

Pivotal®

# Cloud Native Intro

## The Cloud Native Platform

---

Platform Architecture Team



A photograph showing a man with a beard and a woman sitting at a desk, looking at a computer screen together. They appear to be in an office environment with other people and equipment visible in the background.

---

# WE BELIEVE

Transforming how the world builds  
Software will shape the future.

---

Pivotal.

## Then



- Manual assembly line
- Dozens of features

Pivotal.

## Now



- Fully automated assembly lines
- Thousands of features

# Manulife Software Deployment Models

Are you able to do full continuous delivery for every major applications today?

Weeks/Months

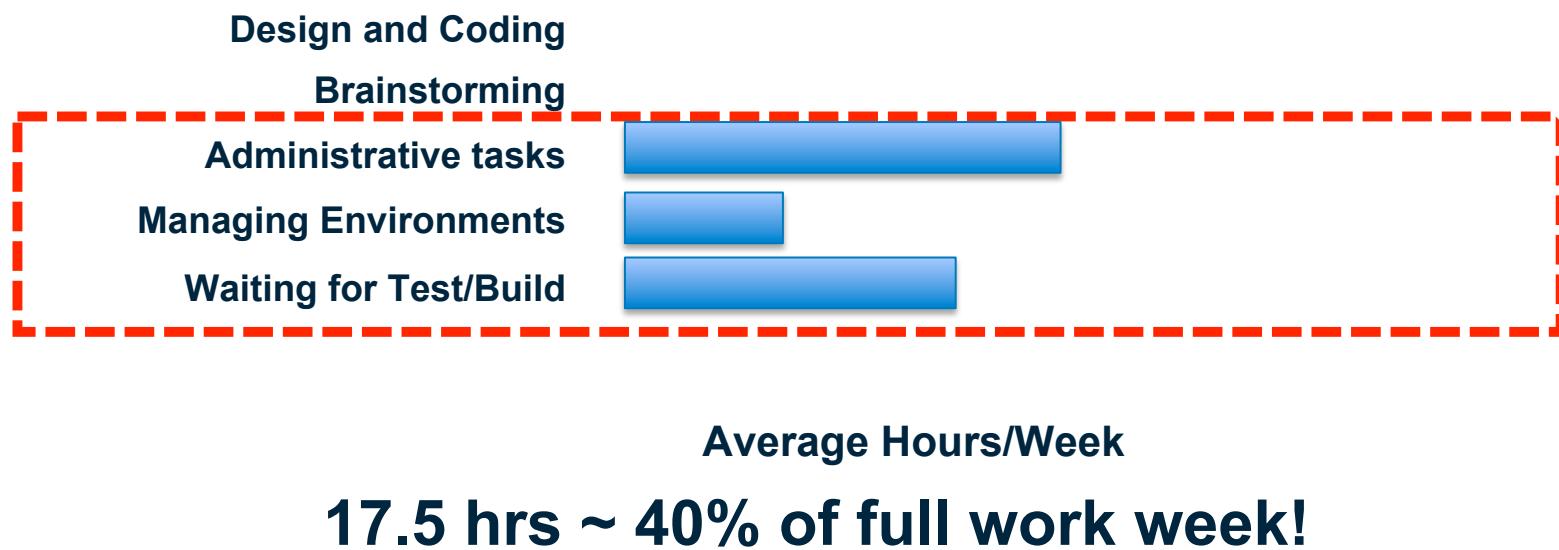
- Dev team unable to iteratively build, deploy and test **early**
- Operations teams **overwhelmed** with requests & processes
- Resources in Dev & Ops **tied up** or they **multitask** between the projects
- Businesses seriously impacted by:

