RED HAT SUMMIT

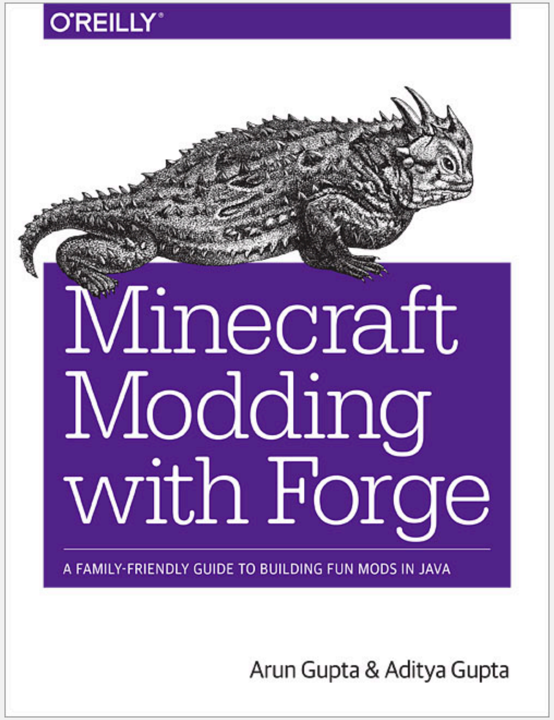
BOSTON, MA
JUNE 23-26, 2015

How DevOps and microservices impact your application architecture and development

Arun Gupta - Director, Developer Advocacy and Technical Marketing, Red Hat
Christina Wong - Principal Product Marketing Manager, Red Hat

redhat.

Arun – avid marathoner, STEM to kids


O'REILLY

Minecraft Modding with Forge

A FAMILY-FRIENDLY GUIDE TO BUILDING FUN MODS IN JAVA

Arun Gupta & Aditya Gupta

Arun Gupta, helping solve problems

Director, JBoss Middleware

- Technical Marketing
- Developer Advocacy

Hot topics

- JBoss Middlware
- Microservices
- DevOps
- Containers
- Developer Tooling

Christina Wong, crazy race car driver

Christina Wong, responsible product marketing manager

JBoss Middleware application runtimes and services

- Red Hat JBoss Enterprise Application Platform
- Red Hat JBoss Data Grid

Hot topics

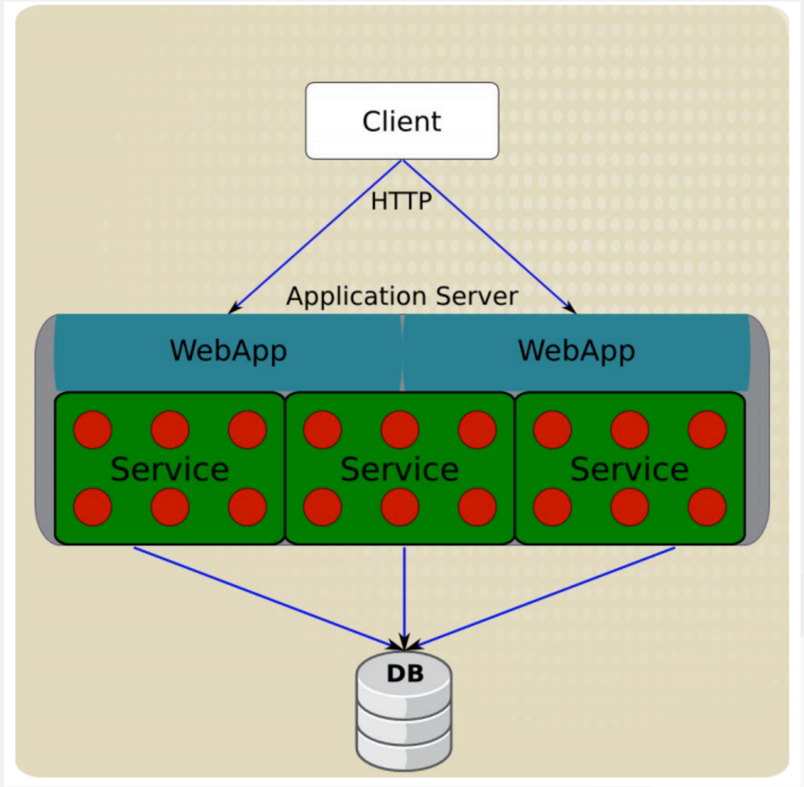
- Devops
- Microservices
- Application platforms
- Fast data

What’s the problem?

