Pivotal

Cloud Native Architecture

Reshmi Krishna

Cloud Native Architecture

Pivotal

Cloud Native Architecture

- Cloud Computing
- New Demands
- Being Reactive
- Speed & Safety

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Cloud Native Architecture

- Cloud Computing
- New Demands
- Being Reactive
- Speed & Safety

- Reactive Design
- 12 Factor Apps
- Microservices
- Design Patterns

Pivotal

Cloud Native Architecture

- Cloud Computing
- New Demands
- Being Reactive
- Speed & Safety

- Reactive Design
- 12 Factor Apps
- Microservices
- Design Patterns

- Tools
- Platform
- Process and Culture

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Cloud Native Architecture

- Cloud Computing
- New Demands
- Being Reactive
- Speed & Safety

Pivotal

Cloud Defined

- Cloud Computing
- Cloud Native Applications
- Cloud Native Application Architecture

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Cloud-Era Demands

- Competitive landscape
- Consumer expectations
- Device Diversity & Ubiquity
- Scale

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Being Reactive

- React to Events
- React to Load
- React to Failure
- React to Users

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Speed & Safety

- Design for quick recovery
- Quick recovery increases risk tolerance
- Risk tolerance enables innovation

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Cloud Native Architecture

- Reactive Design
- 12 Factor Apps
- Microservices
- Design Patterns

Pivotal

Reactive Design

- React to Events → Event-Driven
- React to Load → Scalable
- React to Failure → Resilient
- React to Users → Responsive

http://www.reactivemanifesto.org

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

The Twelve Factor App

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

http://12factor.net

Pivotal

The Twelve Factor App

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

http://12factor.net

Pivotal

The Twelve Factor App

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

http://12factor.net

Pivotal

The Twelve Factor App

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

http://12factor.net

Pivotal

The Twelve Factor App

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

http://12factor.net

Pivotal

The Twelve Factor App

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

http://12factor.net

Pivotal

The Twelve Factor App

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

http://12factor.net

Pivotal

Microservices

If services have to be updated together, they're not loosely coupled!

Loosely coupled service oriented architecture with bounded contexts

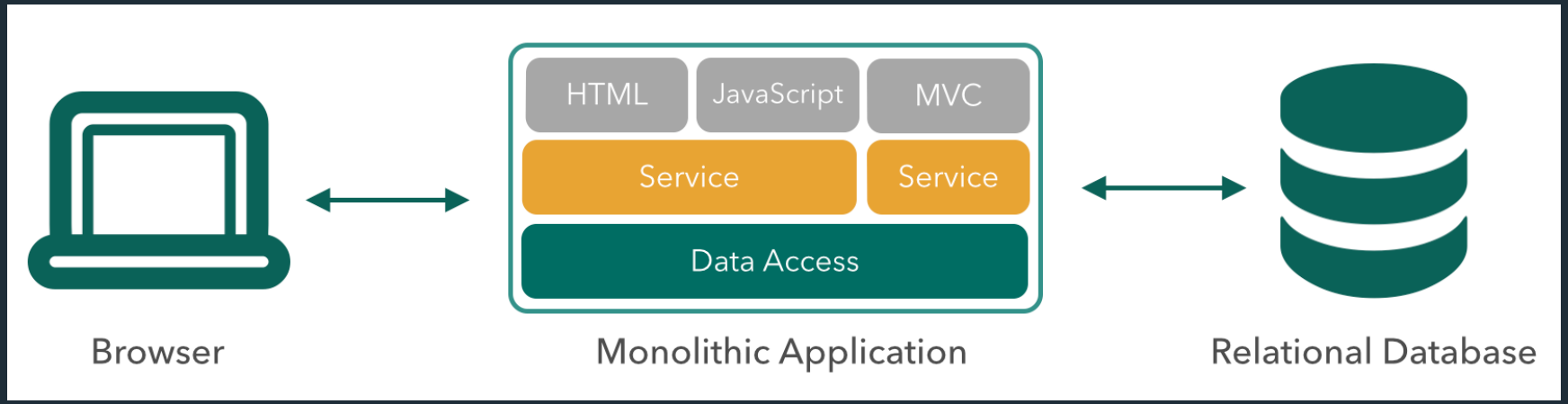
- Adrian Cockcroft

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

Pivotal

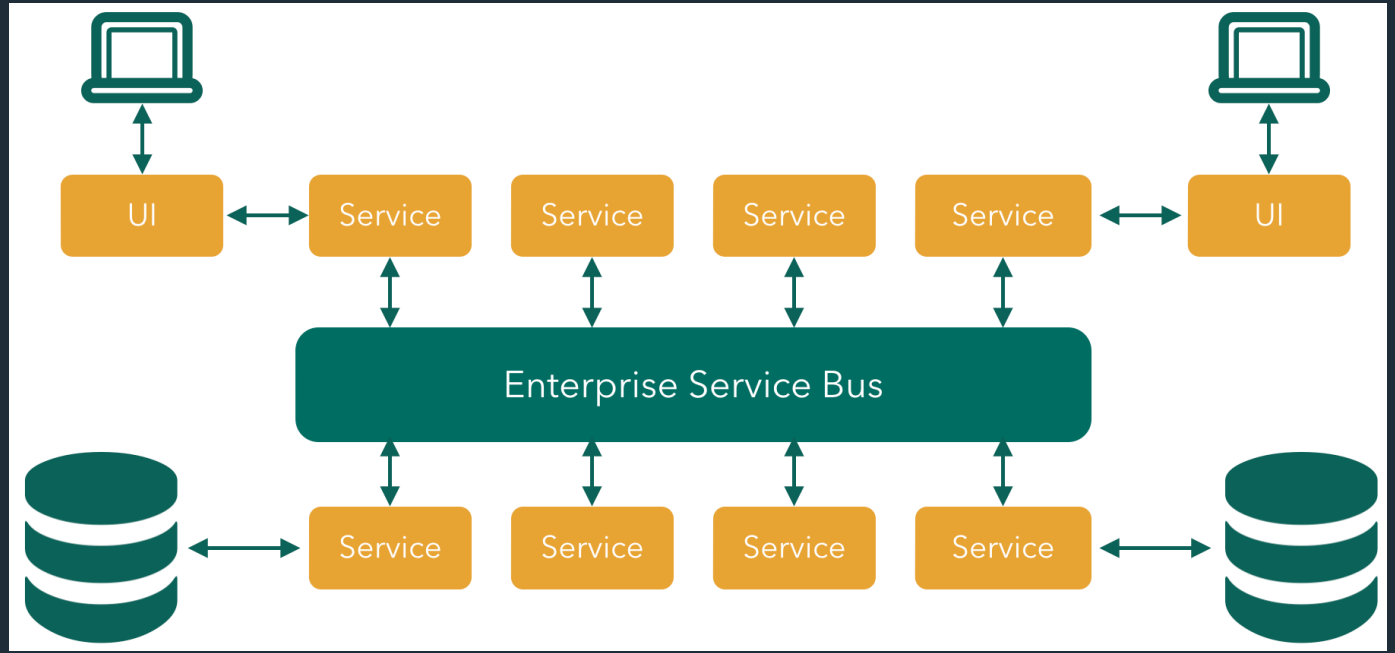
© Copyright 2014 Pivotal. All rights reserved.

Not Monoliths



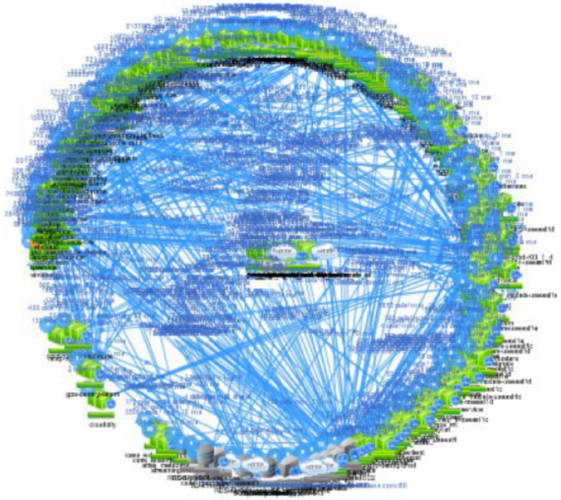
Pivotal

Not ESB-Centric SOA

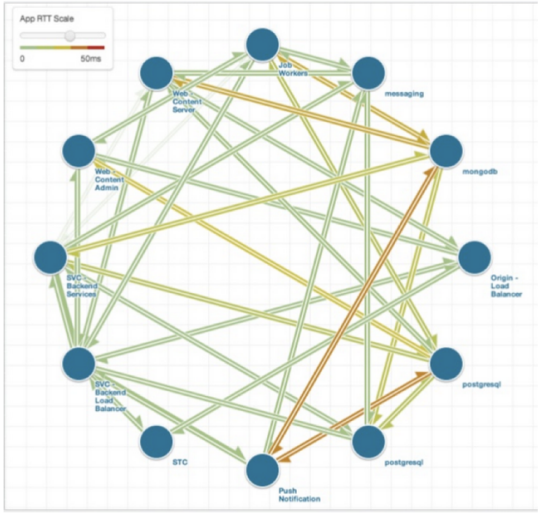


Pivotal

Microservice Pioneers



Netflix



Gilt Groupe (12 of 450)

Twitter

Who Moved My Complexity?

