Pivotal

Accelerating Development Velocity

Pivotal Cloud Foundry Technical Introduction and Overview

Manuel Franco, Felipe Gutierrez

Agenda (Day 1)

The Modern Cloud-Native Platform

Introductions

The Modern Application Landscape

- Capabilities that Advance the State of Software Development
- Modern Application Architectures
- The Rise of “DevOps”

Pivotal Cloud Foundry - The Cloud Native Platform

- Why a Platform?
- Developer and Operator Benefits
- Four Abstractions

Technical Architecture (Pivotal Cloud Foundry)

- Pivotal Application Service (PAS)
- Pivotal Container Service (PKS)
- Pivotal Functions Service (PFS)
- Open Service Broker API
- BOSH

Lean Methodology

- Technical Practices that Enable Software Velocity

The Modern Application Landscape

Business as usual is no longer enough to remain competitive.

You need to be good at software...

Customers expect it.

Meet the demands to operate at scale.

Give you more business options.

Your competitors are improving.

It makes your life better.

Software has become more complex

... real-time

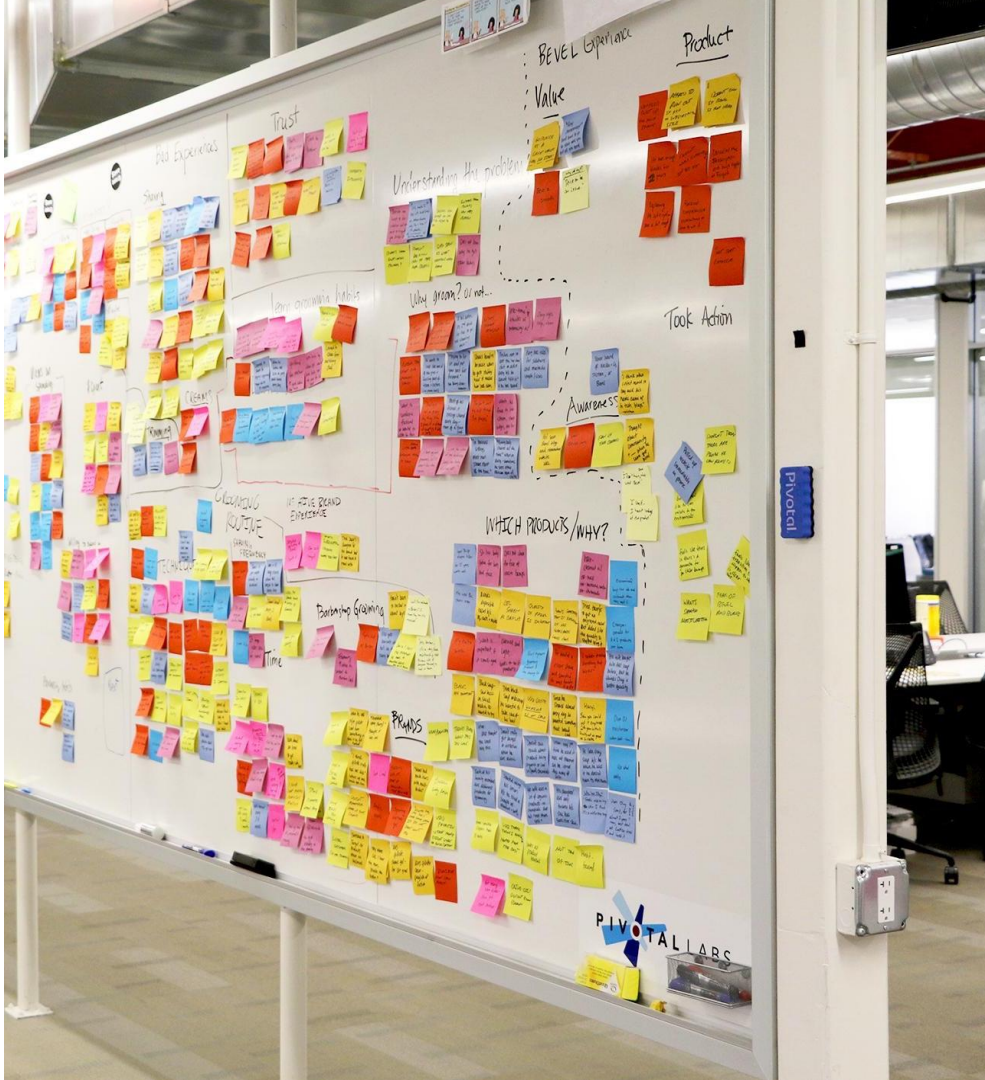
... distributed

... interactive

... customized

... expectations

More important!



Modern Software Organizations

Modern Software Organizations

7 months and 72 steps
to update software, to
same day deployments

ships to production 1,500 times a month

15,000 deploys per month

builds and deploys an MVP in one month

1500 developers with an
operator team of 4 people

Capabilities that advance the state of software development ...

1. Continuous Delivery (CI/CD)
2. Architecture
3. Product and Process
4. Lean Management and Monitoring
5. Culture

Pivotal

Computing Ages

Technology & Process

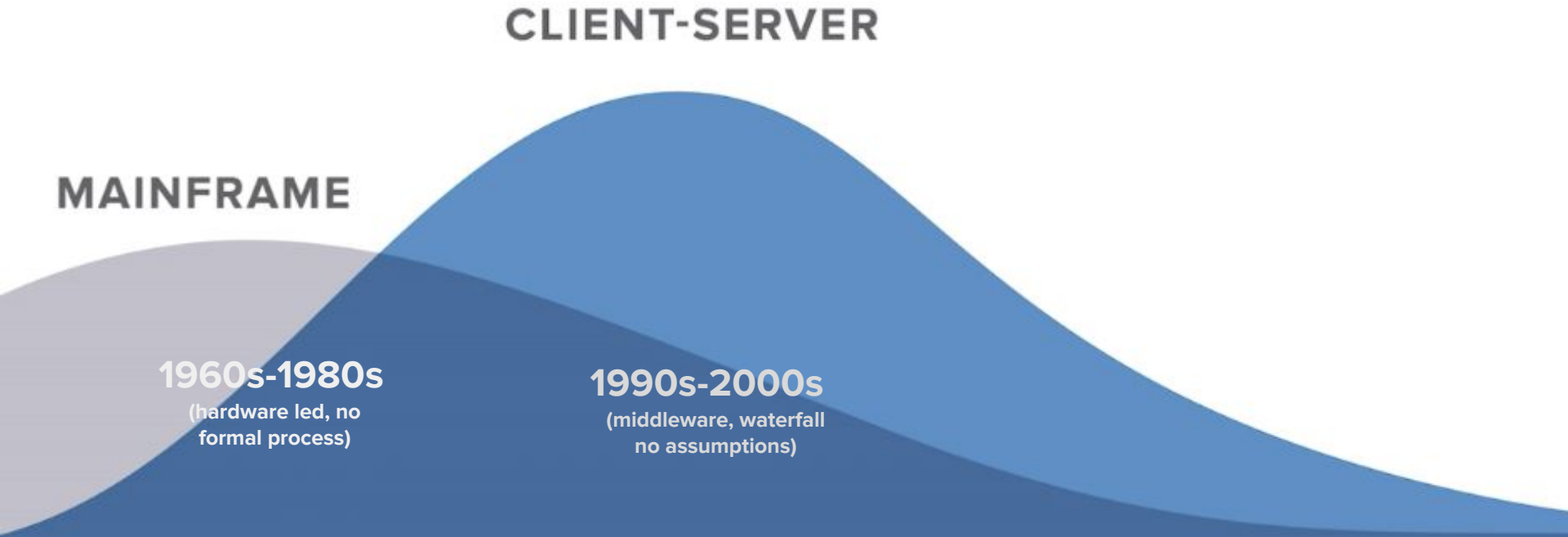
MAINFRAME

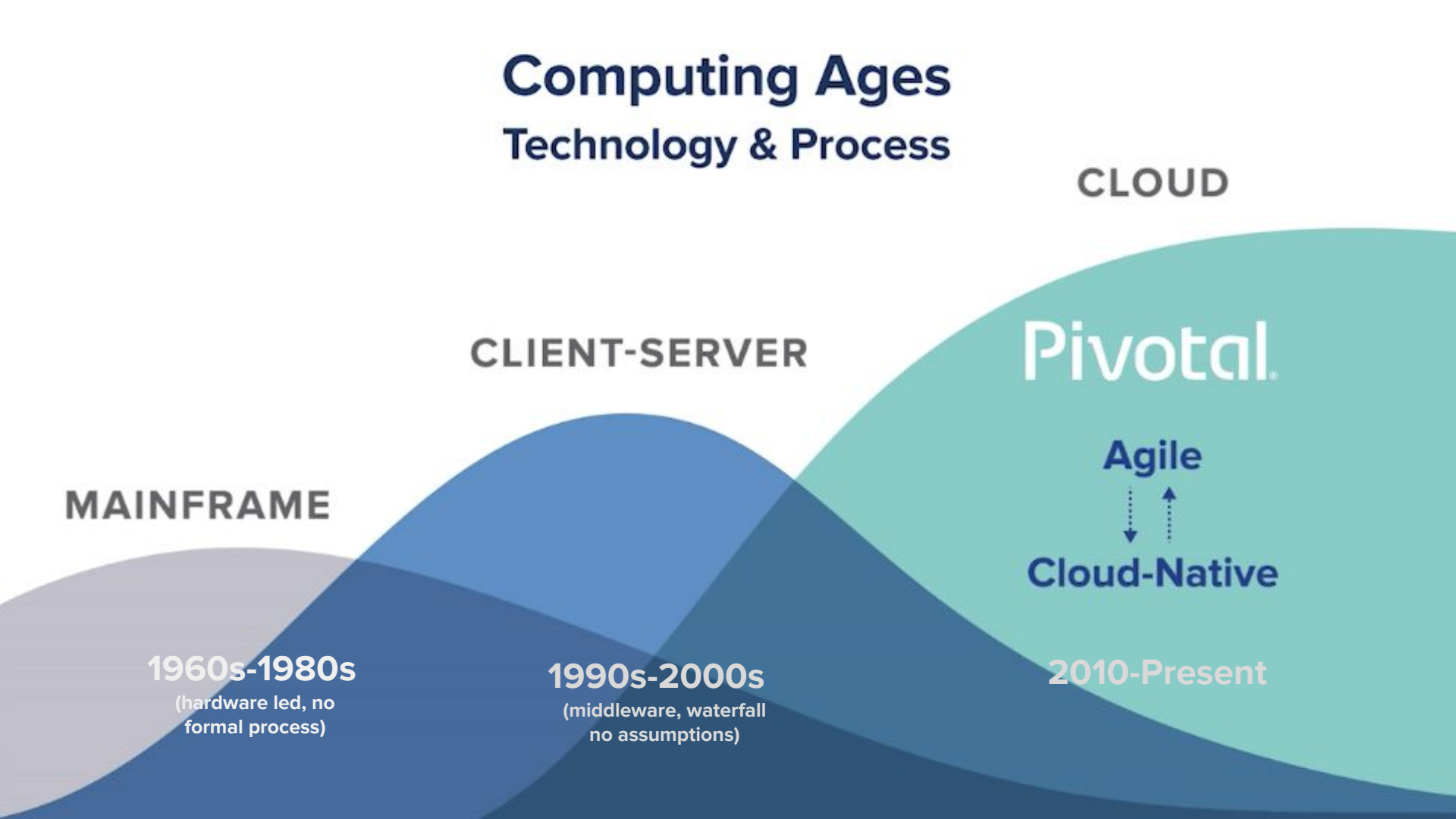
1960s-1980s

(hardware led, no formal process)

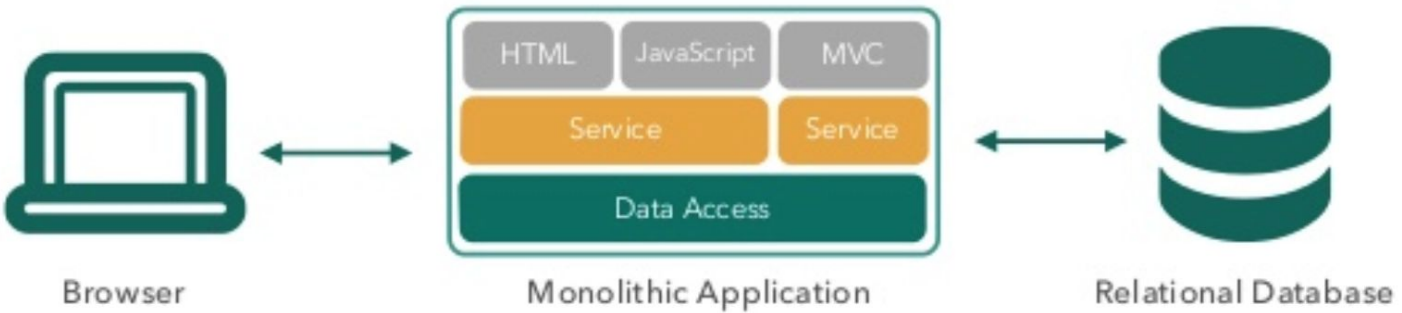
Computing Ages

Technology & Process

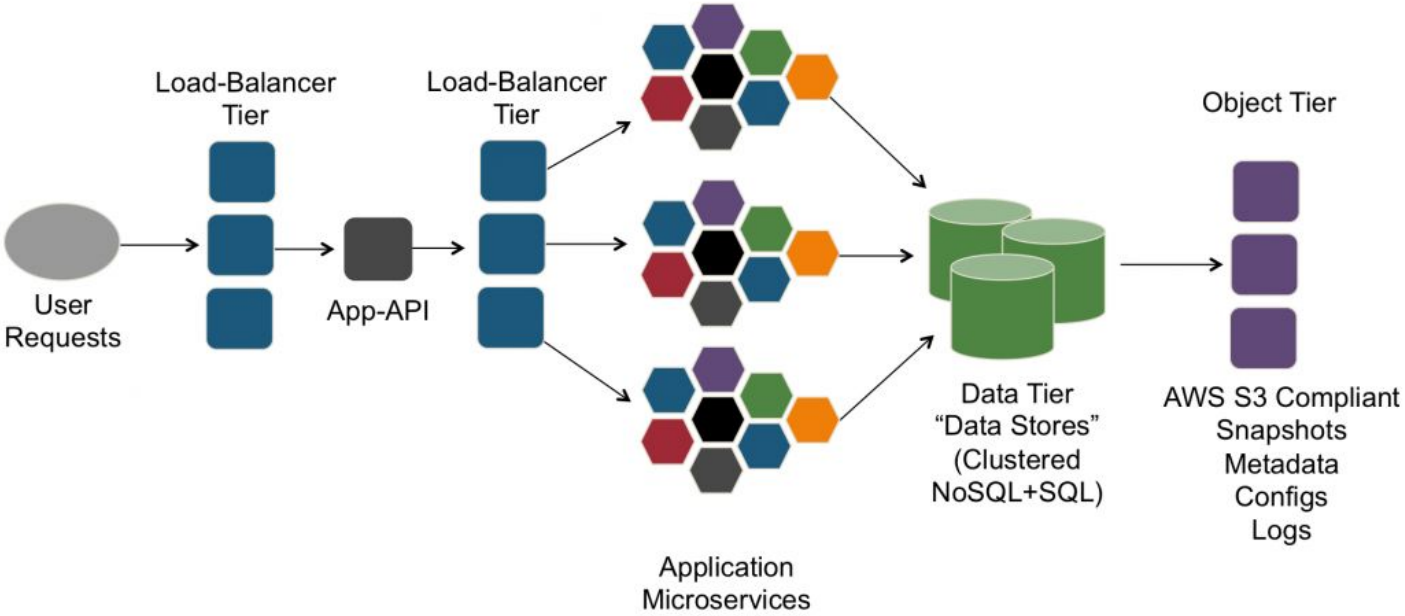


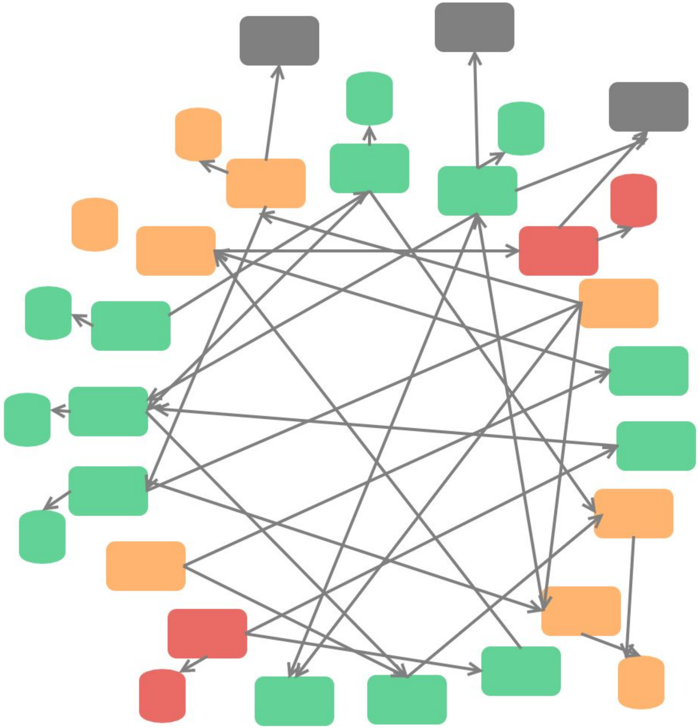

Computing Ages
Technology & Process
CLOUD
CLIENT-SERVER
Pivotal
Agile
MAINFRAME
Cloud-Native
1960s-1980s
(hardware led, no
formal process)
1990s-2000s
(middleware, waterfall
no assumptions)
2010-Present