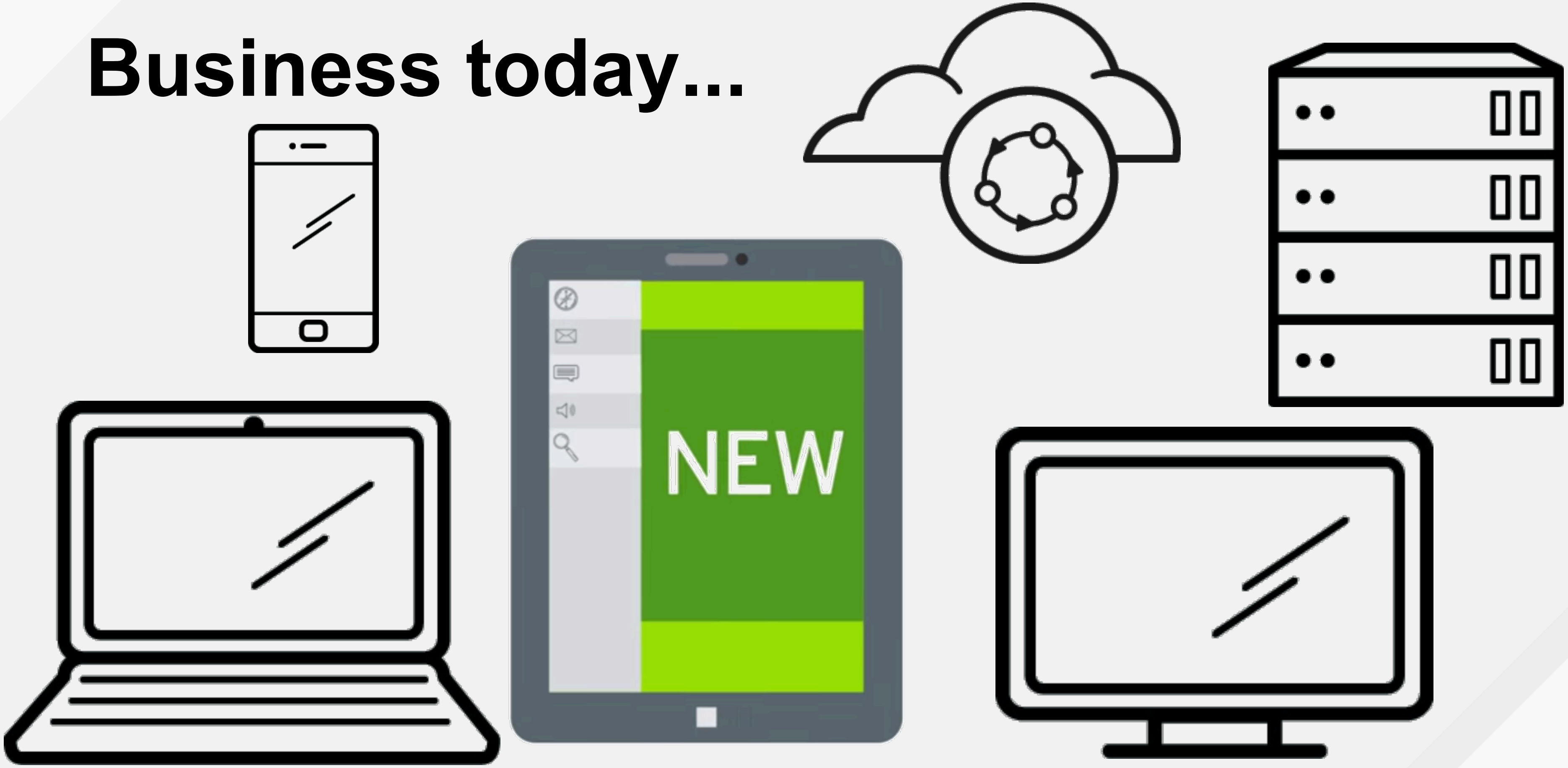
Business back in the day...

Monolithic architecture

- Logically modular, deployed as monolith
- Great for small apps, small teams
- Simple systems
  - Dev, test, deploy
- Optimized for efficiency and latency



Business today...



What’s the problem?

- More, more, more
  - Customers demand more
  - Business demands more
  - FAST!
- But wait!!!
  - Enterprise stability, reliability
  - Predictable processes
  - Manage risk, technical debt

- Large complex monolithic apps
- Difficult maintenance
- Unwieldy
- Hard to test, trial new tech, fix
- Stuck with the original app

The goal?