Difficulty 2 release quickly → Slower time to market → **Lost Opportunity**

Lack of on-demand scaling capabilities → Downtime → **Lost Revenue**

Pivotal

# Software Developers Spend too Much Time NOT Writing Software

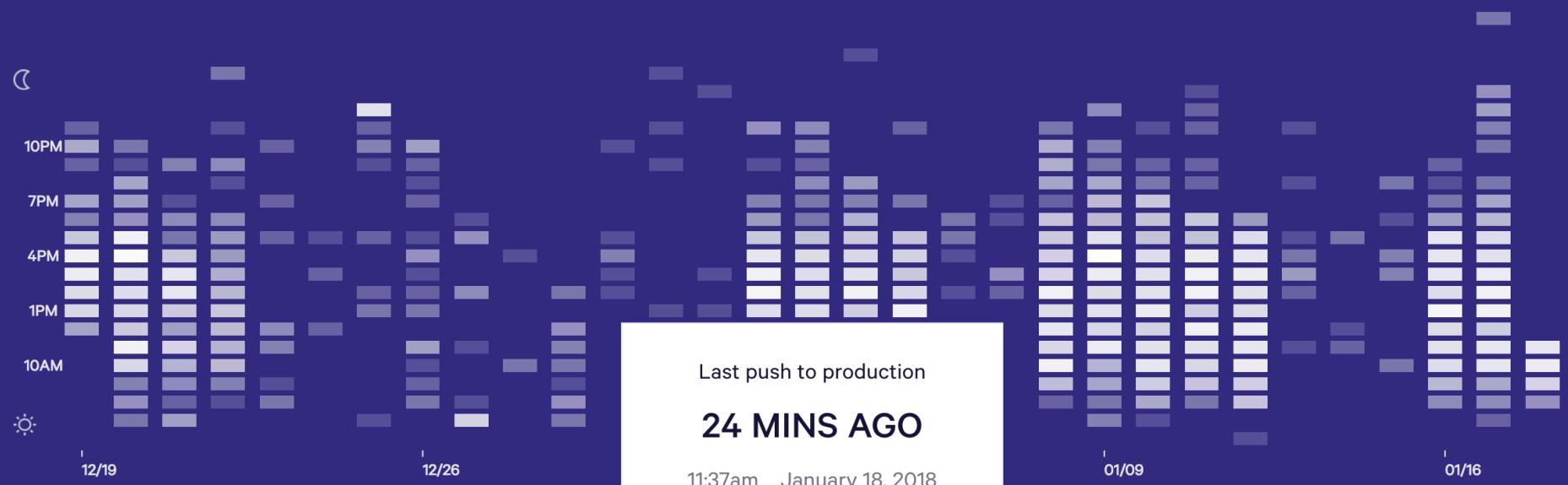


Pivotal

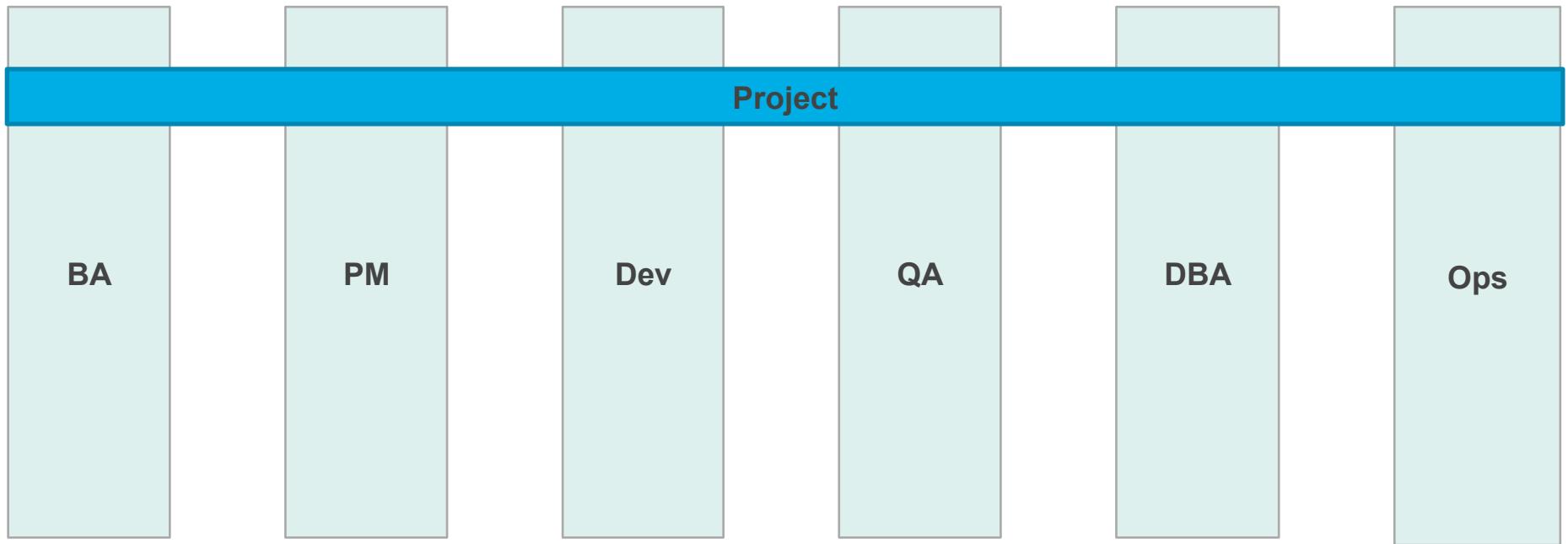
\*Sources: “Electric Cloud LinkedIn Survey to software developers”

[Offerings ▾](#)[Philosophy](#)[Expertise](#)[Origin](#)[Log in](#)[Invest now](#)

