Orienting yourself to continuous delivery & microservices

Pivotal Cloud Roadshow, March 2015

cote@pivotal.io - @cote

Download the slides: http://cote.io/cdtalk

Pivotal

Hello!

- @cote – Director, Technical Marketing at Pivotal
- Former industry analyst at 451 Research and RedMonk
- Corporate Strategy & M&A at Dell
- Recovering code monkey
- More: http://cote.io or cote@pivotal.io

Pivotal

Conclusions

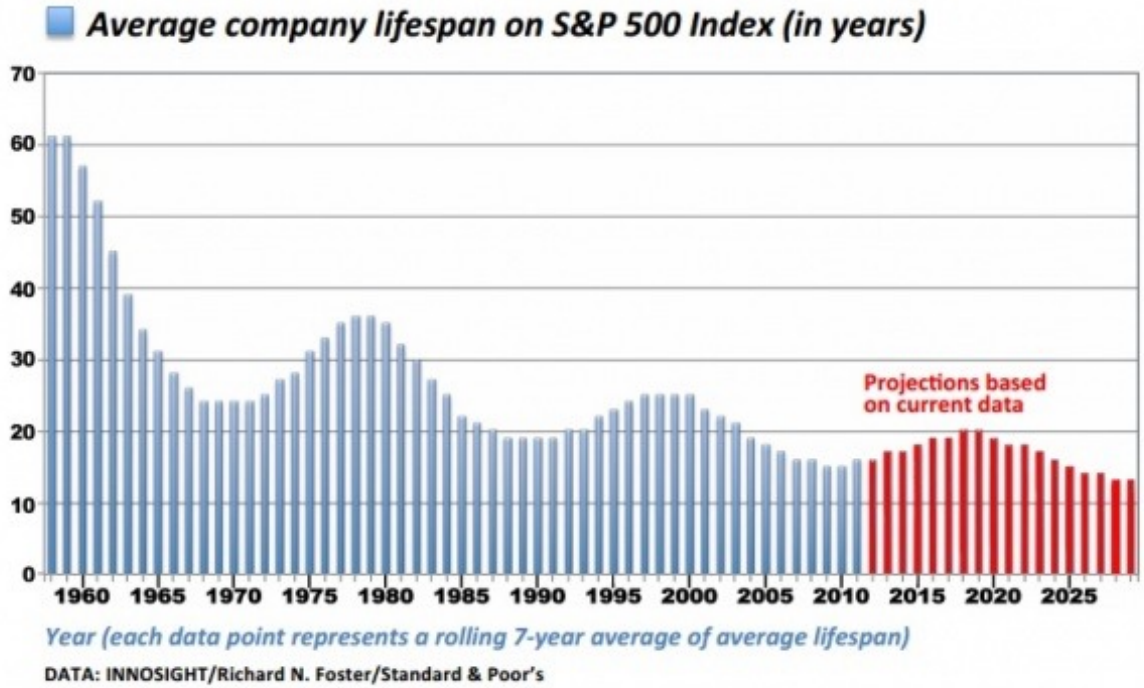
- The *business* goal is to learn your way to a better product that establishes competitive advantage, leading to profit
- "Continuous delivery" is the IT process that enables this goal. It's a horribly leaky abstraction that requires much change.
- Microservices is one of these changes, one of the larger ones
- "A lot of effort went into making this effortless" - look into an operationalized architecture
- The build is not not done until a customer is using it

Pivotal

In case you missed breakfast

Pivotal

Your company is probably not going to last

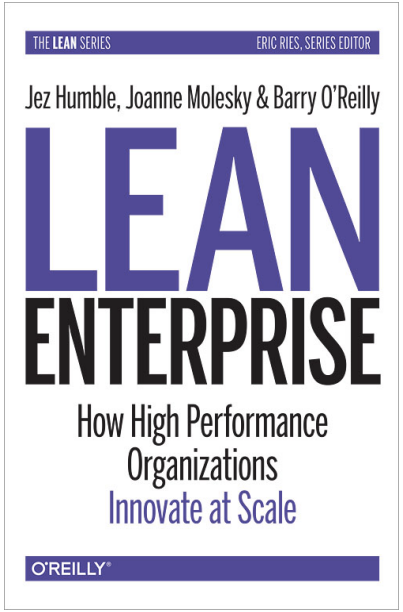


"At the current churn rate, 75% of the S&P 500 firms in 2011 will be replaced by new firms entering the S&P500 in 2027."

Source: "*Creative destruction in the S&P500 index*," Jan 2014.

Pivotal

Achieving the right mixture of risk and agility



we are clear on its purpose: *to discover and operationalize new and potentially disruptive business models, and to quickly discard those that will not work.*

Every entrepreneur, whether they work in a startup or an enterprise, has a vision of their business and the impact it will have on legions of grateful, adoring customers. For this vision to become reality, there are two key assumptions that must be tested: the *value* hypothesis and the *growth* hypothesis. The value hypothesis asks whether our business actually provides value for users by solving a real problem. If so, we can say we have

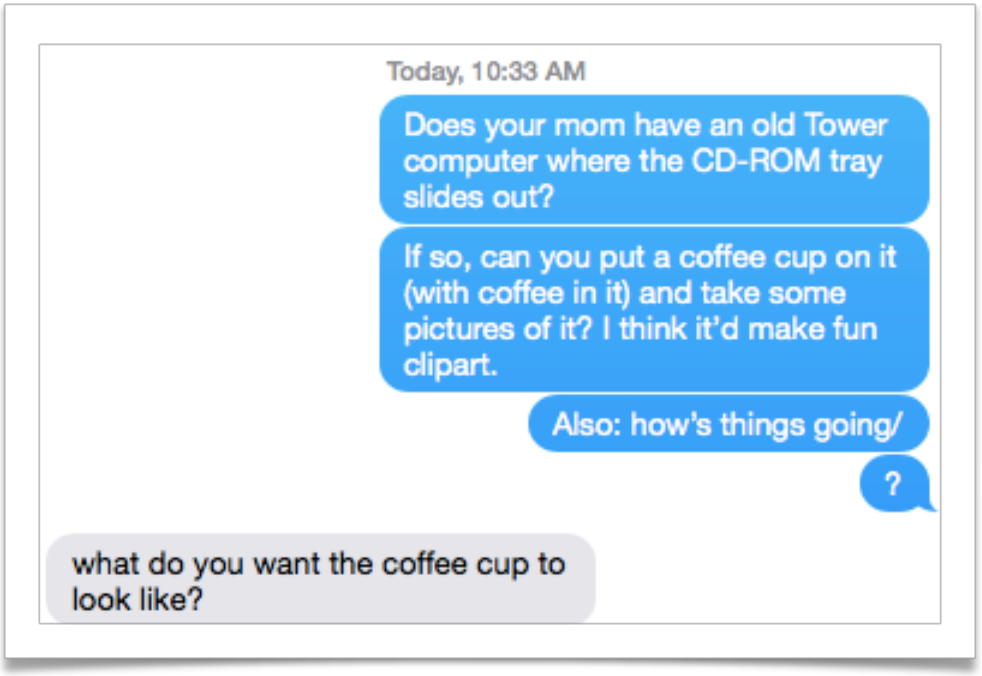
iteration will result in *validated learning*. Validated learning means that we test — to the necessary degree of precision and no further — the key assumptions behind our business model to understand whether or not it would succeed, and then made the decision to persevere, pivot, or stop.

