
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

Pivotal

Disruption to Enterprise IT is already here

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."

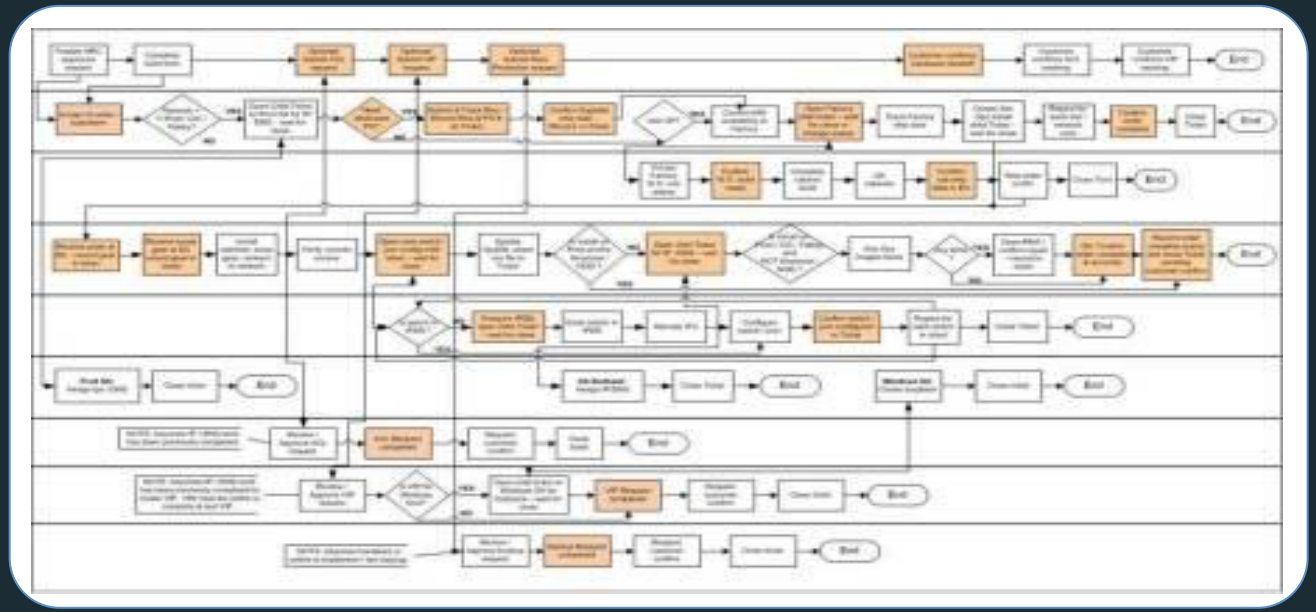
WALL STREET JOURNAL

Pivotal

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?

Pivotal

The Promise of Cloud Native

- Deploy new **features** daily to production
- Loose Coupling (upgrades and fixes)
- Faster Feedback (experiment)
- Self-service, automated middleware
- Free/Cheap horizontal scaling
- No Downtime Upgrades

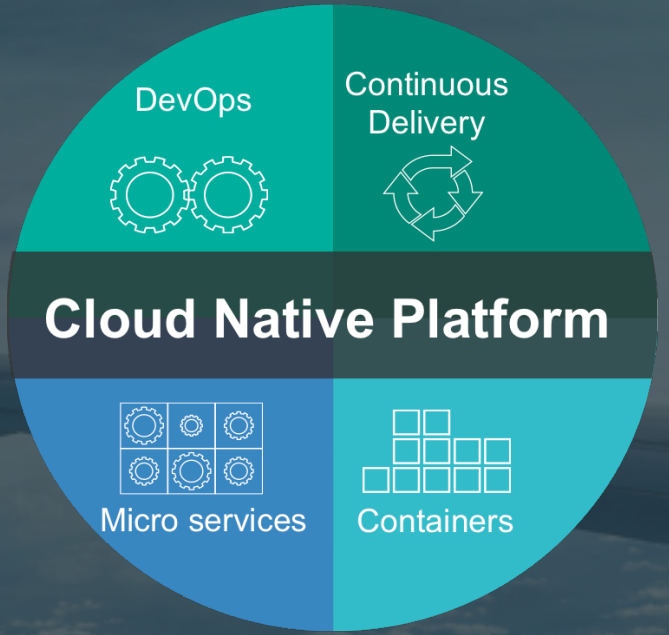
WHAT

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



Pivotal

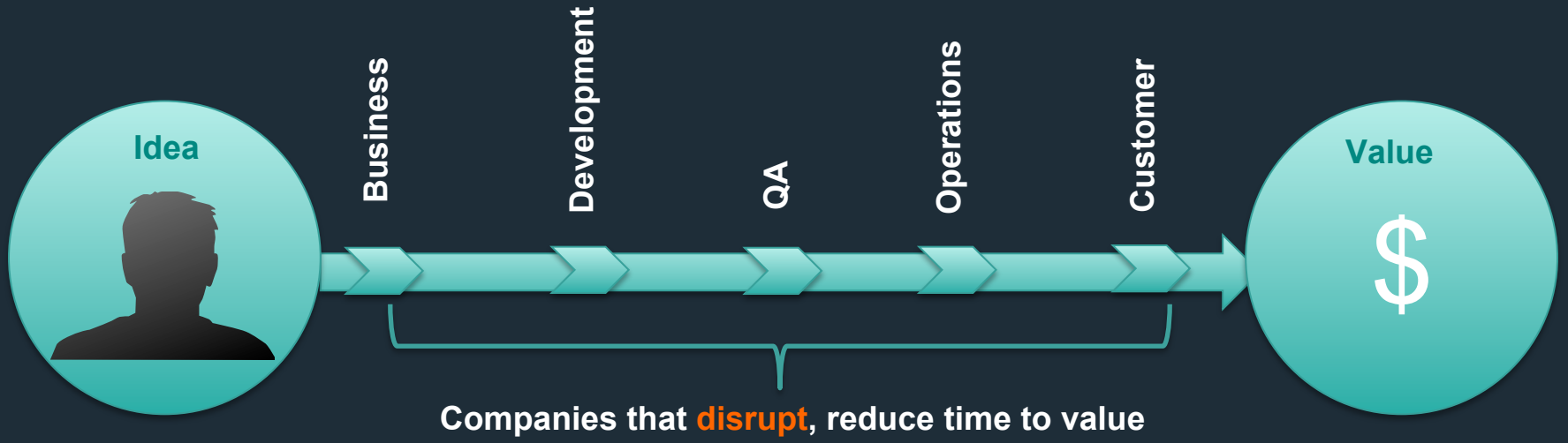
Characteristics of Cloud Native Architectures

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

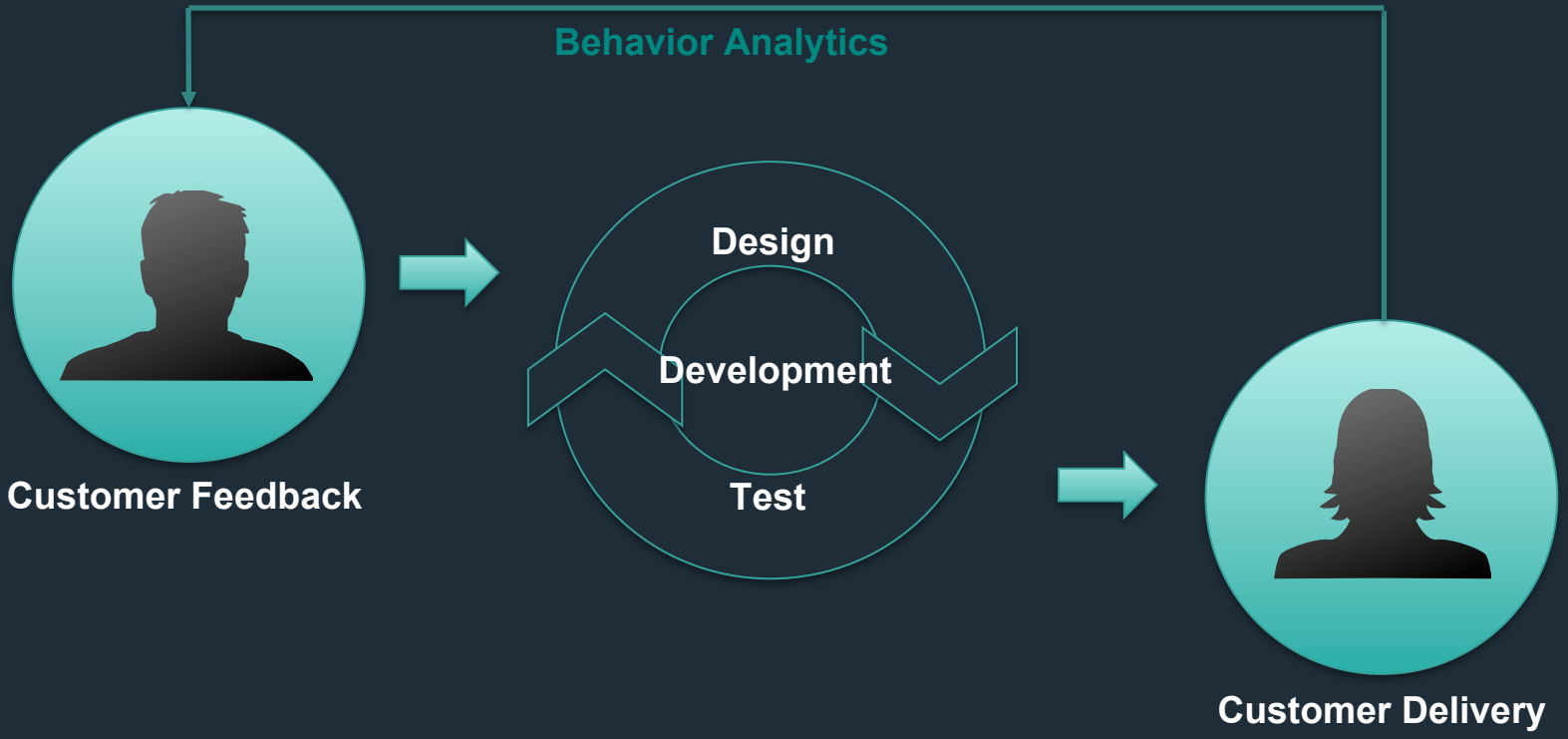
Characteristics of Cloud Native Architectures