- Meet the needs of modern business
    - interconnected
    - immediate
    - high variety of customer touch points
    - data from many sources
    - engaging
    - Personalized
- Experiment, fail fast
- Mobility

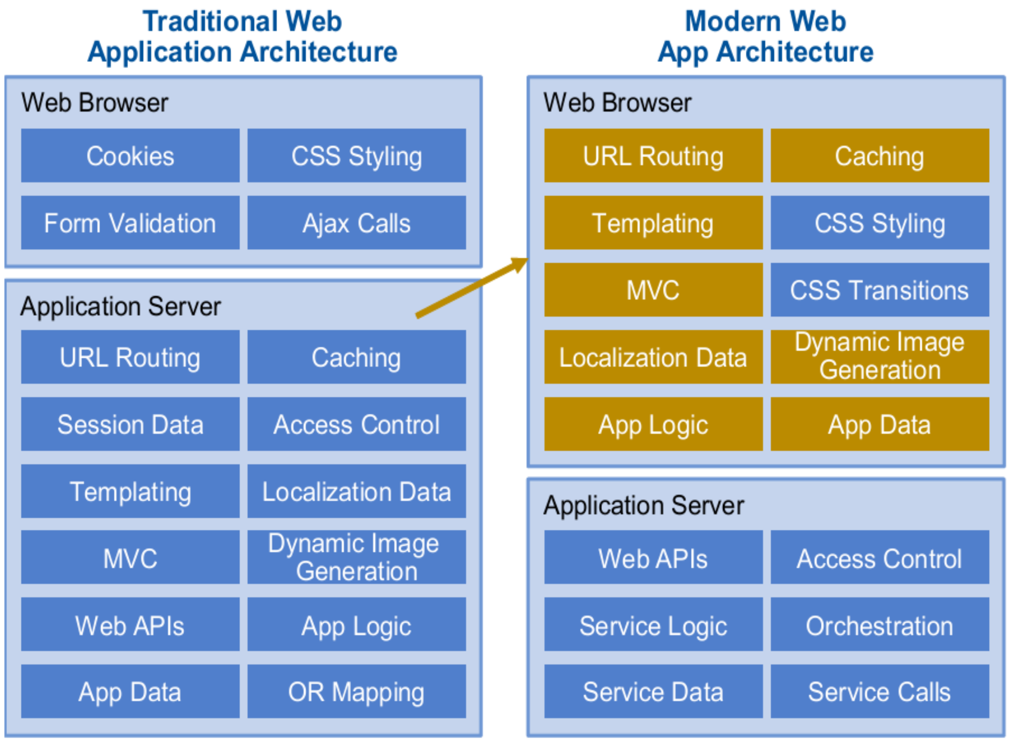
The solution

- Scales out (on x or z axis?)
- Fast to update, refresh
- Hybrid cloud
- Developers can innovate
- Operations can manage, maintain
- Easy to upgrade
- Easy to isolate problems
- Potentially polyglot
- Enables cross team communication

...the "modern application"

A new architecture for a modern business

Figure 2. Comparing Traditional and Modern Web Architectures



Source: Gartner (October 2014)

More than just new approaches to applications...

