

Pivotal

Cloud Native Workshop

Cloud Native Architectures and Frameworks

Pivotal Cloud Foundry

Pivotal

Agenda

- Why?
- What?
- How?

WHY?

Software is changing industries

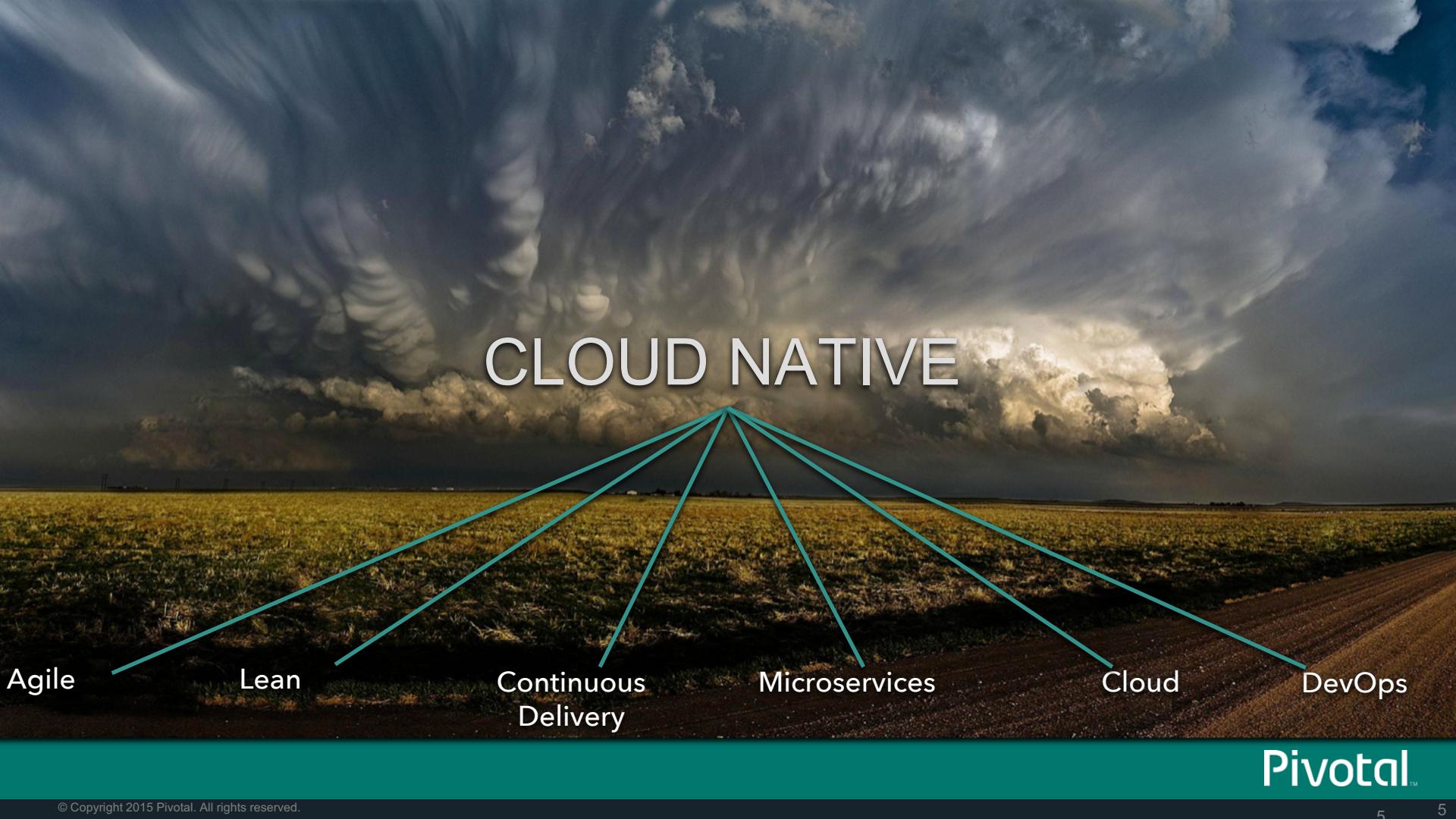


Square

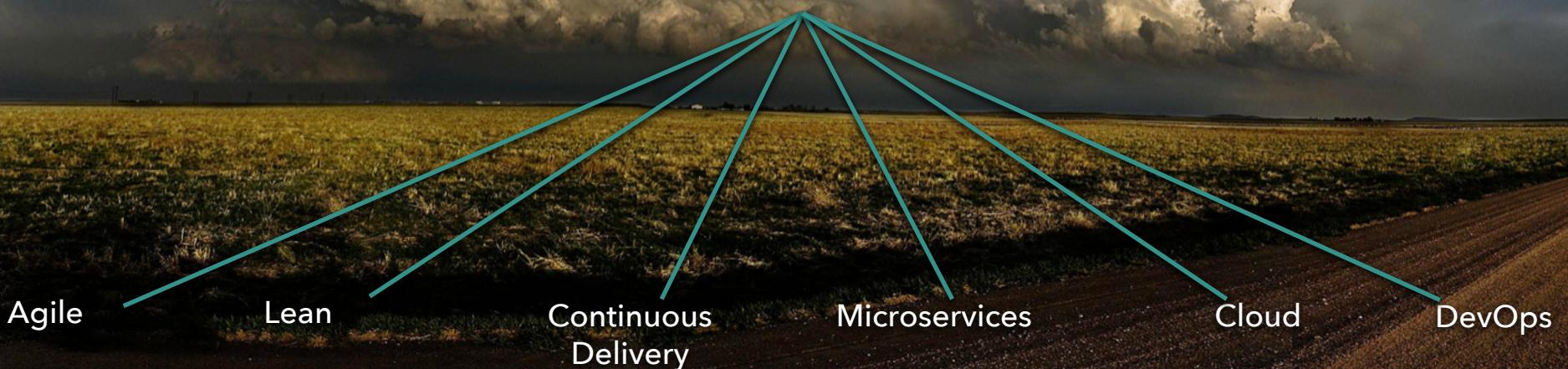