- Microservices are individually simple
- Complexity is transferred to the ecosystem

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

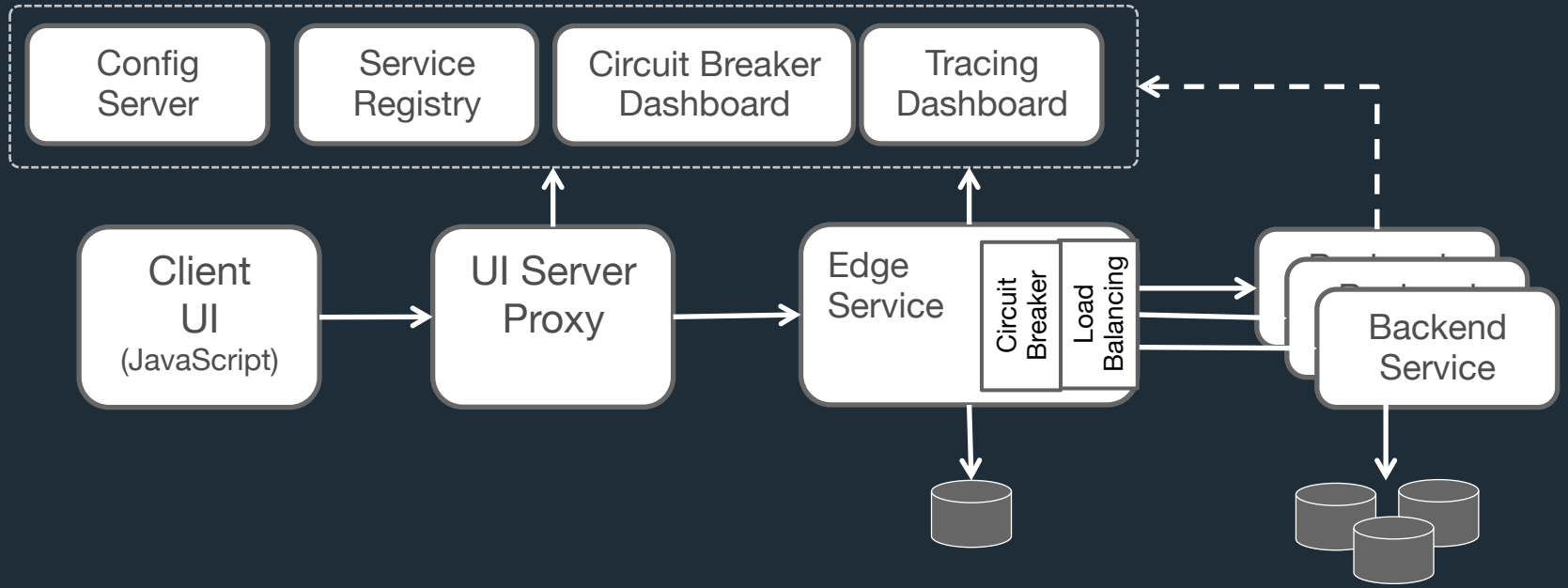
Challenges in a Distributed System

- Configuration management
- Registration and discovery
- Routing and load balancing
- Fault tolerance and isolation
- Aggregation and transformation
- Monitoring and distributed tracing
- Process management

Pivotal

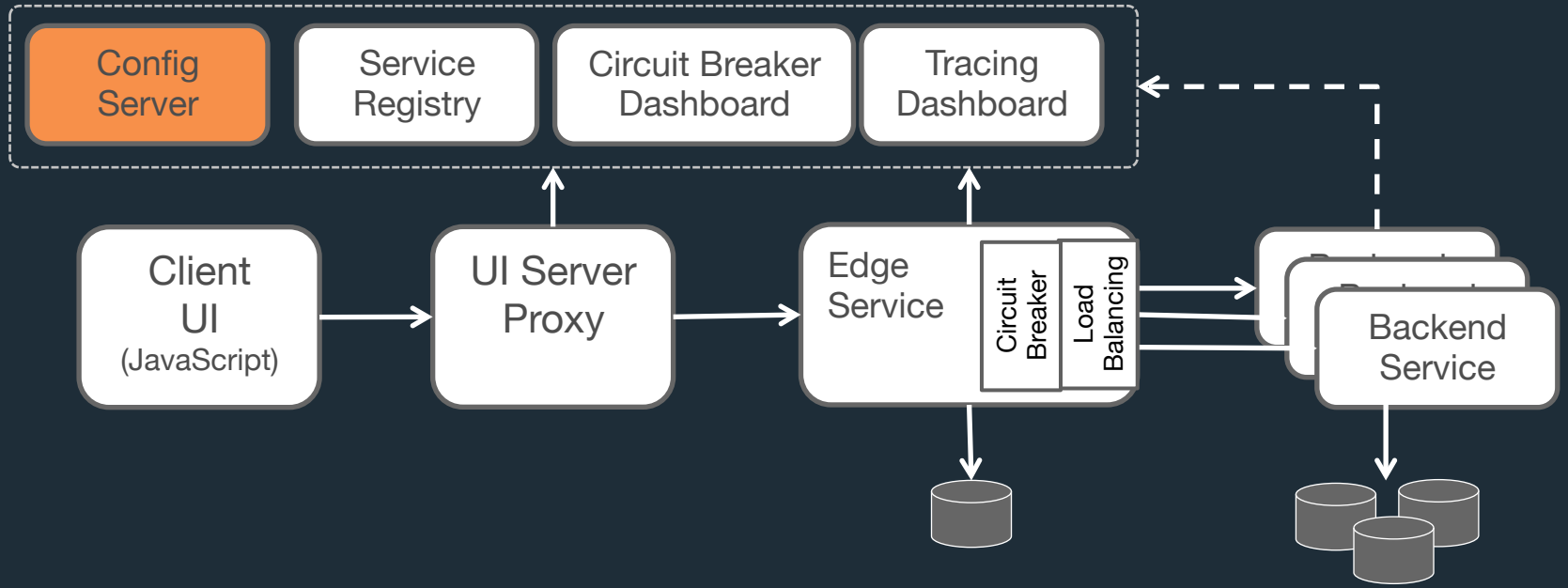
© Copyright 2014 Pivotal. All rights reserved.

Design Patterns & Automation to the Rescue!