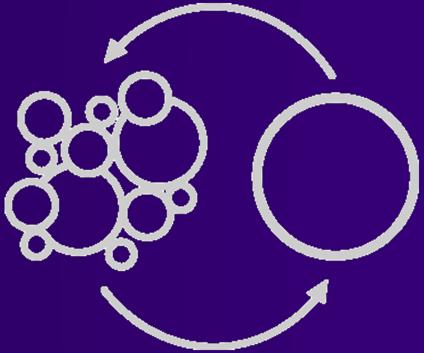
Processes

Architectures

Languages

Platforms

Technologies

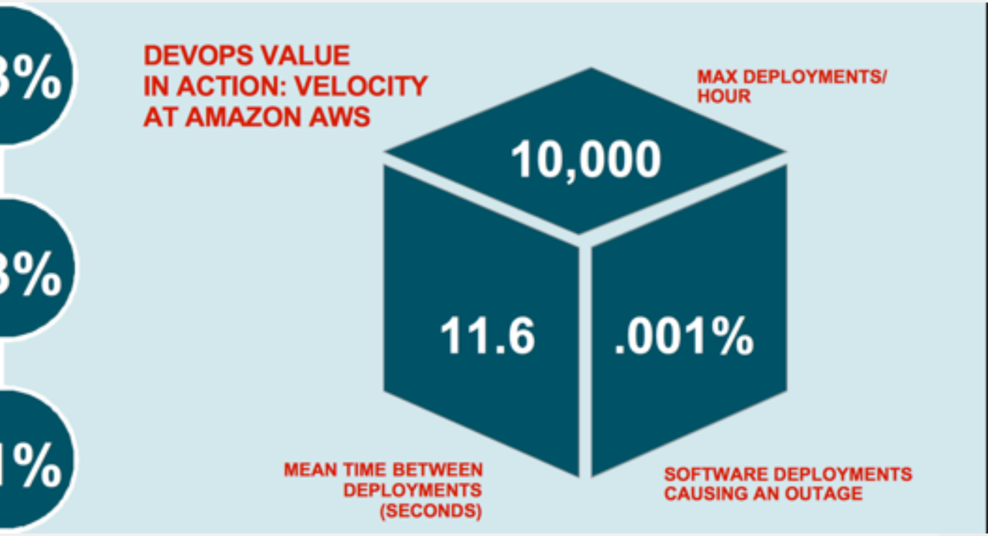


Processes

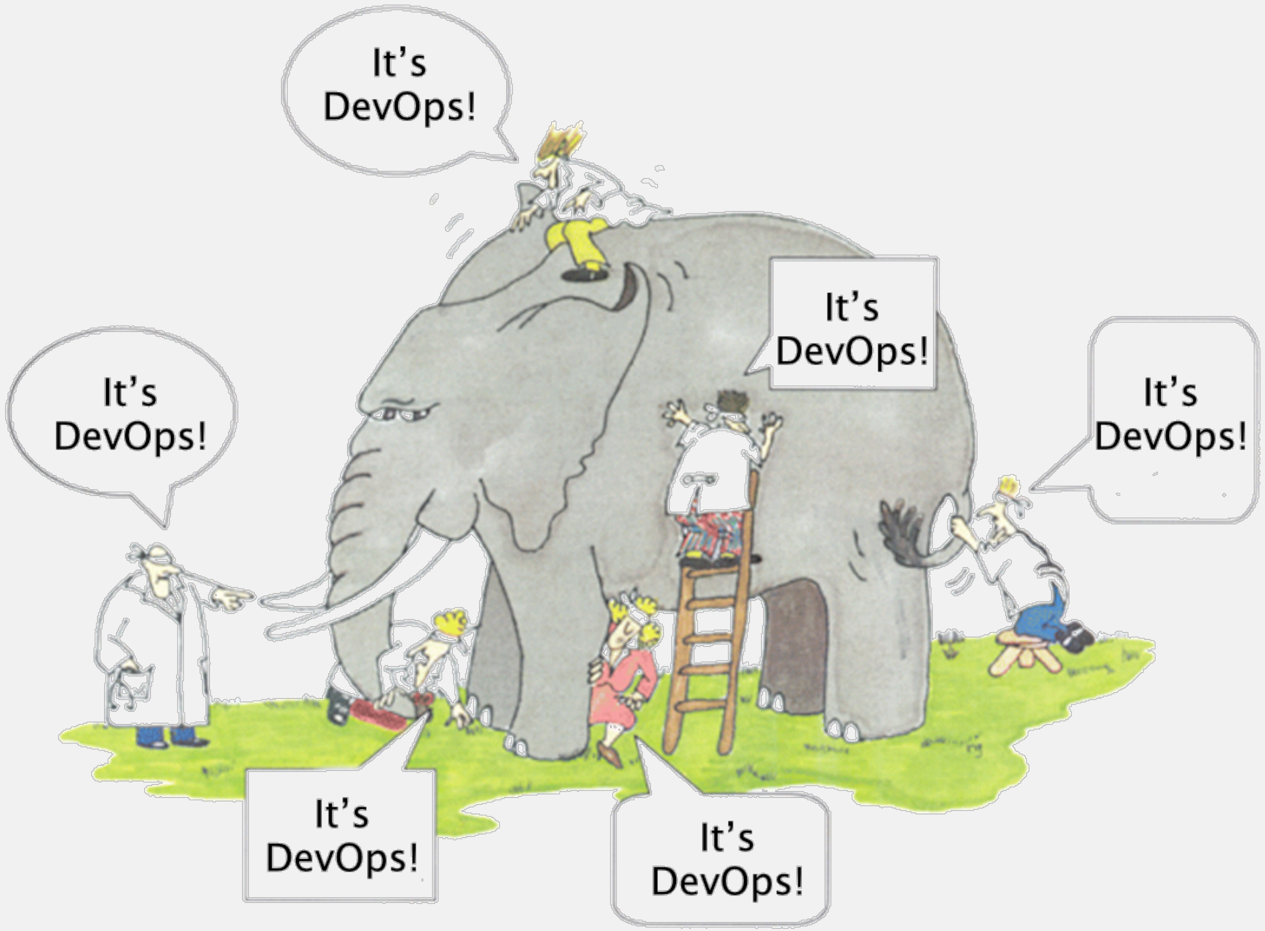
Better deployment quality 63%

Faster release frequency 63%

Improved process visibility 61%


DEVOPS VALUE IN ACTION: VELOCITY AT AMAZON AWS
MAX DEPLOYMENTS/ HOUR
10,000
11.6
.001%
MEAN TIME BETWEEN DEPLOYMENTS (SECONDS)
SOFTWARE DEPLOYMENTS CAUSING AN OUTAGE

WORKED FINE IN DEV
OPS PROBLEM NOW
memegenerator.net


It's DevOps!
It's DevOps!
It's DevOps!
It's DevOps!
It's DevOps!
It's DevOps!

What is DevOps?