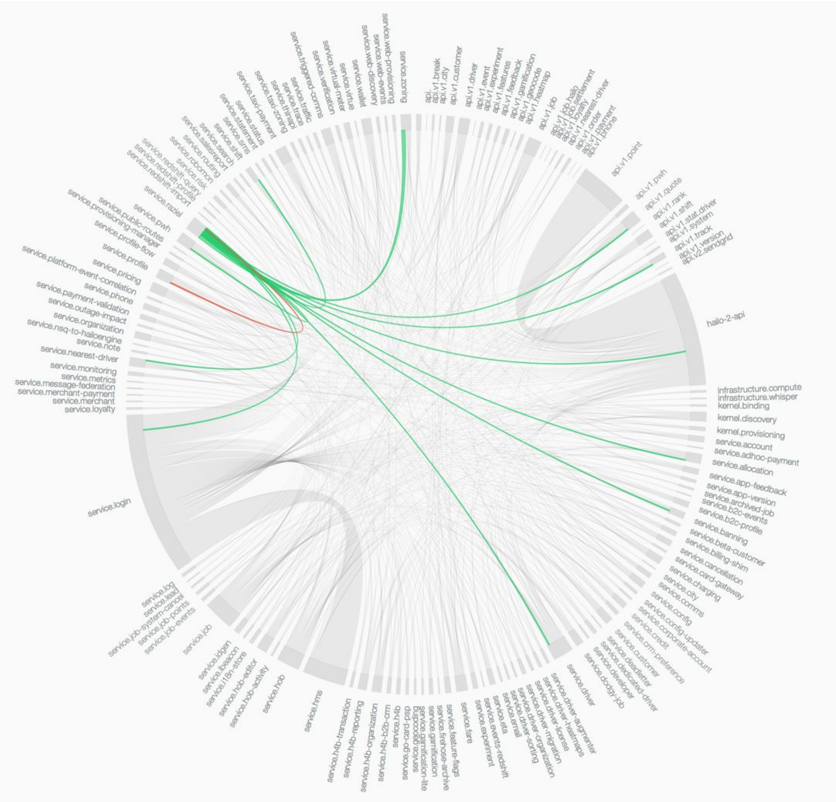
Client-Server Architectures

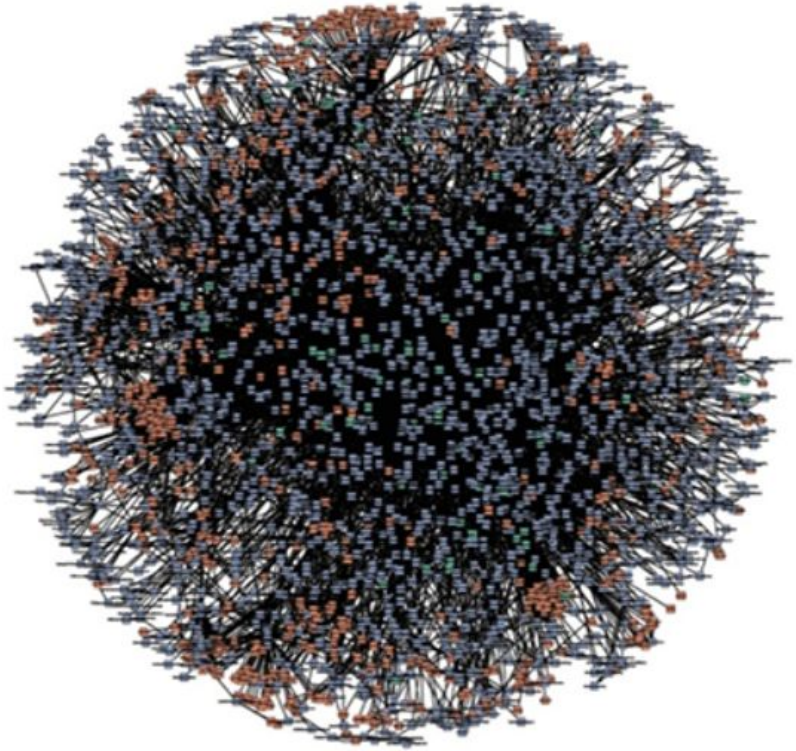


Cloud-Native Architectures

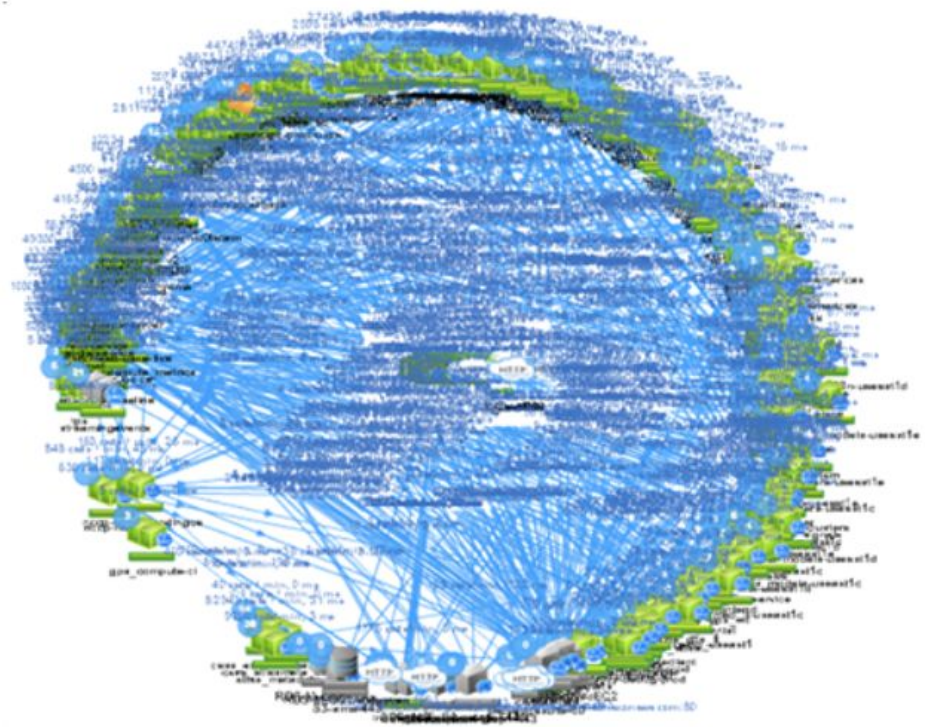





api.v1.point
hailo-2-api
infrastructure.compute
infrastructure.whisper
kernel.binding
kernel.discovery
kernel.provisioning
service.account
service.adhoc-payment
service.allocation
service.app-feedback
service.app-version
service.archived-job
service.b2c-events
service.b2c-profile
service.banning
service.beta-customer
service.login
service.loyalty
service.merchant
service.merchant-payment
service.message-federation
service.metrics
service.monitoring
service.nearest-driver
service.note
service.nsq-to-hailoengine
service.organization
service.outage-impact
service.payment-validation
service.phone
service.platform-event-correlation
service.pricing
service.profile



amazon.com



NETFLIX

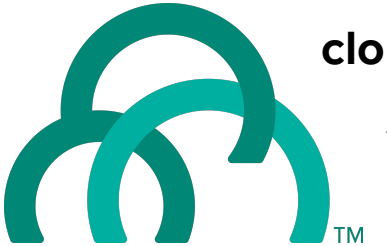
Pivotal

DevOps emerged from companies solving the problem of **how to build secure, resilient and evolving distributed systems at scale.**

Forrester report determined 31% of organizations are not using practices considered necessary for accelerating technology transformations.

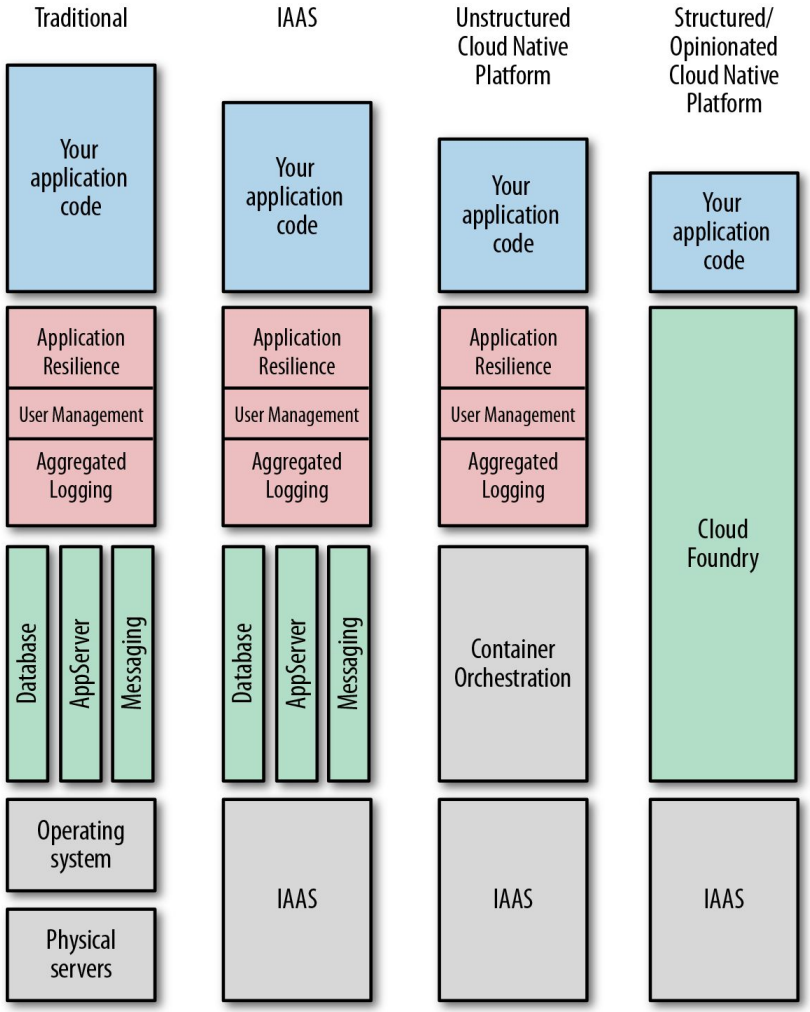
The most innovative and high performing companies are always striving to be better and **never consider themselves done** with their improvement or transformation journey.

You need a cloud-native platform with the right abstractions to capitalize on cloud-native patterns.



cloud-native platform (n)

A platform designed to **reliably, securely and predictably** run applications and services *at scale* on top of potentially unreliable cloud-based infrastructure.


Traditional
IAAS
Unstructured Cloud Native Platform
Structured/ Opinionated Cloud Native Platform
Your application code
Your application code
Your application code
Your application code
Application Resilience
User Management
Aggregated Logging
Application Resilience
User Management
Aggregated Logging
Application Resilience
User Management
Aggregated Logging
Cloud Foundry
Database
AppServer
Messaging
Database
AppServer
Messaging
Container Orchestration
Operating system
Physical servers
IAAS
IAAS
IAAS

Cloud Native Landscape

v2.1

See the interactive landscape at landscape.cncf.io

Greyed logos are not open source

App Definition & Development

Database & Data Warehouse

Vitess (CNCF Incubating)

Streaming

NATS (CNCF Incubating)

Source Code Management

Application Definition

Continuous Integration / Continuous Delivery (CI/CD)

Orchestration & Management

Scheduling & Orchestration

kubernetes (CNCF Graduated)

Coordination & Service Discovery

CoreDNS (CNCF Incubating)

Service Management

gRPC (CNCF Incubating)

envoy (CNCF Incubating)

LINKERD (CNCF Sandbox)

Runtime

Cloud-Native Storage

ROOK (CNCF Sandbox)

Container Runtime

containerd (CNCF Incubating)

rkt (CNCF Incubating)

Cloud-Native Network

CNI

!!!

Provisioning

Host Management / Tooling

Infrastructure Automation

Container Registries

Secure Images

Notary (CNCF Incubating)

TUF (CNCF Incubating)

Key Management

Cloud

Public

Private

Platforms

Certified Kubernetes - Distribution

Certified Kubernetes - Platform

Non-Certified Kubernetes

PaaS/Container Service

Observability & Analysis

Monitoring

Prometheus (CNCF Incubating)

Logging

fluentd (CNCF Incubating)

Tracing

OPENTRACING (CNCF Incubating)

JAEGER (CNCF Incubating)

Serverless

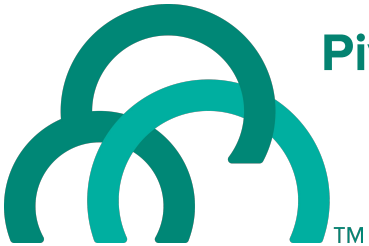
See the separate serverless landscape

Special

Kubernetes Certified Service Provider

Applications

API



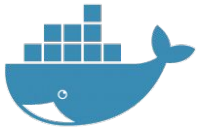
Pivotal Cloud Foundry

An opinionated, structured cloud-native platform that imposes a strict contract between the infrastructure layer underpinning it and the applications and services that it supports.

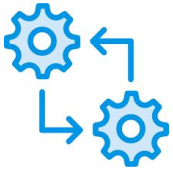
CLOUD FOUNDRY
BOSH

API

Providing Choice for Workloads



Containers



Batches

Event-driven Functions

Microservices

Data Services

Monolithic Applications

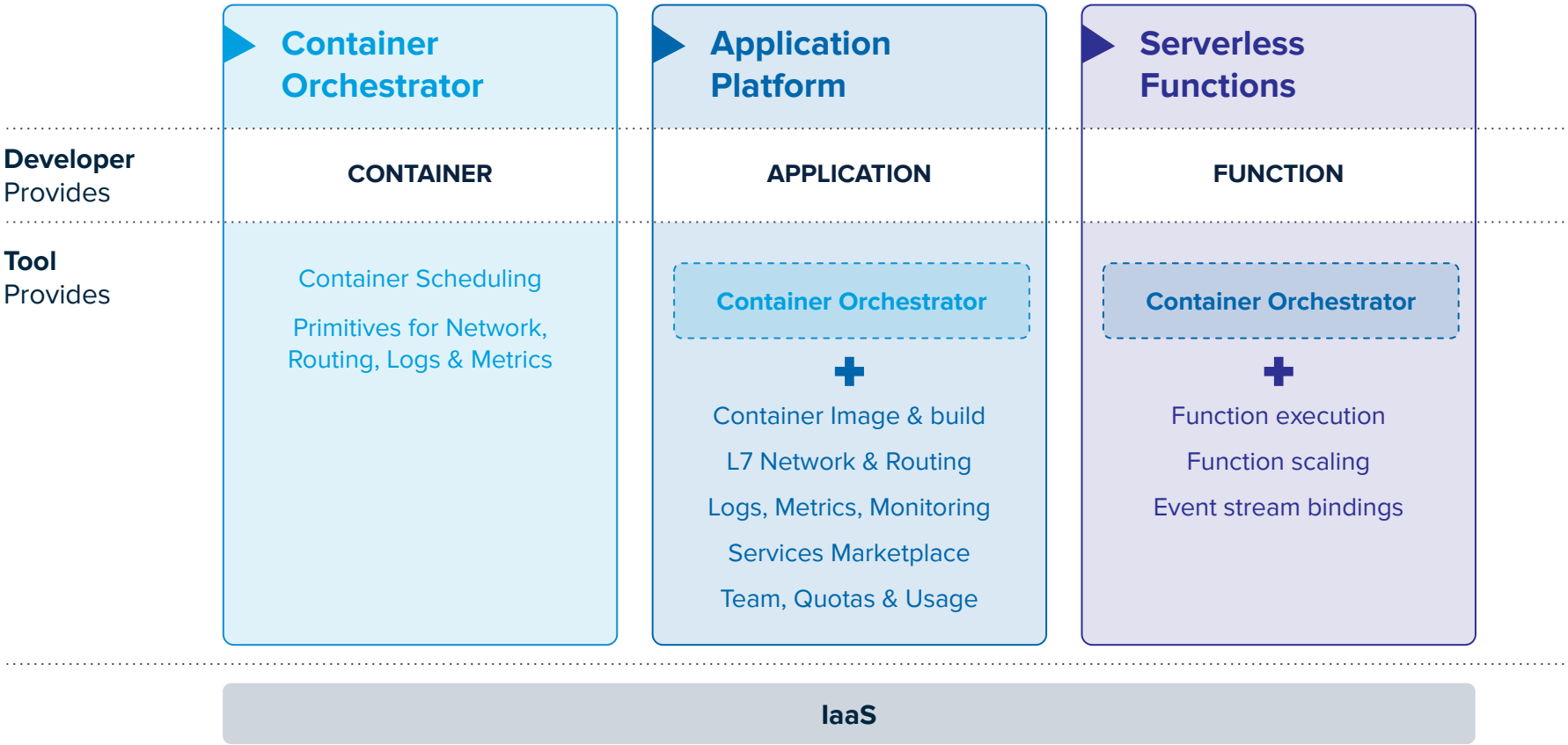
Container Orchestrator (CaaS)

