framework

Application Framework

Contract: 12 Factor App

Runtime Platform

Contract: BOSH Release

Infrastructure Automation

Contract: Cloud Provider Interface

Infrastructure

## Tools



Spring Cloud



Spring Boot



Pivotal  
Cloud Foundry



BOSH



Pivotal



Dev

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



NETFLIX

OSS



Config Server



Service Registry



Circuit Breaker

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

# DEMO

---



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



“

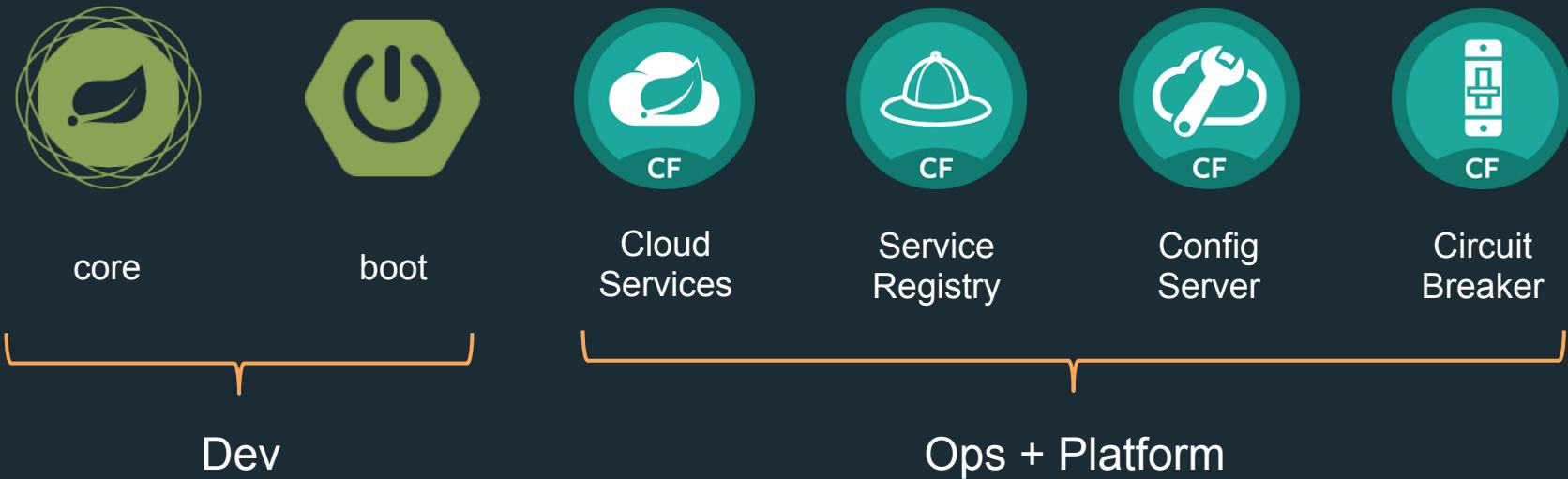
“Velocity on the JVM is the Killer App”

- 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.