DevOps is an approach to process, culture, and tools for delivering increased business value and responsiveness through rapid, iterative, and high-quality IT service delivery.

STUDIES
GAP YEAR
TRAVEL
MARRIAGE
KIDS
DIVORCE
I'm BACK!
DIRTY LAUNDRY

“Dev”
“Ops”
ied

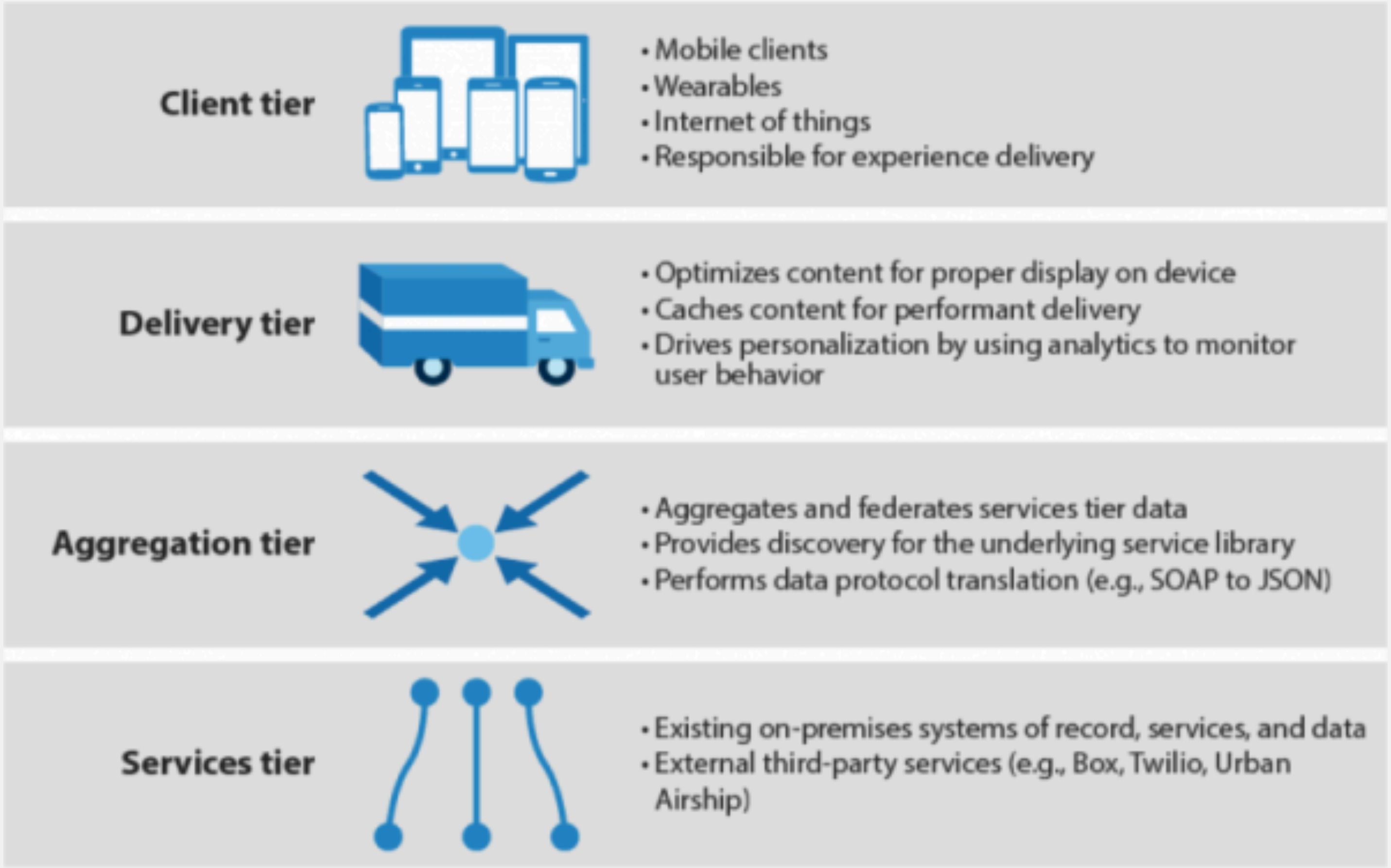
Five "C"s of DevOps

- Collaboration between "dev" and "ops"
- Culture
- Code everything - application and configuration
- Consistency - automation over documentation
- Continuous delivery