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

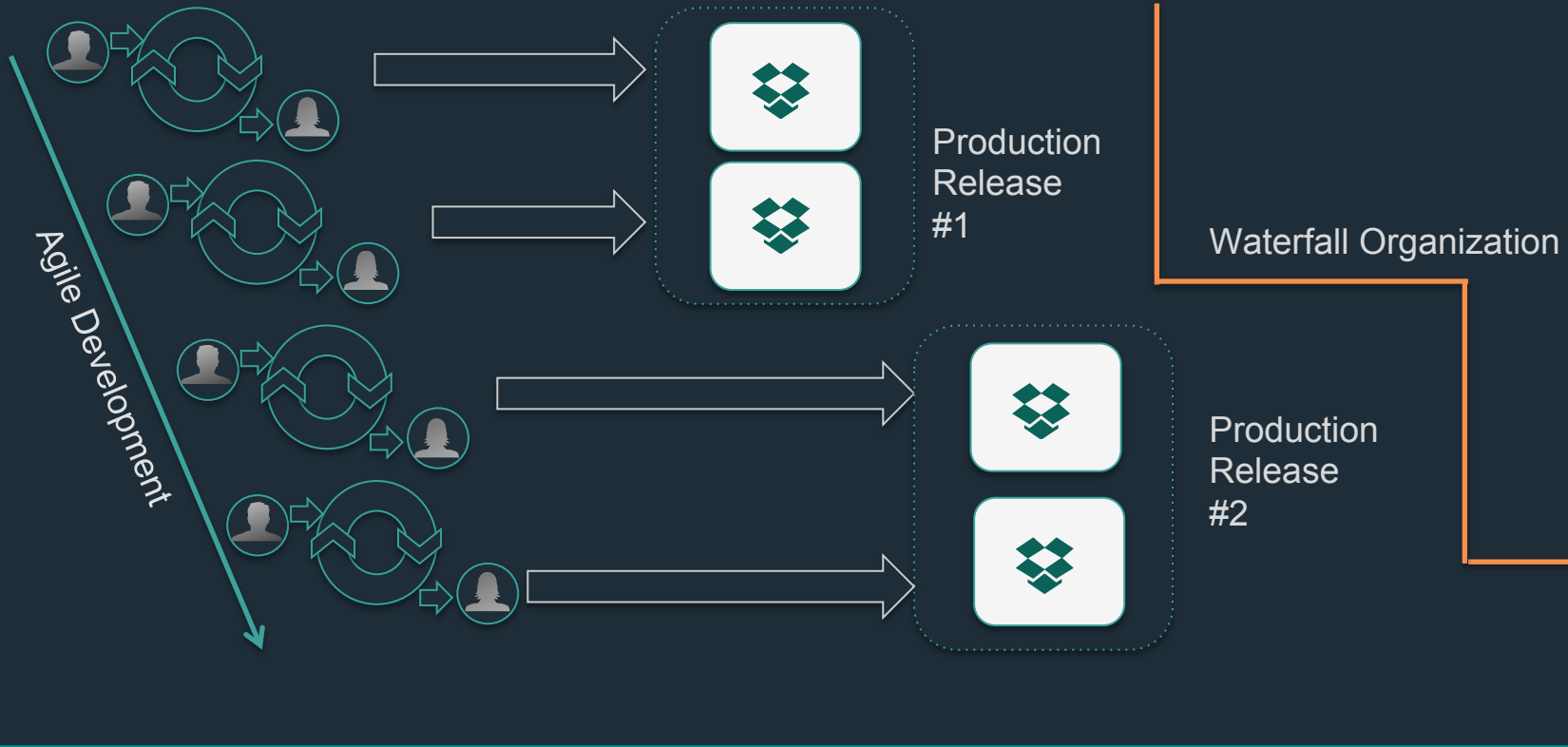
Enabling Continuous Delivery



Enabling Agile Delivery



WaterScrumFall



Application Lifecycle Management: CI/CD

Agile Project Management

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 → **Automate Build & Test** → **Manage Binaries & Build Artifacts** → **Pivotal Cloud Foundry (Elastic Runtime)**

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™

Characteristics of Cloud Native Architectures

- Continuously Delivered
- **DevOps**
- Twelve Factor Apps (http://12factor.net/)
- Microservices
- 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**.