U B E R





CLOUD NATIVE



The promise of Cloud Native

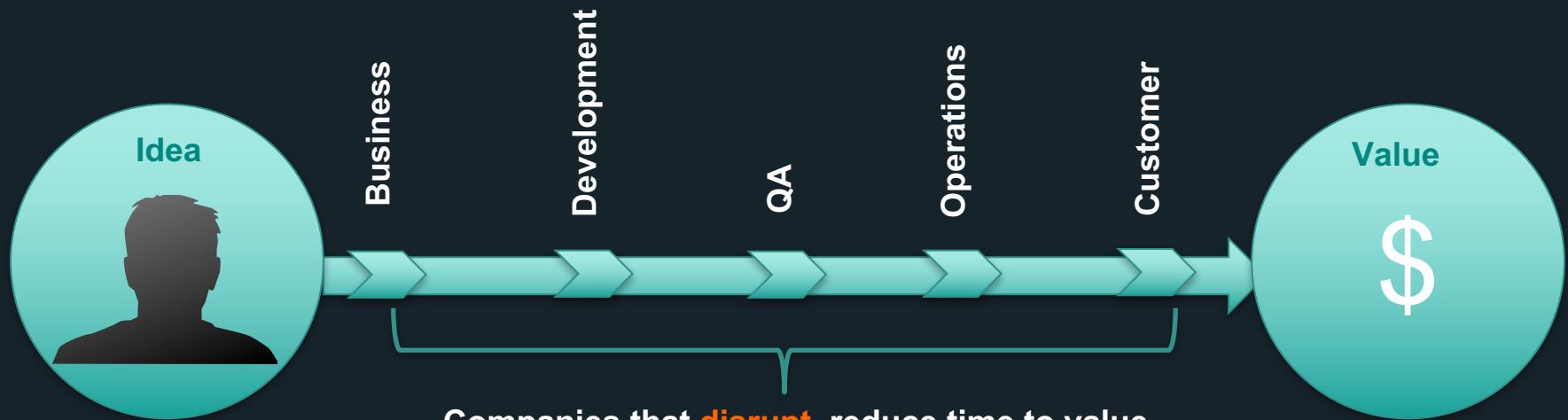
- Deploy new features daily to production
- Automate middleware
- Free/Cheap horizontal scaling
- Contract between App & Platform that creates Promises

WHAT?