## Wealthfront **Engineering**

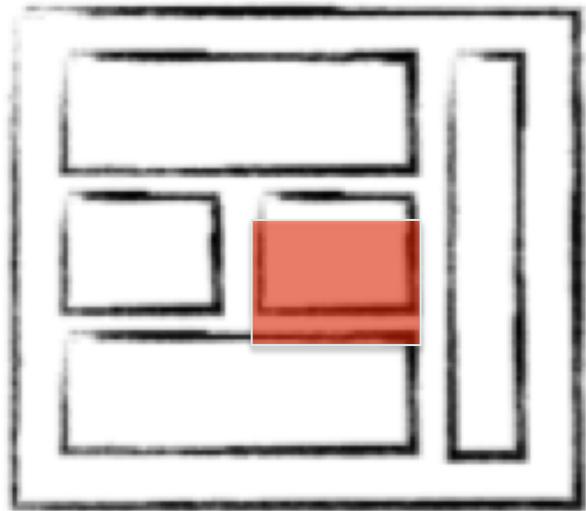
[Overview](#)[Our Team](#)[Blog](#)[Careers](#)

# Traditional Enterprise Silos



Pivotal

# Monolith Applications



Great for the first deployment ...

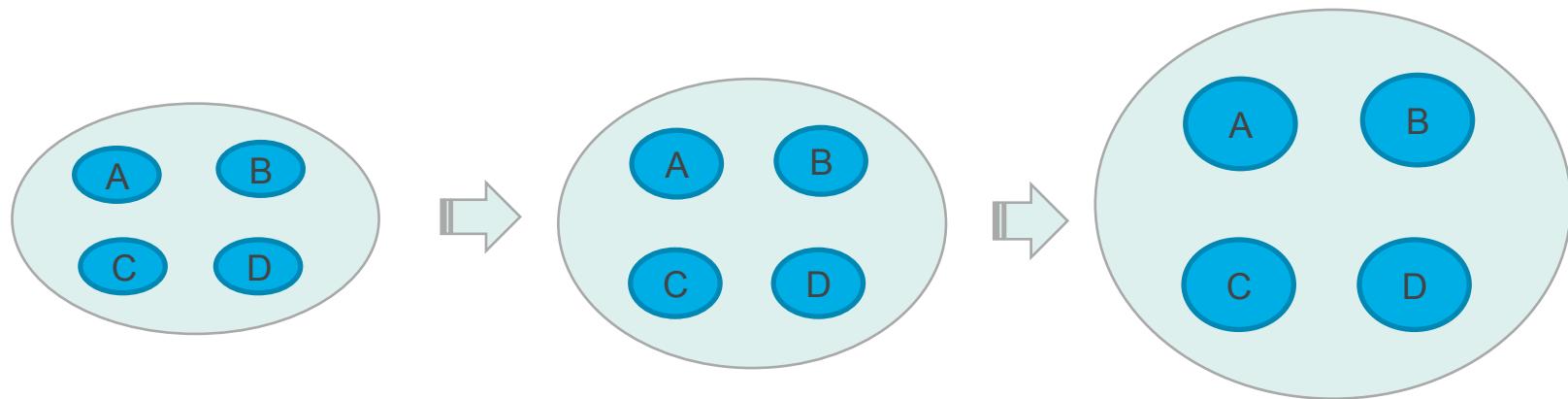
What about the updates ?

- Tightly coupled - Holistic approach
- Reliability - Adding layers & change processes
- Backwards compatibility - Regression testing