Application Platform (PaaS)

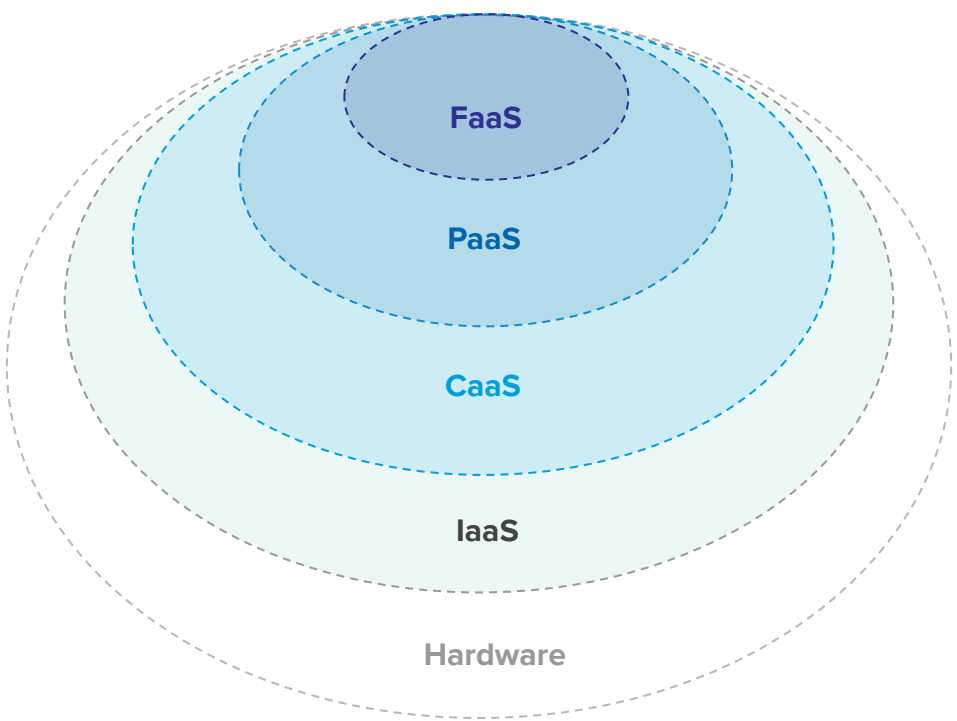
Serverless Functions (FaaS)

IaaS

Choose the right tool for the job



Higher flexibility and less enforcement of standards



Lower development complexity and higher operational efficiency

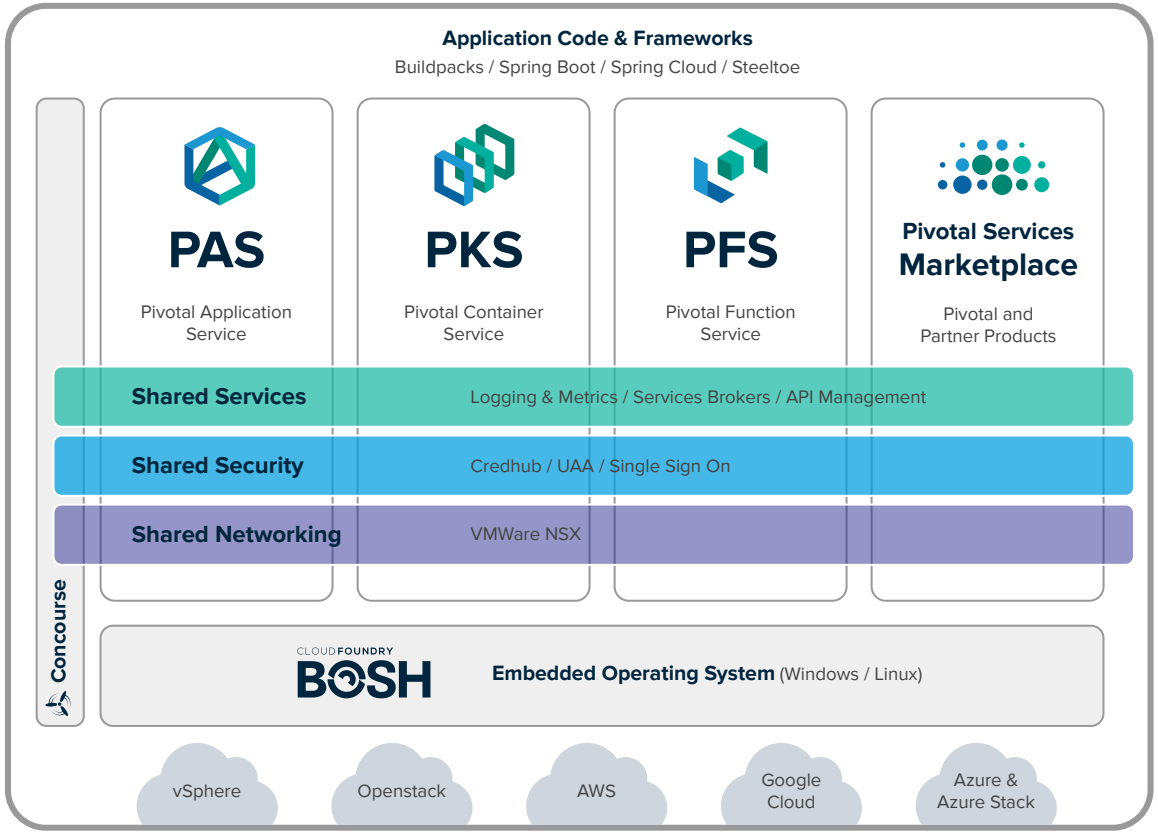
Strategic goal: Push as many workloads as technically feasible to the top of the platform hierarchy

Any App
Every Cloud
One Platform

PCF — for everything
that matters

Pivotal

Pivotal
Cloud Foundry®



App Instance

PCF / PAS

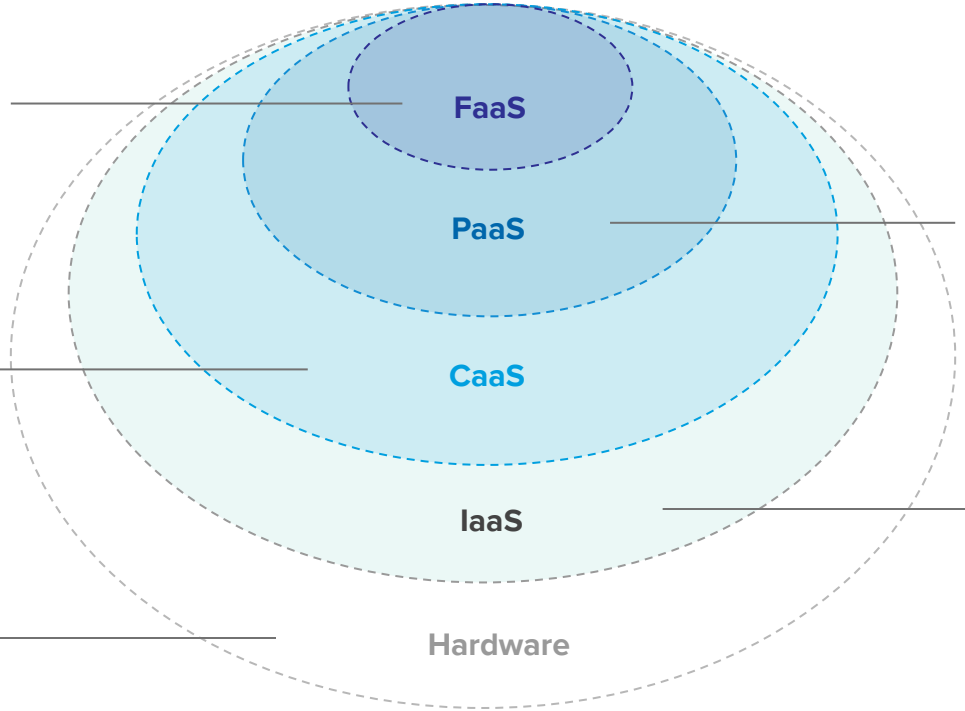
PCF / PAS

App Instance

Developers:
- Polyglot Env.
- CI/CD
- Blue/Green Deploys
- Incremental Changes
- Self-Service Marketplace
- Env. Consistency
- Spring Cloud Services
- Cloud Native & Legacy Apps
- Log Aggregation
- App & Platform Metrics
- App Health Mgmt.
- App Autoscaling Up & Down
- Routing & Routing Services
- Industry Stds Compliance

- ✓ Multi-Cloud
- ✓ Scalability
- ✓ Logging (SIEM)
- ✓ KPIs & Metrics
- ✓ Spring (SCS)
- ✓ .NET Support
- ✓ Containerization
- ✓ Orchestration
- ✓ Service Broker
- ✓ Marketplace
- ✓ Svc Instances
- ✓ Security
- ✓ HA

Operators:
- Zero Downtime
- Bosh Automation
- Health Mgmt.
- Stemcell (OS)
- Middleware
- PCF Metrics
- PCF Health KPIs
- Platform Logs
- 3rd Party Ecosystem
- Environment Parity
- Rolling Upgrades
- Quotas & RBAC
- Windows2012R2/2016
- Secure by Default


AWS Lambda, Azure Functions, OpenWhisk, kubeless, PFS
FaaS
PCF, Azure App Service, Heroku
PaaS
PKS, GKE, OpenShift, AWS Fargate, Kubernetes
CaaS
AWS, Microsoft Azure, GCP, VMware
IaaS
HPE, Dell, IBM, Lenovo
Hardware

The Pivotal value proposition

Developer Productivity

- Accelerate feedback loops by improving delivery velocity
- Focus on applications, not infrastructure
- Give developers the tools and frameworks to build resilient apps

Operational Efficiency

- Employ 500:1 developer to operator ratio
- Perform zero-downtime upgrades
- Runs the same way on every public/private cloud

Comprehensive Security

- Adopt a defense-in-depth approach
- Continuously update platforms to limit threat impact
- Apply the 3 R's → repair, repave, rotate

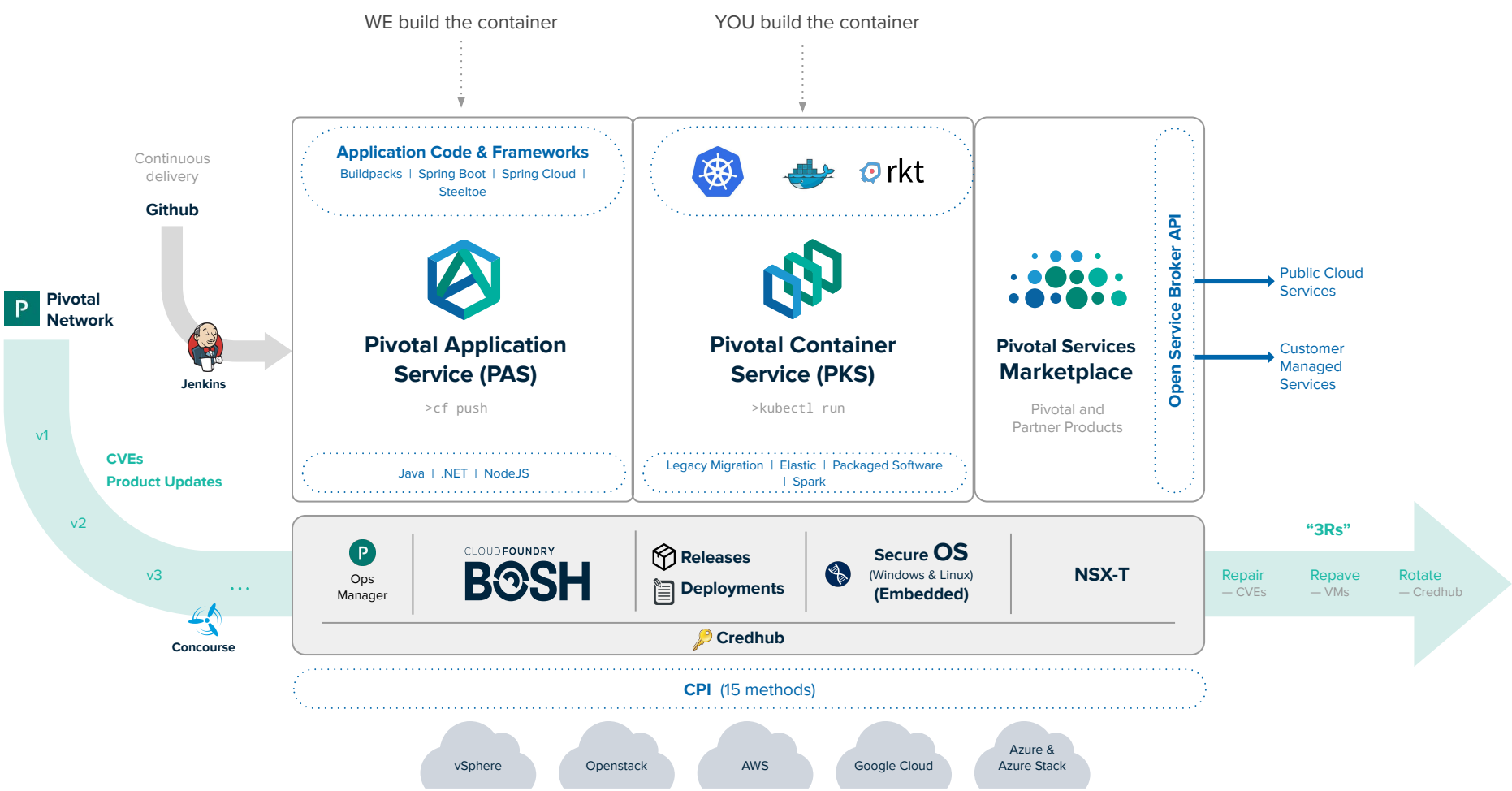
High Availability

- Run platforms that stays online under all circumstances
- Scale up and down, in and out, through automation
- Deploy multi-cloud resilience patterns

Pivotal

Technical Architecture

Pivotal Cloud Foundry Suite


WE build the container
YOU build the container
Continuous delivery
Github
Pivotal Network
Jenkins
v1
CVEs
Product Updates
v2
v3
...
Concourse
Application Code & Frameworks
Buildpacks | Spring Boot | Spring Cloud | Steeltoe
rkt
Pivotal Application Service (PAS)
>cf push
Pivotal Container Service (PKS)
>kubectl run
Pivotal Services Marketplace
Pivotal and Partner Products
Open Service Broker API
Public Cloud Services
Customer Managed Services
Java | .NET | NodeJS
Legacy Migration | Elastic | Packaged Software | Spark
Ops Manager
CLOUDFOUNDRY BOSH
Releases
Deployments
Secure OS (Windows & Linux) (Embedded)
NSX-T
Credhub
"3Rs"
Repair — CVEs
Repave — VMs
Rotate — Credhub
CPI (15 methods)
vSphere
Openstack
AWS
Google Cloud
Azure & Azure Stack
Pivotal