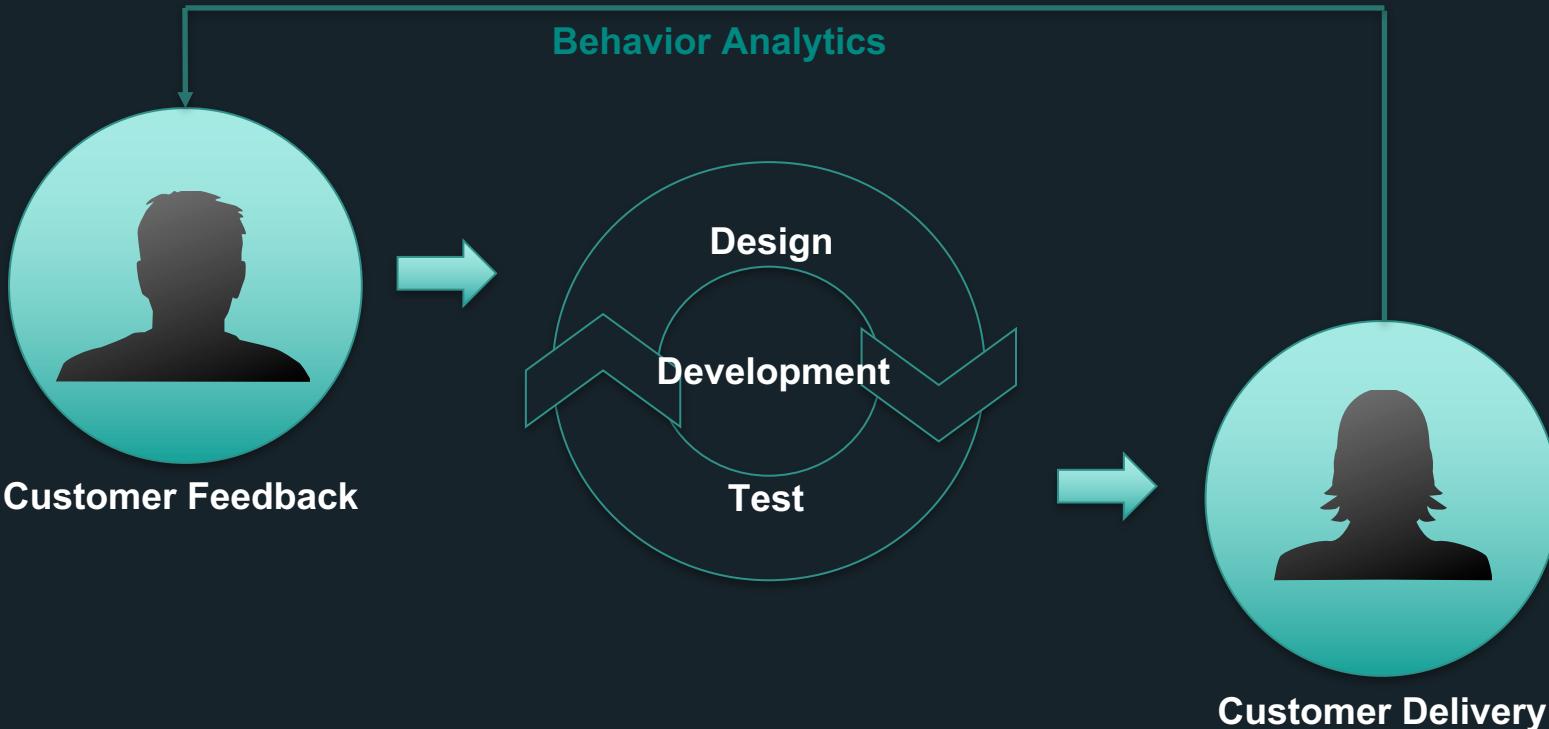
Characteristics of Cloud Native Architectures