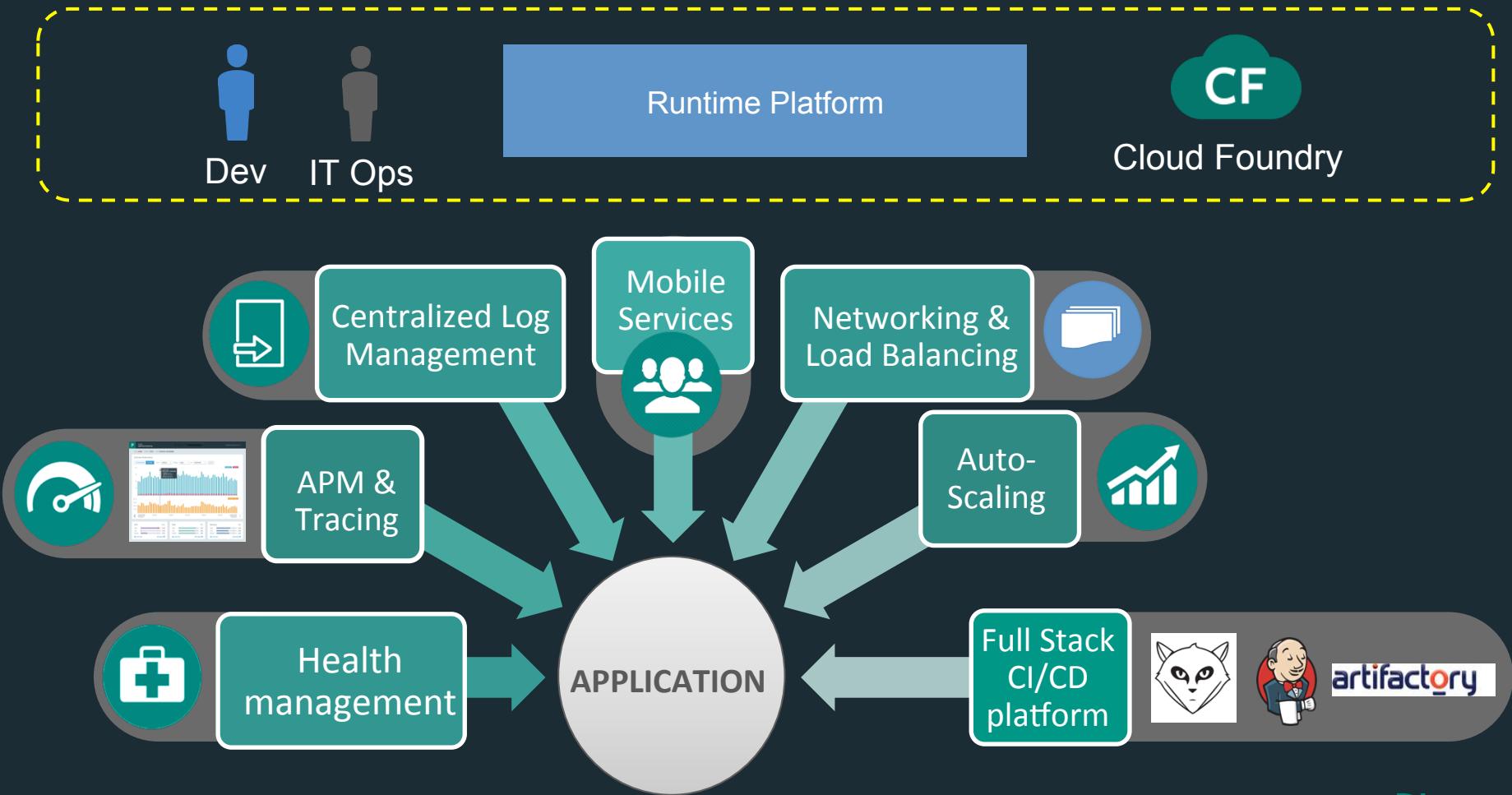


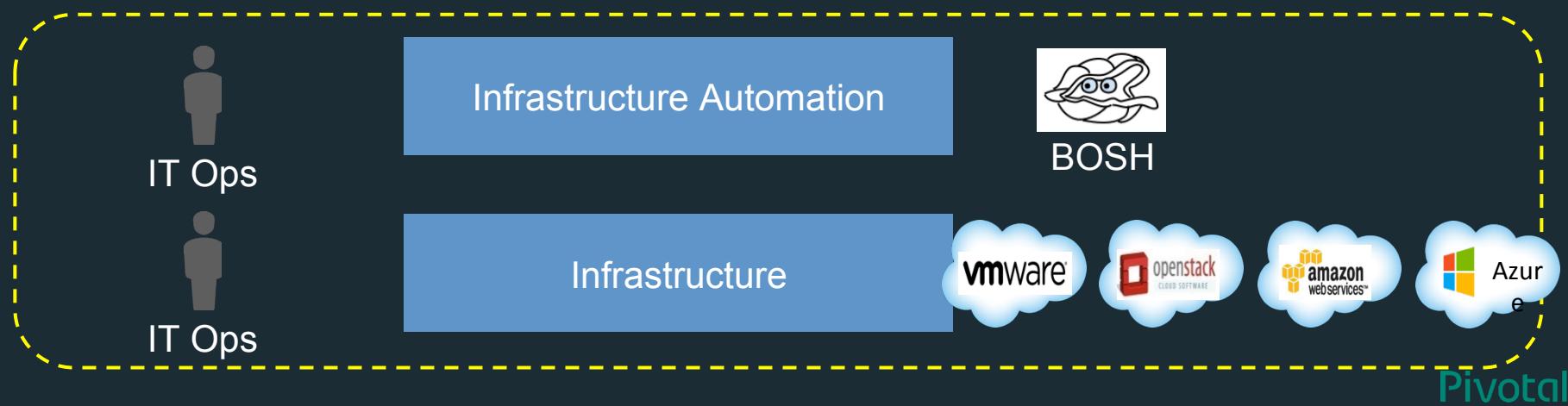