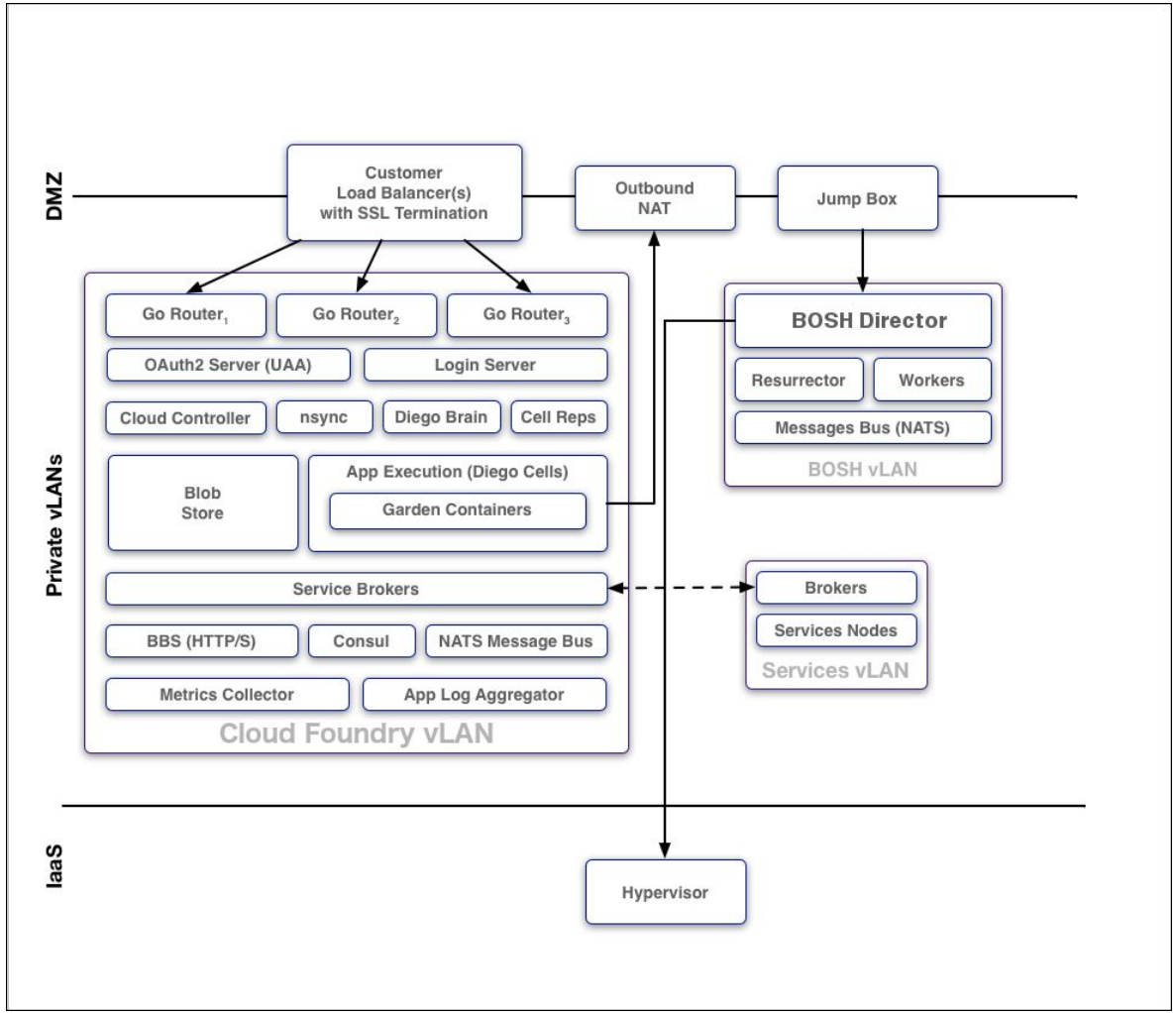
Pivotal Application Service (PAS)

Pivotal

Pivotal Cloud Foundry

PAS Architecture

- Routing
- Authentication
- App Lifecycle
- App Storage and Execution
- Services
- Messaging
- Metrics and Logging



Pivotal

Pivotal Container Service (PKS)

Pivotal Container Service (PKS): A Runtime for Containers

A turnkey solution to provision, operate and manage enterprise grade Kubernetes clusters

Pivotal
+
vmware
+
Google

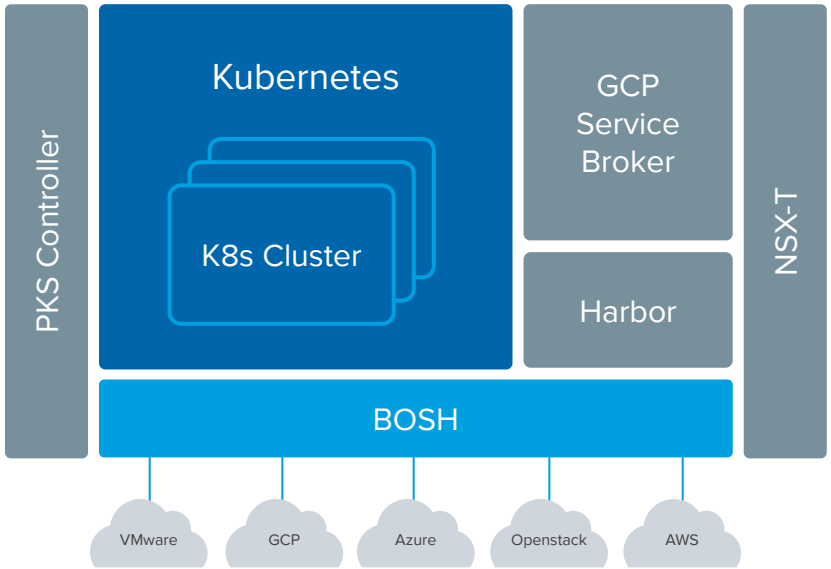
Kubernetes Dial Tone:

- Health management
- Aggregated Metrics and Logging
- Autoscaling
- Persistence interface

Control Plane:

- Provisioning Engine
- T-shirt sized clusters
- Self-service Clusters
- Software Update Automation
- Load balancing
- Networking
- Multi-tenancy

Pivotal Container Service™



Built with open-source Kubernetes — Constant compatibility with the current stable release of Kubernetes, operated by BOSH. No proprietary extensions.

Production-ready — Highly available from apps to infrastructure, no single points of failure. Built-in health checks, scaling, auto-healing and rolling upgrades.

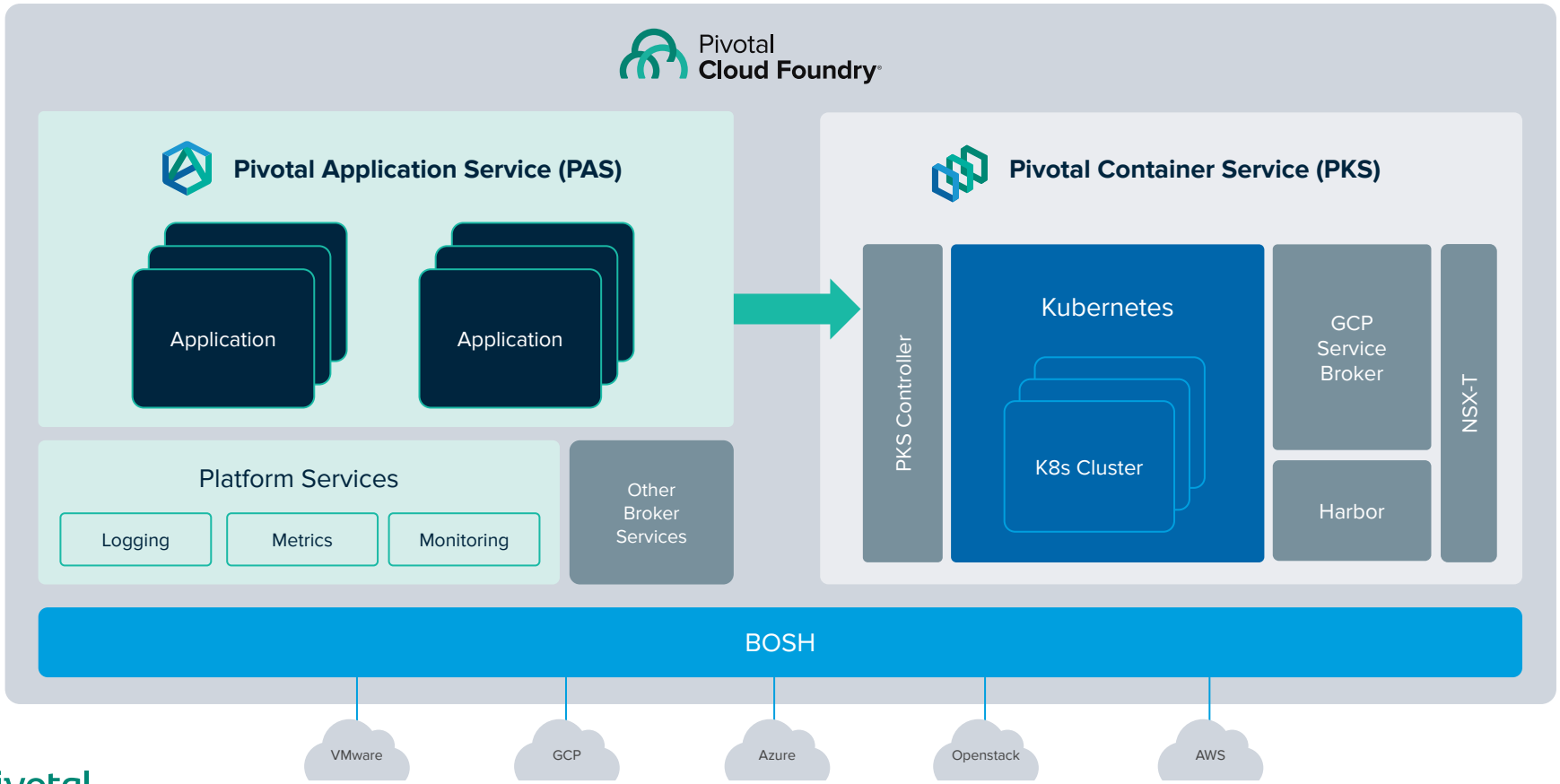
Multicloud — BOSH provides a reliable and consistent operational experience. For any cloud.

Network management and security out-of-the-box with VMware NSX-T. Multi-cloud, multi-hypervisor.

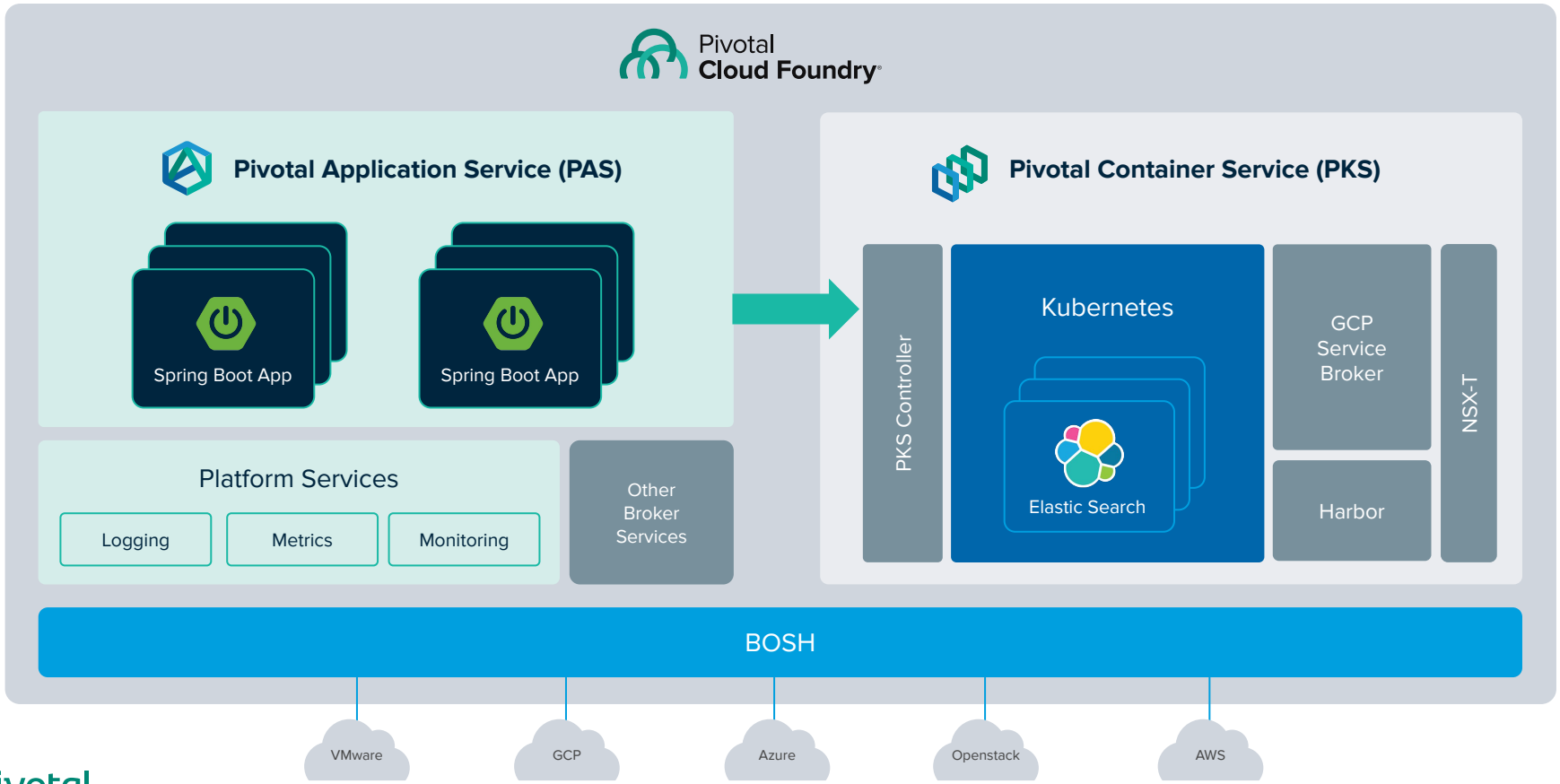
GCP APIs access — The GCP Service Broker allows apps to transparently access Google Cloud APIs, from anywhere. Easily move workloads to/from Google Container Engine (GKE).

Fully automated Ops — Fully automated deploy, scale, patch, upgrade. No downtime. Use CD pipelines to deploy your platform, too.

Leveraging more than one abstraction



Sample Use Case



Pivotal Function Service (PFS)

Pivotal Function Service (PFS): A Runtime for Functions

Execute functions in response to events. Use PFS to handle web events, event-based integration, and large scale streaming data.

Trigger functions via HTTP/Message Broker — PFS is architected to support event stream processing, connecting to message topics via a language-neutral, function container interface.

Run functions anywhere — PFS lets you easily run functions on-premises and in the public cloud for maximum flexibility.

Use modern DevOps workflows — PFS allows you to use familiar, container-based workflows for serverless scenarios.

Pluggable event brokers — PFS can be connected easily with popular message brokers such as Kafka, RabbitMQ, Google Pub/Sub, and AWS Kinesis.

Polyglot — PFS supports the authoring of functions in your chosen framework - Node.js, Spring/Java, or Shell.

Kubernetes Native — PFS runs natively on top of Kubernetes, making it easy to trigger code or containers in response to events.