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

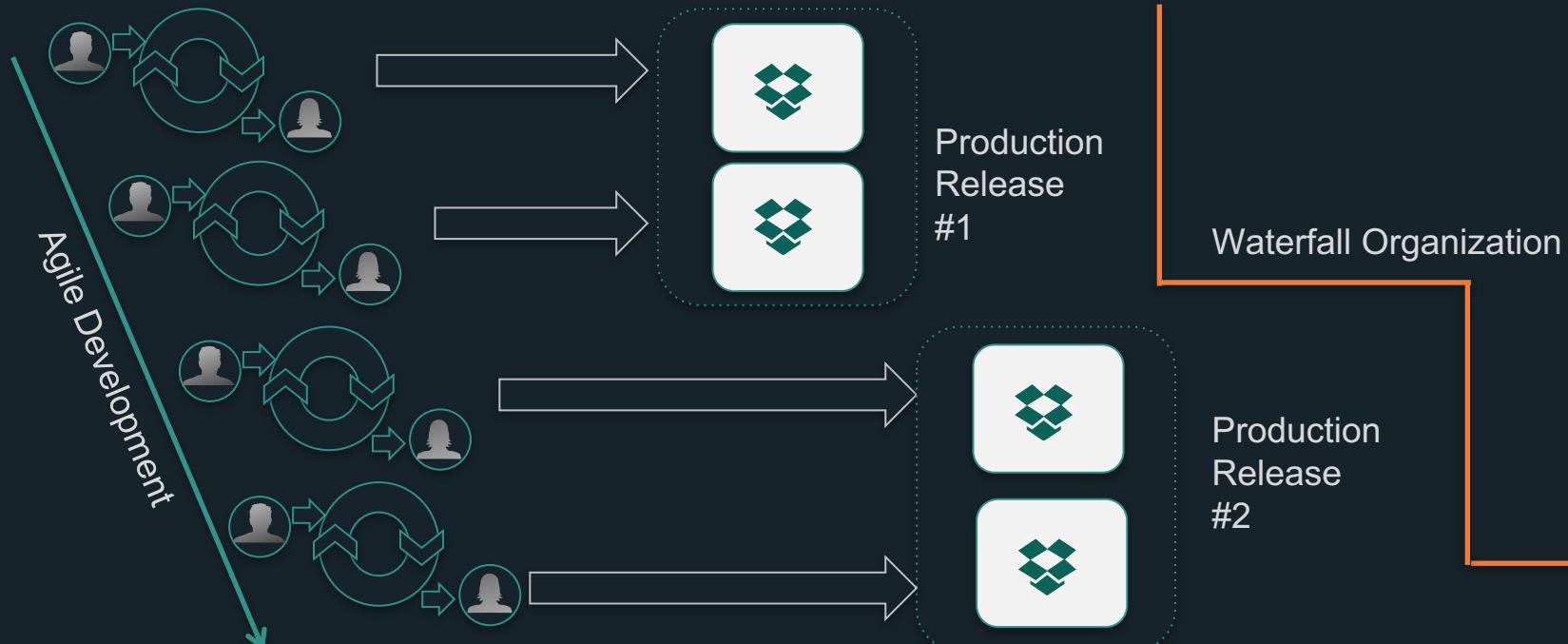
Enabling Continuous Delivery



Enabling Agile Delivery



WaterScrumFall



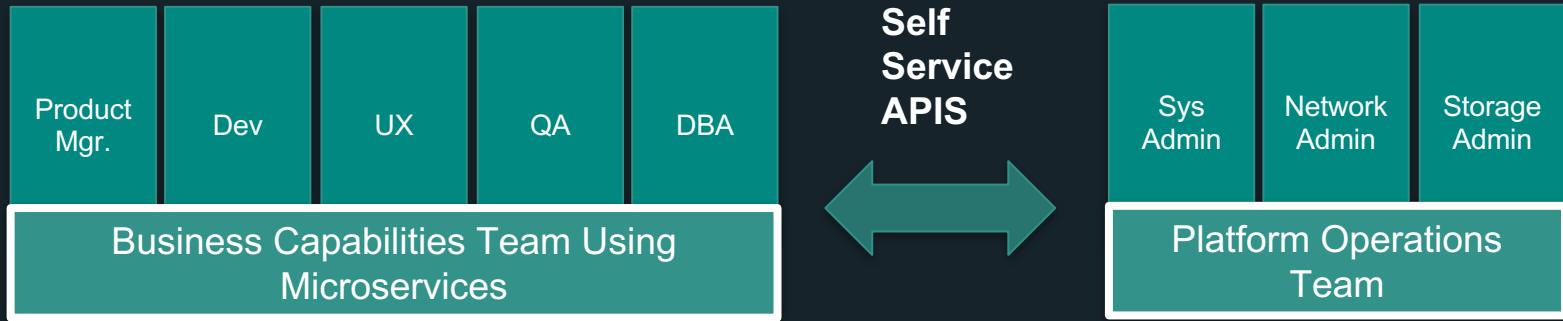
Conway's Law

“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

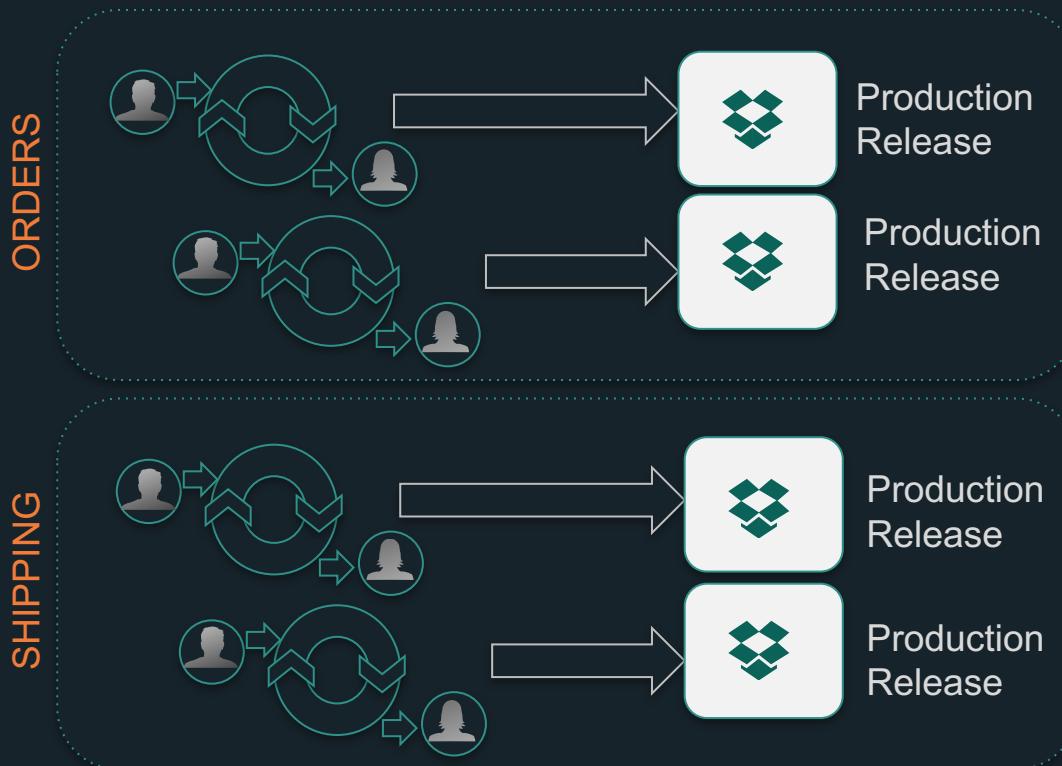
Enable Continuous Delivery



<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

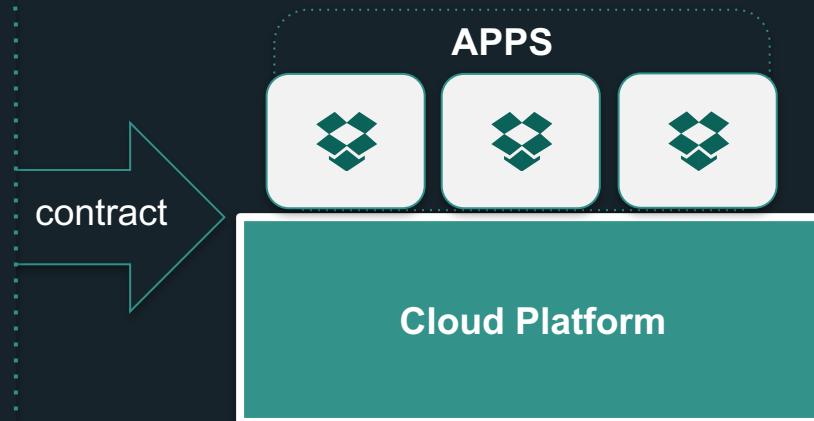


Characteristics of Cloud Native Architectures

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

Twelve-Factor Apps

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

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

Microservice: Definition

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

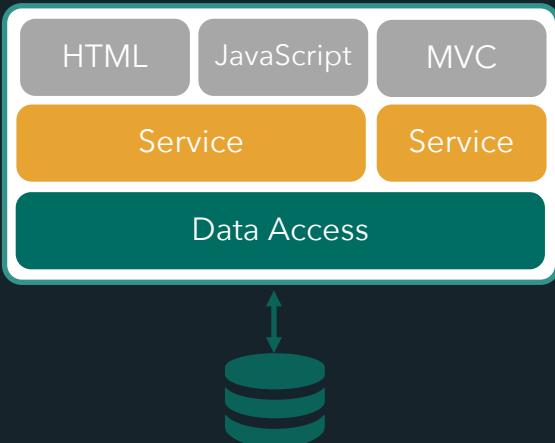
"Loosely coupled service oriented architecture with bounded contexts"