MONOLITHIC/LAYERED

Pivotal

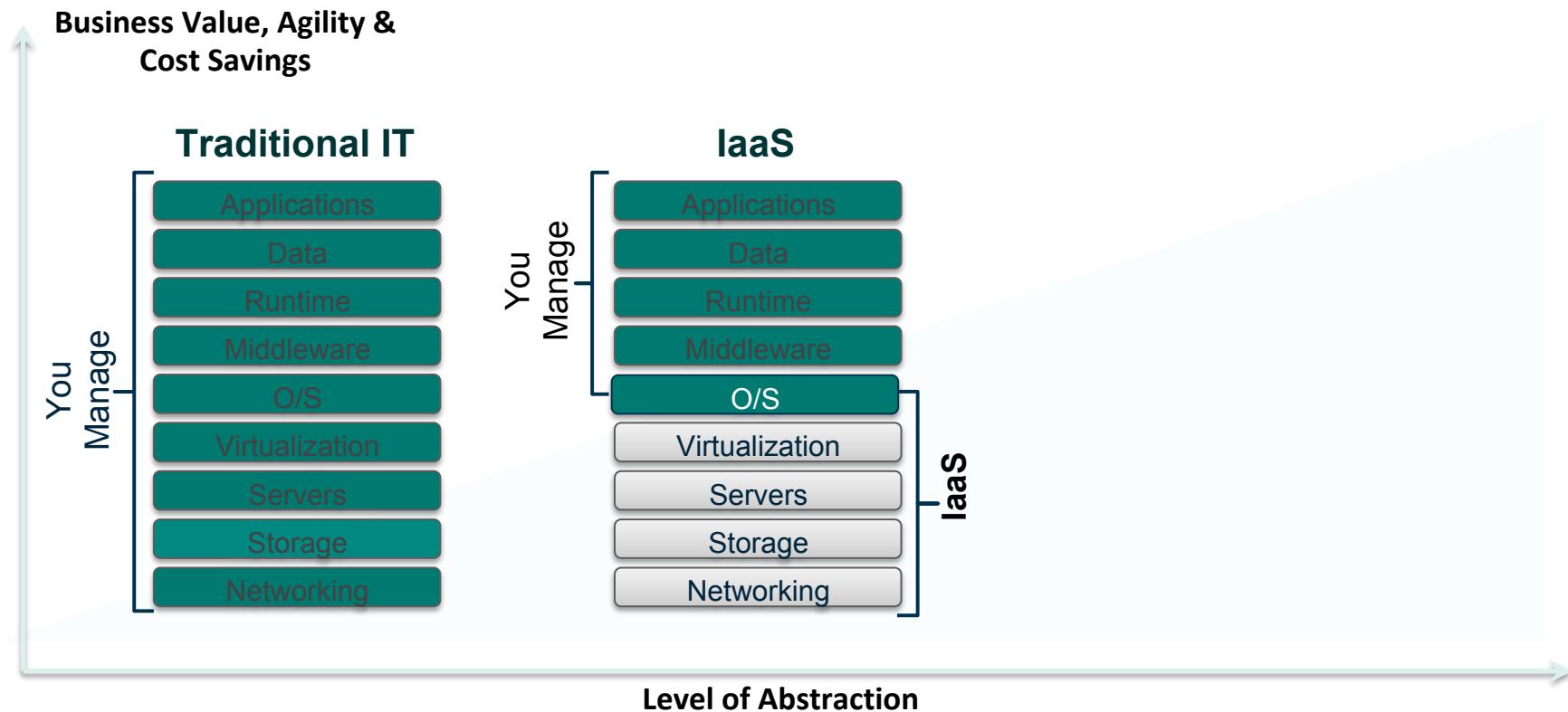
# Monolith Technology Choice & Scaling



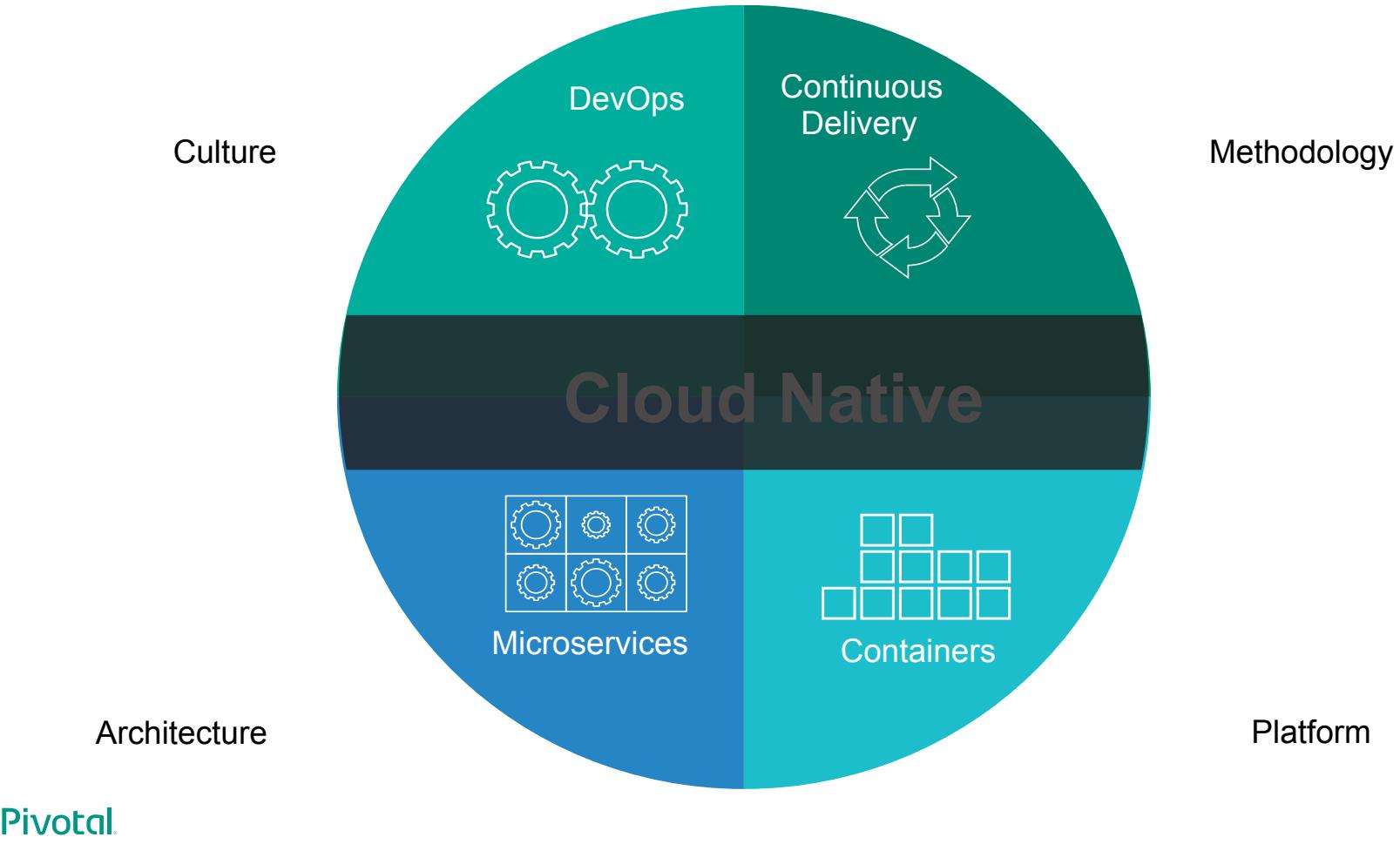
- A is mission critical used all the time with predictable load
- B is mission critical with spikey load
- C is mission critical with low usage
- D is not mission critical and is occasionally used