PFS Scenarios & Use Cases

Web Events

- Website back-end services like form post handlers, authentication, tracking and logging.
- APIs to back-end data services for mobile and web apps e.g GraphQL
- Webhook handlers
- Chat integrations
- Digital assistant services e.g. Alexa skills

Event-based Integration

- Scheduled tasks, ETL
- File processing e.g. images and videos
- Security scanning
- Complex Event Processing and Change Data Capture
- Monitoring, notifications and alerting
- Custom auth e.g. via API Gateway

Stream Processing

- IoT streams
- Log ingestion
- Event streams e.g. with Kinesis
- ML pipelines

App Platforms + Functions

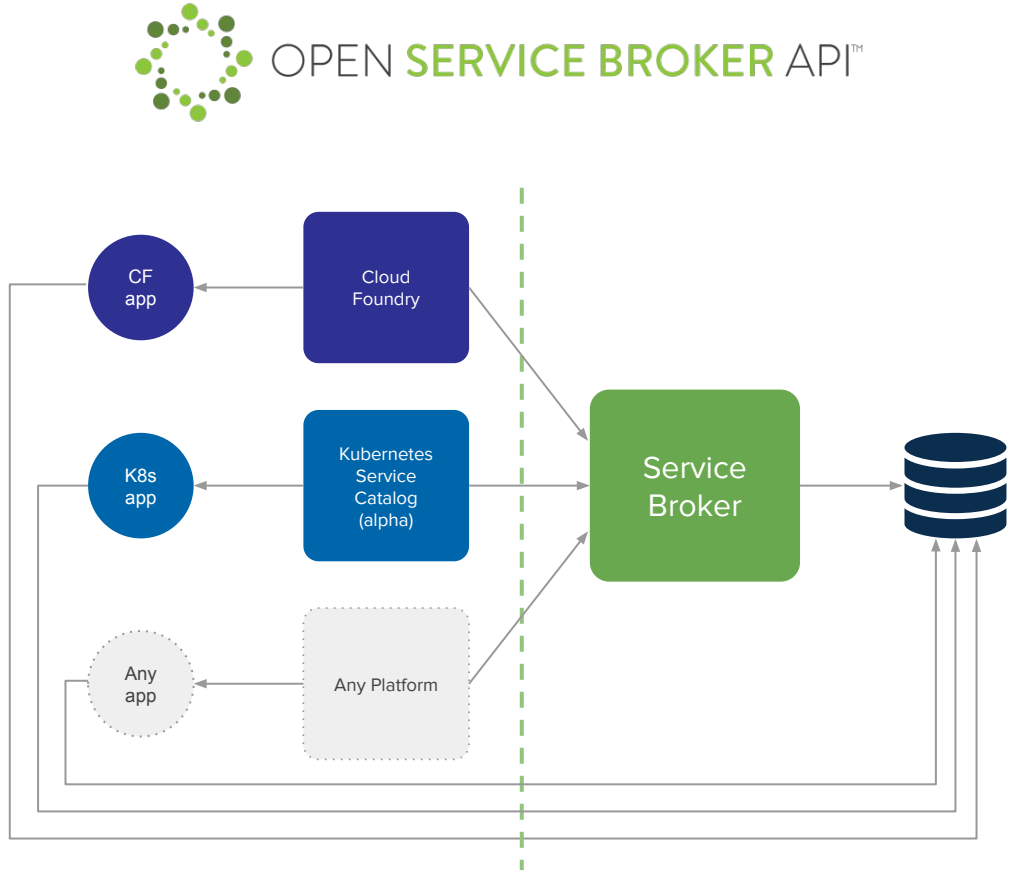
Comparing the experience for Devs + Ops

App Platform Abstraction	Functions Abstraction
runs **apps**	runs **functions**
push app to deploy server	register function and bind it to a trigger
server runs and waits for requests	function doesn't run until triggered
server listens to network	platform deploys and invokes functions
server handles lots of requests	functions handle **events** and then go away
scale out manually or by policy	**auto-scale** based on concurrent event load
pay per instance	pay per use – time & memory

Open Services Broker API

Easy Extensibility

Service Broker Model across Abstractions simplifies ops



Extend Apps with Brokered Services from Pivotal

MySQL for PCF

- Enterprise-ready MySQL for your developers
- Automate database operations in developer workflows
- **NEW:** Leader-follower for multi-site HA

Pivotal Cloud Cache

- High performance, in-memory, data at scale for microservices Look-aside caches & HTTP session state caching
- **NEW:** WAN replication

RabbitMQ for PCF

- Easily connect distributed applications with the most widely deployed open source message broker
- Enable connected scalable, distributed applications
- **NEW:** On-demand clusters

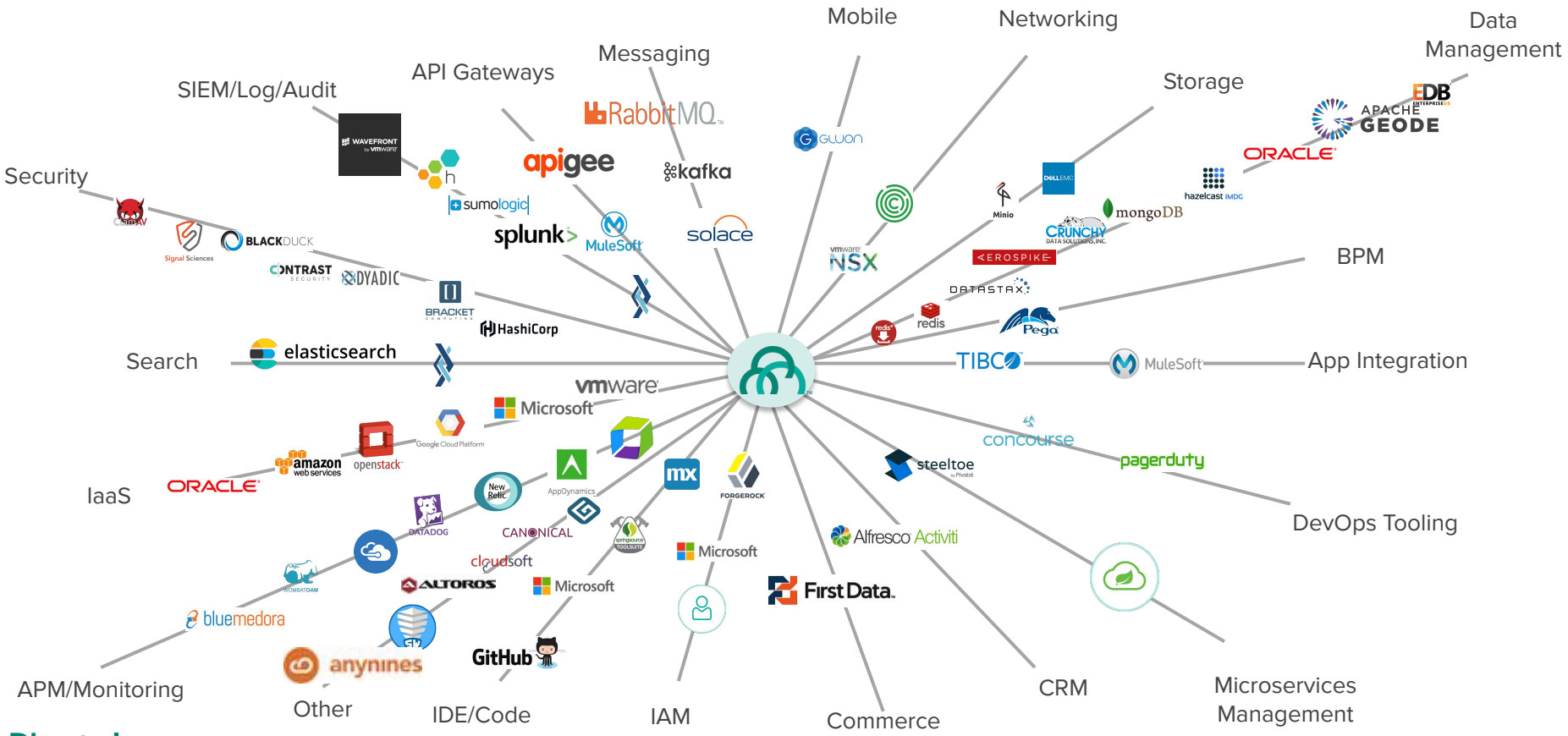
Redis for PCF

- In-Memory cache and datastore, configured for the enterprise
- Efficient provisioning matched to use cases

BOSH Managed | **On-Demand Provisioning** | **Dedicated Instances** | **Custom Service Plans**

Pivotal

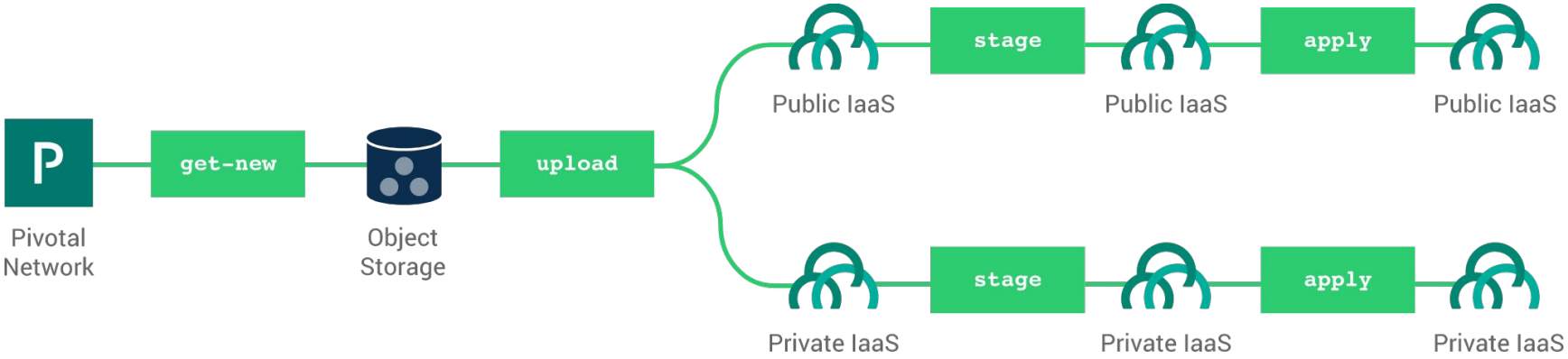
The Growing PCF Ecosystem



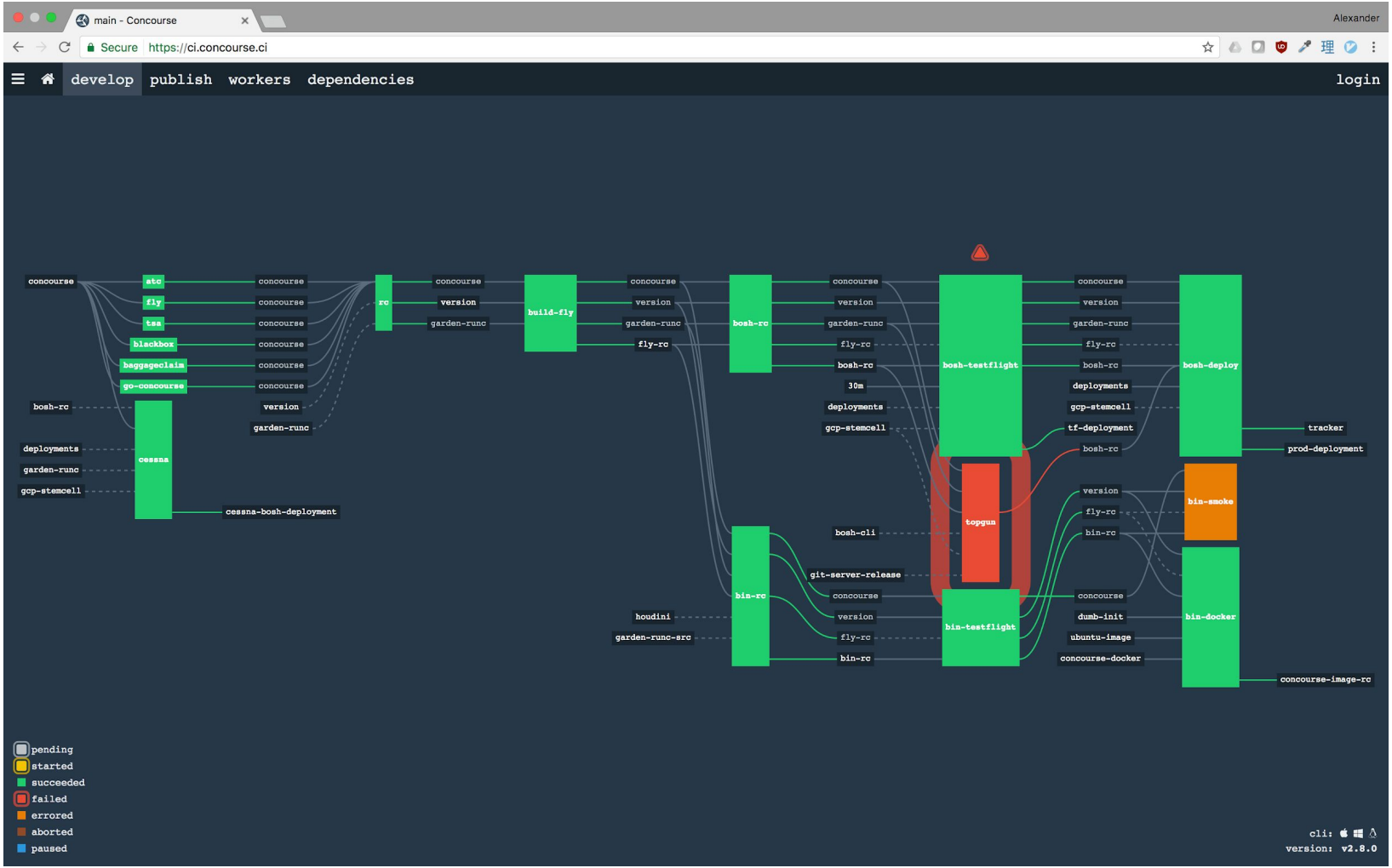
Pivotal

Pipelines ... Pipelines ... Pipelines ...

(CI is the secret sauce to go warp speed)



Pivotal


main - Concourse
Alexander
Secure https://ci.concourse.ci
develop
publish
workers
dependencies
login
concourse
atc
fly
tsa
blackbox
baggageclaim
go-concourse
bosh-rc
deployments
garden-runc
gcp-stemcell
cessna
cessna-bosh-deployment
version
rc
build-fly
fly-rc
bosh-rc
30m
bosh-testflight
topgun
bosh-deploy
tracker
prod-deployment
tf-deployment
bosh-cli
git-server-release
houdini
garden-runc-src
bin-rc
bin-testflight
bin-smoke
bin-docker
dumb-init
ubuntu-image
concourse-docker
concourse-image-rc
pending
started
succeeded
failed
errored
aborted
paused
cli:
version: v2.8.0

Pivotal

Quick Introduction

Pivotal

bosh (n)

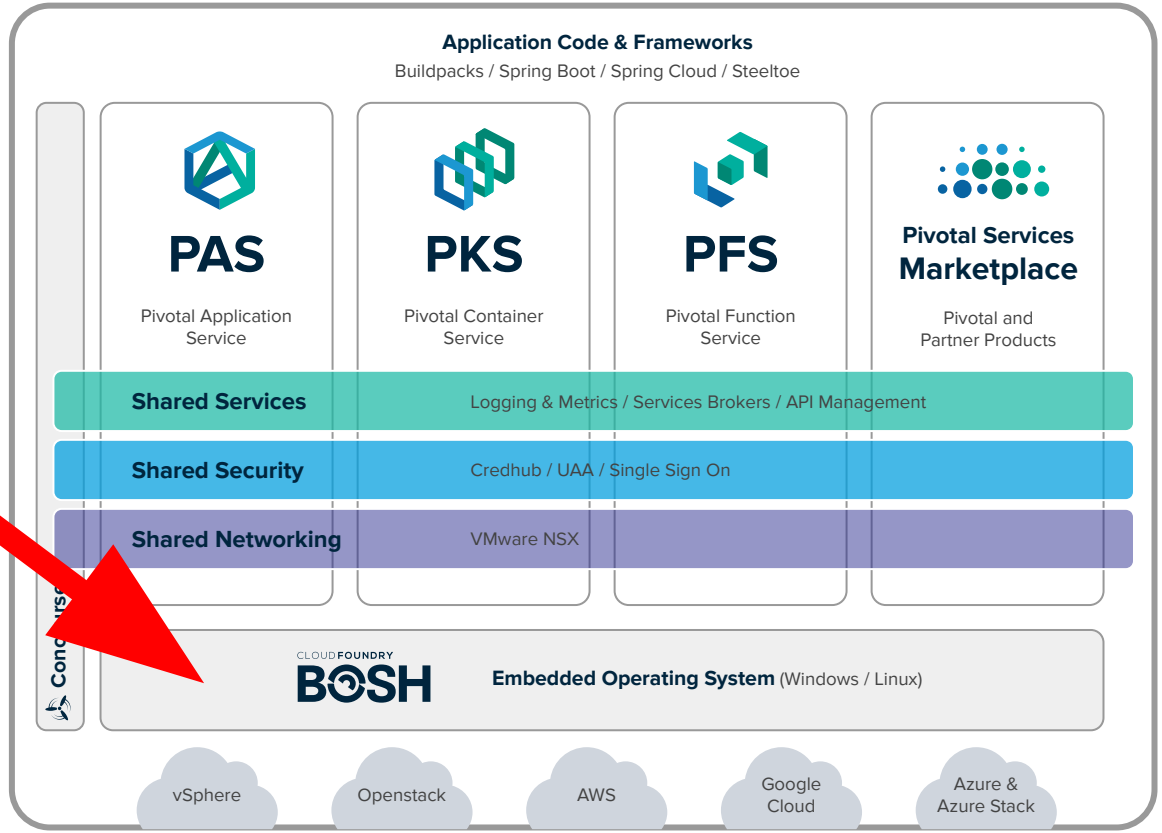
an open source tool chain for release engineering, deployment, and lifecycle management of large scale distributed services.

(syn) borg++

Pivotal

CLOUDFOUNDRY

BOSH™



Pivotal

CLOUD FOUNDRY

BOSH™

What is it?