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

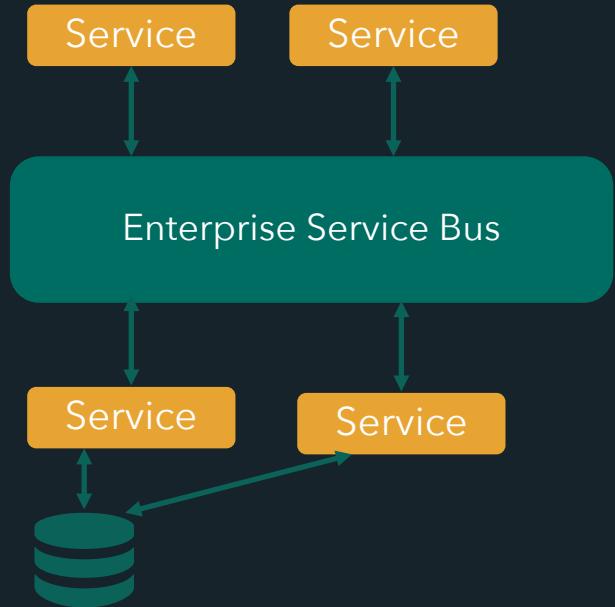
- Adrian Cockcroft

Microservices are NOT

Monolithic Application



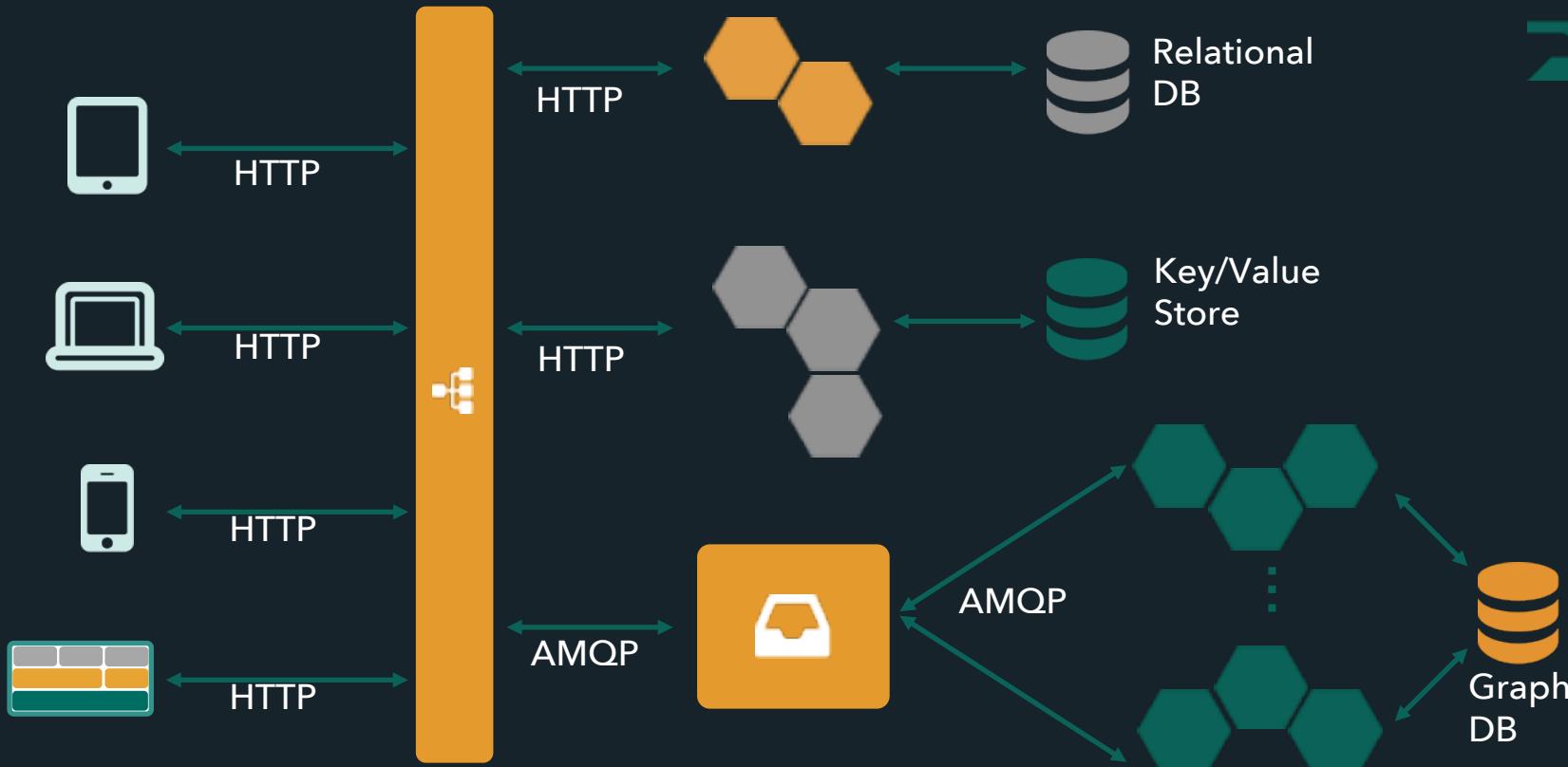
OR



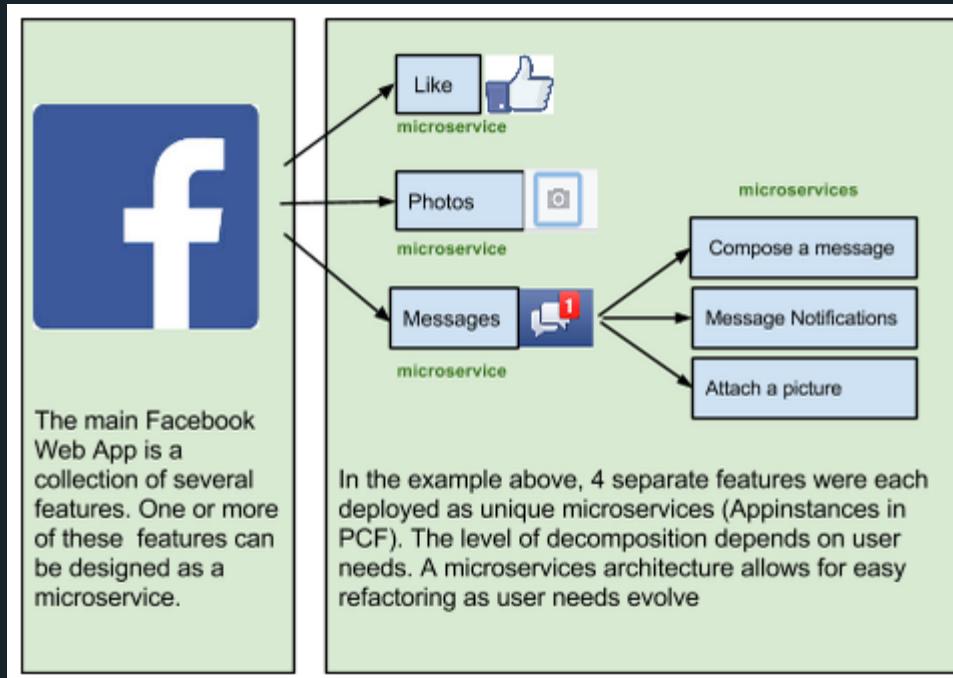
Tightly Coupled

Centralized

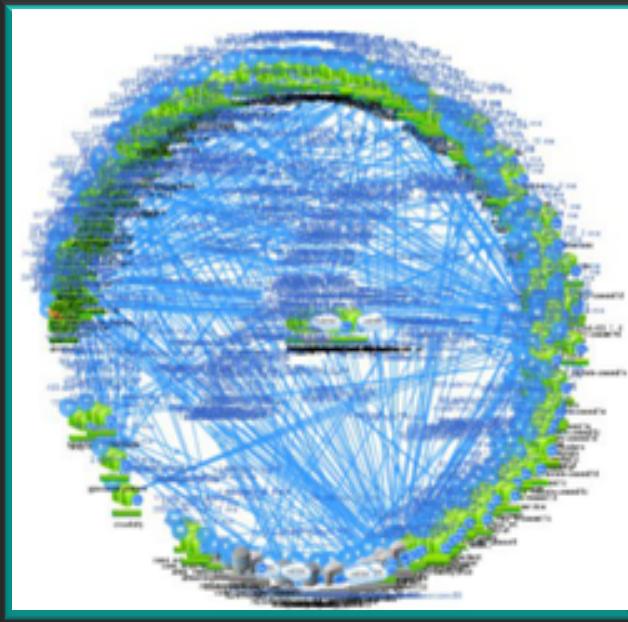
Microservice Architecture



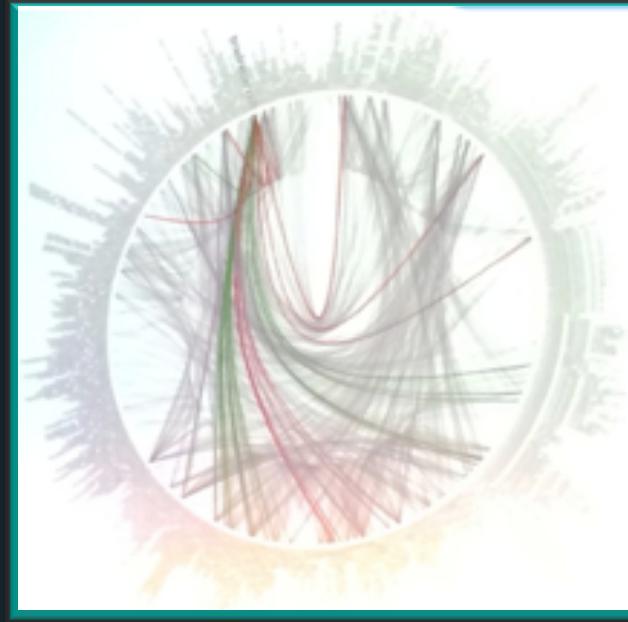
A familiar Microservice example



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 Cloud Native Application Framework



Spring

#1 Enterprise Java App Framework

Netflix

Open Source

Spring Boot

Quick Start

Little to no config.

Opinionated View

Spring Cloud

Distributed Config
Service Registration and Discovery
Fault Tolerance
Routing/Load Balancing
Service Integration

Pivotal is the main custodian of Spring and we hire the main committers for Spring Boot and Spring Cloud. We can bring these resources to bear to help significantly accelerate a customer's journey to becoming a Cloud Native enterprise

Spring Cloud

Application coordination patterns

- Runtimes and their client libraries for:
 - Configuration Management
 - Service Discovery
 - Routing & Load Balancing
 - Circuit Breakers
 - Reverse Proxies
 -and much more
- Security (encrypt, SSO, roles)
- Connectors for CF, Heroku, AWS