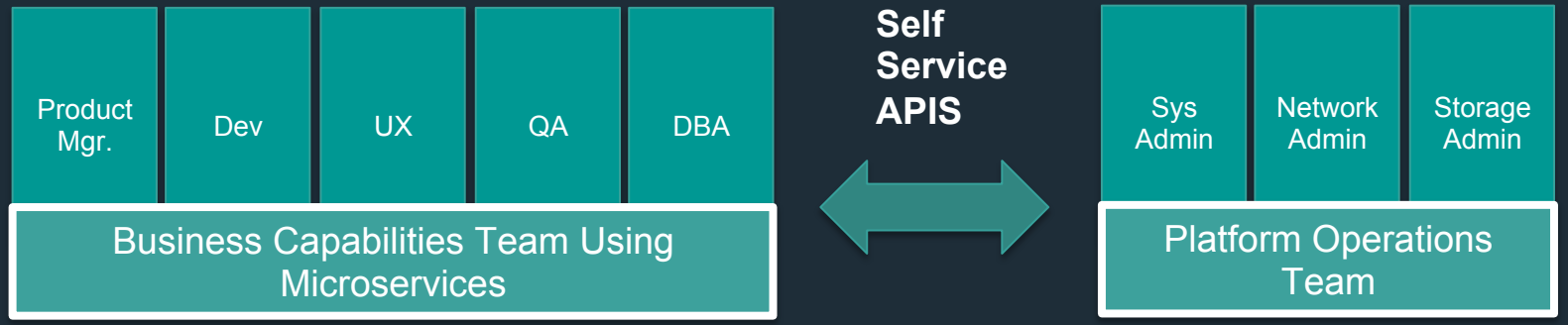
Conway's Law

"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

http://www.melconway.com/Home/Committees_Paper.html

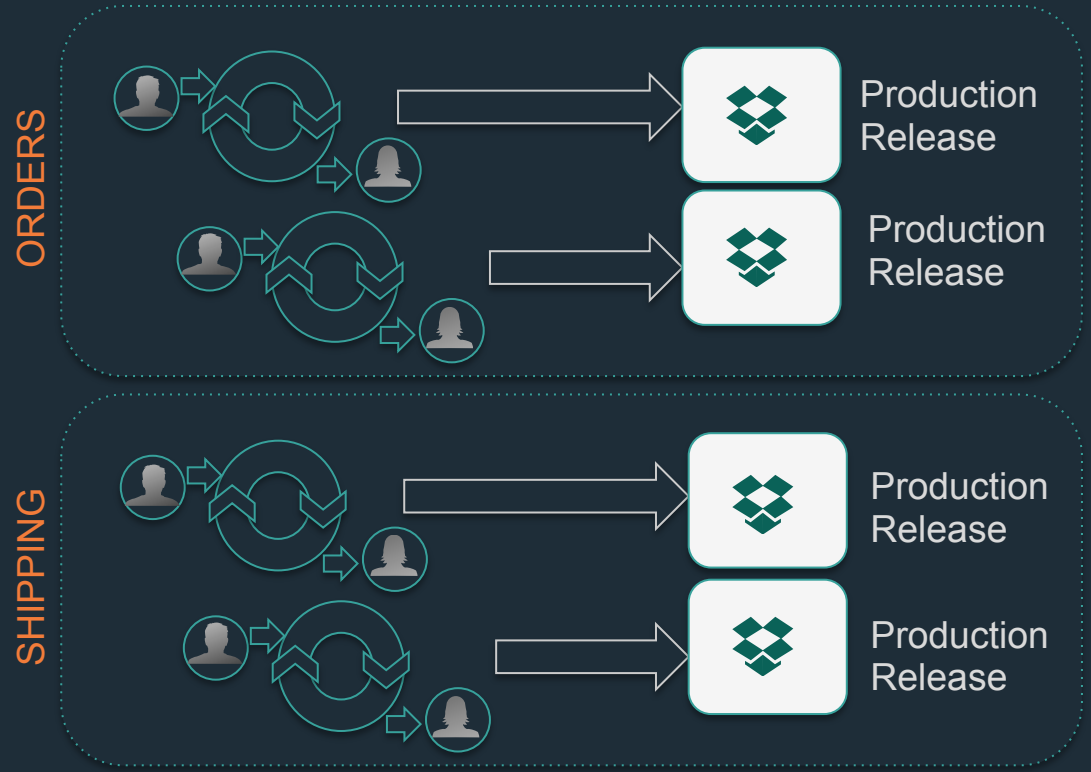
Inverse Conway Maneuver
Enable Continuous Delivery



http://jonnyleroy.com/2011/02/03/dealing-with-creaky-legacy-platforms/
http://www.slideshare.net/adriancockcroft/microservices-the-good-bad-and-the-ugly

Pivotal

Enabling Continuous Delivery



Pivotal

Characteristics of Cloud Native Architectures

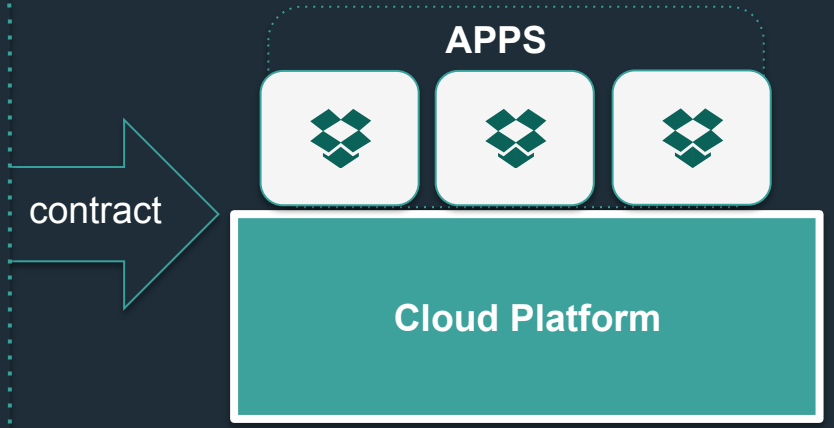
- Continuously Delivered
- DevOps
- **Twelve Factor Apps (http://12factor.net/)**
- Microservices
- Self-Service agile infrastructure
- API-based collaboration
- Anti-fragility

Pivotal

21

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

Pivotal

Characteristics of Cloud Native Architectures

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

What are Microservices?

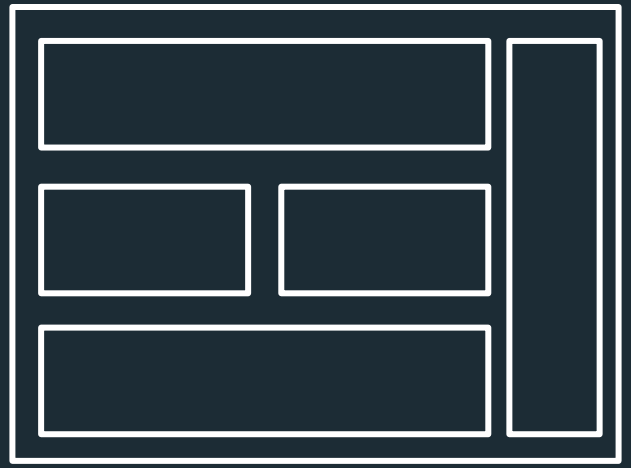
If every service has to be updated in concert,
it's not loosely coupled!

Loosely coupled service oriented architecture with bounded contexts

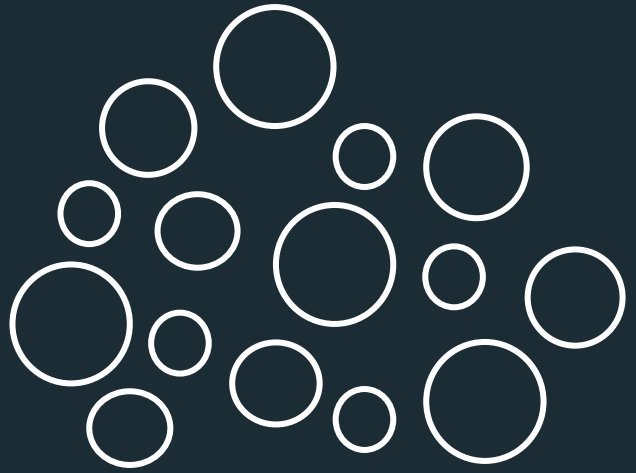
If you have to know about surrounding services you don't have a bounded context.

Trend towards new lightweight architectures

Microservices addressing speed to market and cloud scale

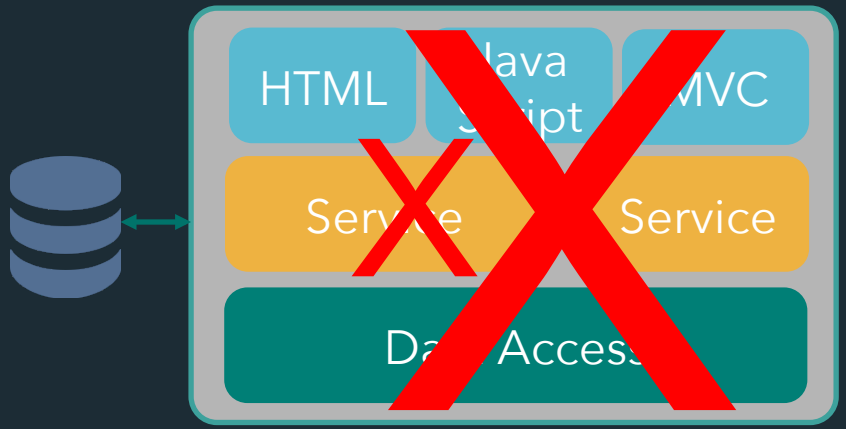


MONOLITHIC/LAYERED



MICROSERVICES

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

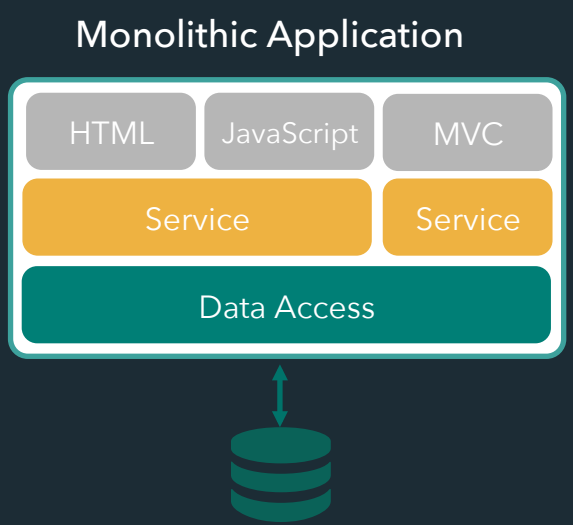


Monolithic Application

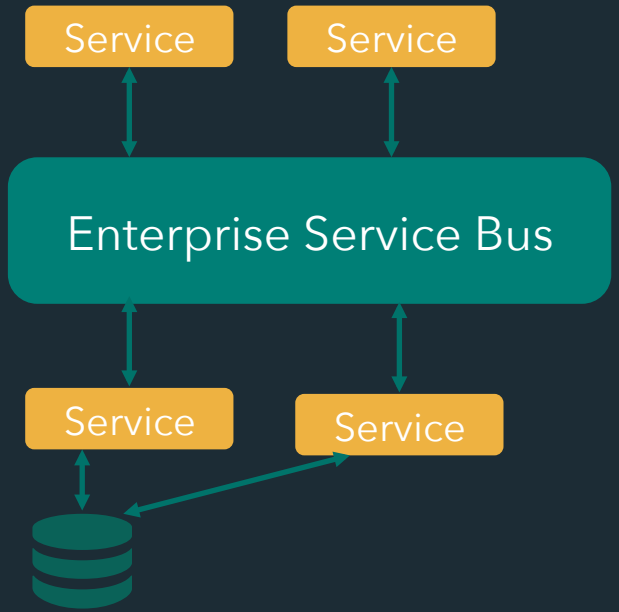
> 2 Pizzas Per Project = Too Many Pizzas