Pivotal

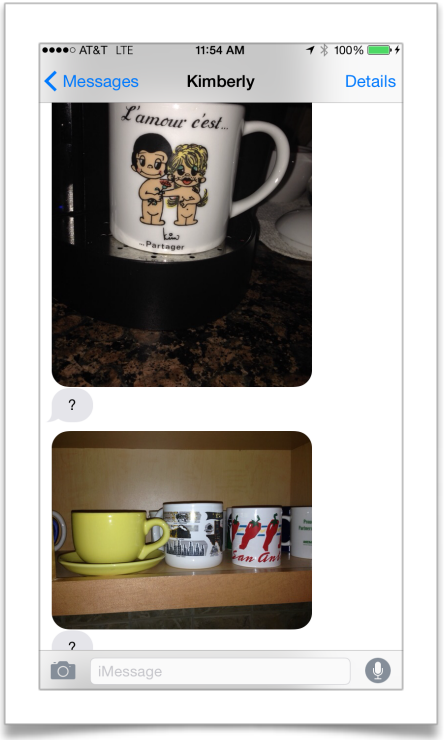
Q: How can IT help?
A: Continuous Delivery

Pivotal

Operating on Internet Time

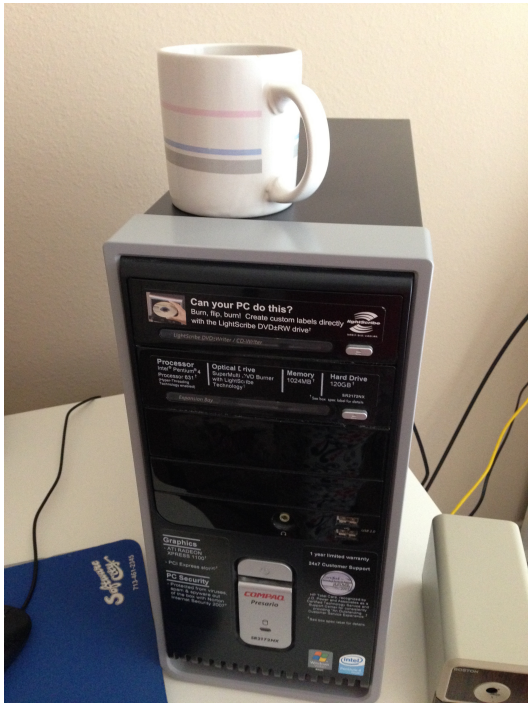


Gathering specifications...



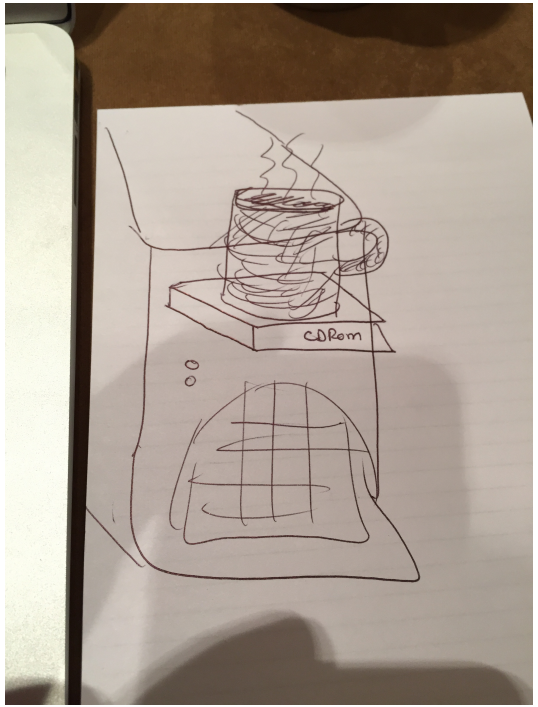
11:18am

Fail fast...



~11:30am

Learning, feedback...

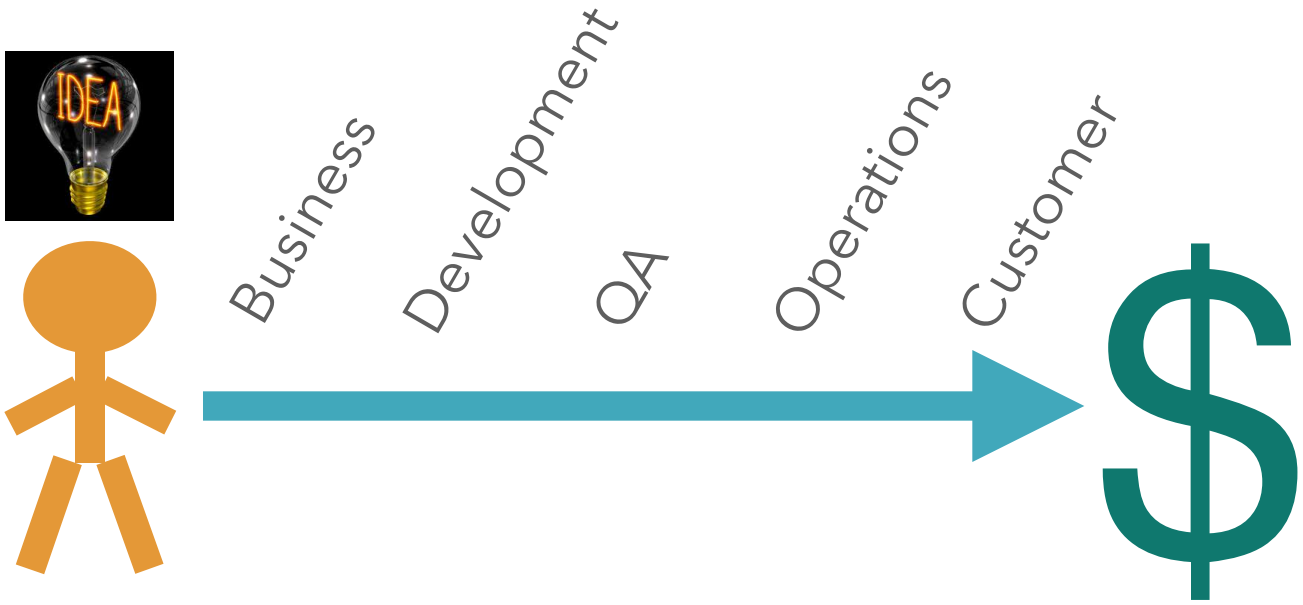


11:33am

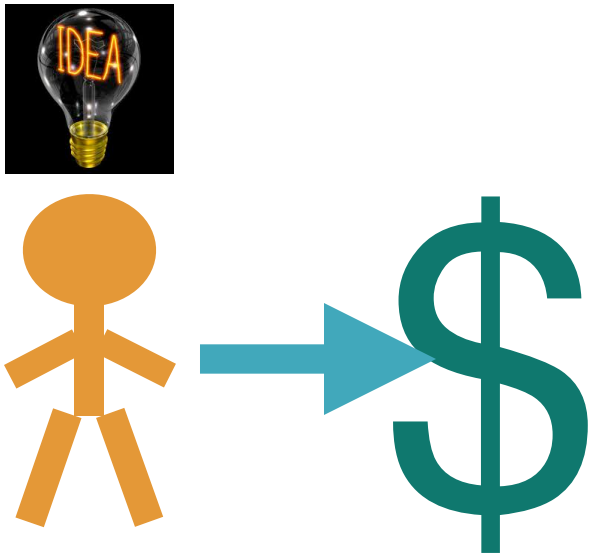
Success! Customer value creation!