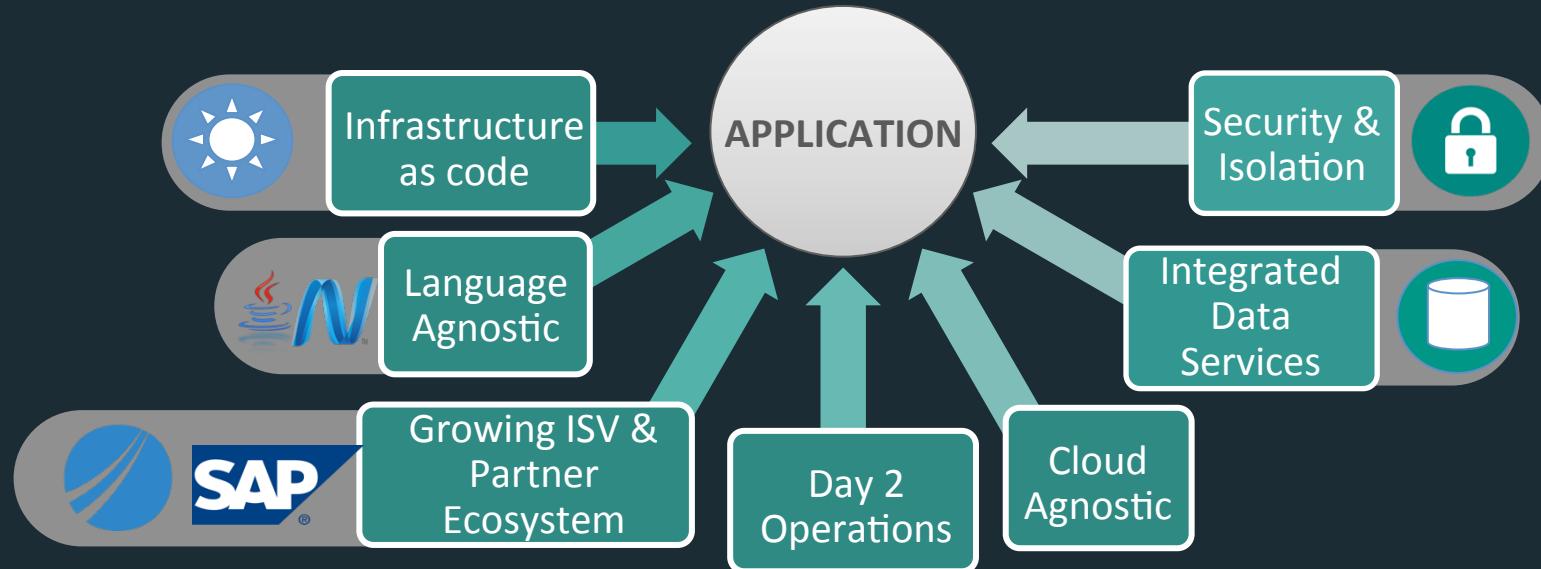
# DEMO