Pivotal

Microservices are NOT



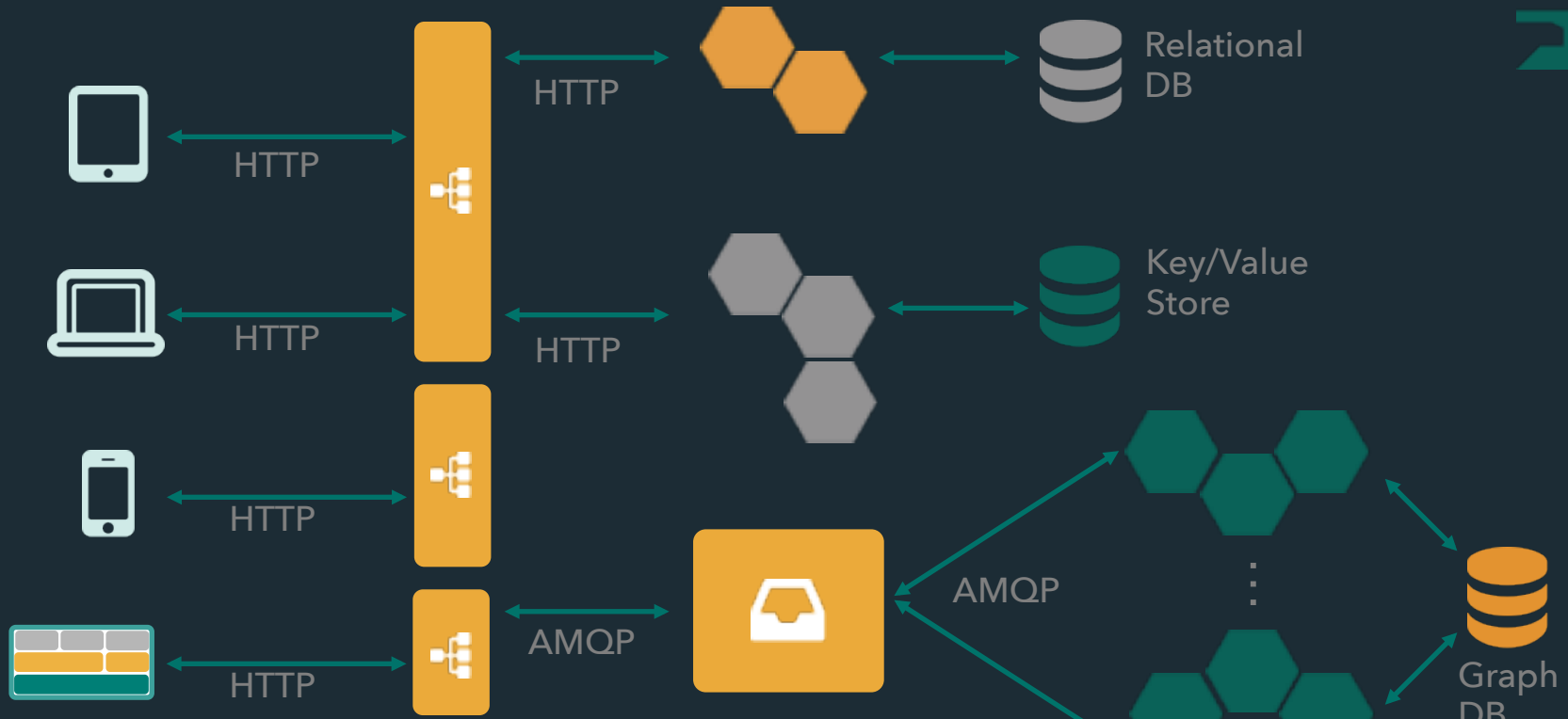
OR



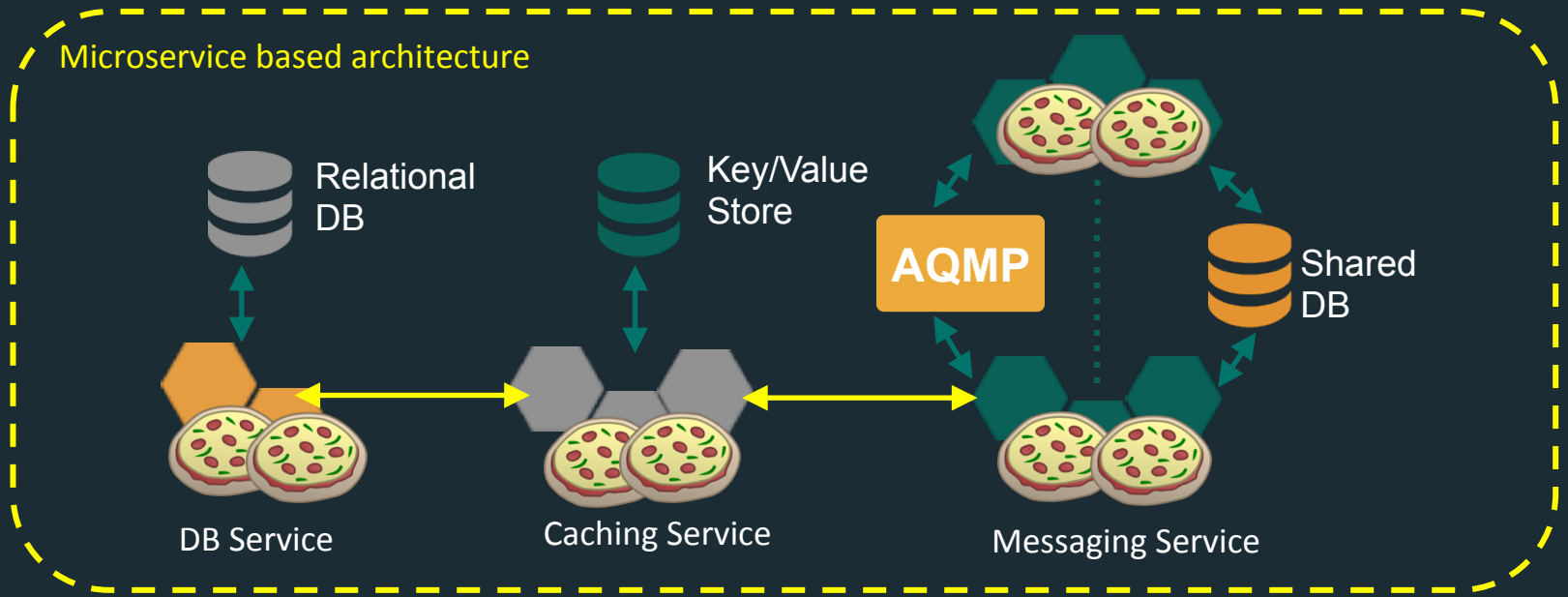
Tightly Coupled

Centralized

Microservice Architecture



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



Two Pizzas Per Microservice = Manageable!

Pivotal

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

Pivotal

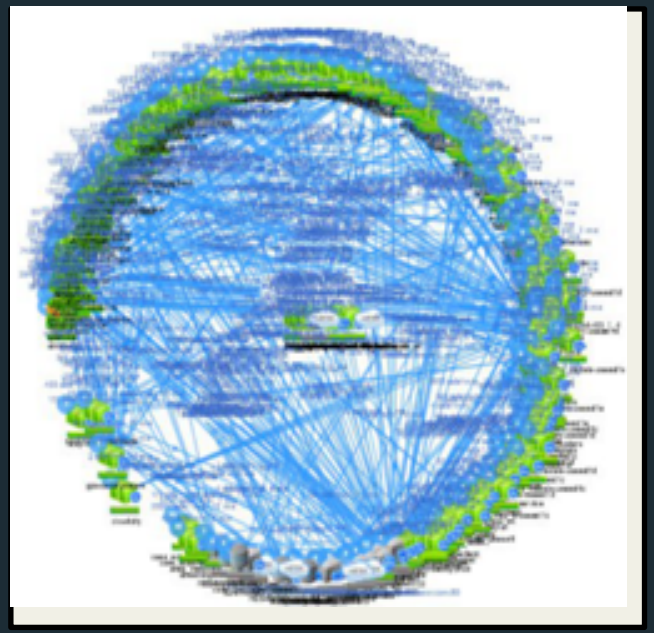
Characteristics of Cloud Native Architectures