Pivotal

# The Cloud Platform Evolution



Pivotal

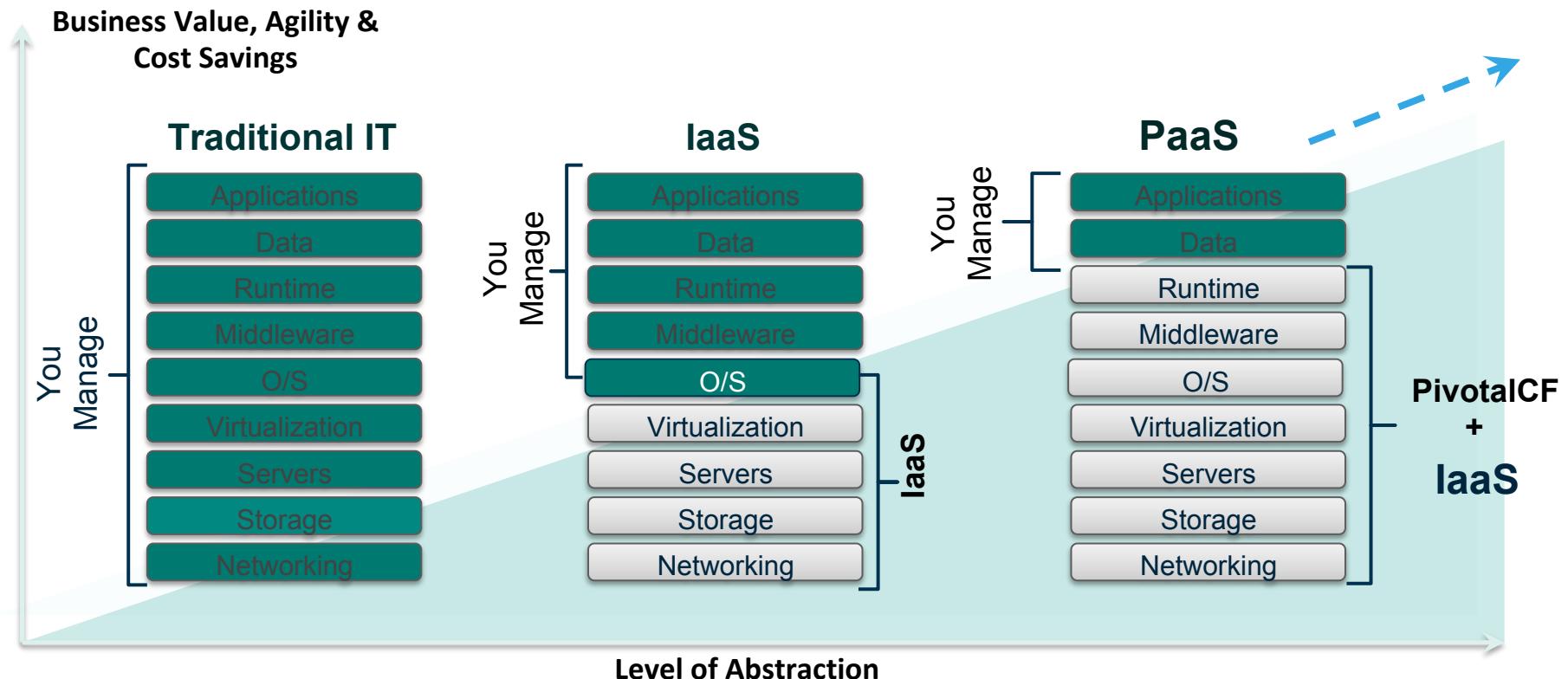


# Cloud Native - Resolving IT Gridlock



Pivotal.