	Initial	Managed	Defined	Quantitatively Managed	Optimizing
Culture & Organization	• Teams organized based on platform/ technology • Defined and documented processes	• One backlog per team • Adopt agile methodologies • Remove team boundaries	• Extended team collaboration • Remove boundary dev/ ops • Common process for all changes	• Cross-team continuous improvement • Teams responsible all the way to production	• Cross functional teams
Build & Deploy	• Centralized version control • Automated build scripts • No management of artifacts • Manual deployment • Environments are manually provisioned	• Polling CI builds • Any build can be re-created from source control • Management of build artifacts • Automated deployment scripts • Automated provisioning of environments	• Commit hook CI builds • Build fails if quality is not met (code analysis, performance, etc.) • Push button deployment and release of any releasable artifact to any environment • Standard deployment process for all environments	• Team priorities keeping codebase deployable over doing new work • Builds are not left broken • Orchestrated deployments • Blue Green Deployments	• Zero touch Continuous Deployments
Release	• Infrequent and unreliable releases • Manual process	• Painful infrequent but reliable releases	• Infrequent but fully automated and reliable releases in any environment	• Frequent fully automated releases • Deployment disconnected from release • Canary releases	• No rollbacks, always roll forward
Data Management	• Data migrations are performed manually, no scripts	• Data migrations using versioned scripts, performed manually	• Automated and versioned changes to datastores	• Changes to datastores automatically performed as part of the deployment process	• Automatic datastore changes and rollbacks tested with every deployment
Test & Verification	• Automated unit tests • Separate test environment	• Automatic Integration Tests • Static code analysis • Test coverage analysis	• Automatic functional tests • Manual performance/ security tests	• Fully automatic acceptance tests • Automatic performance/security tests • Manual exploratory testing based on risk based testing analysis	• Verify expected business value • Defects found and fixed immediately (roll forward)
Information & Reporting	• Baseline process metrics • Manual reporting • Visible to report runner	• Measure the process • Automatic reporting • Visible to team	• Automatic generation of release notes • Pipeline traceability • Reporting history • Visible to cross-silo	• Report trend analysis • Real time graphs on deployment pipeline metrics	• Dynamic self-service of information • Customizable dashboards • Cross-reference across organizational boundaries

http://blog.arungupta.me/continuous-integration-delivery-deployment-maturity-model/

Architectures



Client tier

- Mobile clients
- Wearables
- Internet of things
- Responsible for experience delivery

Delivery tier

- Optimizes content for proper display on device
- Caches content for performant delivery
- Drives personalization by using analytics to monitor user behavior

Aggregation tier

- Aggregates and federates services tier data
- Provides discovery for the underlying service library
- Performs data protocol translation (e.g., SOAP to JSON)

Services tier

- Existing on-premises systems of record, services, and data
- External third-party services (e.g., Box, Twilio, Urban Airship)

http://blogs.forrester.com/ted_schadler/13-11-20-mobile_needs_a_four_tier_engagement_platform

"America is all about speed. Hot, nasty badass speed."

An architectural approach, that emphasizes the decomposition of applications into single-purpose, loosely coupled services managed by cross-functional teams, for delivering and maintaining complex software systems with the velocity and quality required by today's digital business