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

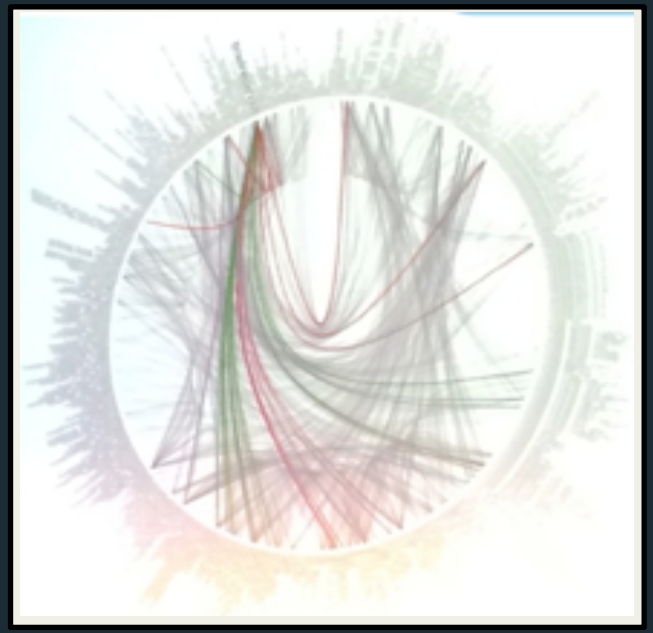
Characteristics of Cloud Native Architectures

- 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

Pivotal

Cloud Native Platform and Contracts

Culture

Dev

Dev

IT Ops

IT Ops

IT Ops

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

vmware

openstack

amazon web services

Azure

Pivotal

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

Pivotal


DEMO
CF

Companies want to be **fast** like Netflix

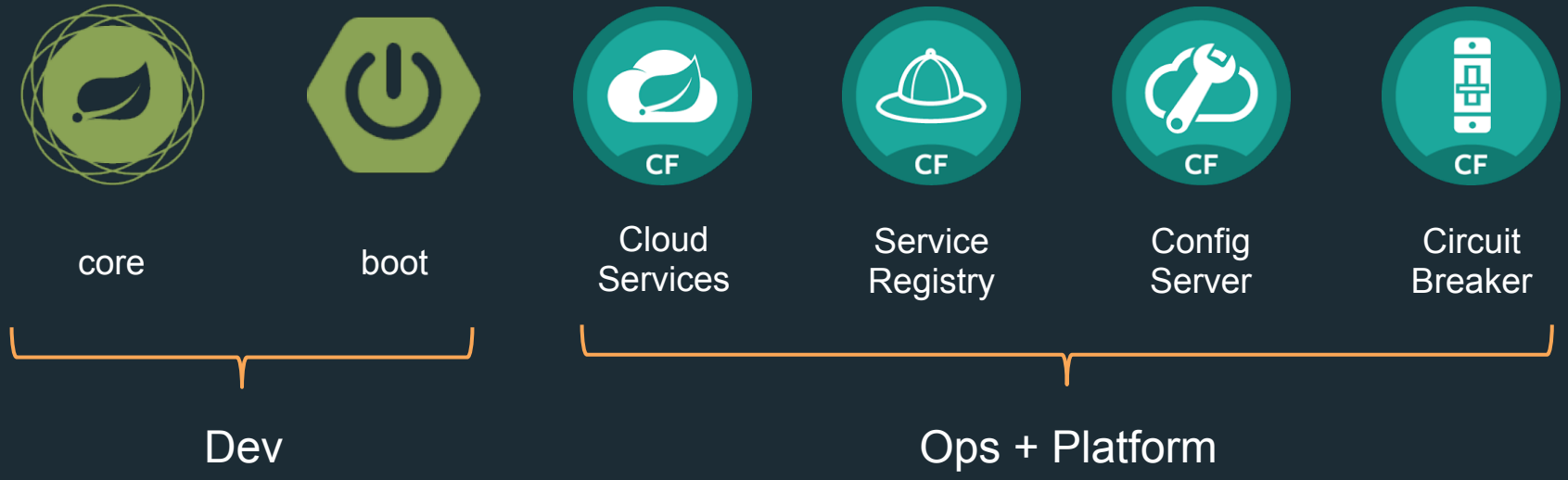
- Netflix needed to be faster to win / disrupt
- Pioneer & vocal proponent of microservices - the key to their speed and success
- Netflix OSS supplies parts, but it's not a solution
- Difficult for enterprises to build it themselves
- Pivotal offers the closest thing to "Netflix in a box" today

> "Velocity on the JVM is the Killer App"
> - Andy Glover (Netflix Eng) @ SpringOne2GX 2014 Keynote
> https://youtu.be/xU267-YHN5c?t=1938

