



# Pivotal.

## BOSH: running platforms so you can run platforms on your platforms

---

Paul Czarkowski / @pczarkowski  
Principal Technologist  
Pivotal software



# Pivotal.

## Successful Patterns for Deploying [and Operating] Platforms.

---

Paul Czarkowski / @pczarkowski  
Principal Technologist  
Pivotal software

A man with glasses, wearing a dark t-shirt with a white geometric logo (the Cloud Foundry logo), is speaking to a group of people. He is gesturing with his hands. The background is dark and out of focus. Two horizontal teal lines are positioned above and below the text.

**Do you know Cloud Foundry?**

A man with glasses, wearing a black t-shirt with a white geometric logo (the Kubernetes logo), is speaking to a group of people. He is gesturing with his hands. The background is dark and out of focus. Two horizontal teal lines are positioned above and below the text.

**Do you know Kubernetes ?**



**Do you know BOSH ?**

# What is a platform ?

A **computing platform** or **digital platform**<sup>[1]</sup> is the environment in which a piece of **software** is executed. It may be the **hardware** or the **operating system** (OS), even a **web browser** and associated **application programming interfaces**, or other underlying software, as long as the **program code** is executed with it. Computing platforms have different **abstraction** levels, including a **computer architecture**, an OS, or **runtime libraries**.<sup>[2]</sup> A computing platform is the stage on which computer programs can run.

A platform can be seen both as a constraint on the **software development process**, in that different platforms provide different functionality and restrictions; and as an assistance to the development process, in that they provide low-level functionality ready-made. For example, an OS may be a platform that abstracts the underlying differences in hardware and provides a generic command for saving **files** or accessing the **network**.

[https://en.wikipedia.org/wiki/Computing\\_platform](https://en.wikipedia.org/wiki/Computing_platform)

A person is seated at a table in a meeting, gesturing with their hands while speaking. They are wearing a dark t-shirt with a geometric logo. Two other people are visible in the foreground, seen from the back, listening to the speaker. The background is a plain wall.

**A modern software  
platform provides  
API driven compute  
resources.**

---

"Every IT team  
is a  
Platform Team."

---



**Gabe Monroy**

@gabrtv Follows you

Head of Product, Azure Containers.  
CNCF board member. "My life is dope  
and I do dope shit."



# Kubernetes is a Platform Platform



**Joe Beda** ✓

@jbeda

Joe Beda  
CTO, Heptio



Photo by U.S. Navy photo by Chief Journalist Joe Furell

1/2/21



**Kelsey Hightower** 

@kelseyhightower

Follow



Kubernetes is a platform for building platforms. It's a better place to start; not the endgame.