Business-driven microservices

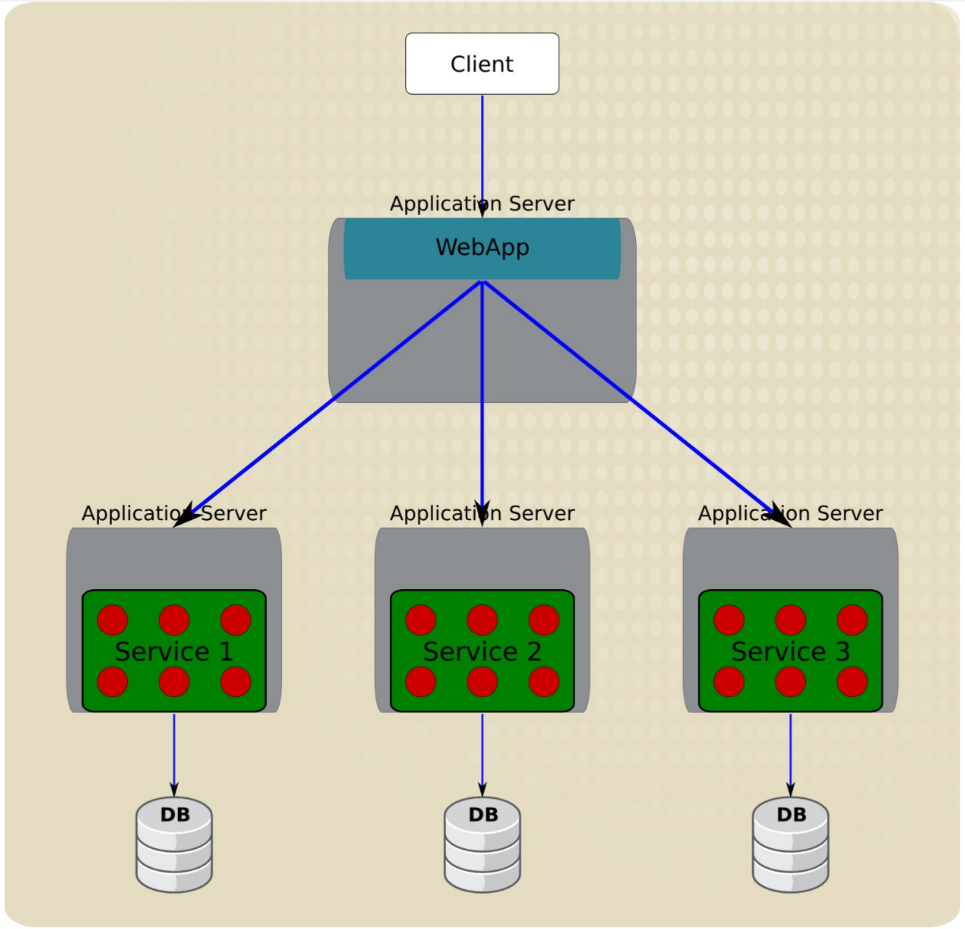


Figure 2.3.5-1: Business-Driven Microservices

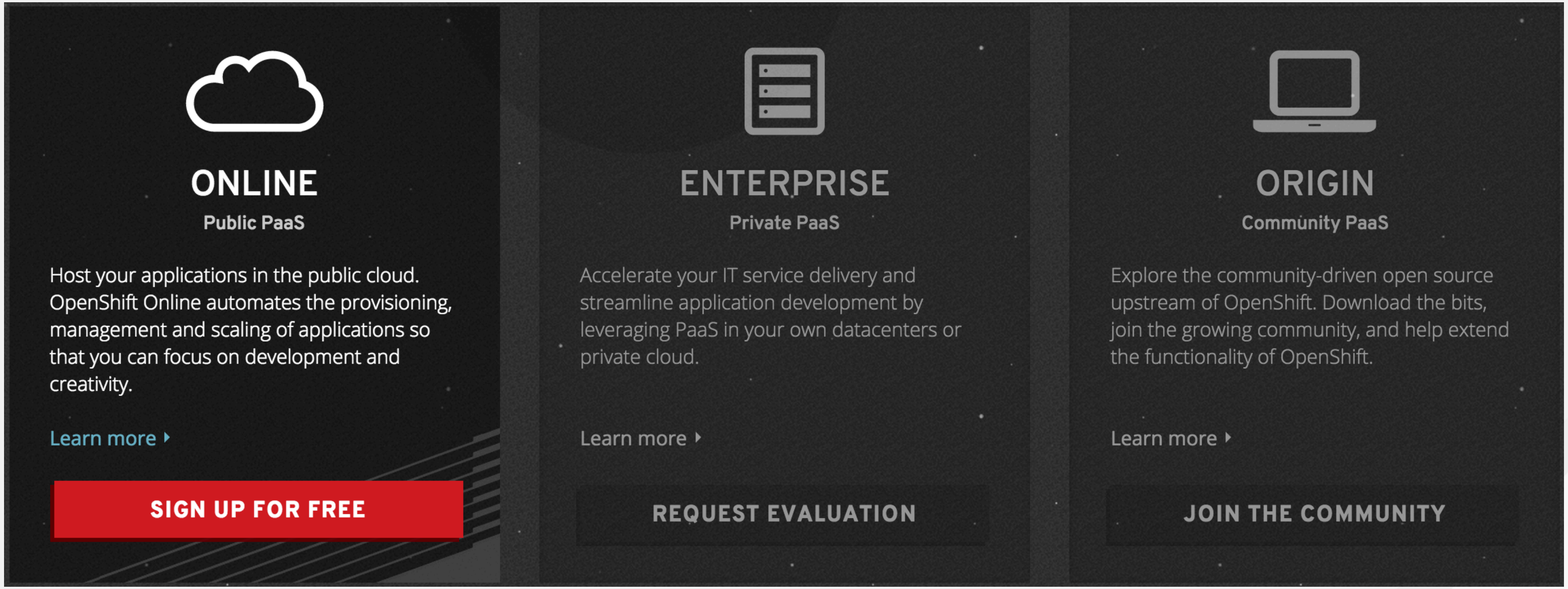
Platforms

Technologies and tools

Languages

Platforms

PaaS


ONLINE

Public PaaS

Host your applications in the public cloud. OpenShift Online automates the provisioning, management and scaling of applications so that you can focus on development and creativity.

Learn more ›

SIGN UP FOR FREE

ENTERPRISE

Private PaaS

Accelerate your IT service delivery and streamline application development by leveraging PaaS in your own datacenters or private cloud.

Learn more ›

REQUEST EVALUATION

ORIGIN

Community PaaS

Explore the community-driven open source upstream of OpenShift. Download the bits, join the growing community, and help extend the functionality of OpenShift.