Pivotal

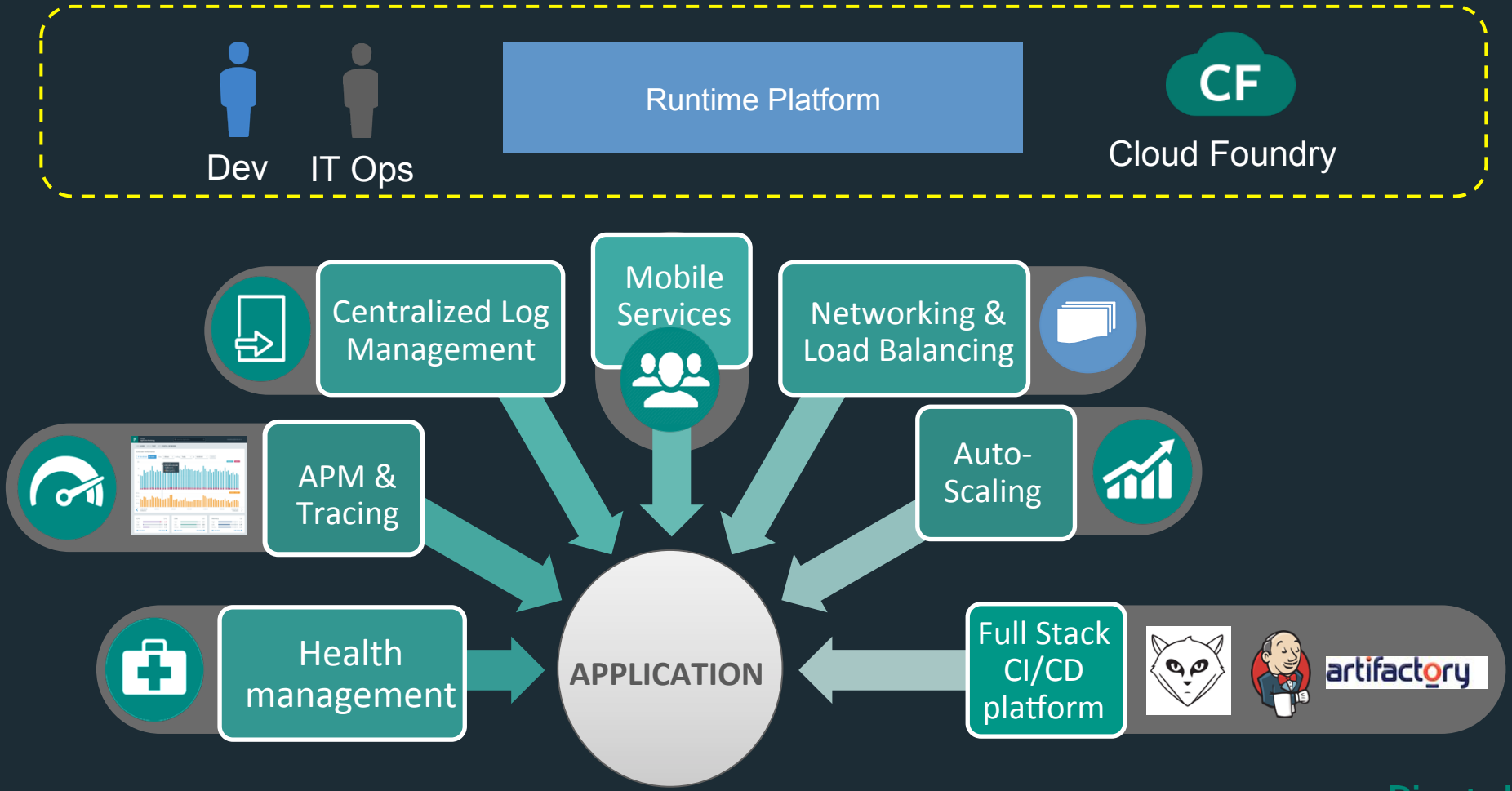
Spring Cloud Services

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

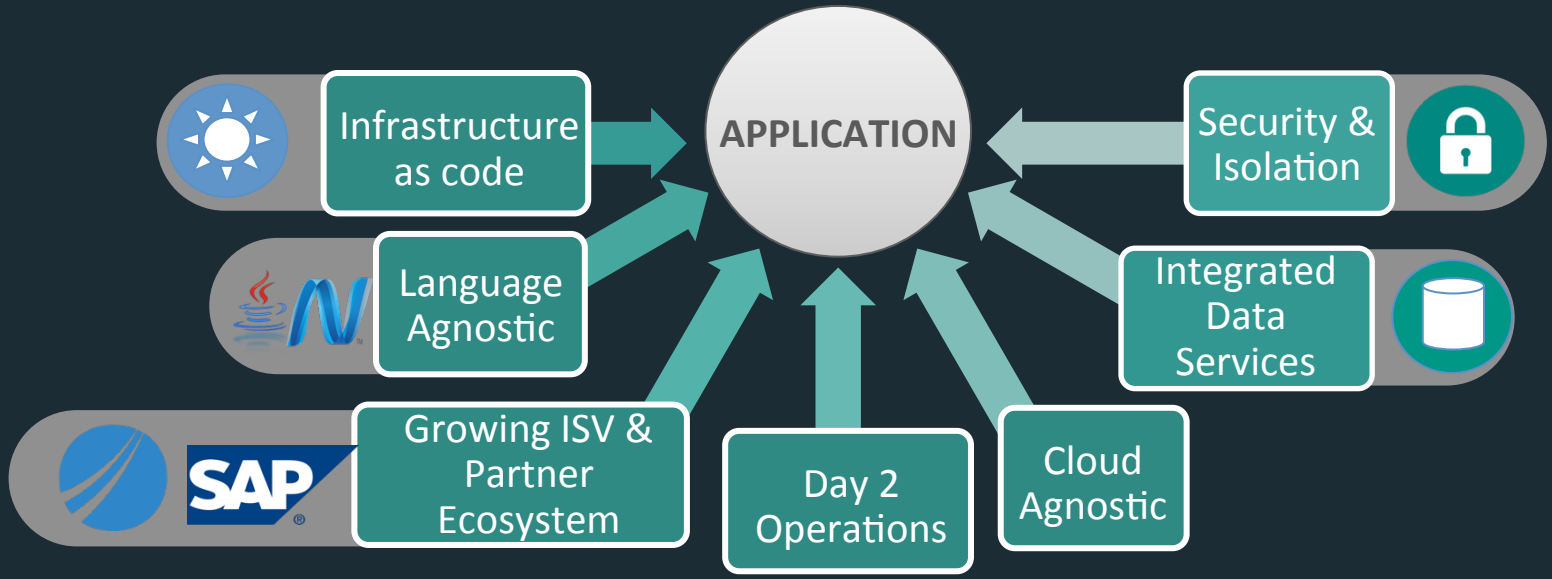


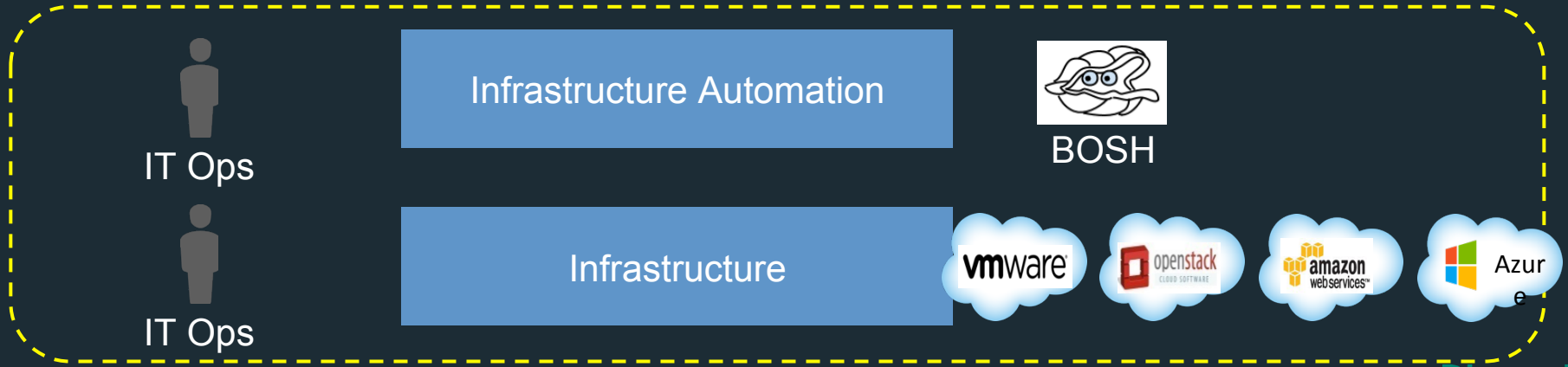
Pivotal


DEMO
CF


Dev
IT Ops
Runtime Platform
CF
Cloud Foundry
Centralized Log Management
Mobile Services
Networking & Load Balancing
APM & Tracing
Auto-Scaling
Health management
APPLICATION
Full Stack CI/CD platform
artifactory

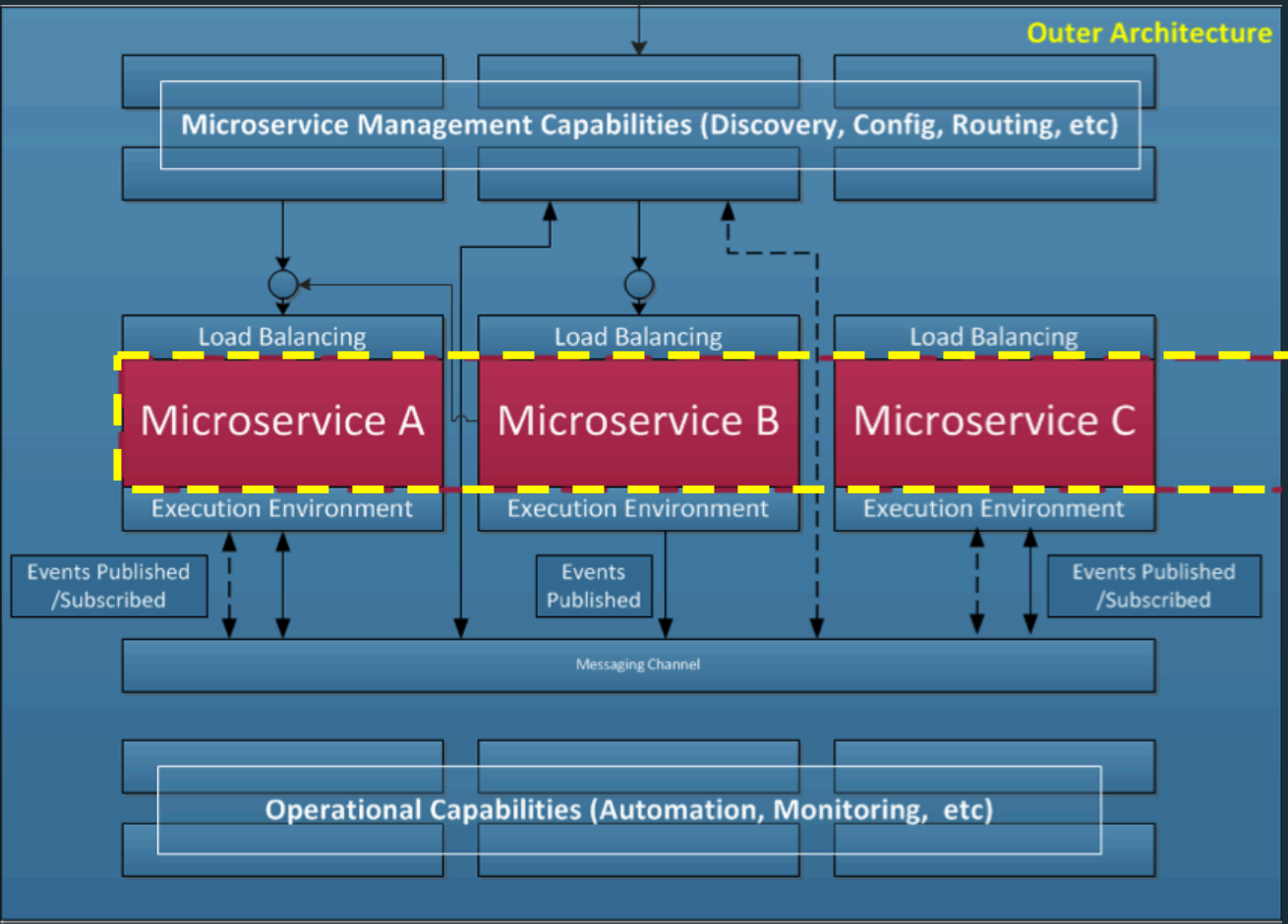
Pivotal


APPLICATION
Infrastructure as code
Language Agnostic
Growing ISV & Partner Ecosystem
Day 2 Operations
Cloud Agnostic
Security & Isolation
Integrated Data Services
SAP