The screenshot shows the 'Circuit Breaker' service page. It features a circular icon with a circuit breaker symbol and the text 'Circuit Breaker'. Below it is a brief description: 'Provides aggregation and visualization of circuit breaker metrics for a distributed system deployed to Pivotal Cloud Foundry.' A link to 'Documentation | Support' is also present.

The screenshot shows the 'Service Registry' service page. It features a circular icon with a hat and the text 'Service Registry'. Below it is a brief description: 'Provides application service registration and discovery for a distributed system deployed to Pivotal Cloud Foundry.' A link to 'Documentation | Support' is also present.

The screenshot shows the 'Config Server' service page. It features a circular icon with a gear and the text 'Config Server'. Below it is a brief description: 'Provides server and client-side support for external configuration in a distributed system deployed to Pivotal Cloud Foundry.' A link to 'Documentation | Support' is also present. On the right side, there is a sidebar titled 'PLAN FEATURES' with two checked items: 'Single-tenant' and 'Netflix OSS Eureka'. A button labeled 'Select this plan' is highlighted with a mouse cursor.

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>

It takes a Cloud Native Platform

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

Essential elements of a Cloud-Native Platform

- Rapid Provisioning of infrastructure components
- Rapid Application Deployment & Management
- Scales Horizontally
- Basic Monitoring, Logs & Metrics
- No-downtime platform upgrades
- Dynamic load balancing and routing
- Manage supporting services (data, messaging)
- Failure detection and health remediation



Pivotal Cloud Foundry

A complete framework for delivering software rapidly, consistently and reliably at scale.

Application Framework



spring

**NETFLIX
OSS**

Language frameworks designed for microservice architectures with components for service discovery, metrics, circuit breakers and more.

Platform Runtime



Pivotal
Cloud Foundry

Application container runtime with production-ready backing services, automated build pipeline, routing, health management and logging.

Infrastructure Automation



Pivotal
Cloud Foundry

API-based infrastructure provisioning that delivers bit-for-bit, consistent, self-healing deployments across private and public cloud.

Pivotal

Cloud Foundry Haiku

Here is my source code

Run it on the cloud for me

I do not care how

Pivotal

Bosh Haiku

Here are my servers

Go make them a cloud foundry

I do not care how

Pivotal

Pivotal

A NEW PLATFORM FOR A NEW ERA