---

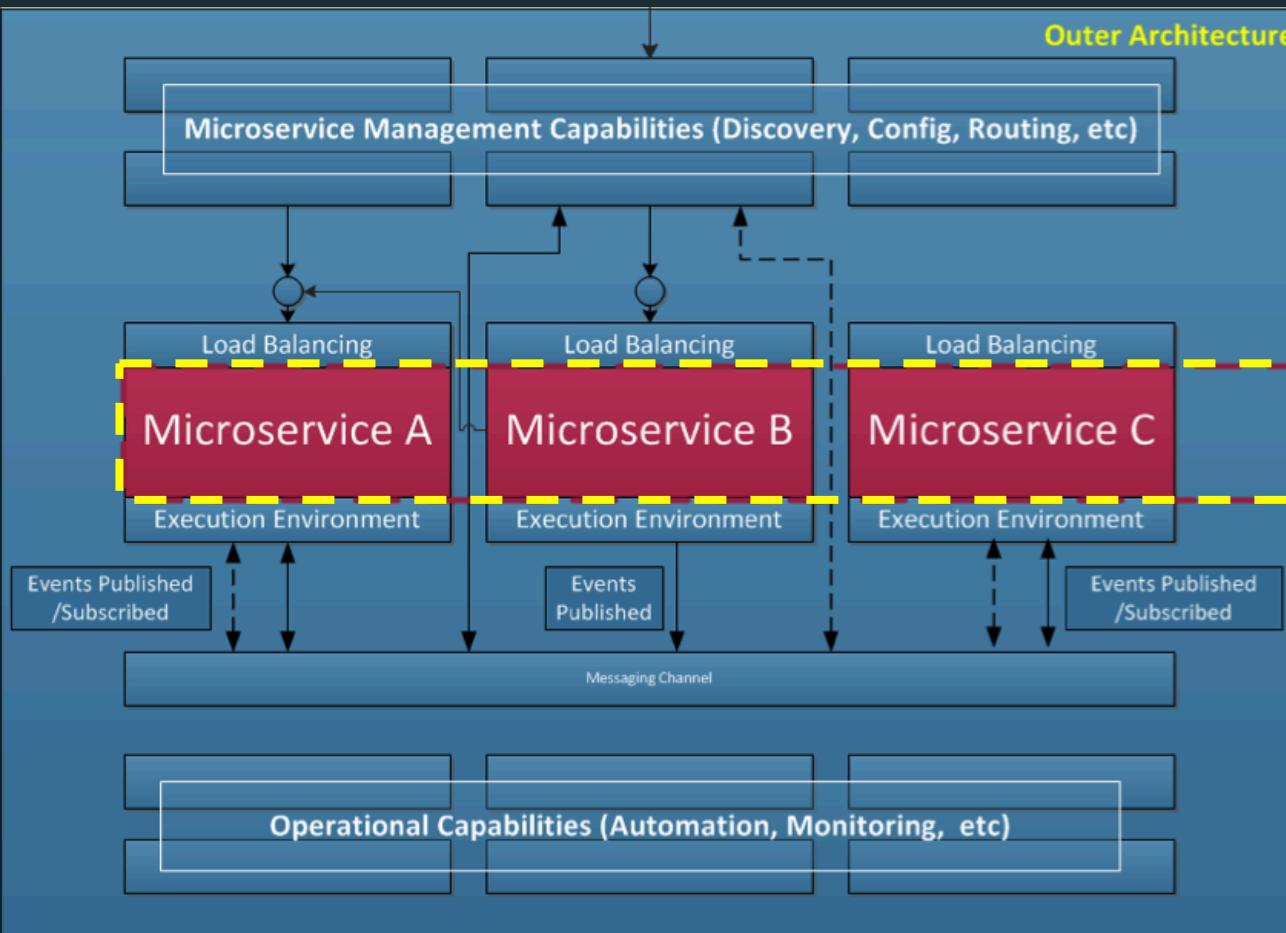


CF





# Cloud Native Platforms Provide the Outer Architecture

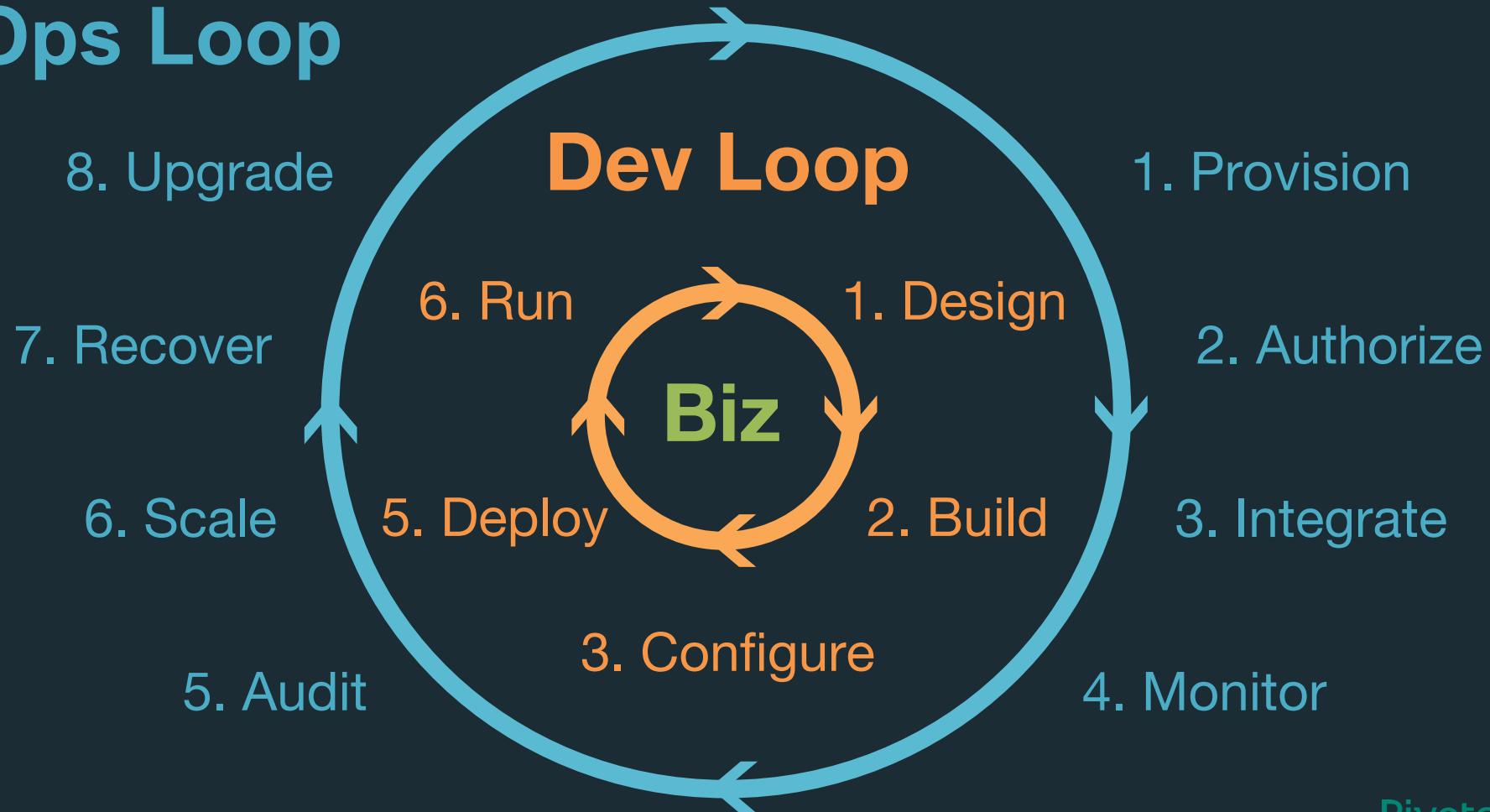


"The hardest part of building microservices is in the outer architecture (All of the blue boxed components)."

Inner  
Architecture

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

# 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  
Tool Chain



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



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



Share binaries and manage distributions. Manage artifact lifecycle.

# Pivotal™

Develop, Test, QA and Production on the same platform. Horizontal scaling, high availability, security, logging, update management 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