IT Ops
IT Ops
Infrastructure Automation
Infrastructure
BOSH
vmware
openstack
amazon web services
Azur
e

Pivotal

Cloud Native Platforms Provide the Outer Architecture

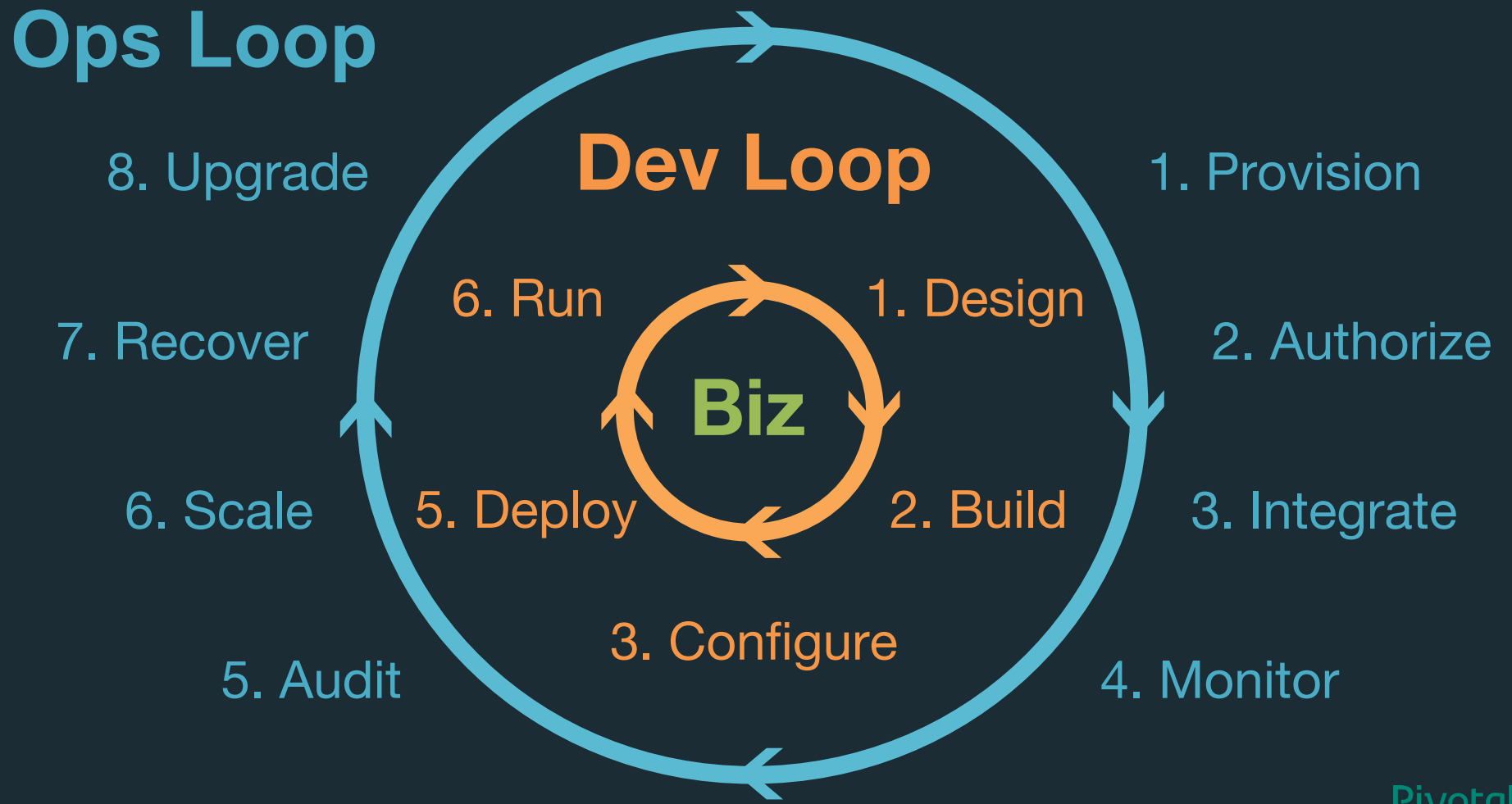


Inner Architecture

"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."

Pivotal


Ops Loop
8. Upgrade
7. Recover
6. Scale
5. Audit
Dev Loop
1. Provision
2. Authorize
3. Integrate
4. Monitor
3. Configure
6. Run
1. Design
5. Deploy
2. Build
Biz
Pivotal

Application Lifecycle Management: CI/CD

Agile Project Management

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 → **Automate Build & Test** → **Manage Binaries & Build Artifacts** → **Pivotal Cloud Foundry (Elastic Runtime)**

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.

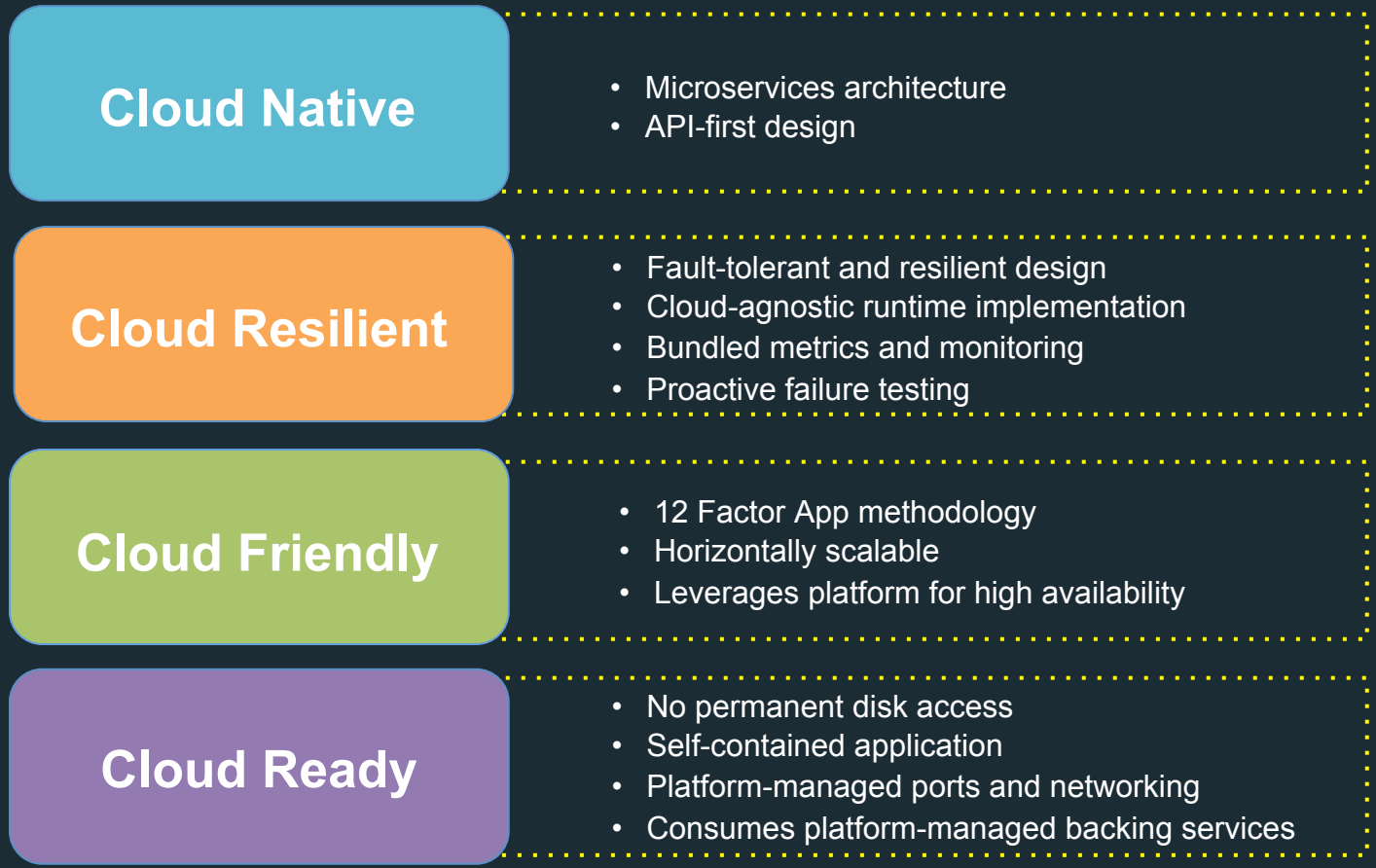
Share binaries and manage distributions. Manage artifact lifecycle.