Learn more ›

JOIN THE COMMUNITY

php

Java

ceylon

Scala

JS

JRuby

Groovy

ORACLE
DATABASE

MySQL

PostgreSQL

mongoDB

Couchbase

redis

cassandra

Technologies and tools

CHEF SALTSTACK

puppet labs ANSIBLE

docker

Rocket

cruisecontrol

Jenkins

Travis CI

shippable

logstash

elasticsearch kibana

kubernetes mesosphere

VERT.X

MEAN.IO

node JS

METEOR

MEAN.JS

WildFly Apache TomEE

Moving forward…

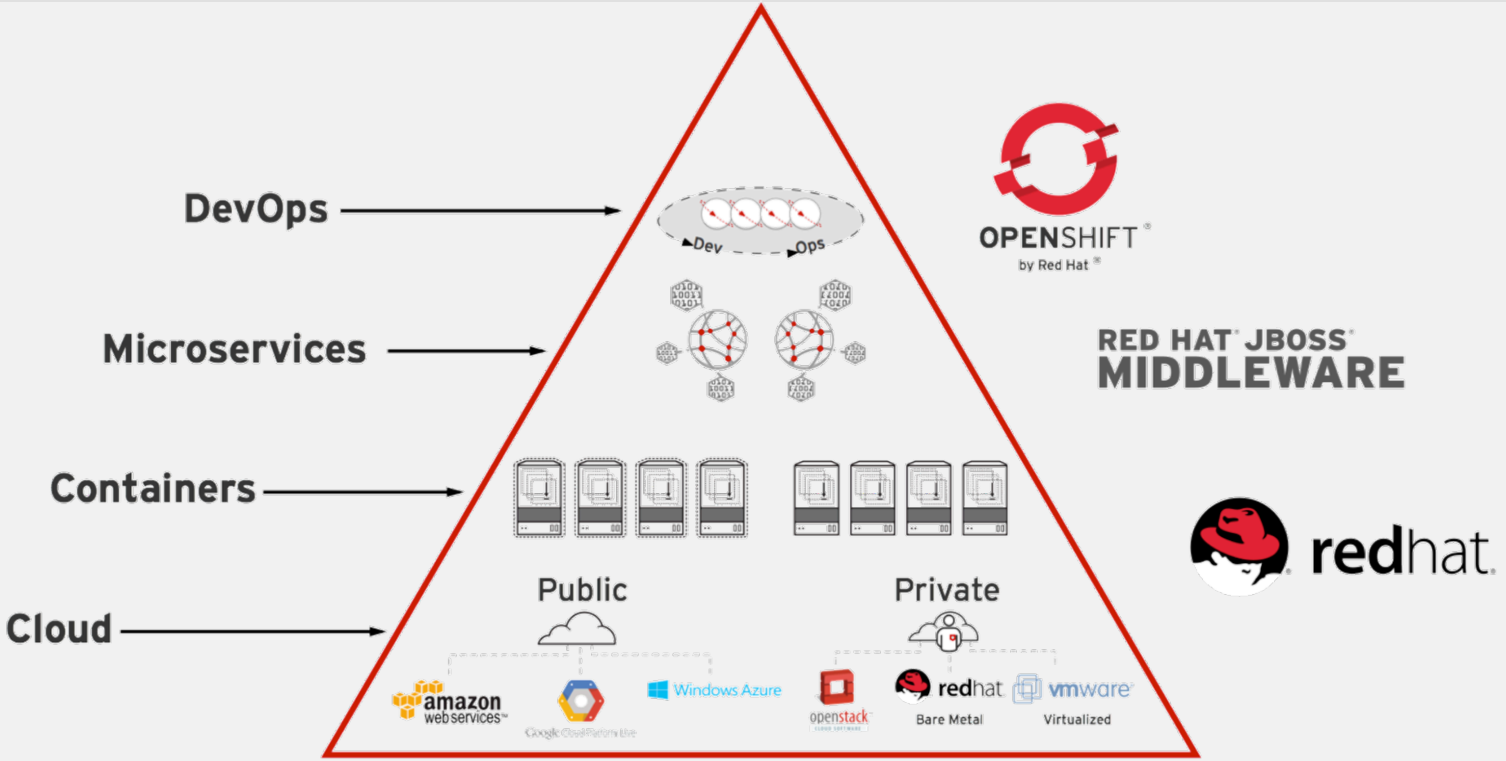
The effect on development teams

- What changes to implement?
- What scope?
- Timing?
- Who?
- Expectations?
- Tradeoffs?

Some guidance needed?

- Prioritize devops
- Evaluate needs according to the maturity matrix
- Determine tools, tech, etc
- Microservices – proceed with care!

Bringing it all together



redhat

RED HAT SUMMIT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.