- Handles packaging, distribution, orchestration and management
- Allows for complex clustered deploys
- Captures the skills required for decade+ cluster lifecycle management
- **Director** - the main actor
  - The “**stemcell**” - Ubuntu image with an agent
  - The “**release**” - like an SRPM
  - The “**manifest**” - details about your particular deployment of a release

CLOUD FOUNDRY

BOSH™

Fundamental Outcomes

- Packaging with embedded OS
- Server provisioning on any IaaS
- Software deployment across clusters

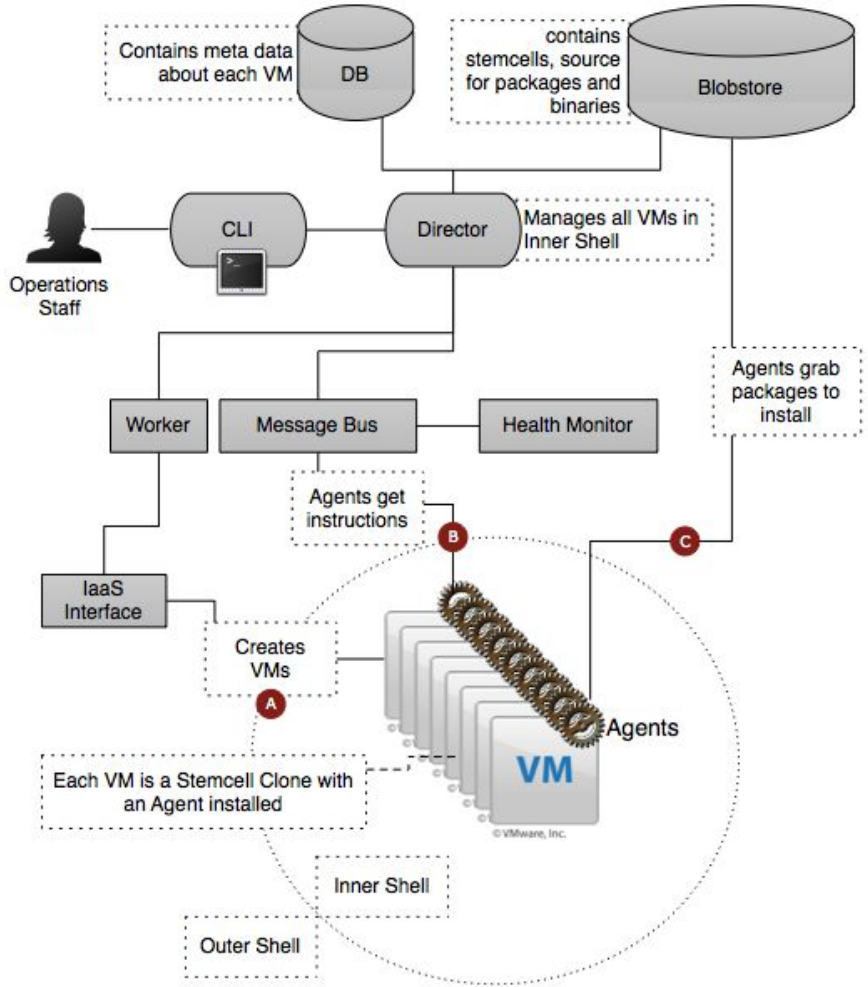
- Service & server state monitoring
- Self-healing w/ Resurrector
- Rolling upgrades via canaries
- Dynamic scaling up or down

Pivotal

CLOUD FOUNDRY

BOSH™

Architecture



Pivotal

CLOUDFOUNDRY BOSH™

Declarative YMLs

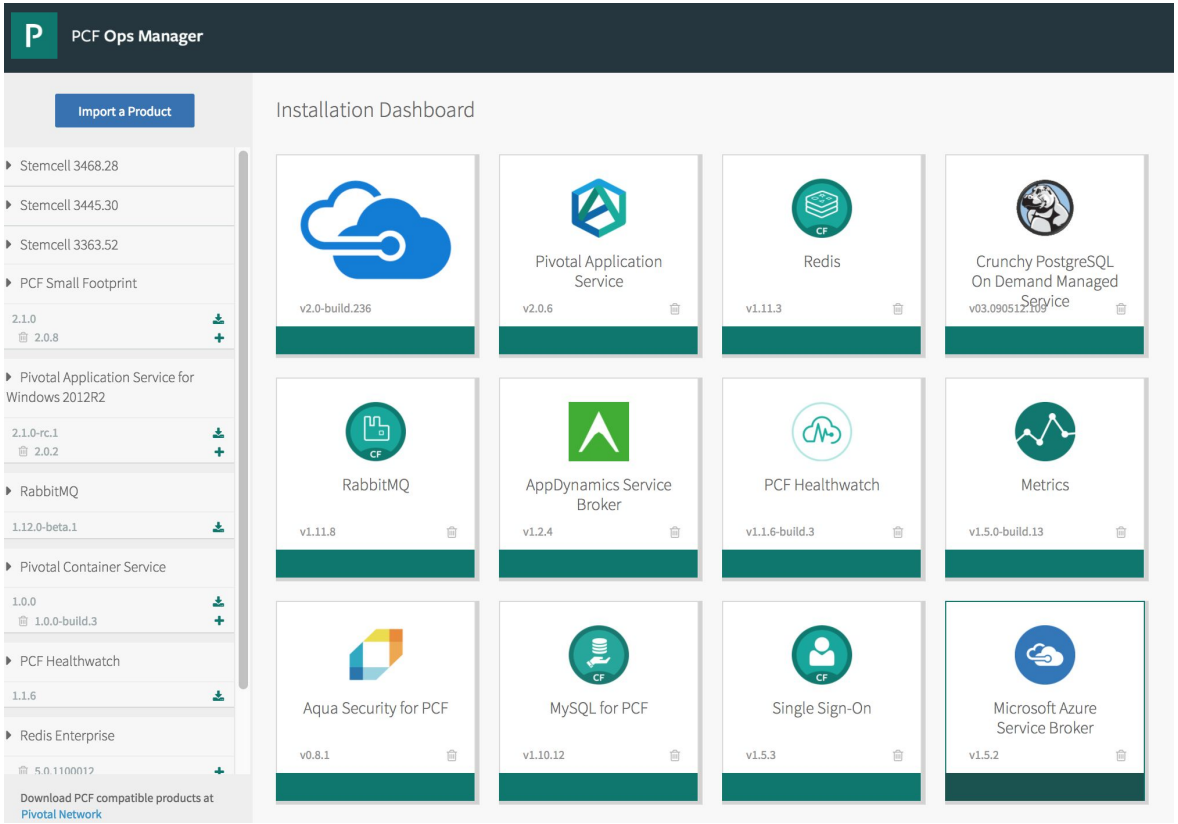
```
9332 cf-0c20d18b3ed338e0e0a2.yml
```

```
    enable: false
- name: mysql
  release: cf-mysql
  consumes: {}
  provides:
    internal-mysql-database:
      as: direct-mysql
  properties:
    cf_mysql:
      mysql:
        port: 3306
        cli_history: true
        interrupt_notify_cmd: "/var/vcap/jobs/send-email/bin/run"
        innodb_buffer_pool_size: 2147483648
        innodb_strict_mode: true
        max_open_files: 1048576
        startup_timeout: 120
        cluster_probe_timeout:
        cluster_health:
          password: "((mysql-cluster-health-credentials.password))"
        disable_auto_sst: false
        remote_admin_access: false
        seeded_databases:
        - name: account
          username: "((pivotal-account-db-credentials.username))"
          password: "((pivotal-account-db-credentials.password))"
        - name: app_usage_service
          username: "((app-usage-db-credentials.username))"
          password: "((app-usage-db-credentials.password))"
        - name: autoscale
          username: "((autoscale-db-credentials.username))"
          password: "((autoscale-db-credentials.password))"
        - name: ccdb
          username: "((cc-db-credentials.username))"
          password: "((cc-db-credentials.password))"
        - name: credhub
          username: "((credhub-db-credentials.username))"
          password: "((credhub-db-credentials.password))"
```

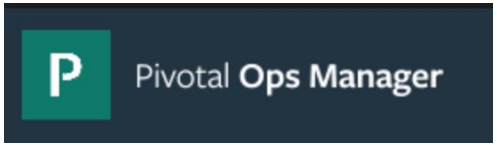
Pivotal

Pivotal Ops Manager

A gateway to managing PCF (via BOSH)



Pivotal



Demo

Pivotal

Lean Methodology

August, 2017


through lean, agile & user centered design immersion
We enable organizations
to deliver value fast, forever.
meeting business & user needs through live features
releasing early & often
ongoing at a sustainable pace
Pivotal

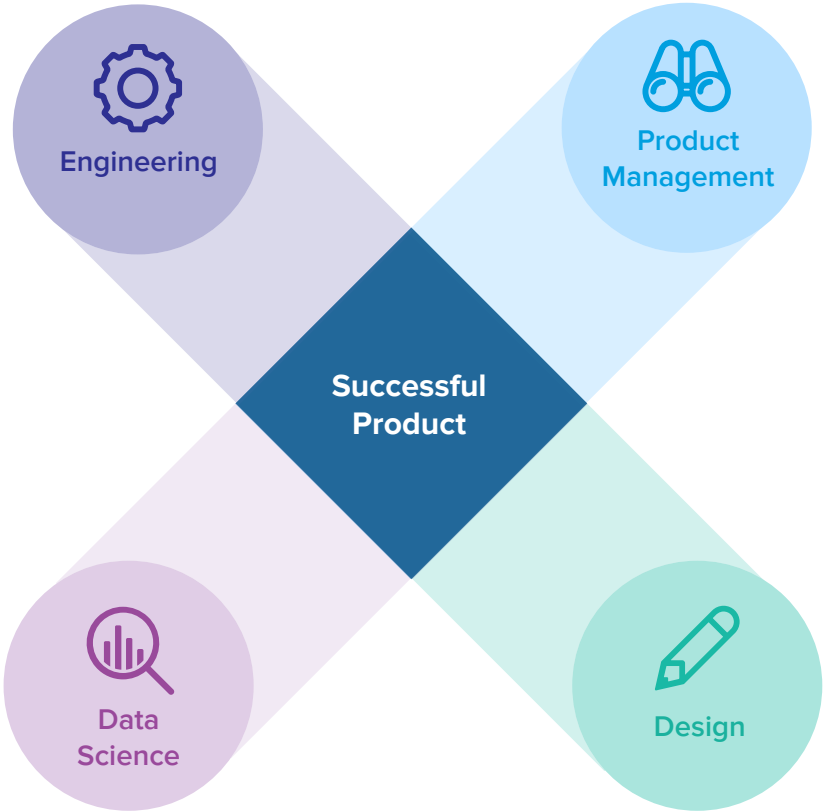
Common problems we hear from clients...

“It takes forever to release new features. We see opportunities to sell to new customers, but we **can’t respond to change** fast enough.”

“We built something for our customers but it doesn’t meet their needs. **Users aren’t adopting our product”**

“We handed off the requirements to IT, but then 6 months later we got a **product that didn’t meet our expectations.”**

Our Solution: a product focused ‘Balanced Team’



Pivotal

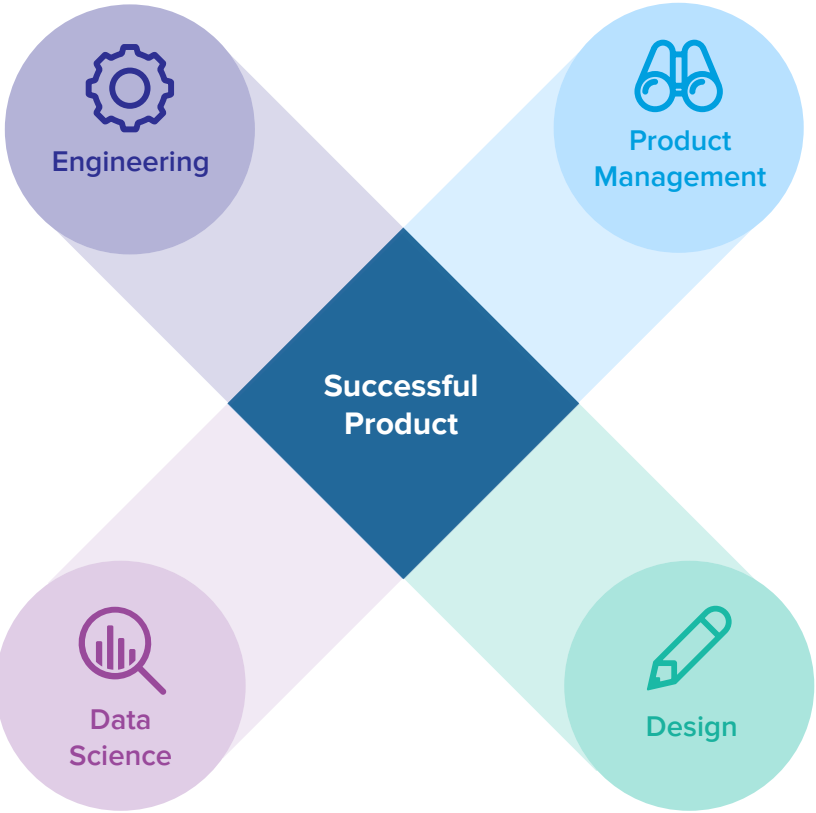
Our Solution: a product focused ‘Balanced Team’

Feasible

Can we build this?

Viable

Will this help the business?



Measurable

Can data help us make better decisions?

Desirable

Will this resonate with users?

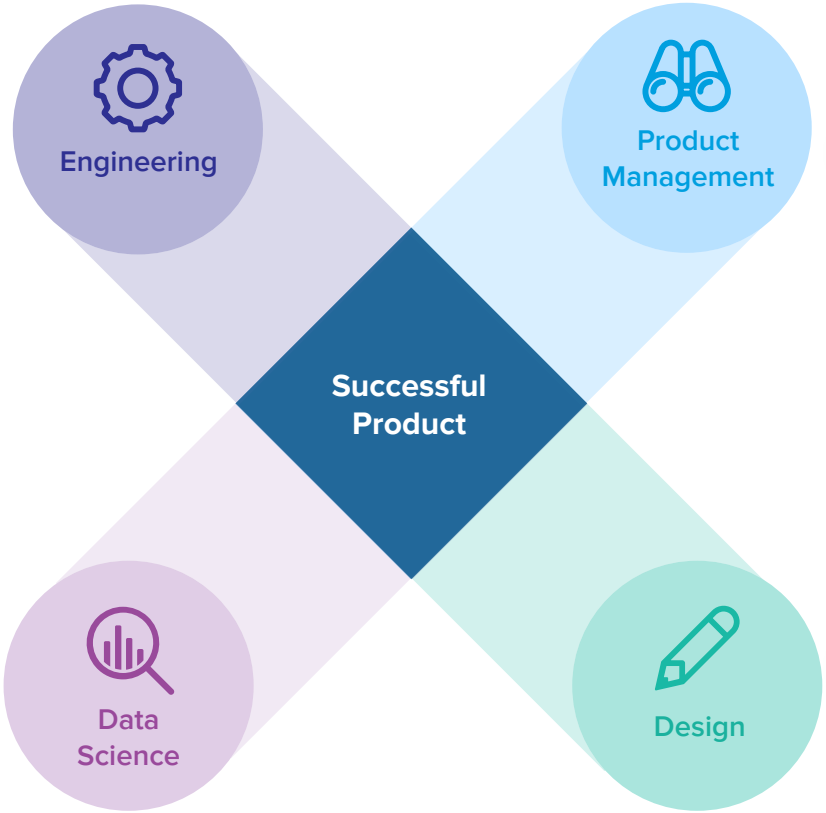
Our Solution: a product focused ‘Balanced Team’

Feasible

“How can we build a system that will respond in the face of changing requirements?”

Viable

“Are we creating valuable business outcomes? How might we measure those outcomes?”



Measurable

“Where can data add value? What actionable insights can we learn from data?”

Desirable

“What problems are users facing today? What would solve them? Will they adopt this product?”

Engineering

Extreme Programming

Building working software at a consistent speed and quality in the face of changing requirements.

PRACTICES

- Pair Programming
- Test-Driven Development
- Short iterations
- Continuous Integration / Continuous Deployment

Design

User Centered Design

Ensuring the software solves a real problem for real users in a desirable and usable product.

PRACTICES

- User Interviews
- Ethnographic studies
- Persona definition
- Prototype creation

Product Management

Lean

Reducing the risk of building the wrong thing while comfortably changing direction.