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



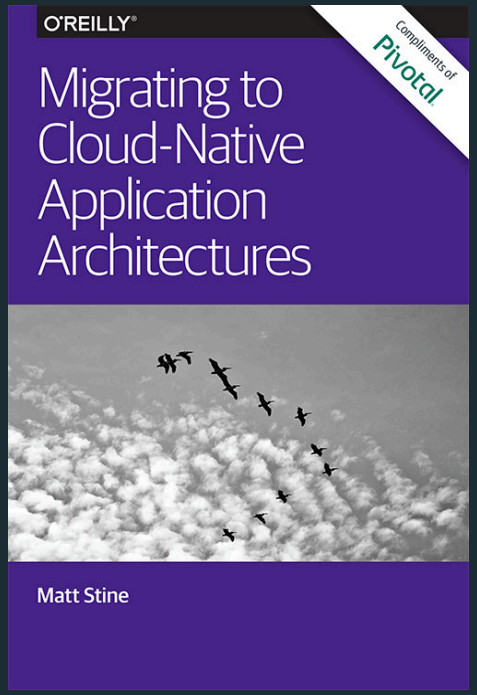
Pivotal

HOW

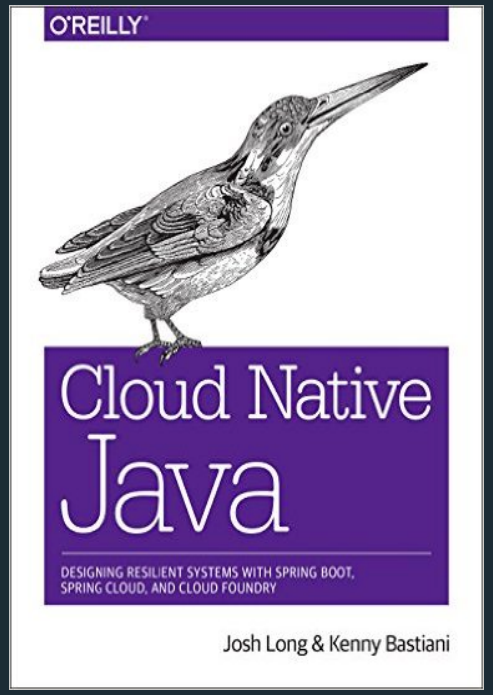
Do we get started

Pivotal

Read

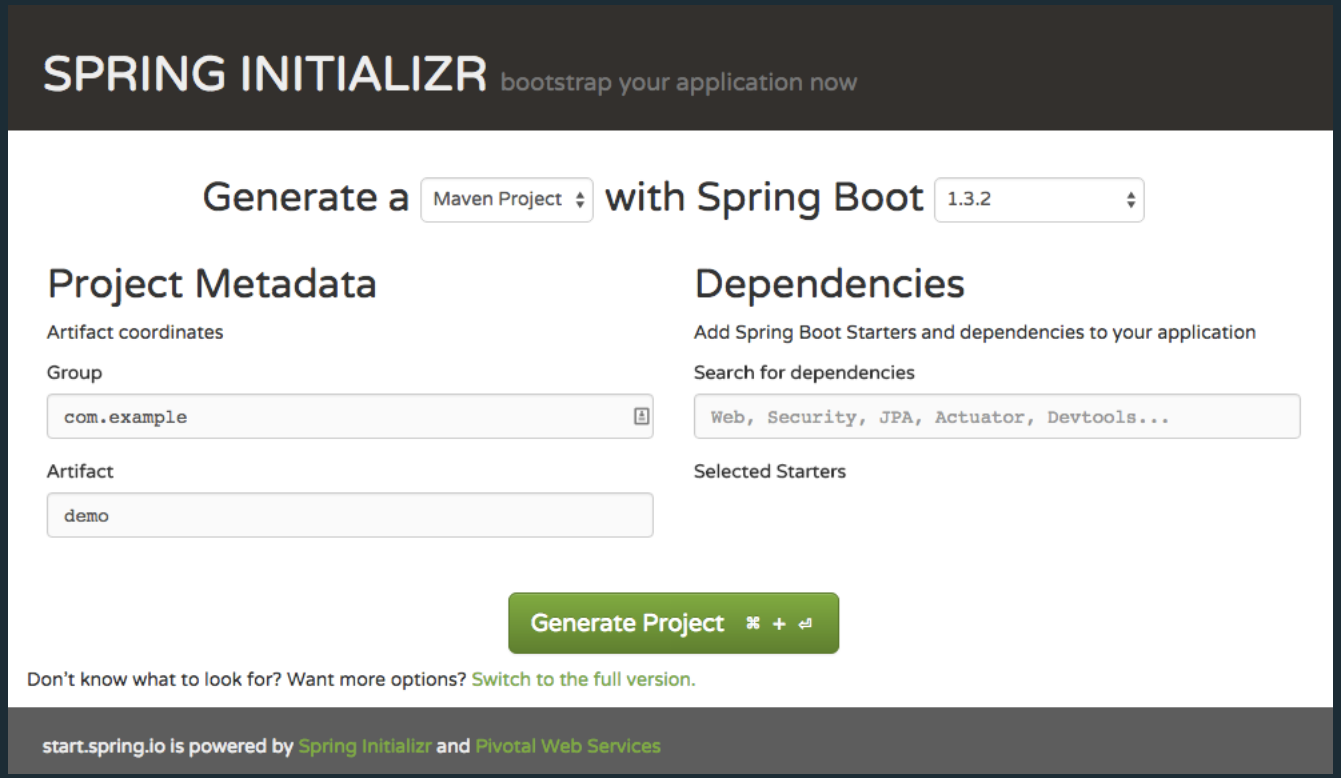


By Matt Stine (@mstine)



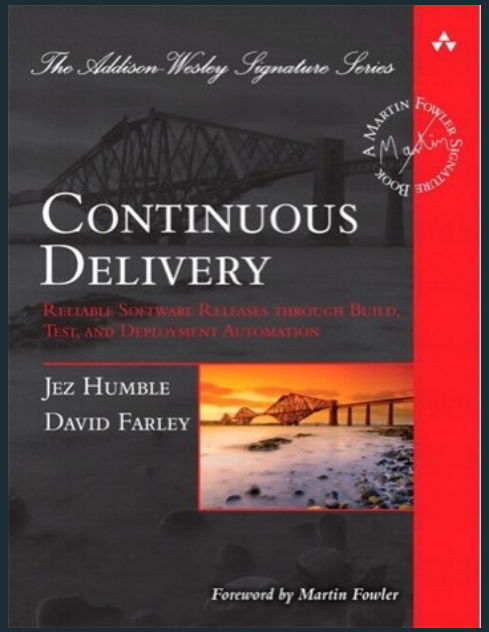
By Josh Long (@starbuxman) and Kenny Bastani

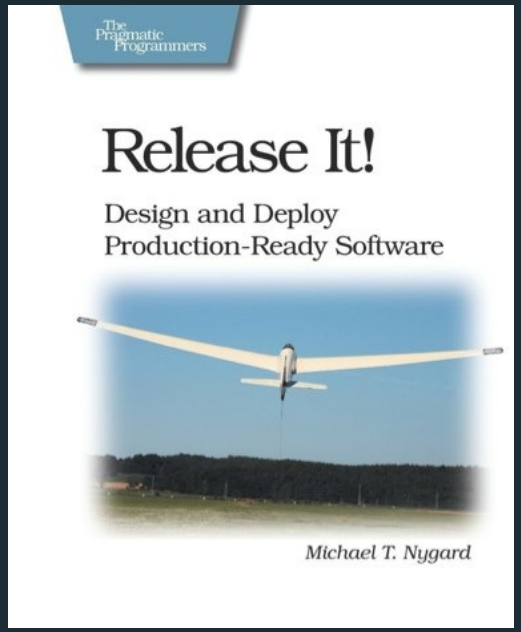
Bootstrap your Application Now: http://start.spring.io

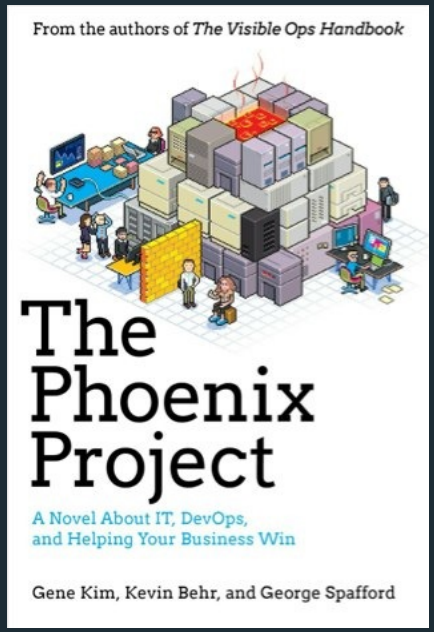


Pivotal

Foundations







Foundations