# The Cloud Platform Evolution



Pivotal

# Deploying Apps Shouldn't Be Painful

## Traditional

- Provision a VM, IP, DNS
- Install Web Server
- Install Frameworks
- Deploy Application
- Configure Load Balancer
- Configure SSL Termination
- Configure Firewall
- Configure Monitoring
- Configure Logging

## Pivotal Cloud Foundry

`cf push`

Pivotal

## The developer dream haiku

Here is my source code,

Run it in the Cloud for me,

I do not care how

# What Happened ?

```
D:\pcf\PCF-demo-1>cf push
Using manifest file D:\pcf\PCF-demo-1\manifest.yml
Updating app pcfdemo-1 in org PCF-Org-01 / space development as H141869...
OK
Using route pcfdemo.apps.pcf.dce [REDACTED]
Uploading app files from: D:\pcf\PCF-demo-1\target\pcfdemo.war
Uploading 615.4K, 66 files
Done uploading
OK
Binding service myrabbit to app pcfdemo-1 in org PCF-Org-01 / space development as H141869...
OK
Binding service mylogger to app pcfdemo-1 in org PCF-Org-01 / space development as H141869...
OK
Binding service myscaler to app pcfdemo-1 in org PCF-Org-01 / space development as H141869...
OK
STOPPING app pcfdemo-1 in org PCF-Org-01 / space development as H141869...
OK
STARTING app pcfdemo-1 in org PCF-Org-01 / space development as H141869...
----> Downloaded app package (8.5M)
----> Downloaded app buildpack cache (4.6K)
it#3bd15e1
    > Downloading Open Jdk 1.8.0_40 from https://download.run.pivotal.io/openjdk/trusty/x86_64/openjdk-1.8.0_40.tar.gz (found in cache)
        Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (0.9s)
    >>> Downloading Spring Auto Reconfiguration 1.7.0.RELEASE from https://download.run.pivotal.io/auto-reconfiguration/auto-reconfiguration-1.7.0.RELEASE.jar (found in cache)
        Modifying /WEB-INF/web.xml for Auto Reconfiguration
    > Downloading Tomcat 8.0.21 instance stored from https://download.run.pivotal.io/tomcat-8.0.21.tar.gz (found in cache)
        Expanding Tomcat 8.0.21 instance to .java-buildpack/tomcat-8.0.21
    -lifecycle-support/tomcat-lifecycle-support-2.4.0.RELEASE.jar (found in cache)
    ----> Downloading Tomcat Logging Support 2.4.0.RELEASE from https://download.run.pivotal.io/tomcat-logging-support/tomcat-logging-support-2.4.0.RELEASE.jar (found in cache)
    -->>> Downloading Tomcat Access Logging Support 2.4.0.RELEASE from https://download.run.pivotal.io/tomcat-access-logging-support/tomcat-access-logging-support-2.4.0.RELEASE.jar (found in cache)

    > Uploading droplet (80M)
1 of 1 instances running
App started
OK

app pcfdemo-1 was started using this command: JAVA_HOME=$PWD/.java-buildpack/open_jdk_jre JAVA_OPTS=-Djava.io.tmpdir=$TMPDIR -XX:OnOutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh -Xmx382293K -Xms382293K -XX:MaxMetaspaceSize=64M -XX:MetaspaceSize=64M -Xss995K -Djava.security.egd=file:///dev/urandom -Daccess.logging.enabled=false -Dhttp.port=$PORT" $PWD/.java-buildpack/tomcat/bin/catalina.sh start
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: pcfdemo.apps.pcf.dce [REDACTED]
last uploaded: Mon Aug 3 00:19:12 UTC 2015
stack: cflinuxfs2
#0  state      since
#0  running  2015-08-02 07:19:38 PM  CPU 0.0%  memory 345.8M of 512M  disk 138.3M of 1G  details
D:\pcf\PCF-demo-1>
```