11:39am

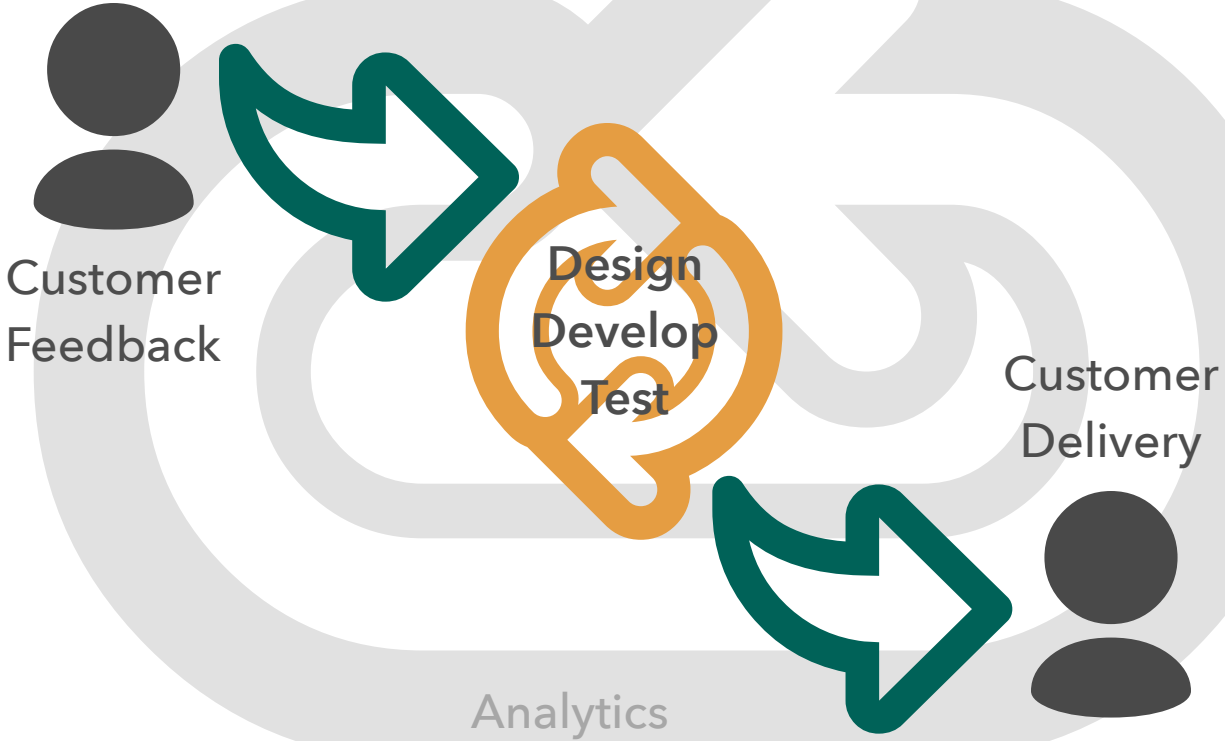
The IT cycle is too for slow concept to cash



Continuous delivery aims to shorten it



Keep the pipeline flowing!



Warner Music: Software Factories

New applications and major updates

- **Before**: 6 months, team of 10 developers
- **After**: 6 weeks, same team
- **Speed/Agility**: 400% faster on new platform
- **HR Hard Savings:** $1.1M per application update delivered

Jonathan Murray, a real person!

See more background in the Cloud Foundry 2013 talk and 2014 interview.

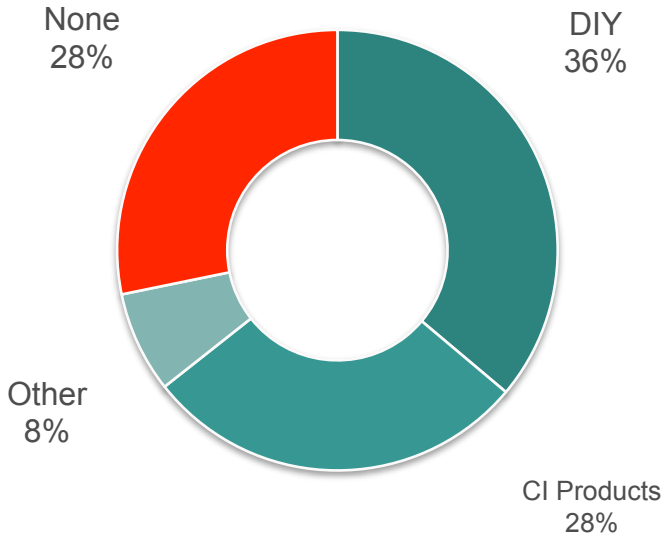
Pivotal

But things are not so nifty...

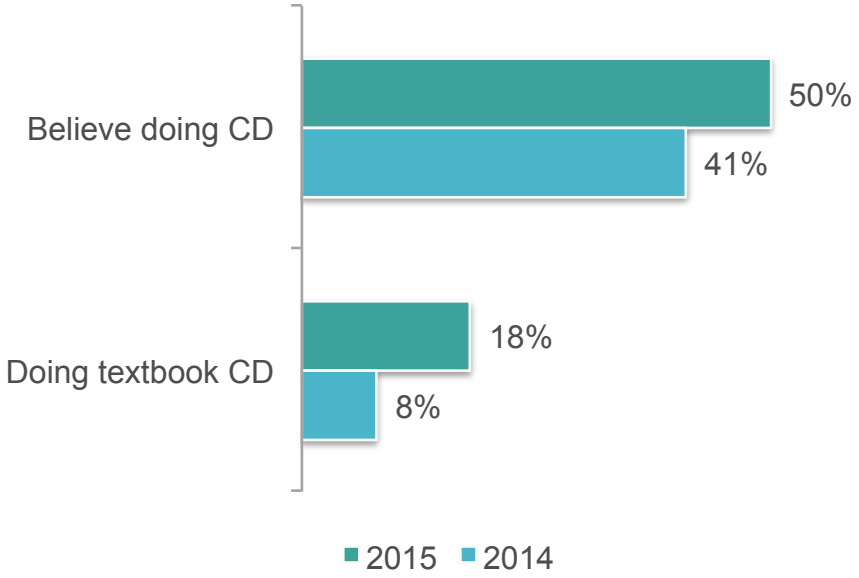
Pivotal

Usage is low, but improving

What build automaton or CI/CD tools are you using?
(451 Research study)



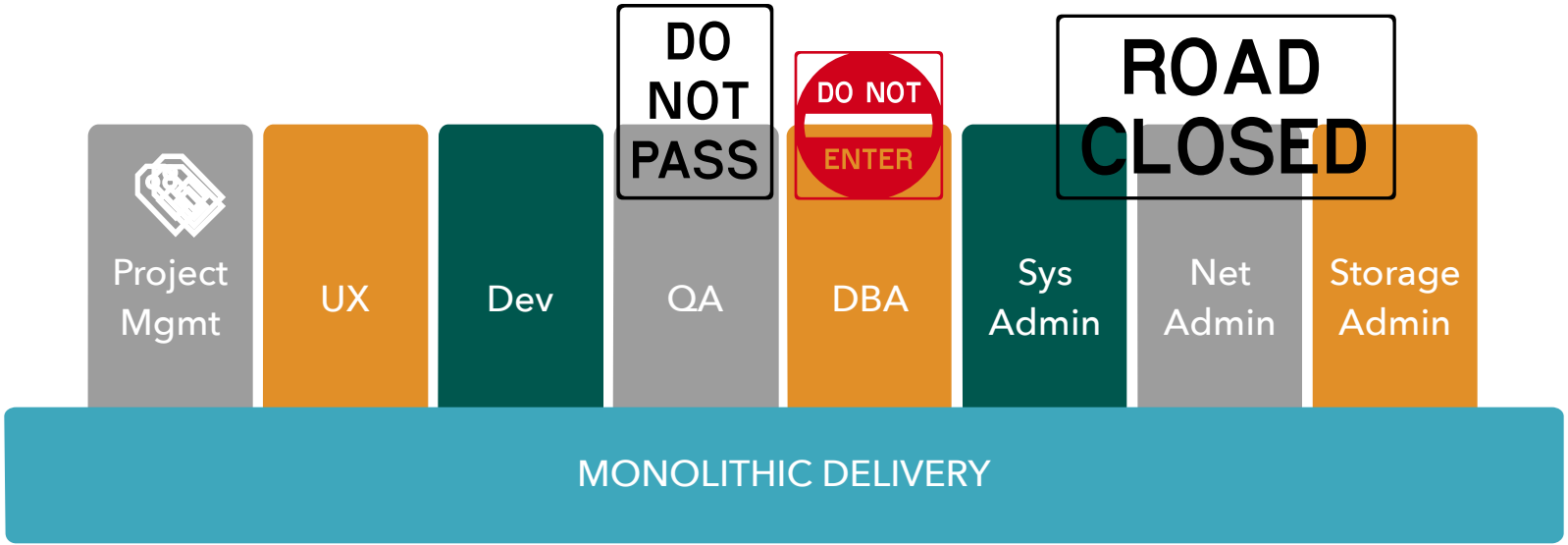
Use of CD is growing
(DZone studies)