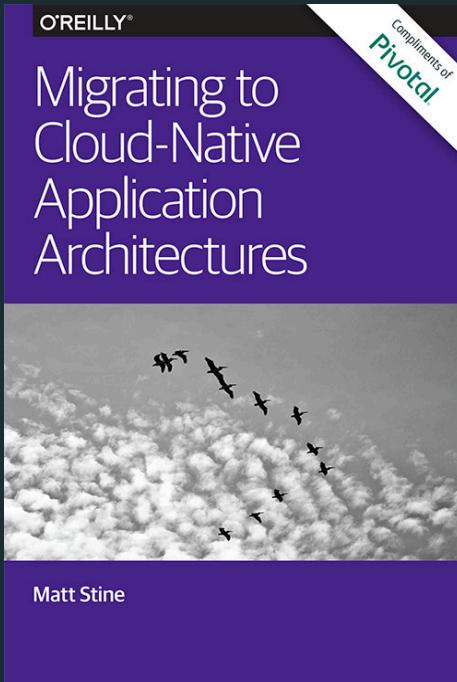
## Cloud Ready

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

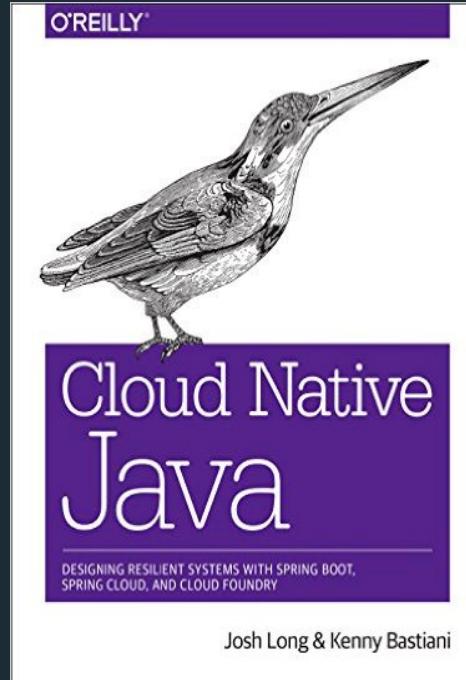
# HOW

## 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  clear

Artifact

**Dependencies**

Add Spring Boot Starters and dependencies to your application

Search for dependencies

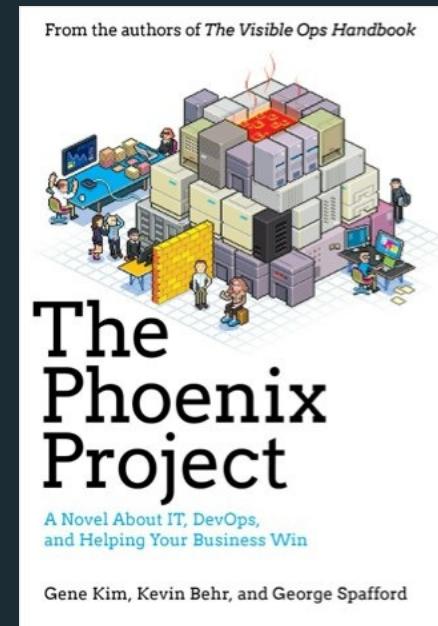
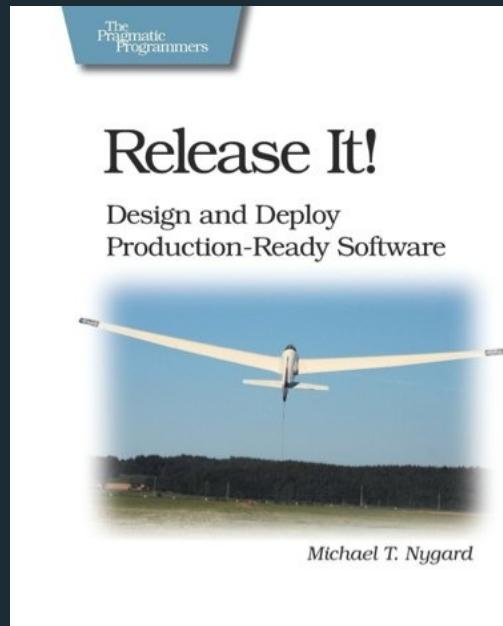
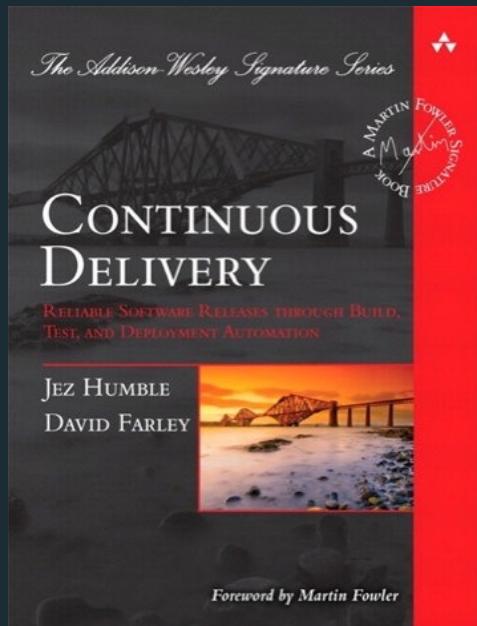
Selected Starters

**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



# Foundations