Firewall & Routes

Service Bindings

App Lifecycle

Runtime Installation & Config

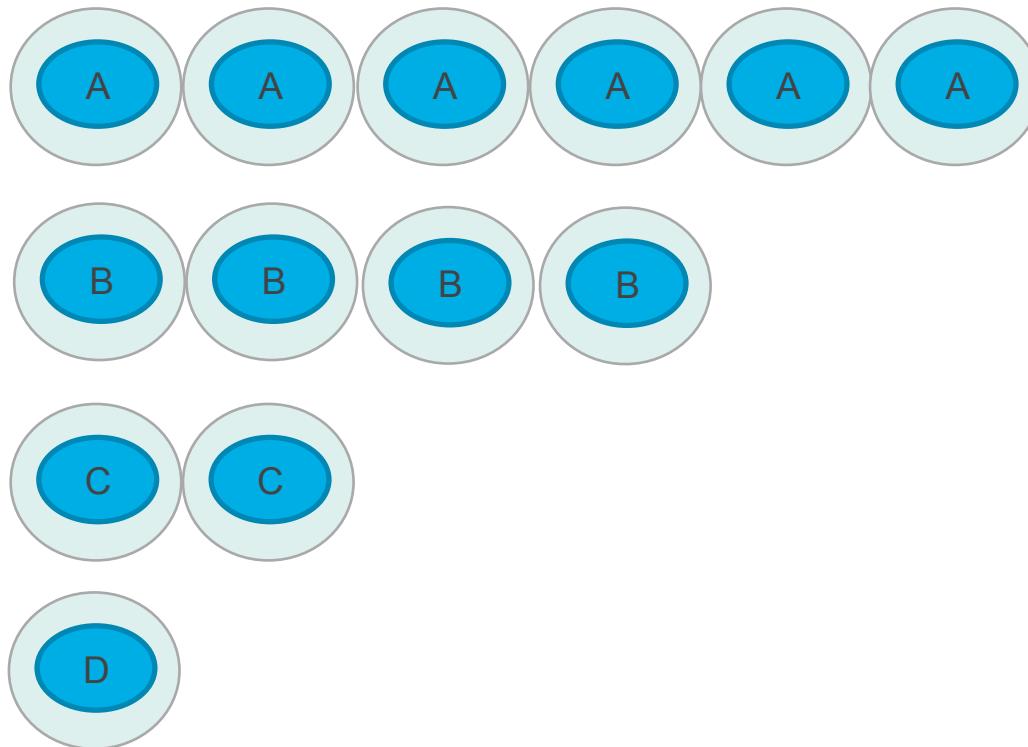
Middleware Installation & Config

Application Installation & Config

App Lifecycle

Logging, Health, Telemetry

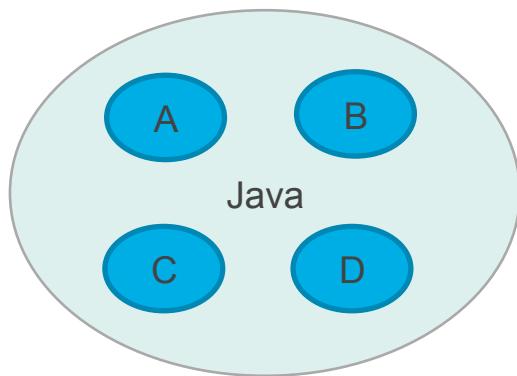
# Microservices Scaling Out



- mission critical 24/7 with predictable load
- mission critical with spikey load
- mission critical with low usage
- not mission critical & occasionally used