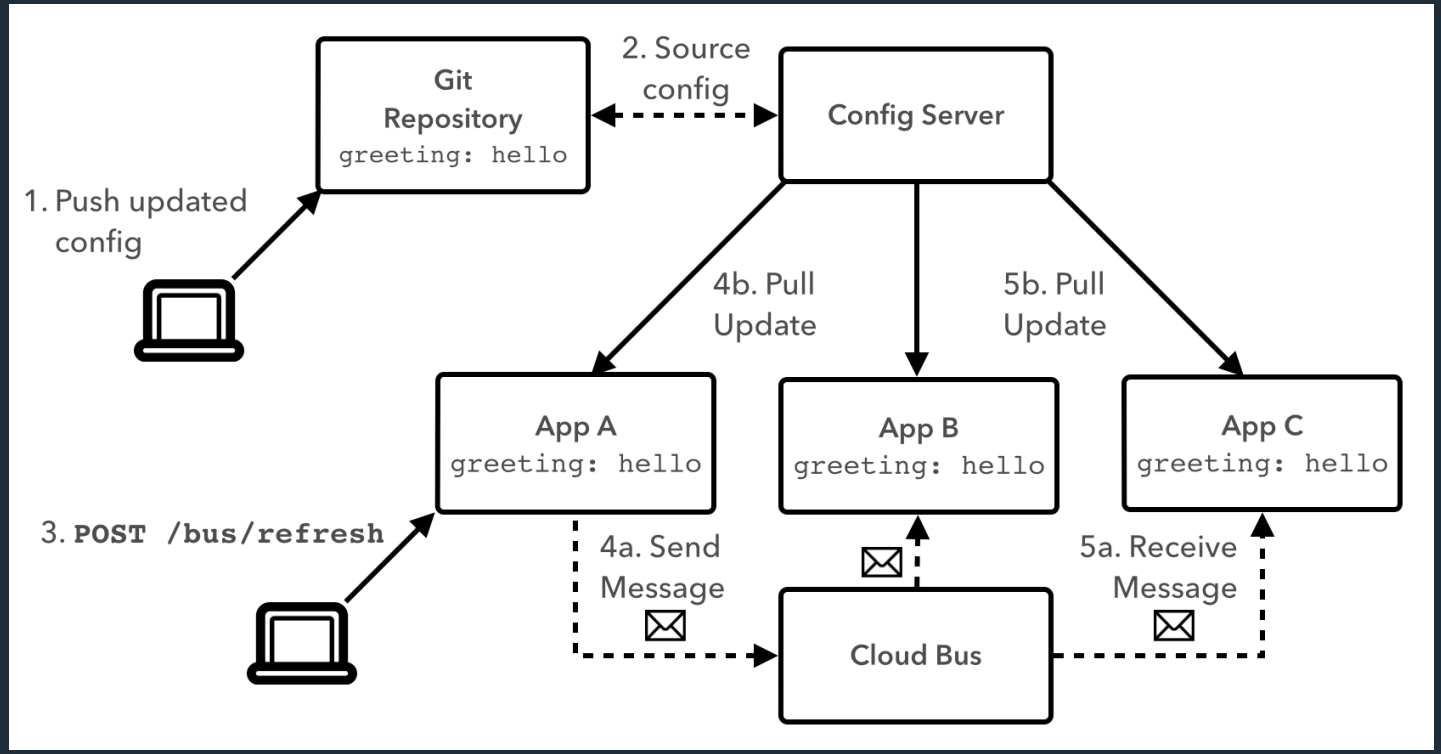
Pivotal

Configuration Management

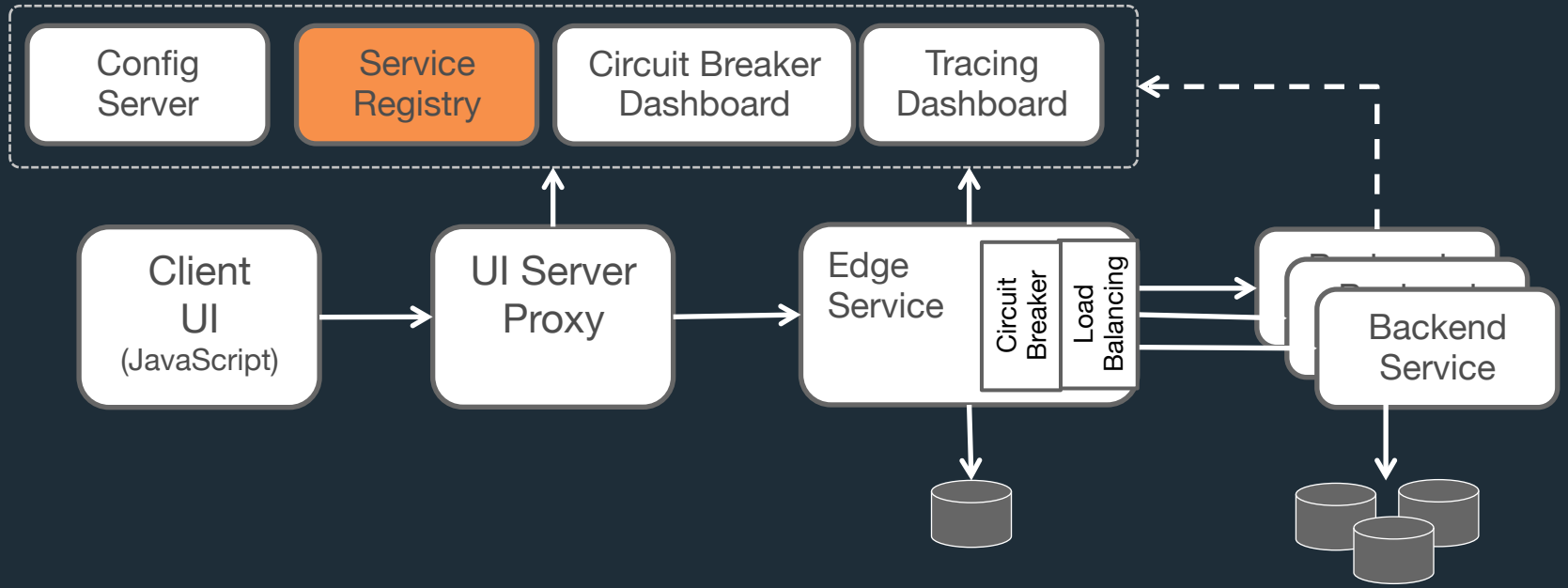


Pivotal

Configuration Server



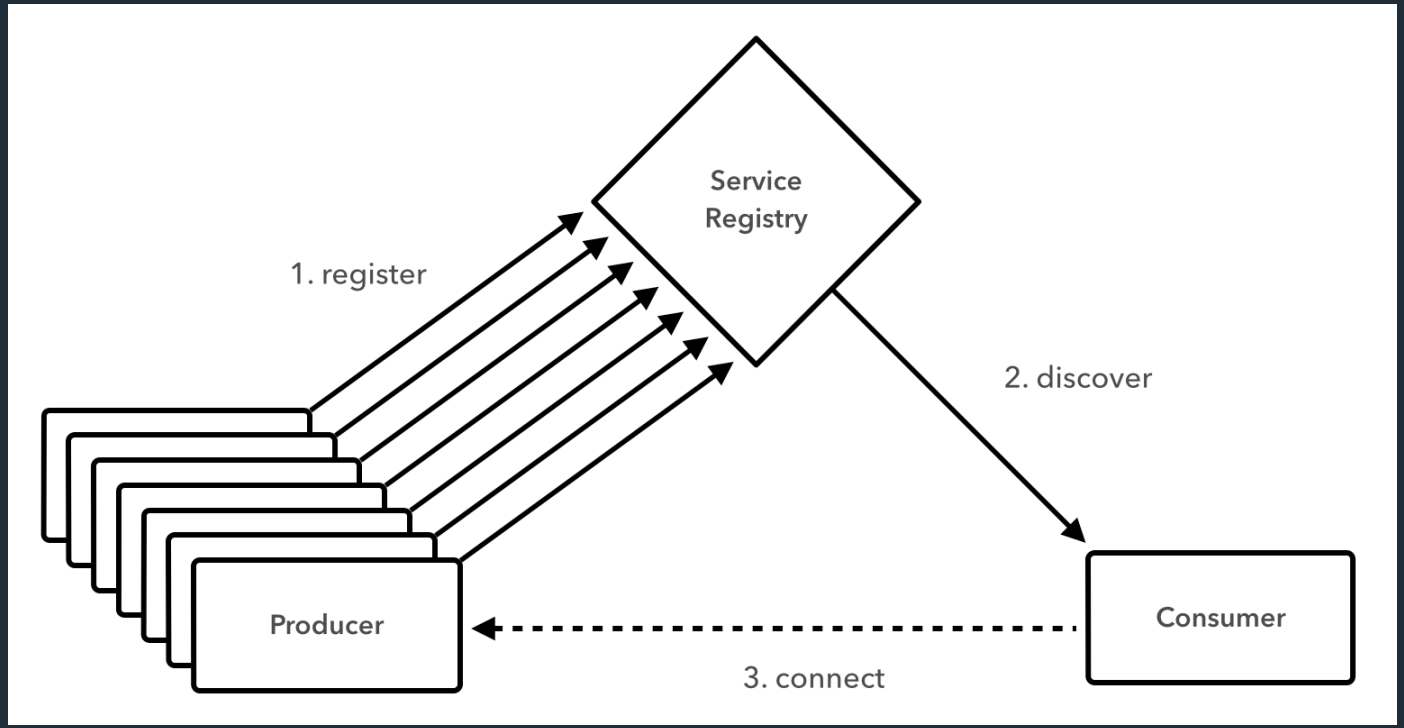
Registration and Discovery



Pivotal

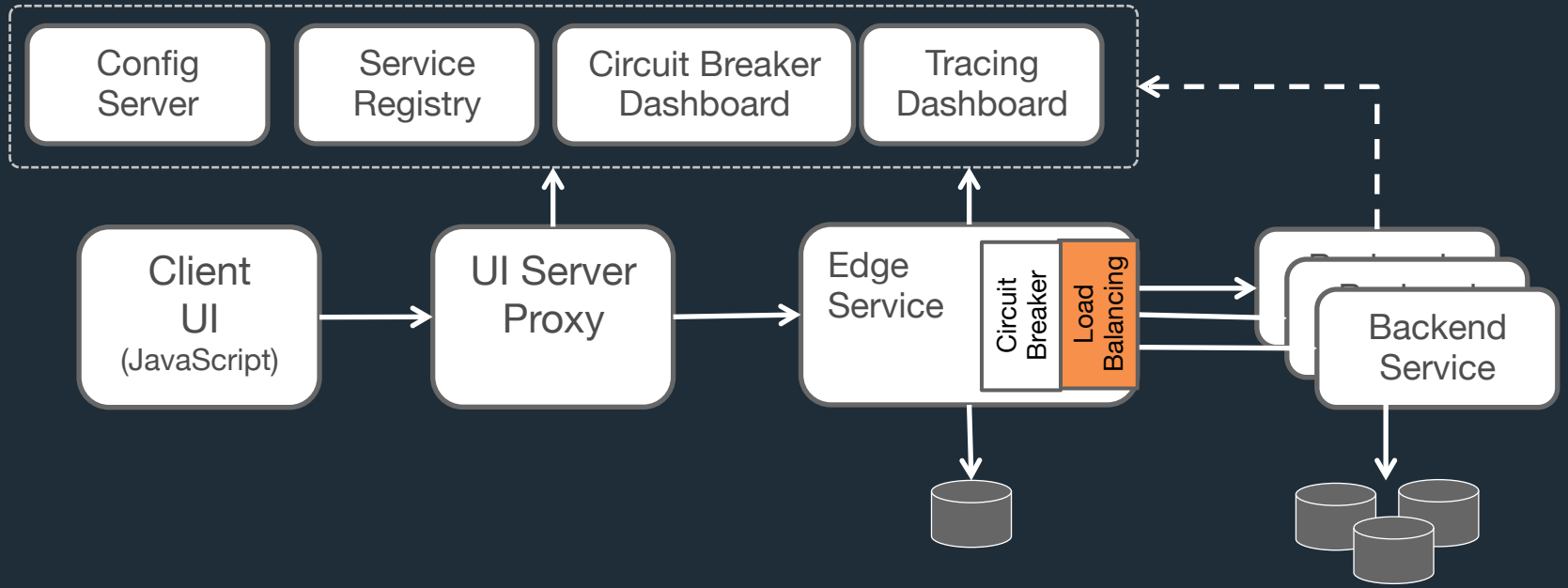
© Copyright 2014 Pivotal. All rights reserved.