Sources: 2014Q1 451 Research DevOps Study, n=201. In second study (n=300), 38% used “build and continuous integration tools”; "DZone's 2014 Guide to Continuous Delivery," n=500; The [DZone] Guide to Continuous Delivery, Vol. 2," Feb, 2015, n=900.

Pivotal

We've trained IT to be paranoid & slow



Adapted from: http://www.slideshare.net/adriancockcroft/goto-berlin

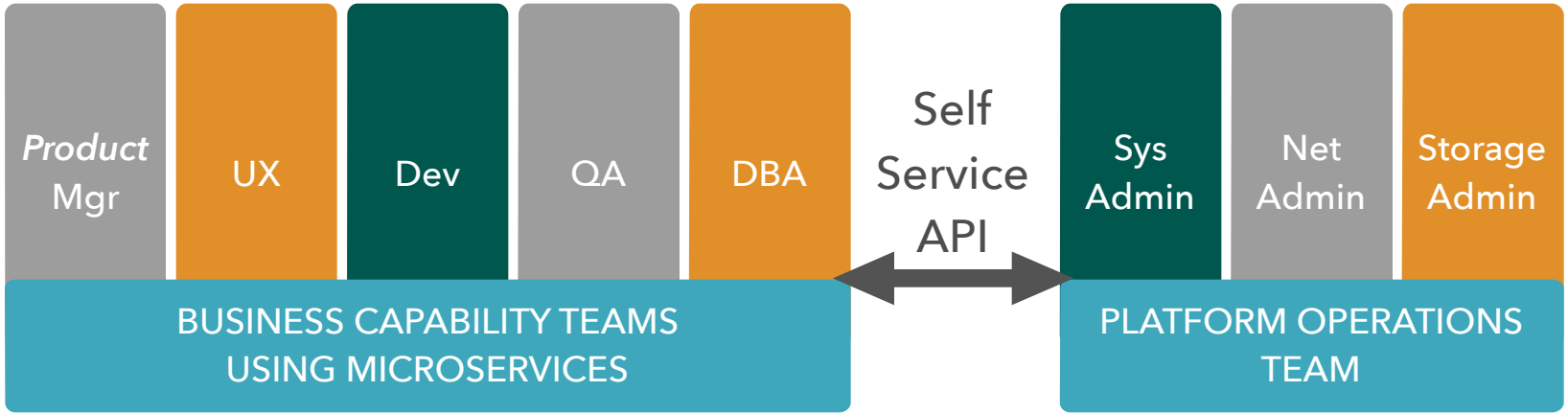
Pivotal

So let's re-train them...

Adapted from: http://www.slideshare.net/adriancockcroft/goto-berlin

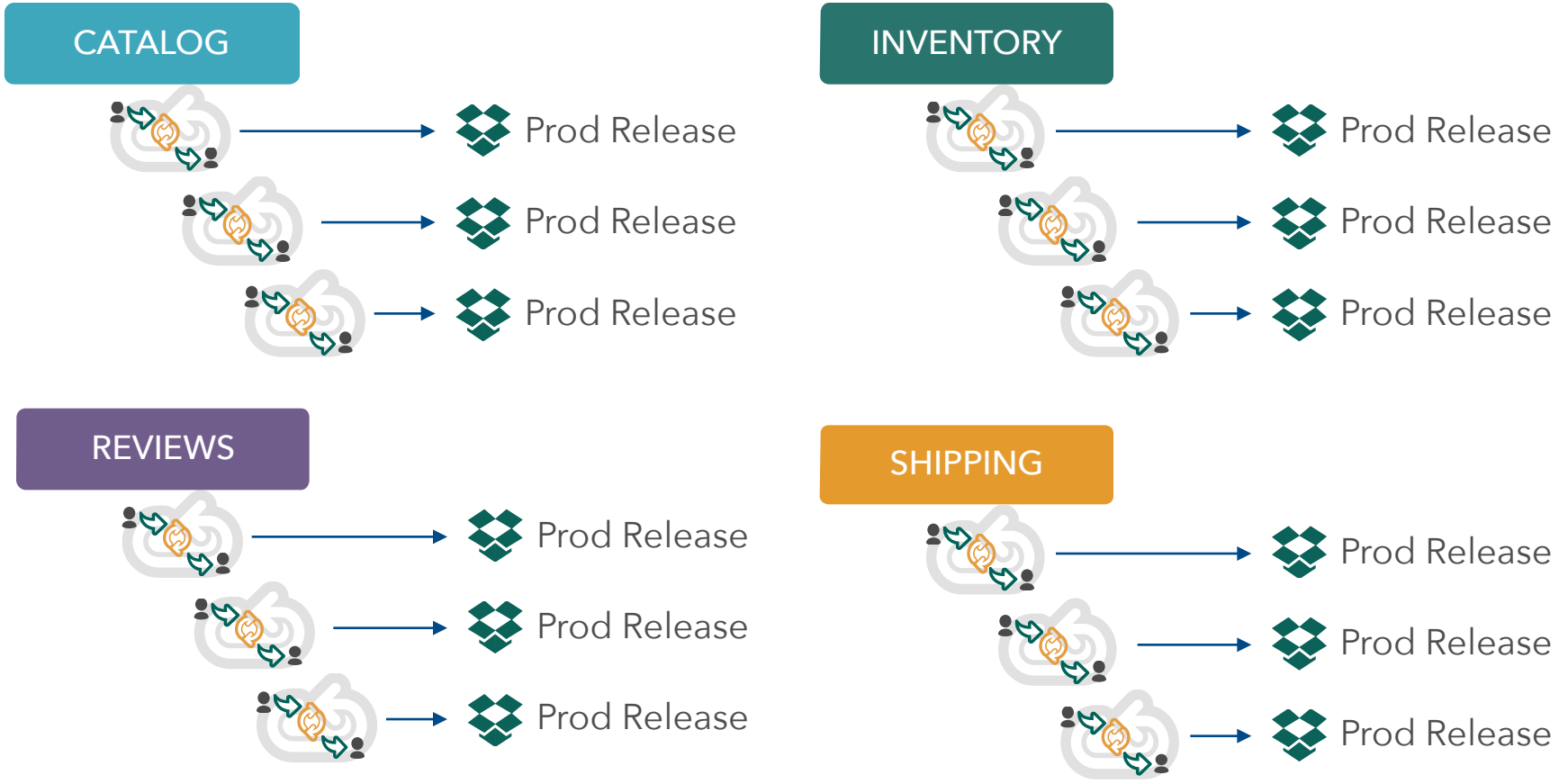
Pivotal

De-coupling: it actually works this time!