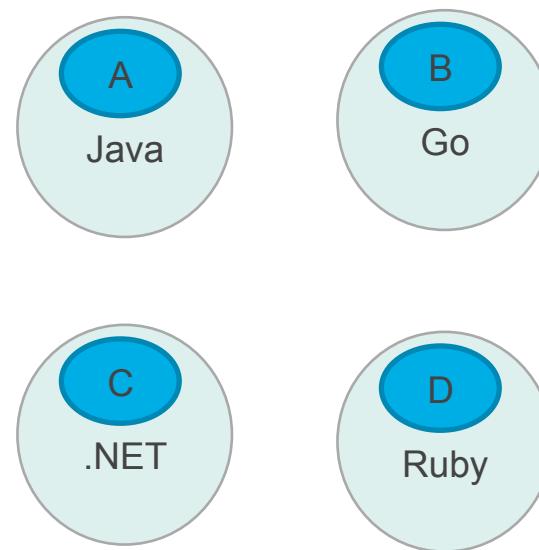
# Technology Choices

Then



Monolithic Application  
implemented using a  
**single technology**

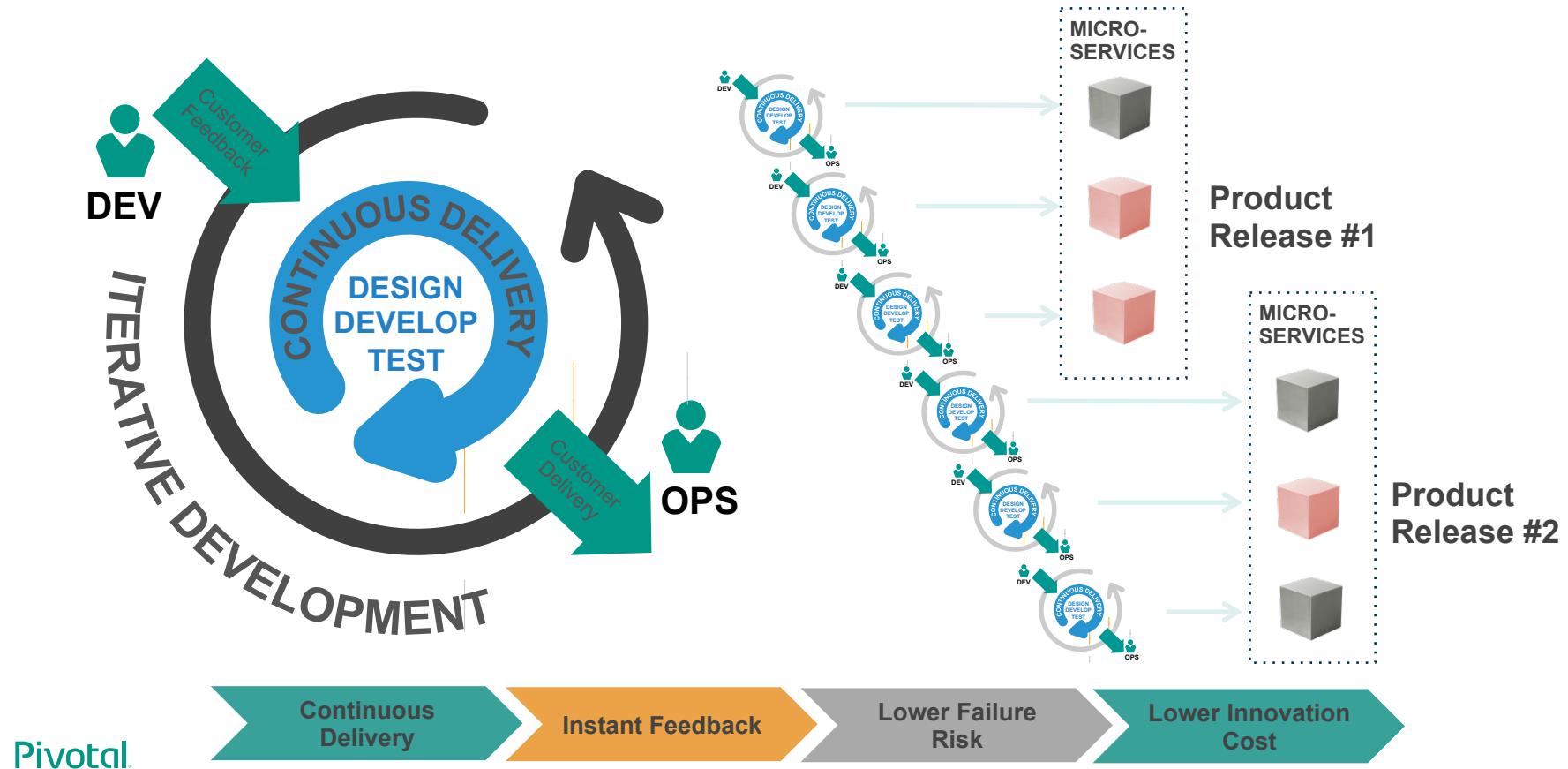
Now



Each microservice can be implemented using  
the **best technology** for the feature

Pivotal

# Continuous Delivery → Time To Market & Flexibility



# DevOps – Separation of Concerns

## DEVELOPER

target <my cloud>  
push <my app>  
bind <my services>  
scale <my app> +1000

The screenshot shows the Pivotal CF interface. At the top, there are tabs for 'development', 'stage', and 'test'. Below this, the 'APPLICATIONS' section lists 'Spring Music' with 100 instances and 128MB memory. The 'SERVICES' section lists 'mysql\_staging' (Pivotal MySQL Dev 100mb), 'rabbitdev' (Pivotal RabbitMQ), and 'rabbitmq' (Pivotal RabbitMQ). On the right, the 'TEAM' section shows members: James.Jameson@gmail.com, admin, frank.zaylor@gopivotal.com, jsmith@gmail.com, martin.fowles@yahoo.com, mohanand@gopivotal.com, rboohman@gopivotal.com, and susan.branson@gmail.com. There are buttons for 'EDIT' and 'INVITE'.



App Deployment: 30-90 seconds

## OPERATOR

provision cloud <Public/Private>  
provision service <PaaS,Hadoop...>  
upgrade/update <my cloud>  
scale <my cloud>

The screenshot shows the Pivotal Ops Manager interface. The left sidebar lists 'Available Products': Pivotal RabbitMQ (No upgrades available), Ops Manager Director for VMware vSphere (No upgrades available), Pivotal Elastic Runtime (No upgrades available), HD (No upgrades available), Metrics (No upgrades available), Pivotal MySQL Dev (No upgrades available). The right side shows the 'Installation Dashboard' with tiles for 'Pivotal RabbitMQ' (v1.1.0), 'Pivotal Elastic Runtime' (v1.1.0, highlighted with a red dashed border), 'HD' (v1.1.0), 'MySQL Dev' (v1.1.0), and 'Ops Metrics' (v1.1.0).



Cloud Deployment: 2-4 hours

Pivotal



# Pivotal®

---

## Transforming How The World Builds Software

© Copyright 2017 Pivotal Software, Inc. All rights Reserved.