Registration and Discovery Server



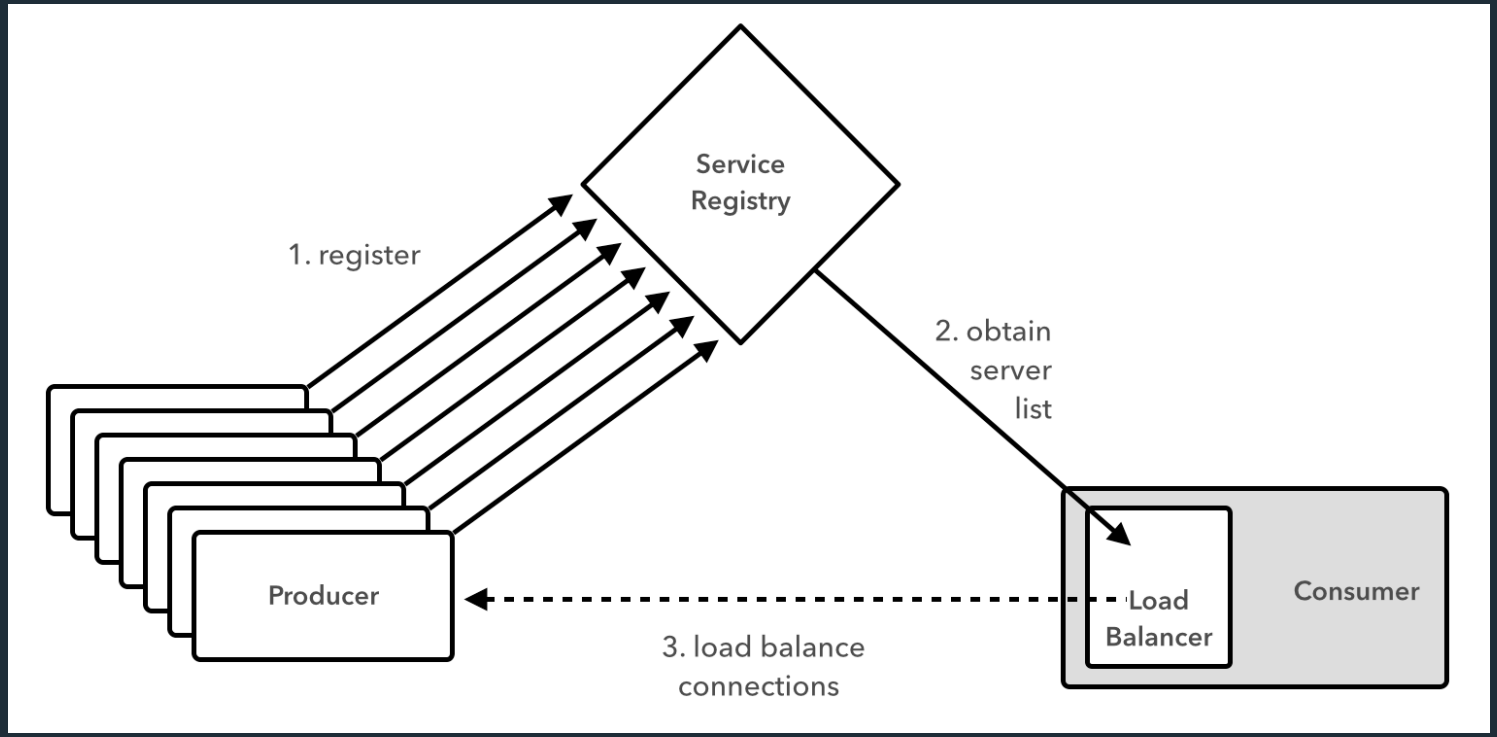
Pivotal

Routing and Load Balancing



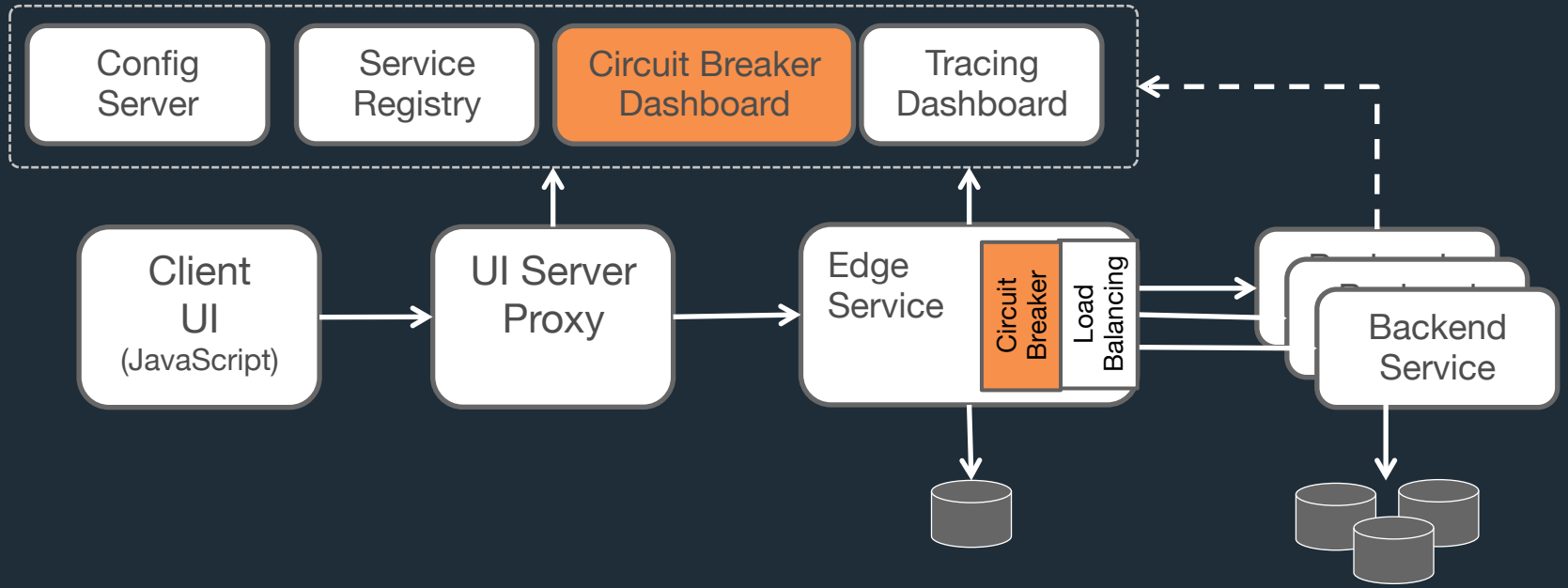
Pivotal

Client-Side Load Balancing