Adapted from: http://www.slideshare.net/adriancockcroft/goto-berlin

Pivotal



Pivotal

Goals & Needs

Continuous Delivery

Software Factories

Feedback

Rapid Iteration

Horizontal Scale

Diversity of Clients

Supporting Changes

Monoliths

Microservices

Applications

Infrastructure

Physical/Virtual

Pivotal CF

Pivotal

Microservices overview

Pivotal

DEFINE: Microservice

Loosely coupled service oriented architecture with bounded contexts

Adrian Cockcroft
Technology Fellow, Battery Ventures
Former Netflix Chief Cloud Architect

Pivotal

DEFINE: Microservice

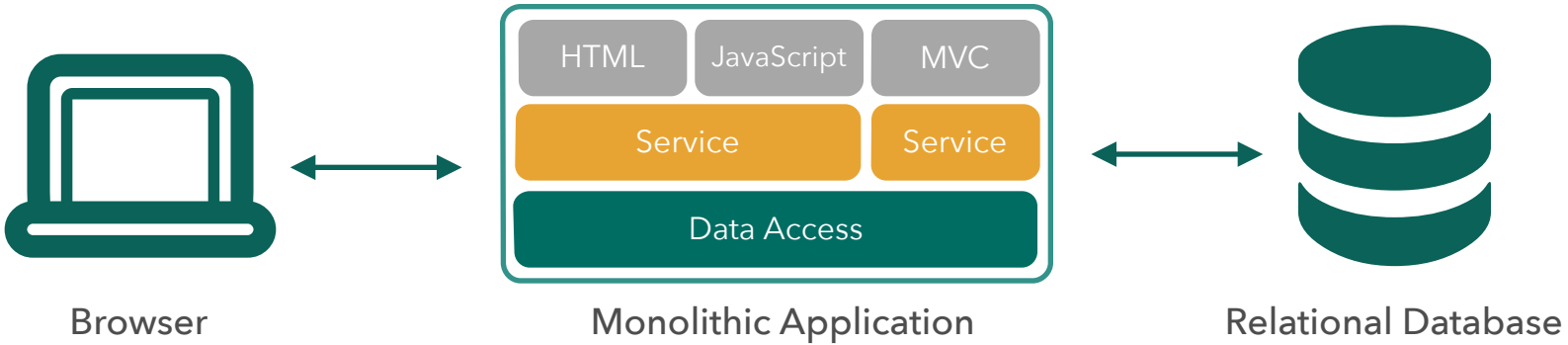
If every service has to be updated in concert, it's not loosely coupled!

Loosely coupled service oriented architecture with bounded contexts

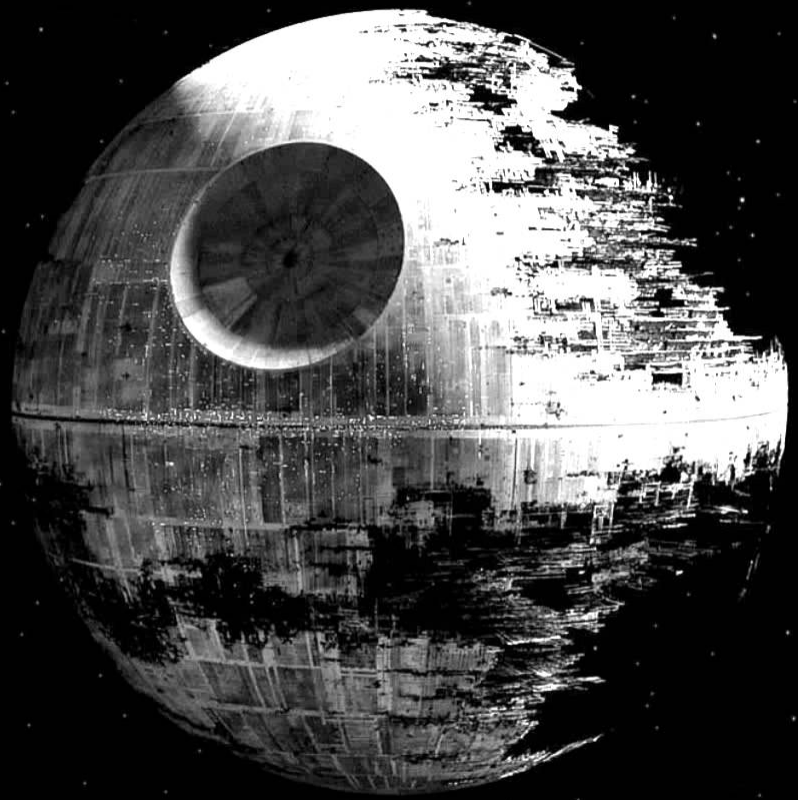
If you have to know about surrounding services you don't have a bounded context.