PRACTICES

- Minimum Viable Product (MVP) definition
- Lean experiments
- Identify & test assumptions
- Data driven decisions

Data Science

Data Driven

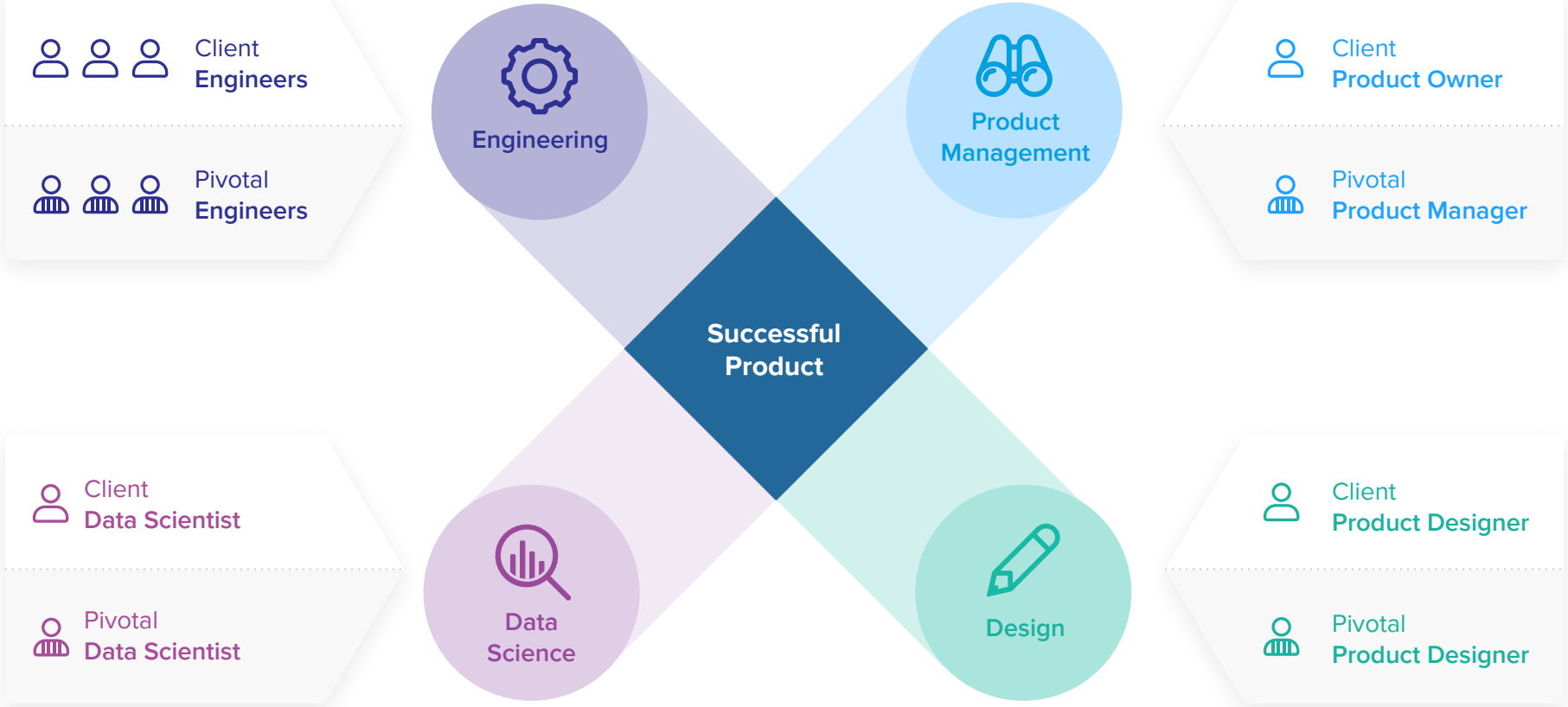
Informed decision making through data to improve the viability of the product

PRACTICES

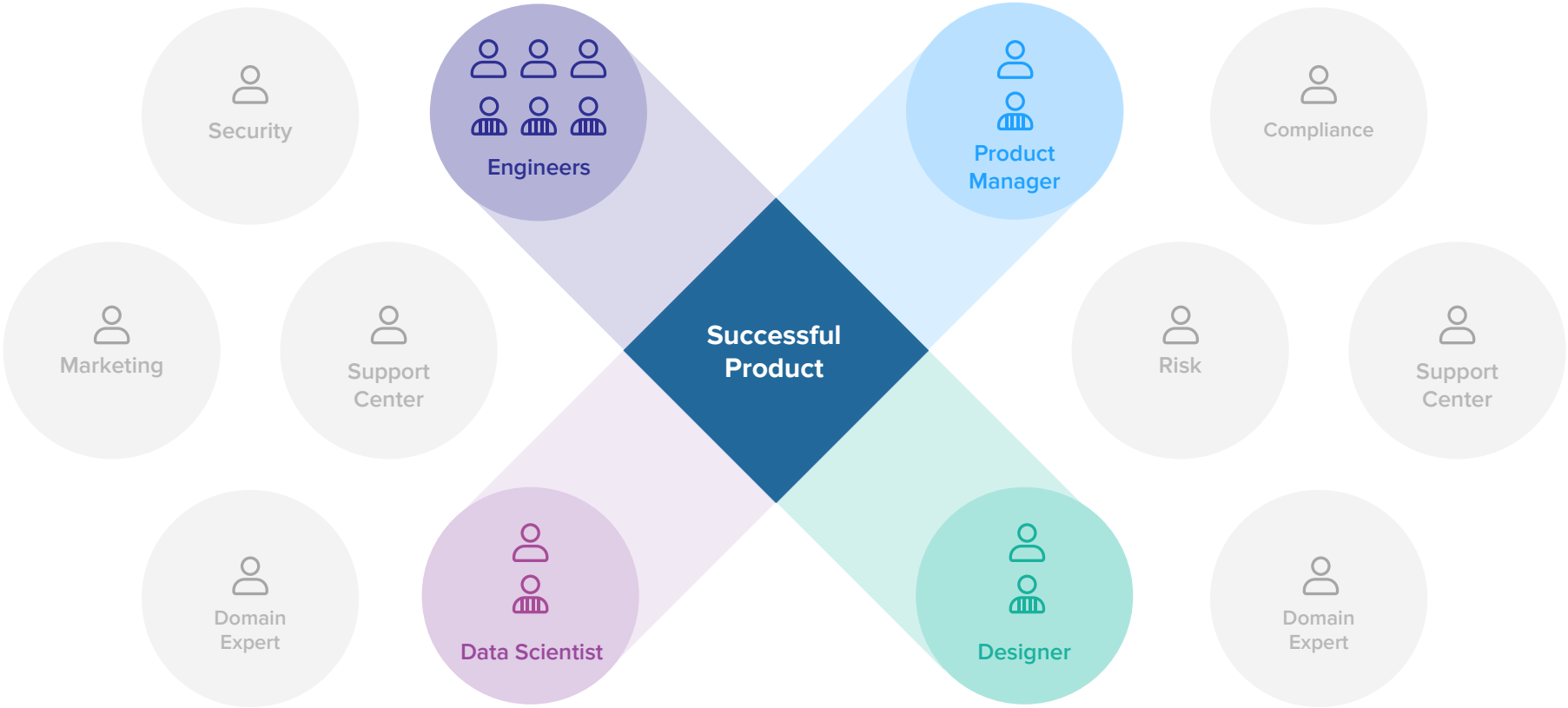
- Artificial Intelligence
- Data discovery
- Deployment to production
- Preventative analytics
- Personalisation