Pivotal

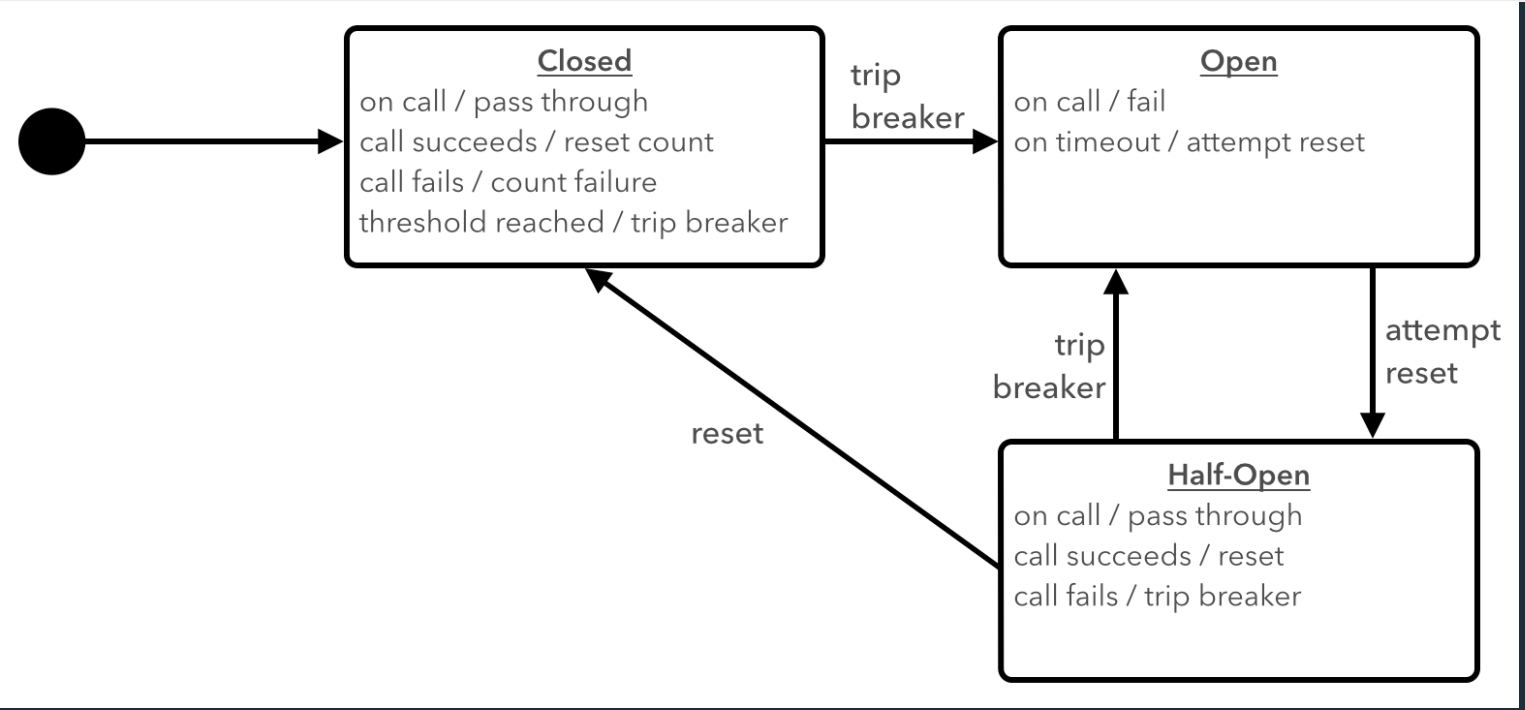
Fault Tolerance and Isolation



Pivotal

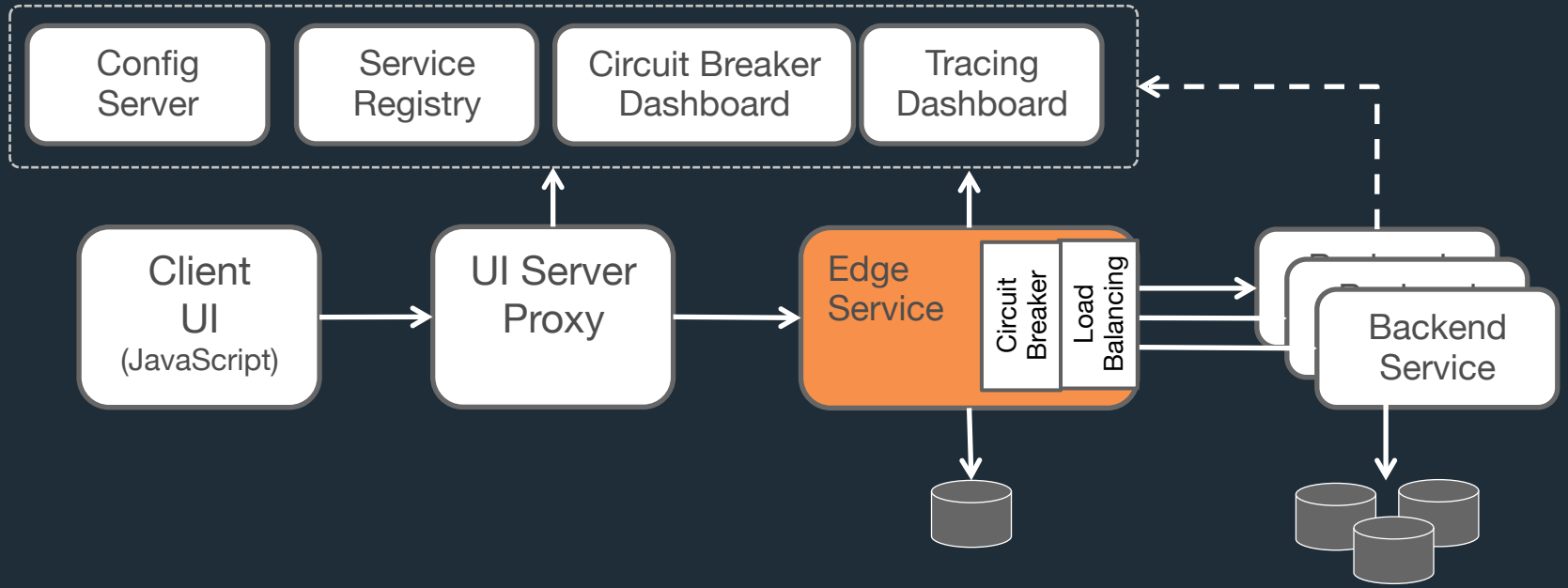
© Copyright 2014 Pivotal. All rights reserved.