Pivotal

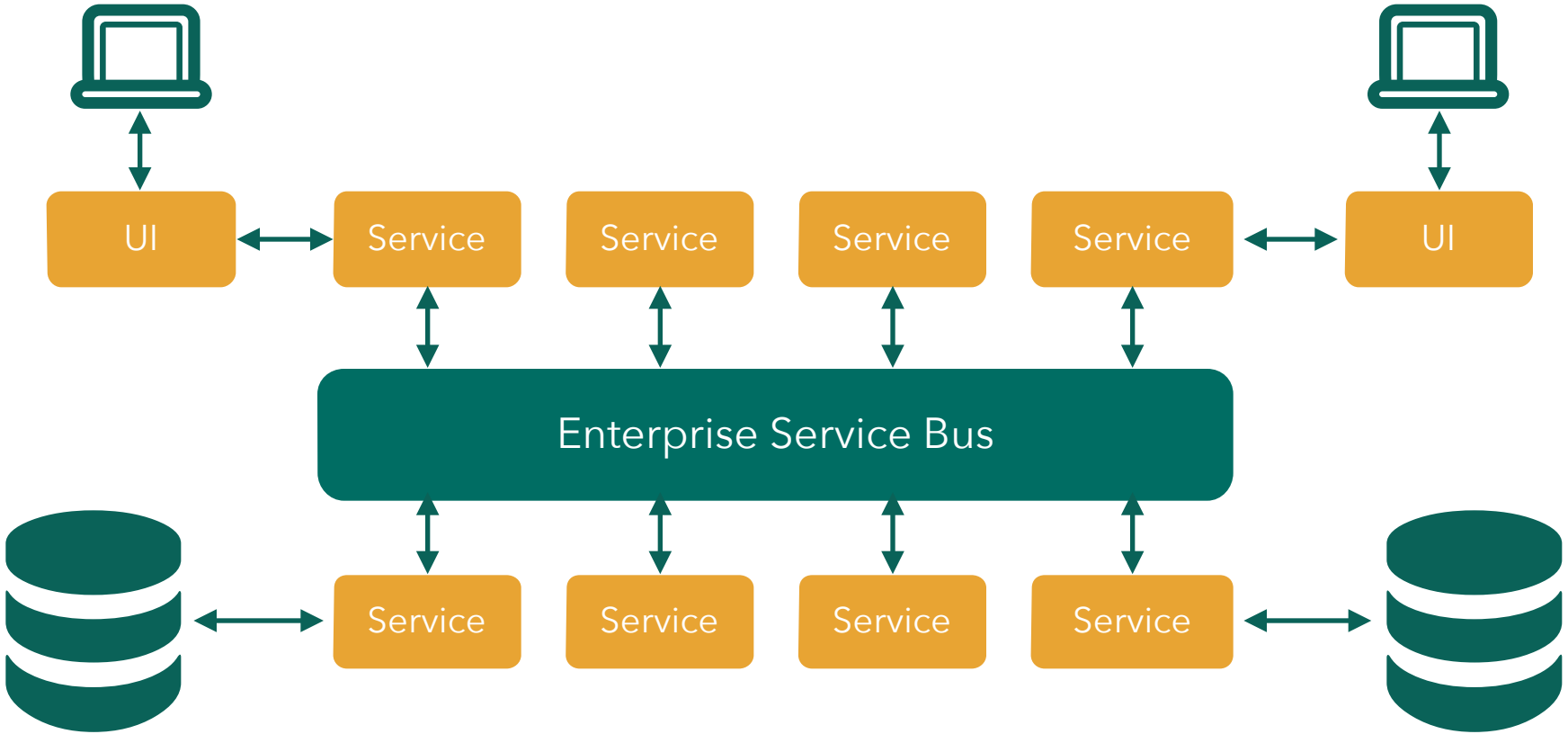
Not Monoliths



WS -

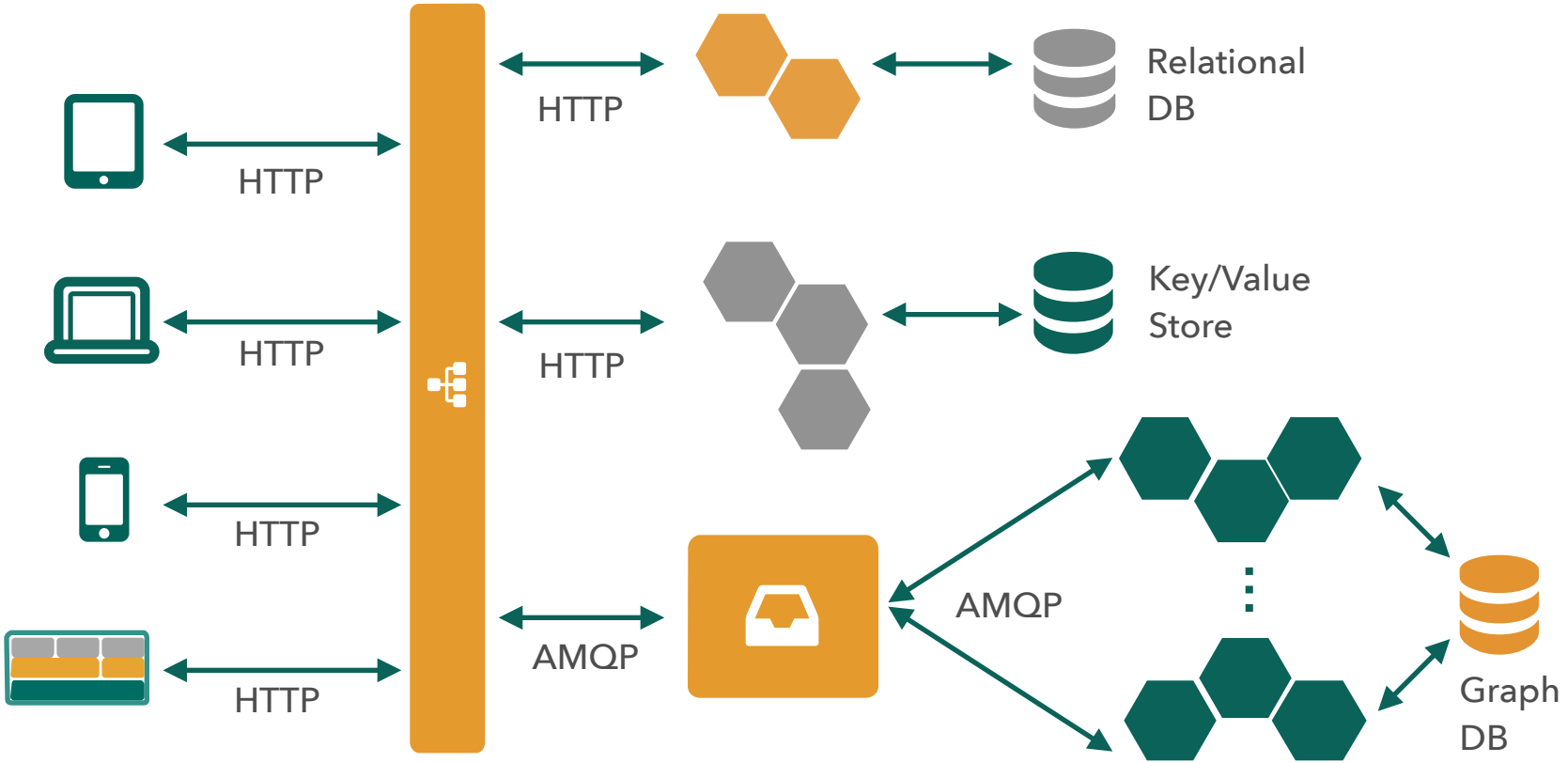


Not Traditional (ESB-centric) SOA



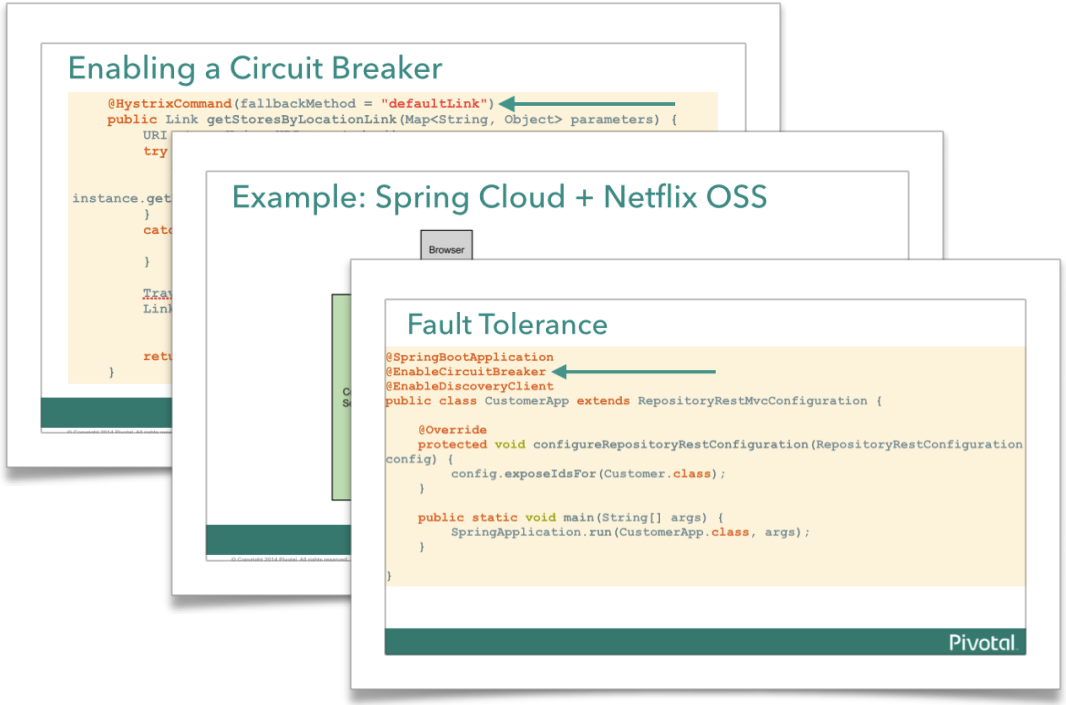
Pivotal

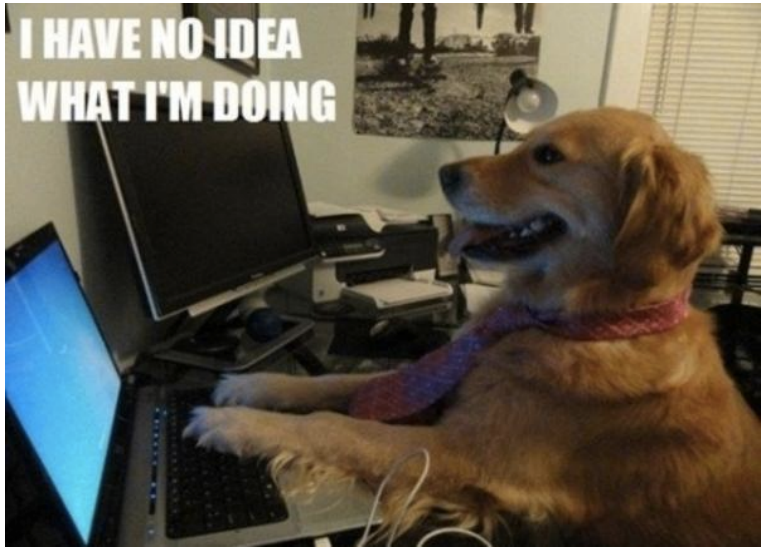
Microservice Architecture



Pivotal

Normally, here, you'd see some awesome code…





(This is me.)