Pivotal

Paired Staffing Model

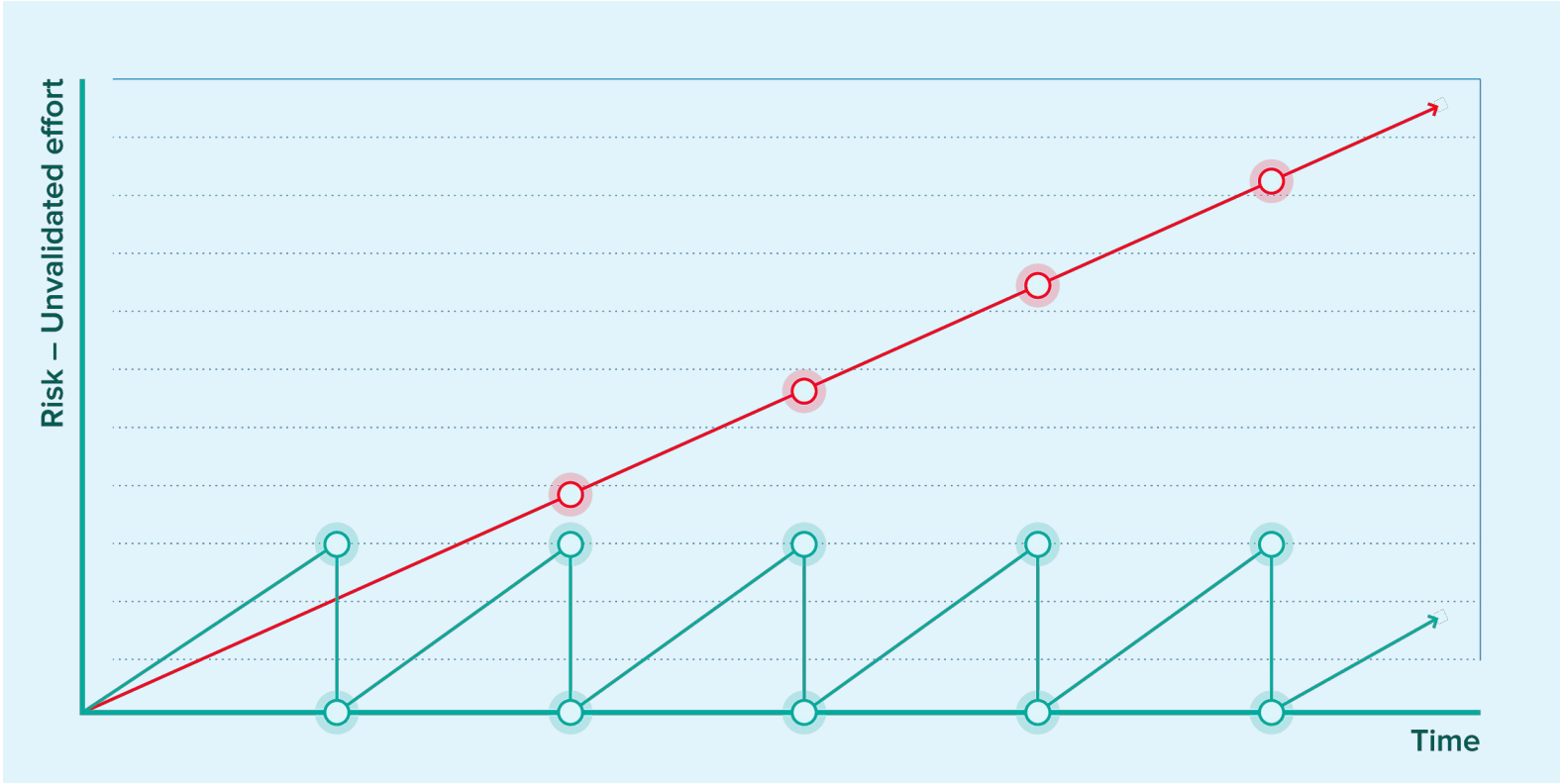


The product team composition can vary



Pivotal

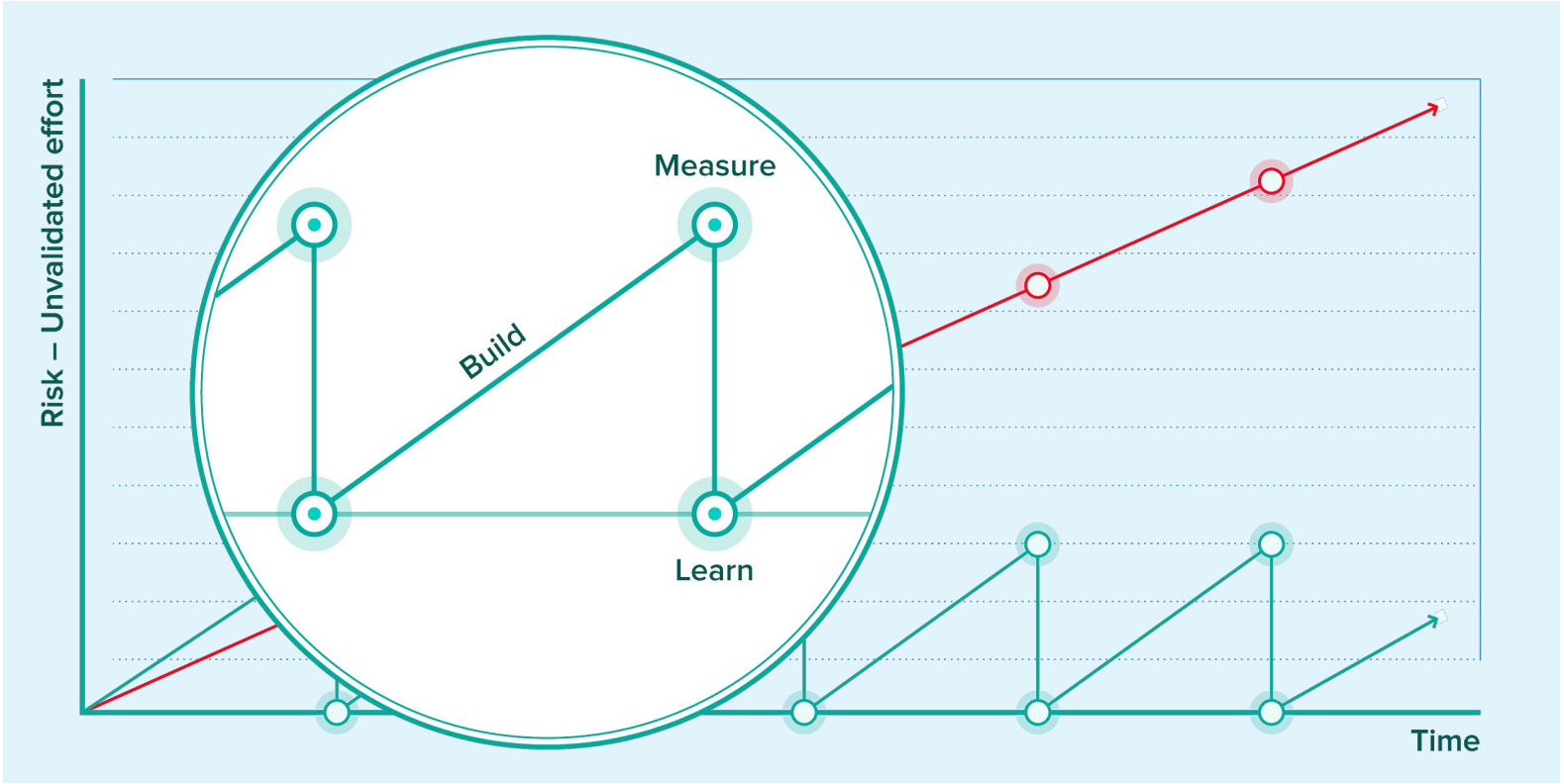
Lean Product Management Practices



Product Management

Pivotal

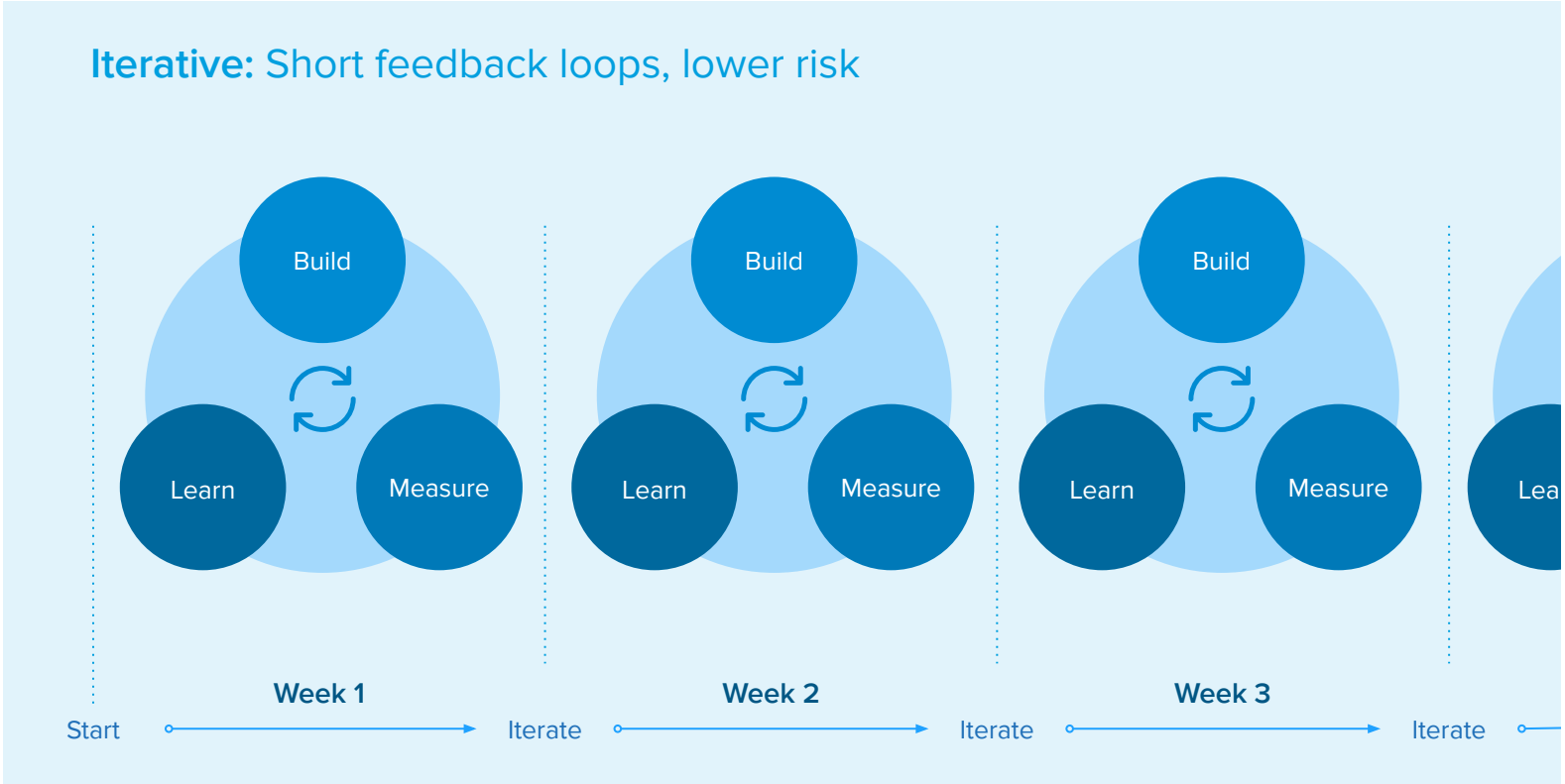
Lean Product Management Practices



Product Management

Pivotal

Agile Development

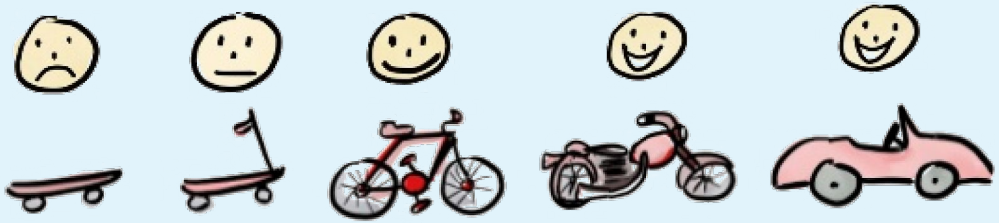


Product Management

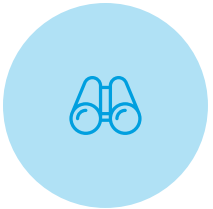
Lean Product Management Practices

Not like this...

Like this...



Created by Henrick Kniberg

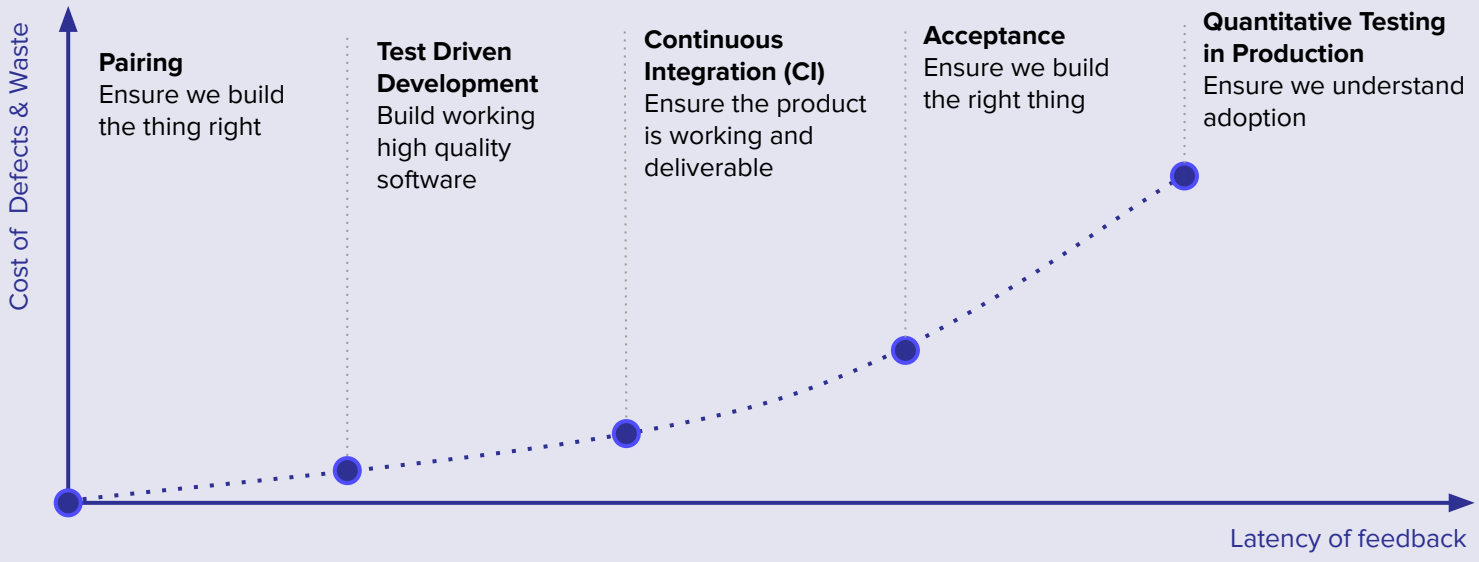


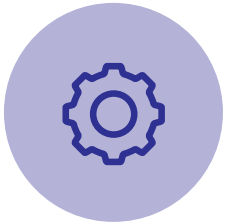
Product Management

Pivotal

Extreme Programming

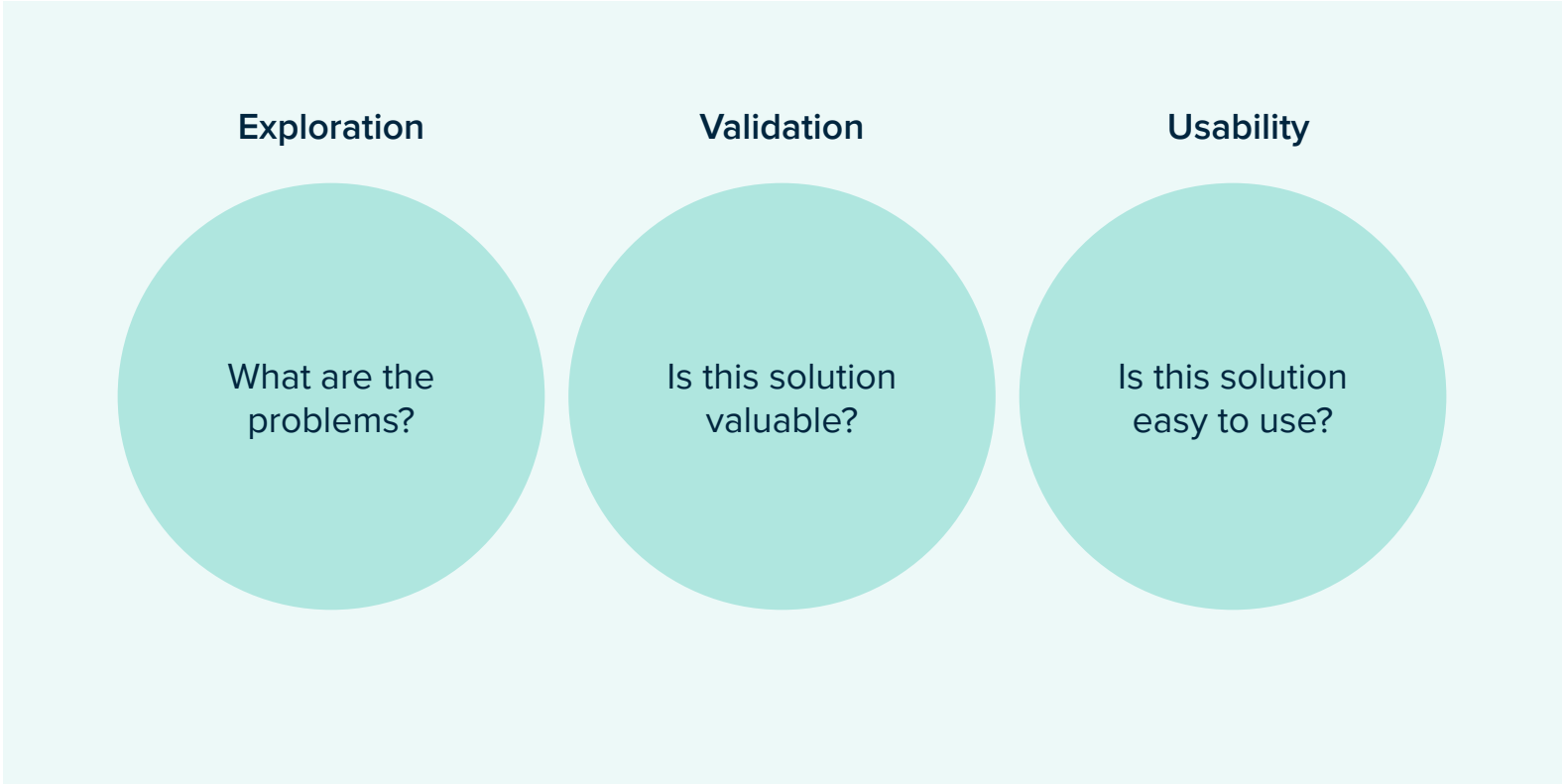
Practices to ensure development of high quality software and catch defects and reduce waste early in the development cycle



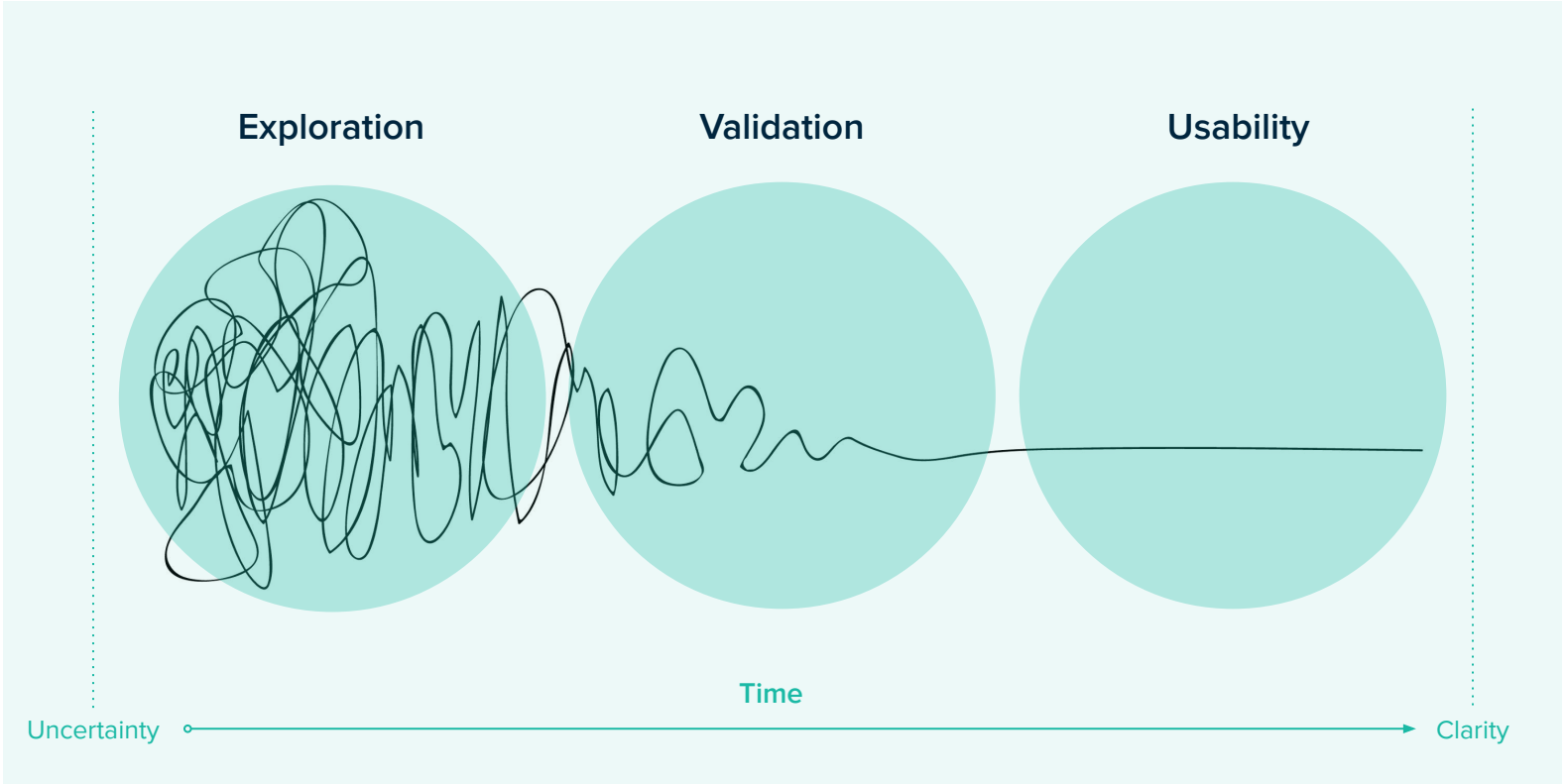


Development

User-Centered Design (UCD) Practices



User-Centered Design (UCD) Practices



Design

Pivotal

A Week-in-the-Life

Daily Standup

- One-minute meeting to discuss daily activities
- Team discuss what they did yesterday and what they'll do today

IPM

- The product manager leads the team through the backlog for that week
- The team clarifies and ensures consistency
- Stories are estimated

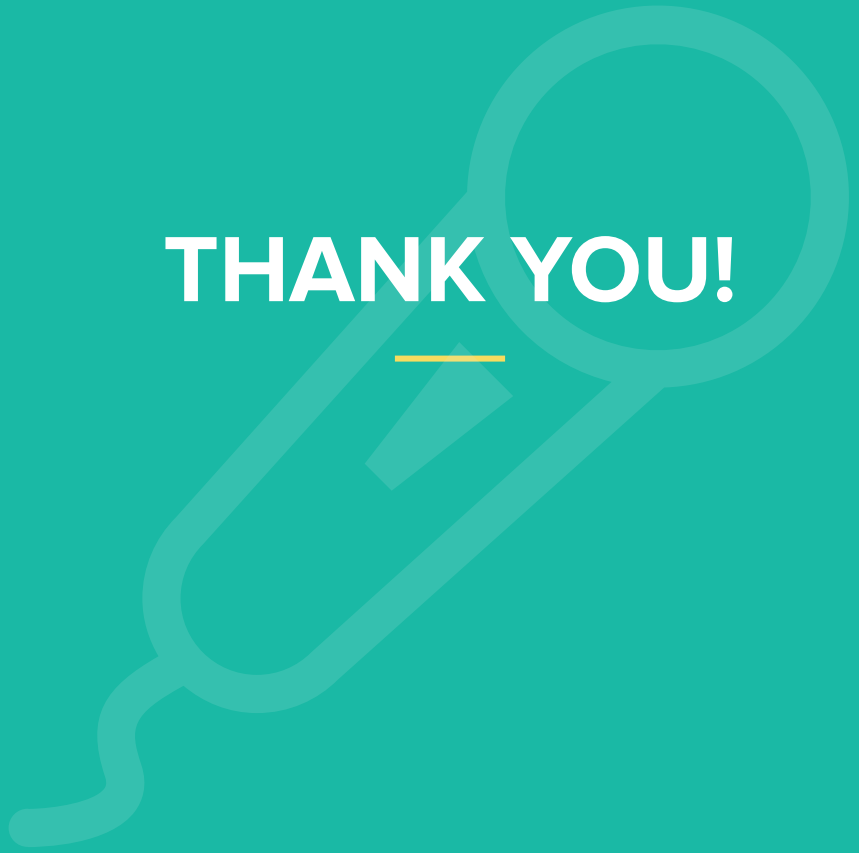
Iteration

- Product backlog and user stories are written and prioritised daily by the product manager.
- The team sit together, self-organise, and are highly collaborative
- Prototypes are built, tested, and refined by the designer
- User research eliminates unnecessary features

Retrospective

- The team meets to decompress, identify issues, and discuss areas for improvement
- Actions are captured for and reviewed weekly
- Retros allow teams to continuously improve and iterate the agile process

THANK YOU!



Pivotal