Circuit Breaker



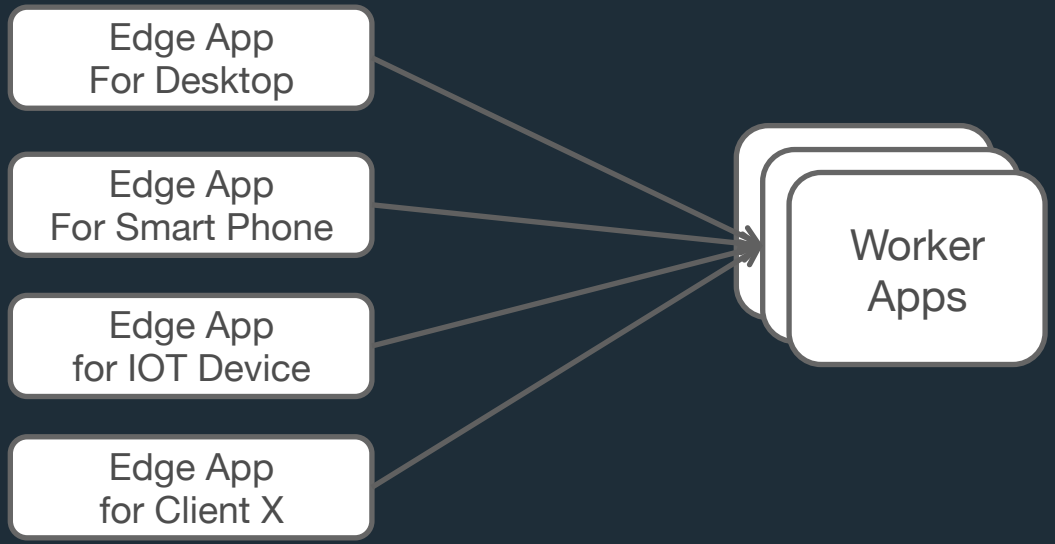
Pivotal

Aggregation and Transformation



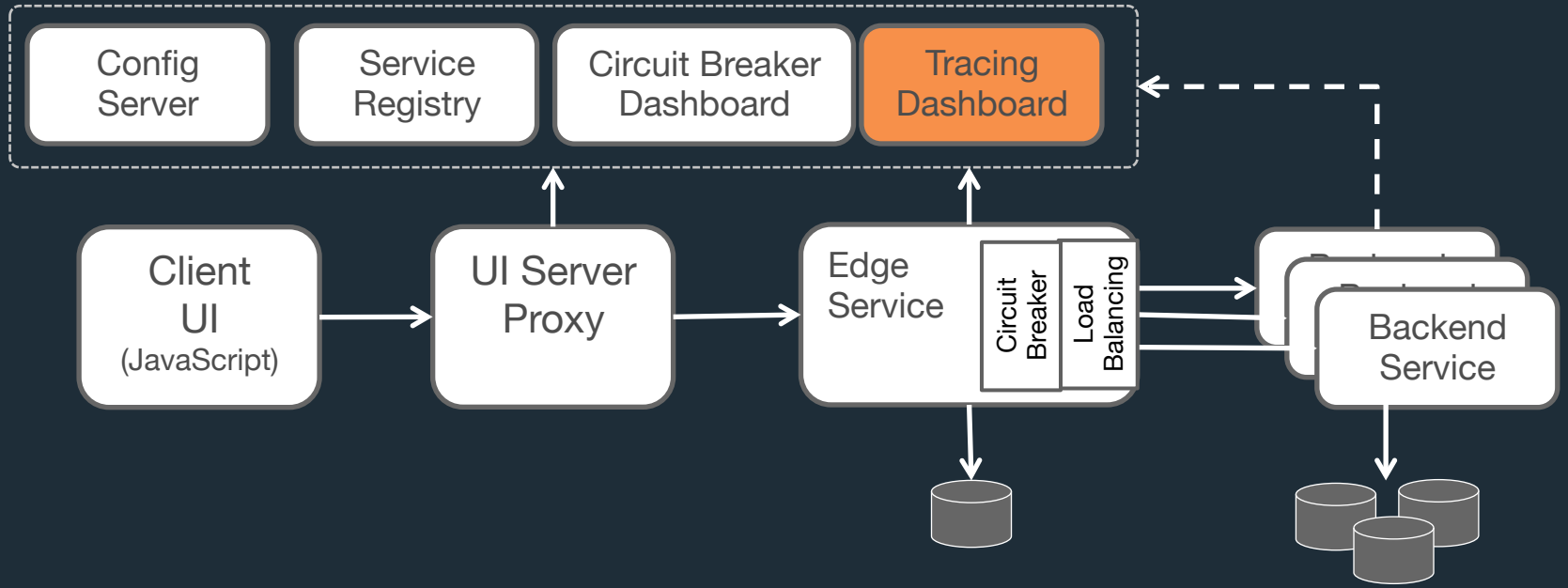
Pivotal

API Gateway



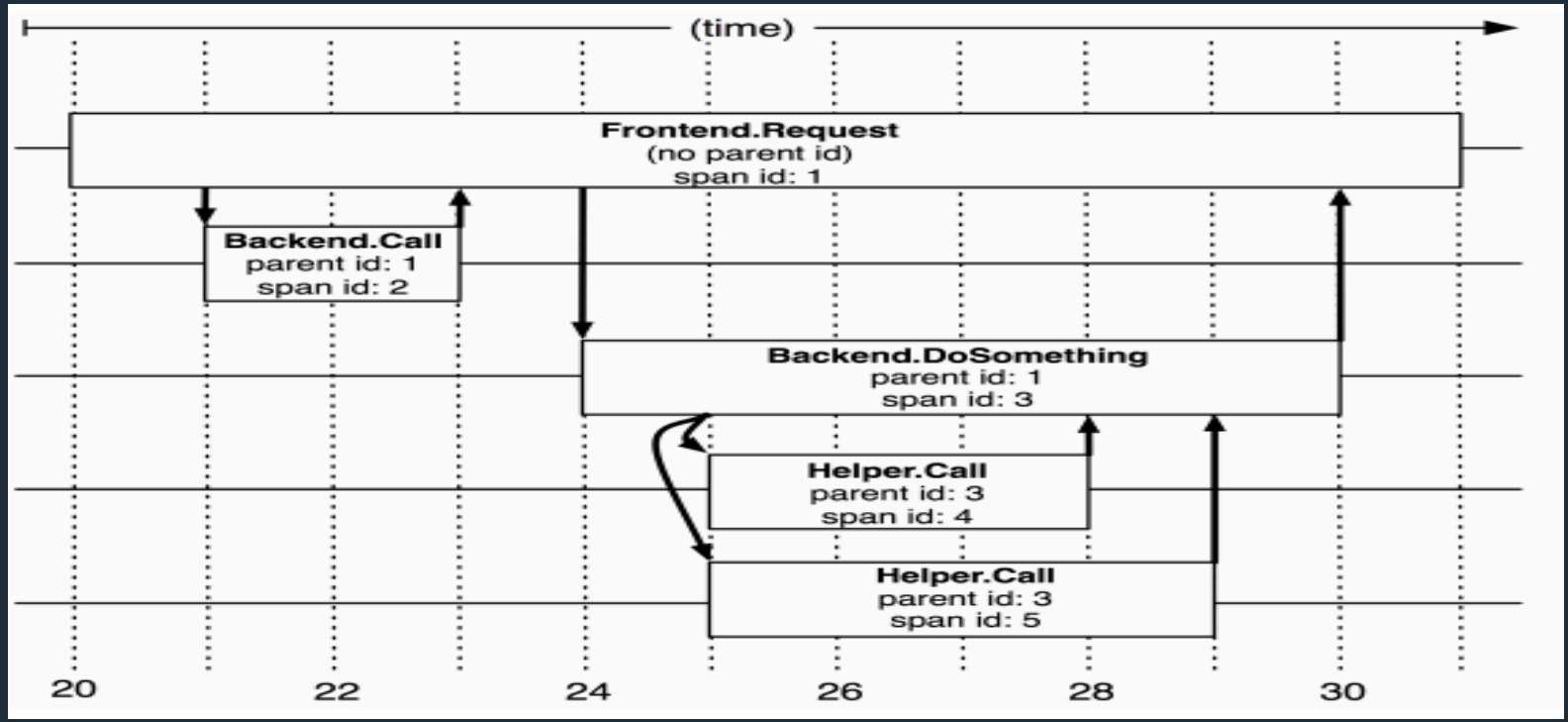
Pivotal

Monitoring and Distributed Tracing

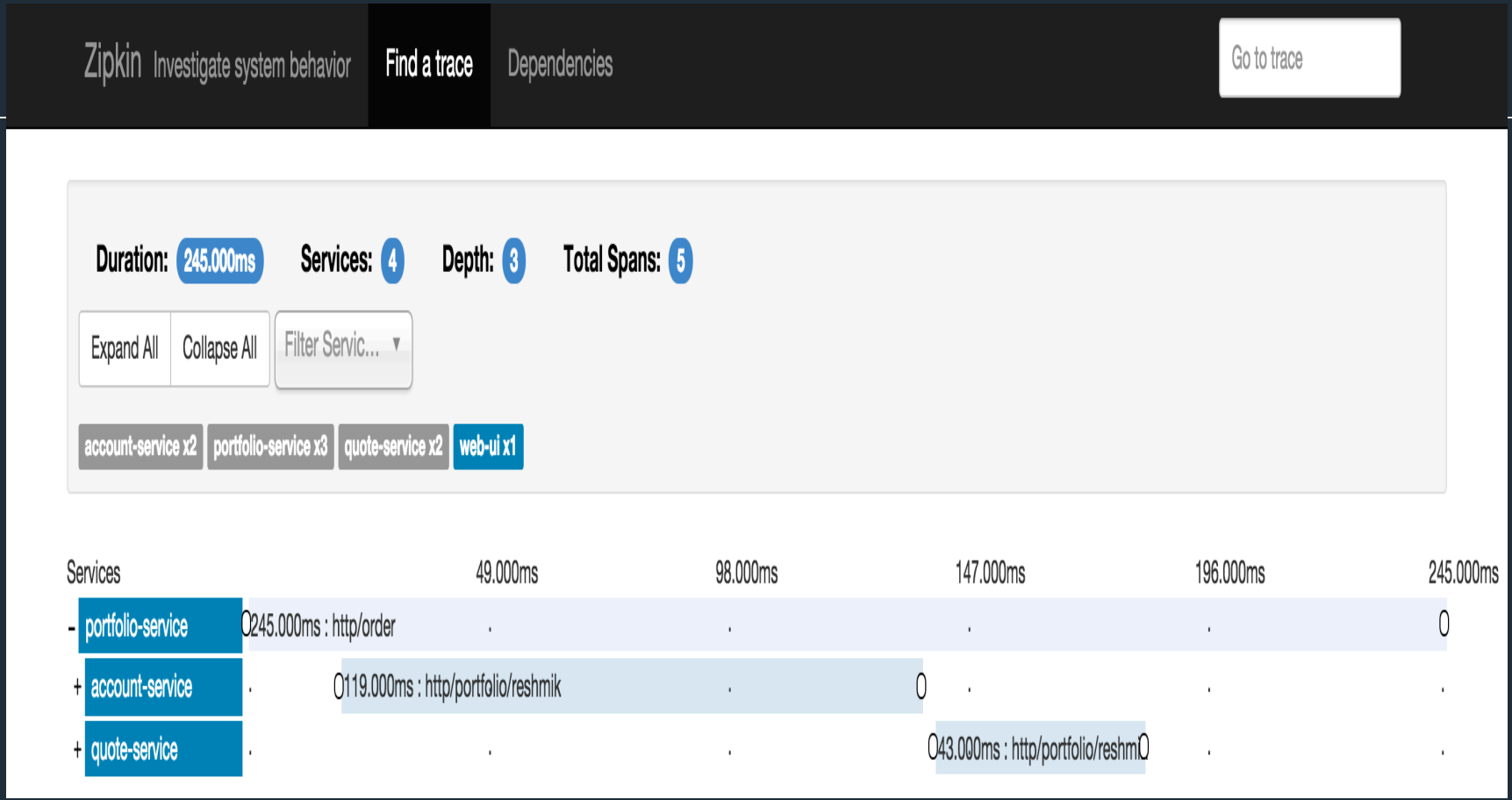


Pivotal

Distributed Tracing

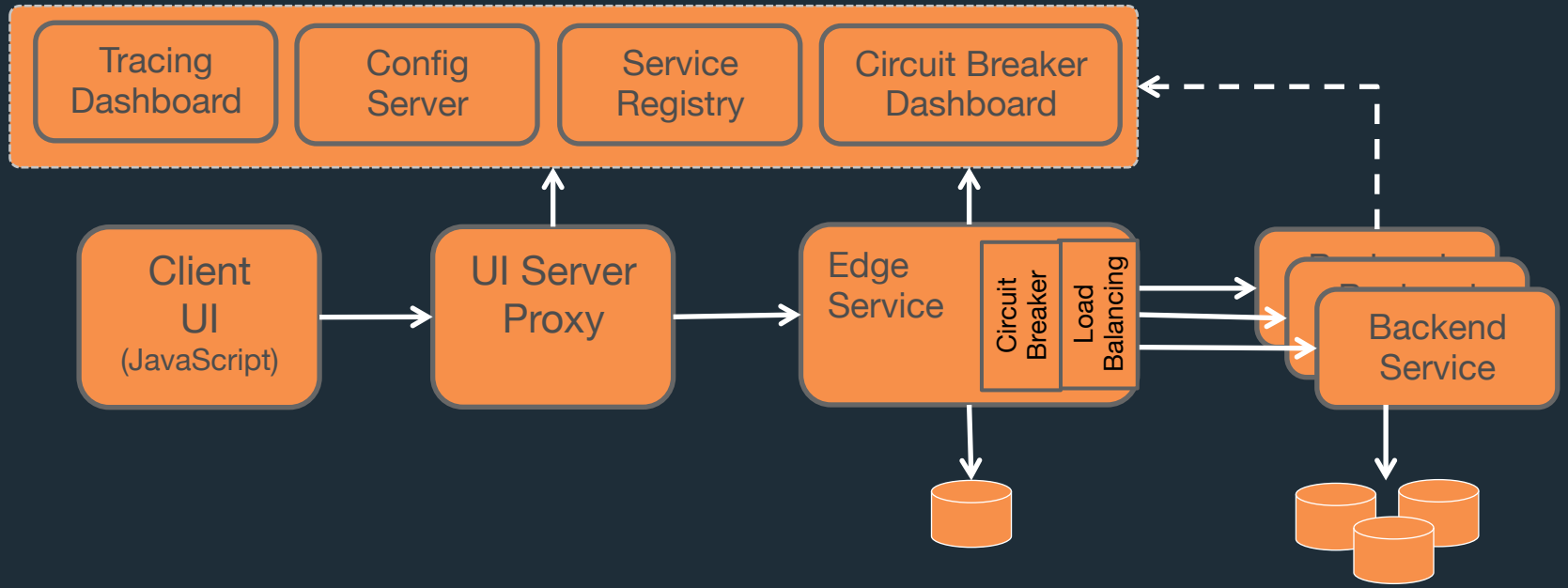