Pivotal

Time to freak out ops...

Pivotal

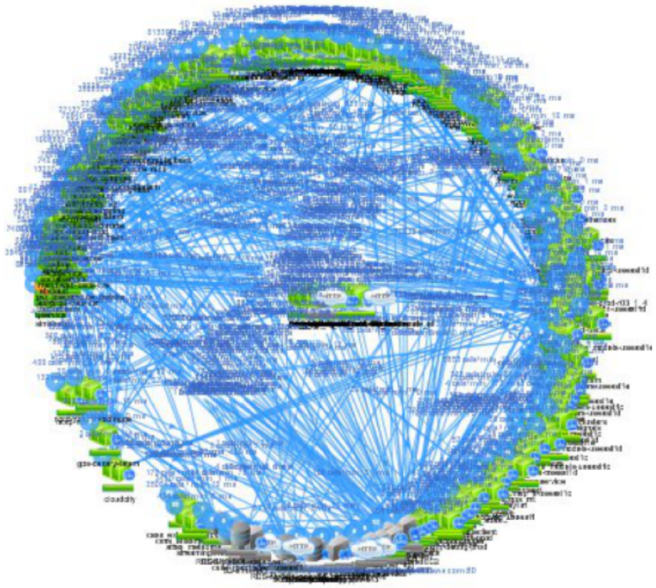
What we're telling ops...

1. Draw some circles

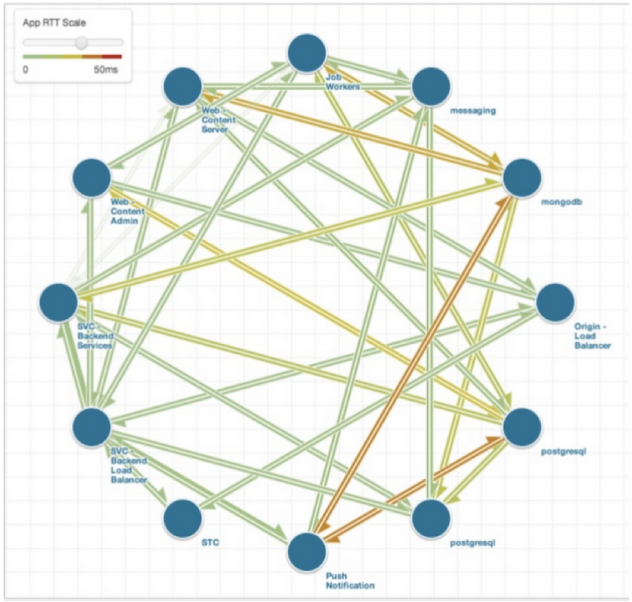
2. Draw the rest of the owl.

Pivotal

This is what ops sees!



Netflix



Gilt Groupe (12 of 450)

Twitter

High Scalability

Building bigger, faster, more reliable websites.

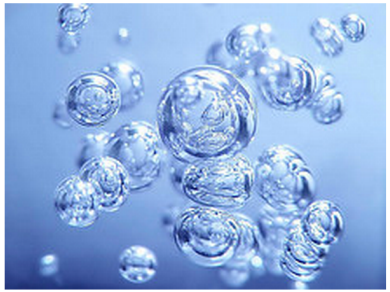
Home Real Life Architectures Strategies All Posts Advertising Book Store Start Here Contact
All Time Favorites RSS Twitter Facebook G+

« Paper: Scalable Atomic Visibility with RAMP Transactions - Scale Linearly to 100 Servers | Main | Google Finds: Centralized Control, Distributed Data Architectures Work Better than Fully Decentralized Architectures »

Microservices - Not A Free Lunch!

TUESDAY, APRIL 8, 2014 AT 8:54AM

This is a guest post by Benjamin Wootton, CTO of Contino, a London based consultancy specialising in applying DevOps and Continuous Delivery to software delivery projects.



Microservices are a style of software architecture that involves delivering systems as a set of very small, granular, independent collaborating services.

http://highscalability.com/blog/2014/4/8/microservices-not-a-free-lunch.html

Pivotal

Paying for your lunch…

- Significant Operations Overhead
- Substantial DevOps Skills Required
- Implicit Interfaces
- Duplication of Effort
- Distributed System Complexity
- Asynchronicity is Difficult!
- Testability Challenges

Pivotal

You must be this tall to use Microservices…

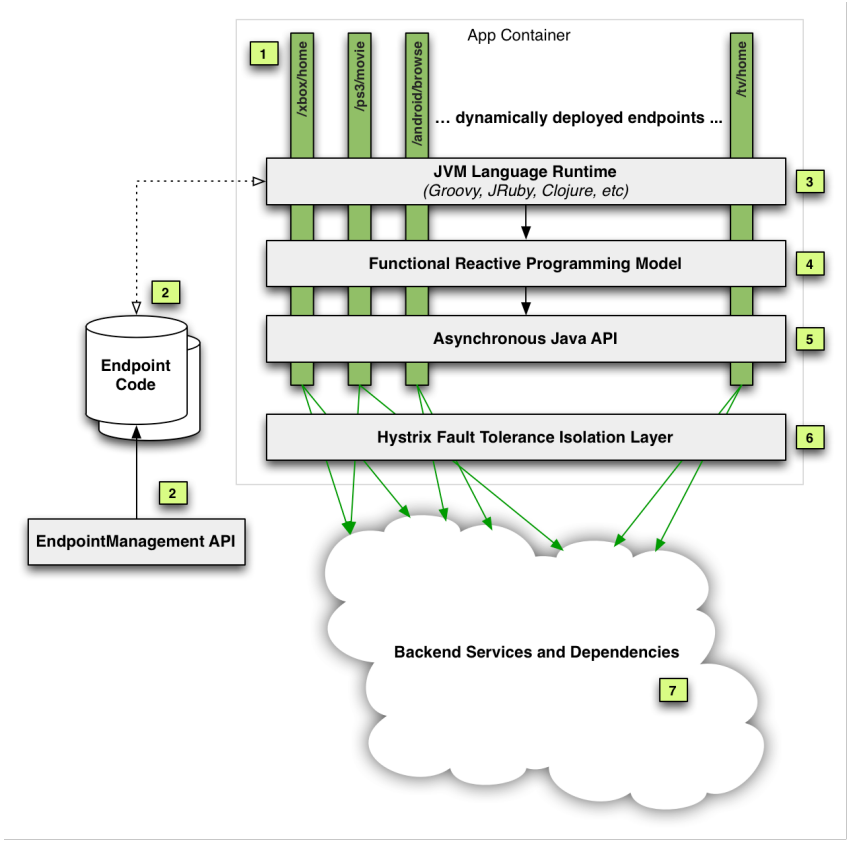
- Rapid provisioning
- Basic monitoring
- Rapid application deployment
- DevOps culture

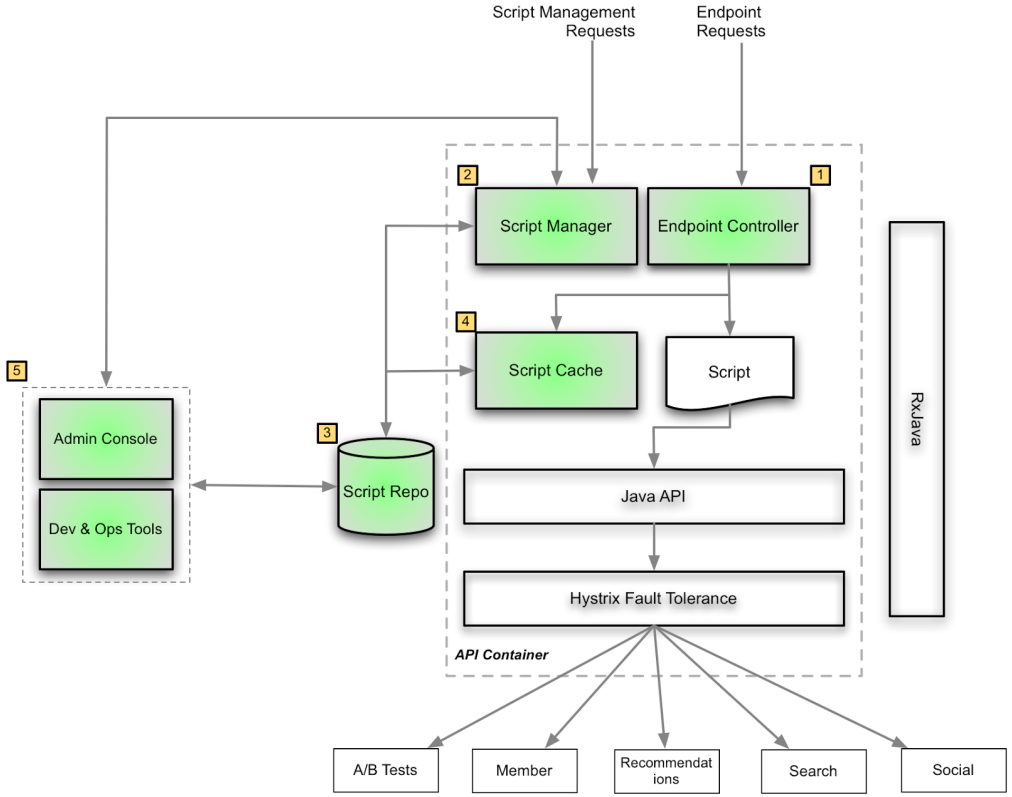
Source: "Microservice Prerequisites," Martin Fowler, August 2014.

Pivotal

It takes a platform...

NETFLIX

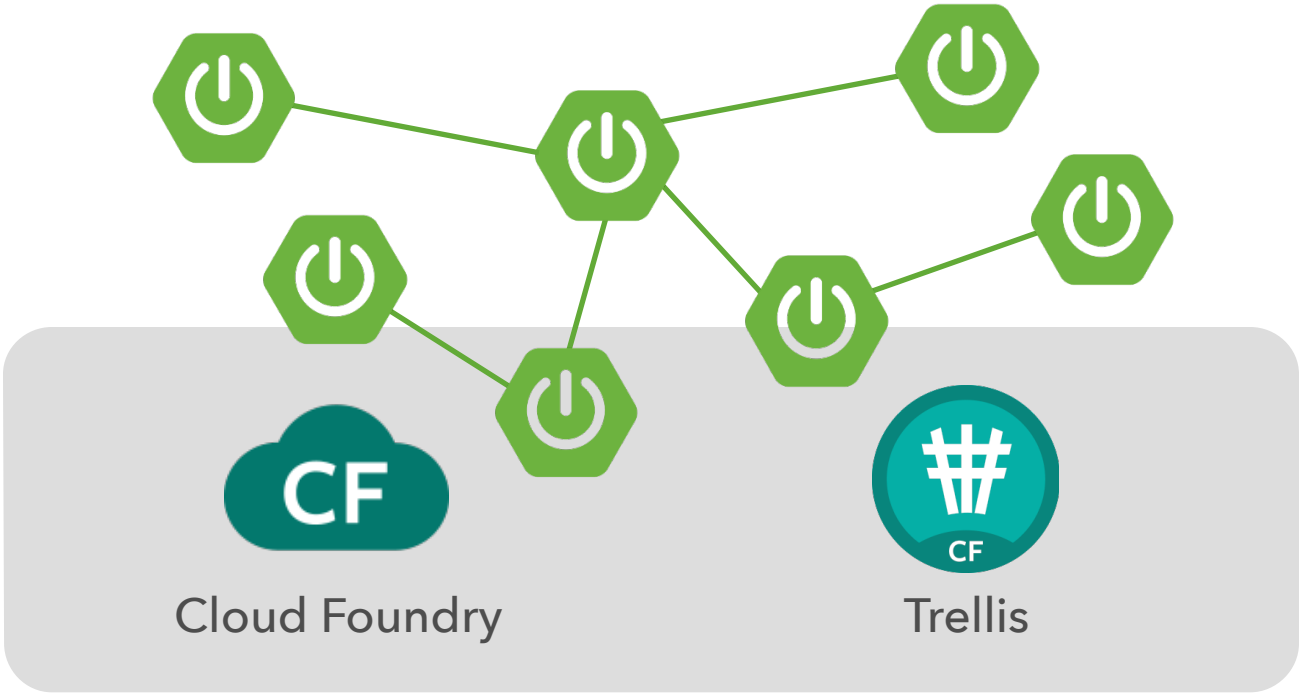




http://techblog.netflix.com/2013/01/optimizing-netflix-api.html
http://techblog.netflix.com/2014/03/the-netflix-dynamic-scripting-platform.html

Pivotal

It takes a platform…



Platform Features

- Environment Provisioning
- On-Demand/Automatic Scaling
- Failover/Resilience
- Routing/Load Balancing
- Data Service Operations
- Monitoring

Pivotal

Trellis Services Suite

Trellis
for Pivotal Cloud Foundry

Config Server
for Pivotal Cloud Foundry

Service Registry
for Pivotal Cloud Foundry

Circuit Breaker Dashboard
for Pivotal Cloud Foundry

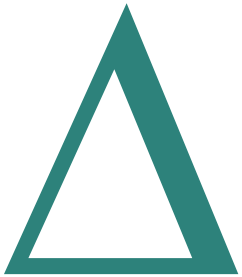
Pivotal

It seems like a lot…but it'll be fun (and valuable)

Pivotal

Supporting Rapid Change

Microservices is the first architectural style developed:



POST Continuous Delivery

&

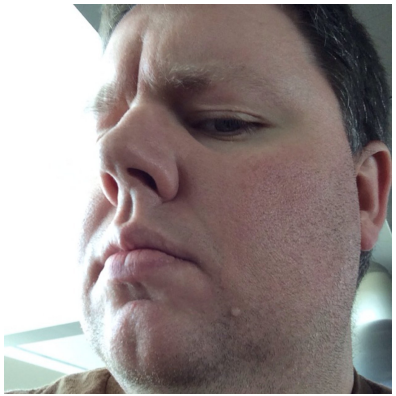
POST DevOps

Architecture is abstract until it is operationalized.

Neal Ford

Architectures that aren't operationalized exist only on whiteboards.

Matt Stine



Pivotal

An operationalized architecture