Pivotal



Pivotal

Process Management

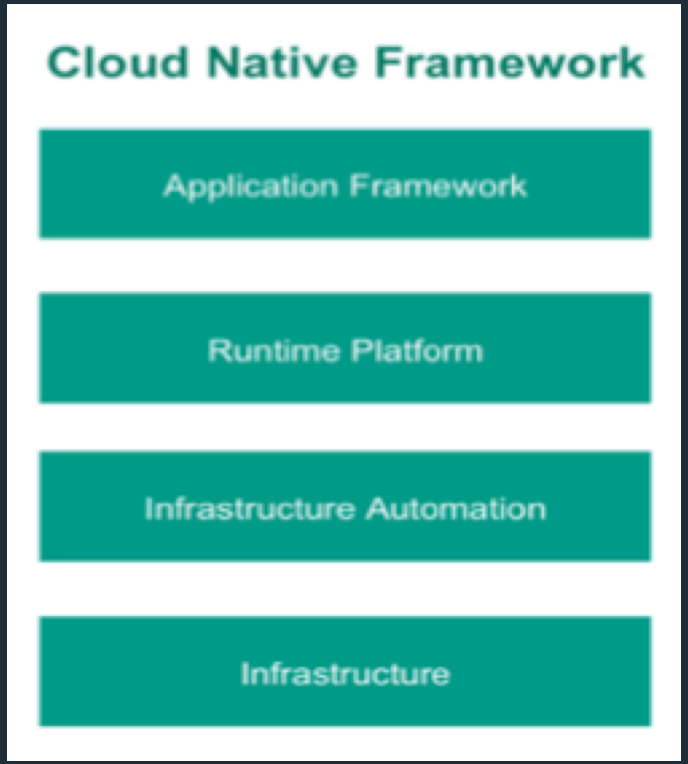


Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Structured Automation

- Groundwork for devops
- Self-service
- Rapid, automated provisioning
- Predictability and consistency
  - not ad-hoc!
- Visibility (monitoring & metrics)
- Continuous delivery on Day 2, too!



Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Cloud Native Architecture

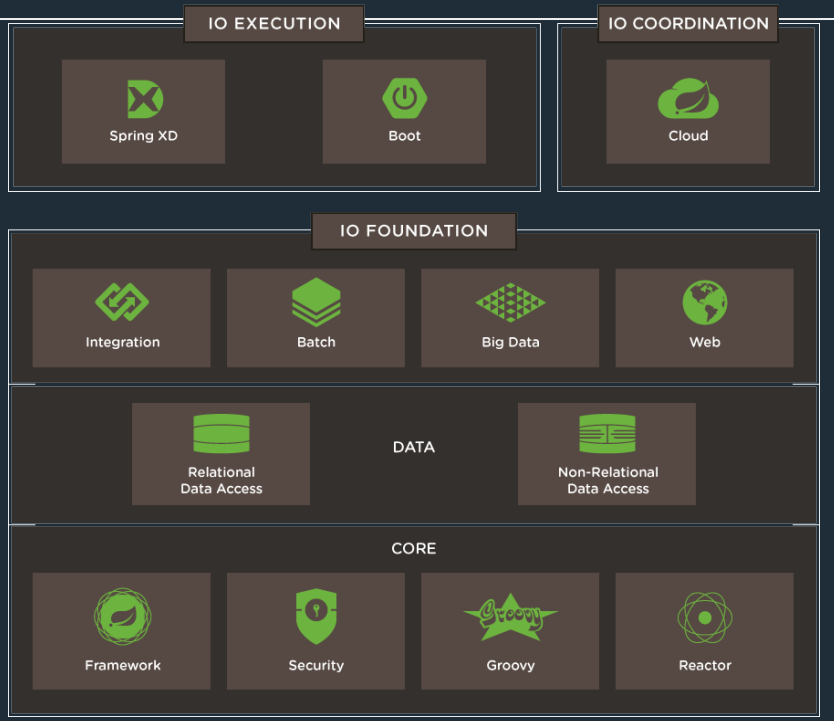
- Tools
- Platform
- Process and Culture

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Spring I/O

- Spring Boot
  - How you build an app
- Spring Cloud
  - Pieces for Cloud Native
- Spring Core



Pivotal

Spring Boot

- “Microframework”
- Opinionated
- Convention over configuration
- Production ready
- Ops Friendly
  - Self contained
  - Health and metrics
  - Externally configurable

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Spring Cloud

- Distributed/versioned configuration
- Service registry and discovery
- Routing
- Service-to-service calls
- Load balancing
- Circuit Breakers
- More…

Pivotal

Spring Cloud Netflix Components

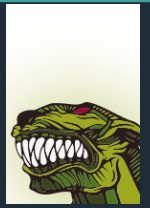
Ribbon

Archaius

Hystrix

Feign

Zuul



Client

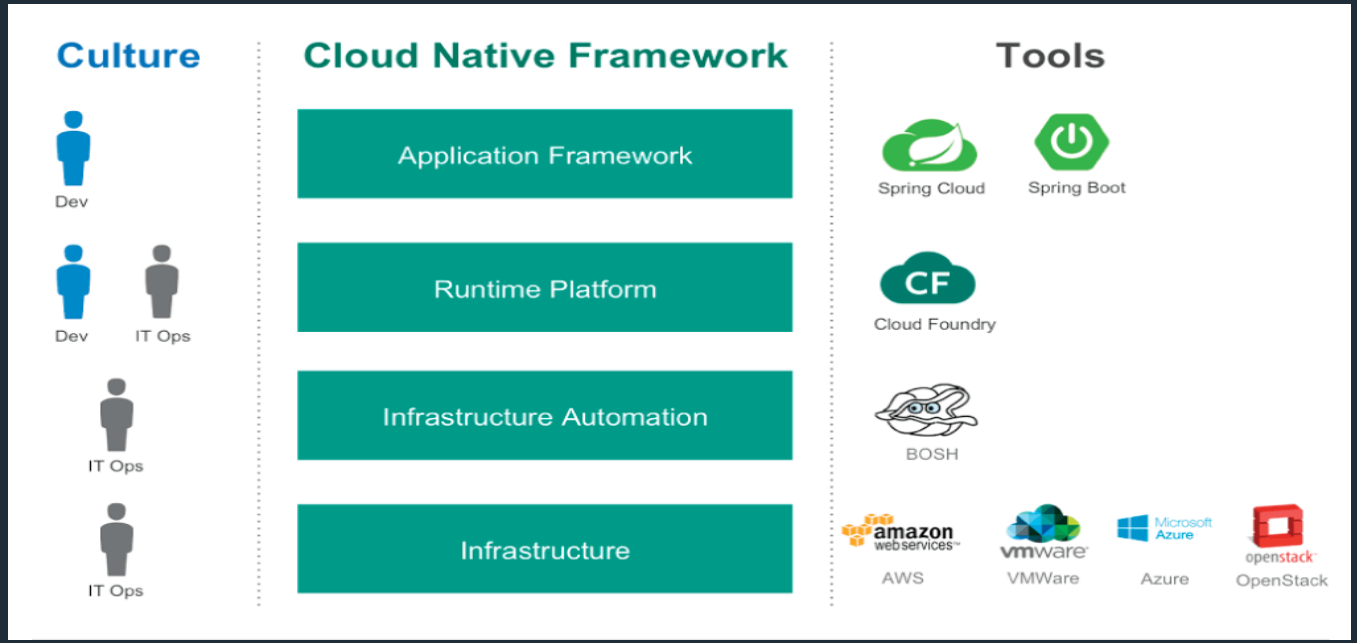
Eureka

Turbine

Server

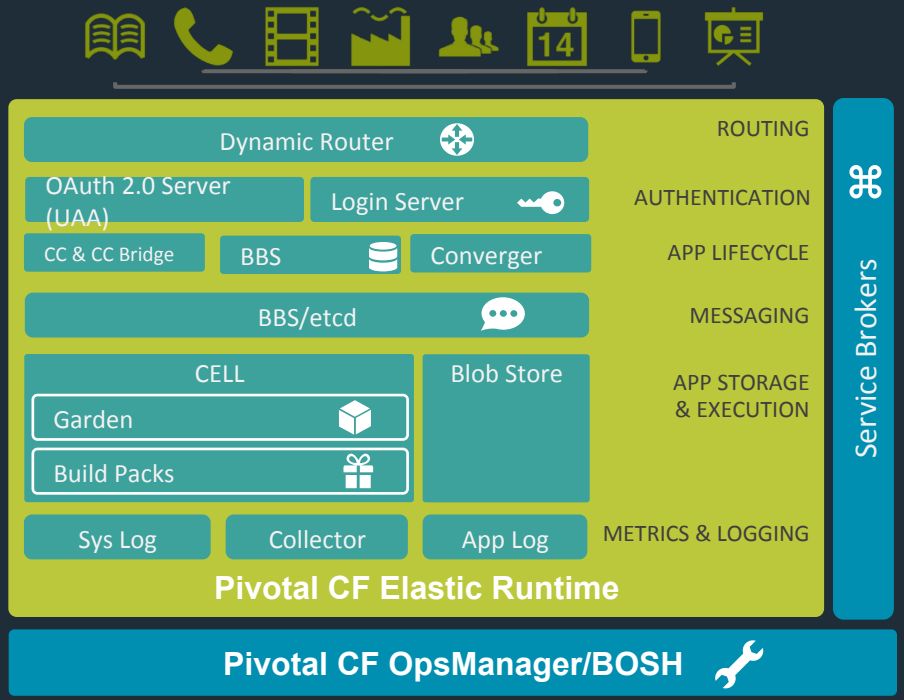
Pivotal

Pivotal Cloud Foundry



Pivotal

Pivotal Cloud Foundry



- Turnkey, fully automated Platform-as-a-Service
- Scalable runtime environment, extensible to most modern frameworks and languages running on Linux
- Instant expansion or upgrade with no downtime
- Deploy, scale and manage applications with bindable services using simplified semantics and APIs

Pivotal

The Journey: Technology

- Provide the right tools
  - Platform, design pattern foundations, continuous delivery…
- Start small
  - Right-size for your needs
  - For migrations, don’t boil the ocean
- Target applications that:
  - Have high business/competitive impact
  - Need to scale or change frequently

Pivotal

The Journey: Culture & Organization

- Small, cross-functional teams
- Products, not projects
- Test-driven development
- Short iterations
- Transparency and visibility

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Cloud Native Architecture

- Cloud Computing
- New Demands
- Being Reactive
- Speed & Safety

- Reactive Design
- 12 Factor Apps
- Microservices
- Design Patterns

- Tools
- Platform
- Process and Culture

Pivotal

© Copyright 2014 Pivotal. All rights reserved.

Pivotal

Thank You

Pivotal

Open.
Agile.
Cloud-Ready.