1:04 PM - 27 Nov 2017

153 Retweets 461 Likes



# Generic Platform

Users



API

Artifacts

Database

Storage

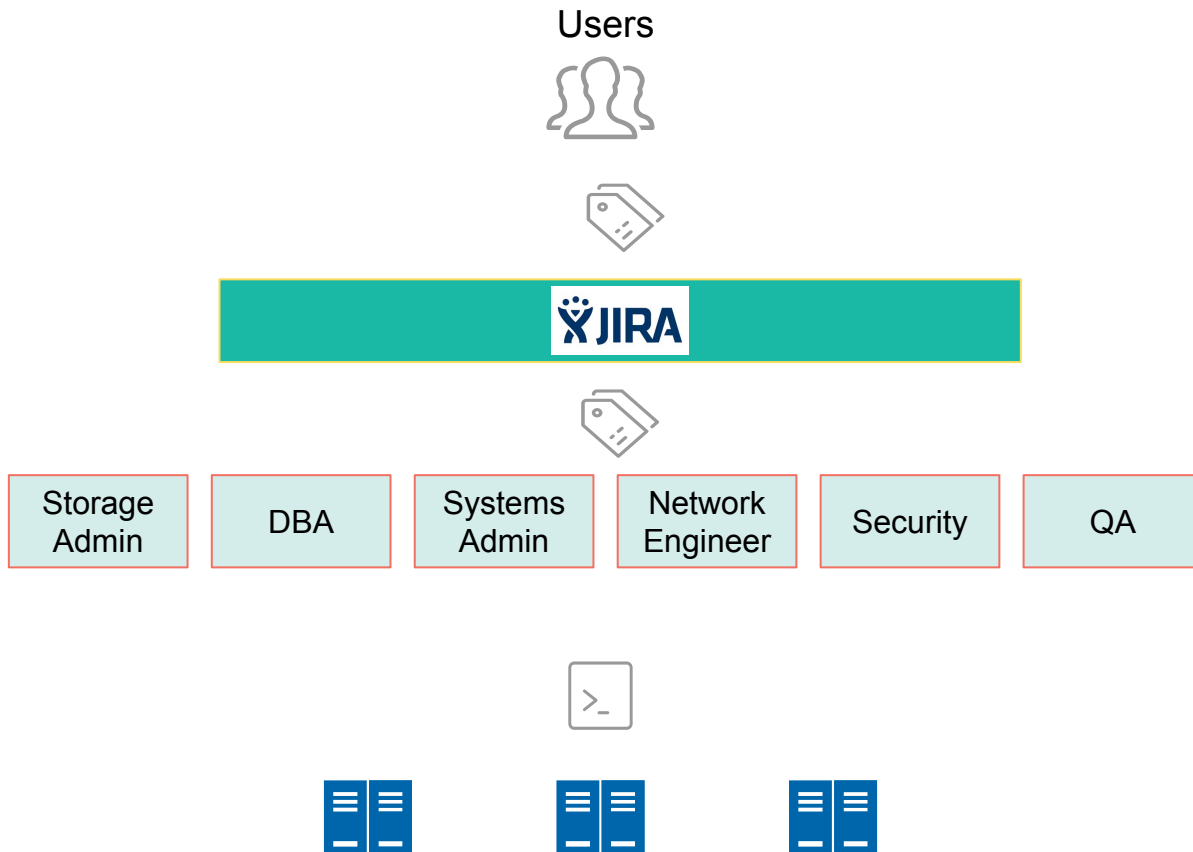
Compute

Network

Access



# Enterprise IT



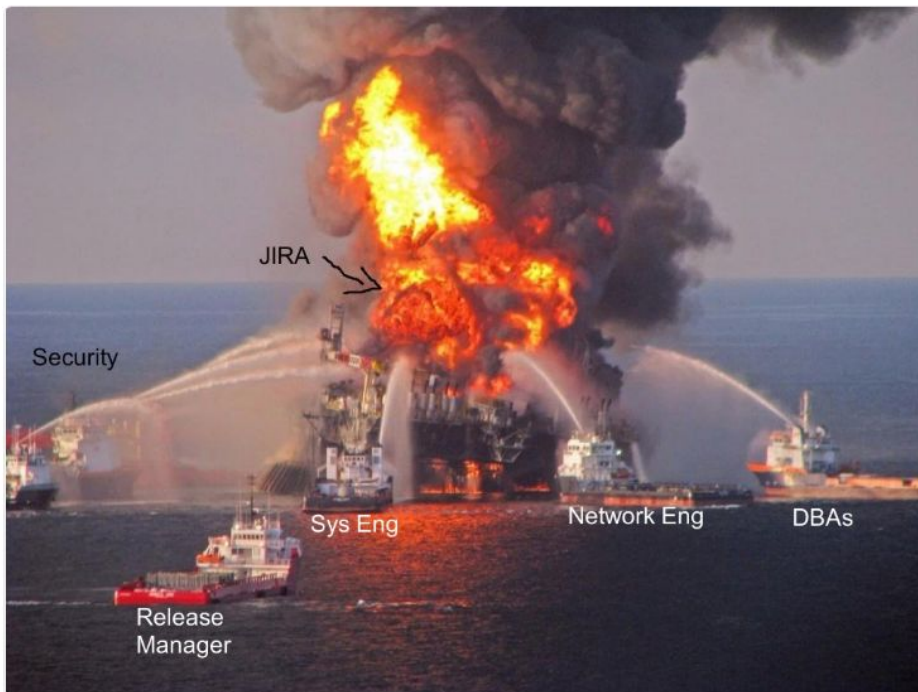


Czarcloudski

@pczarkowski




## Enterprise DevOps



11:16 AM - 11 Jun 2018

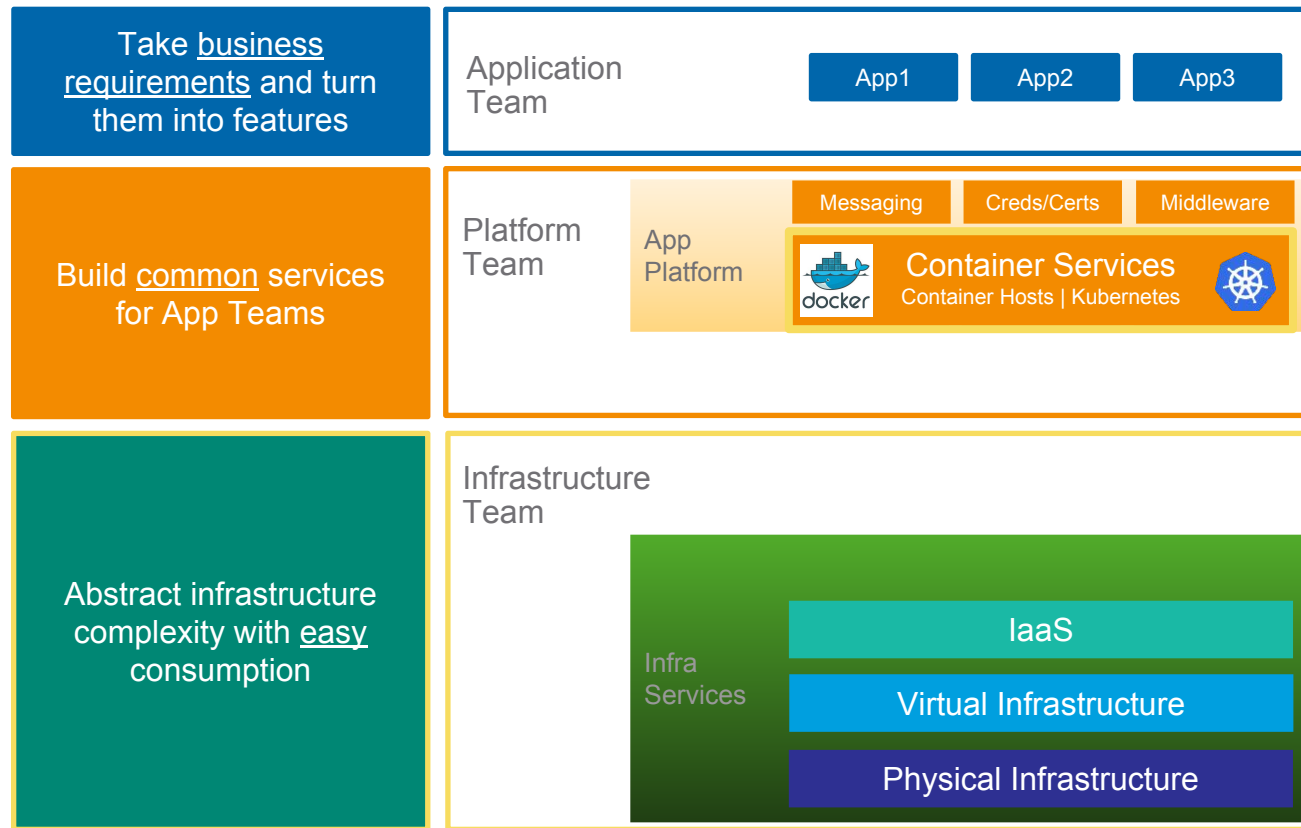
1,217 Retweets 2,307 Likes



A man with glasses is speaking to a group of people in a meeting. He is wearing a dark t-shirt with a geometric logo. The background is dark and out of focus. The text is overlaid on the image, centered, and is white. There are two horizontal teal lines, one above and one below the text.

**“You don’t have a platform  
problem, you have a culture  
problem.”**

# Evolve your IT teams!

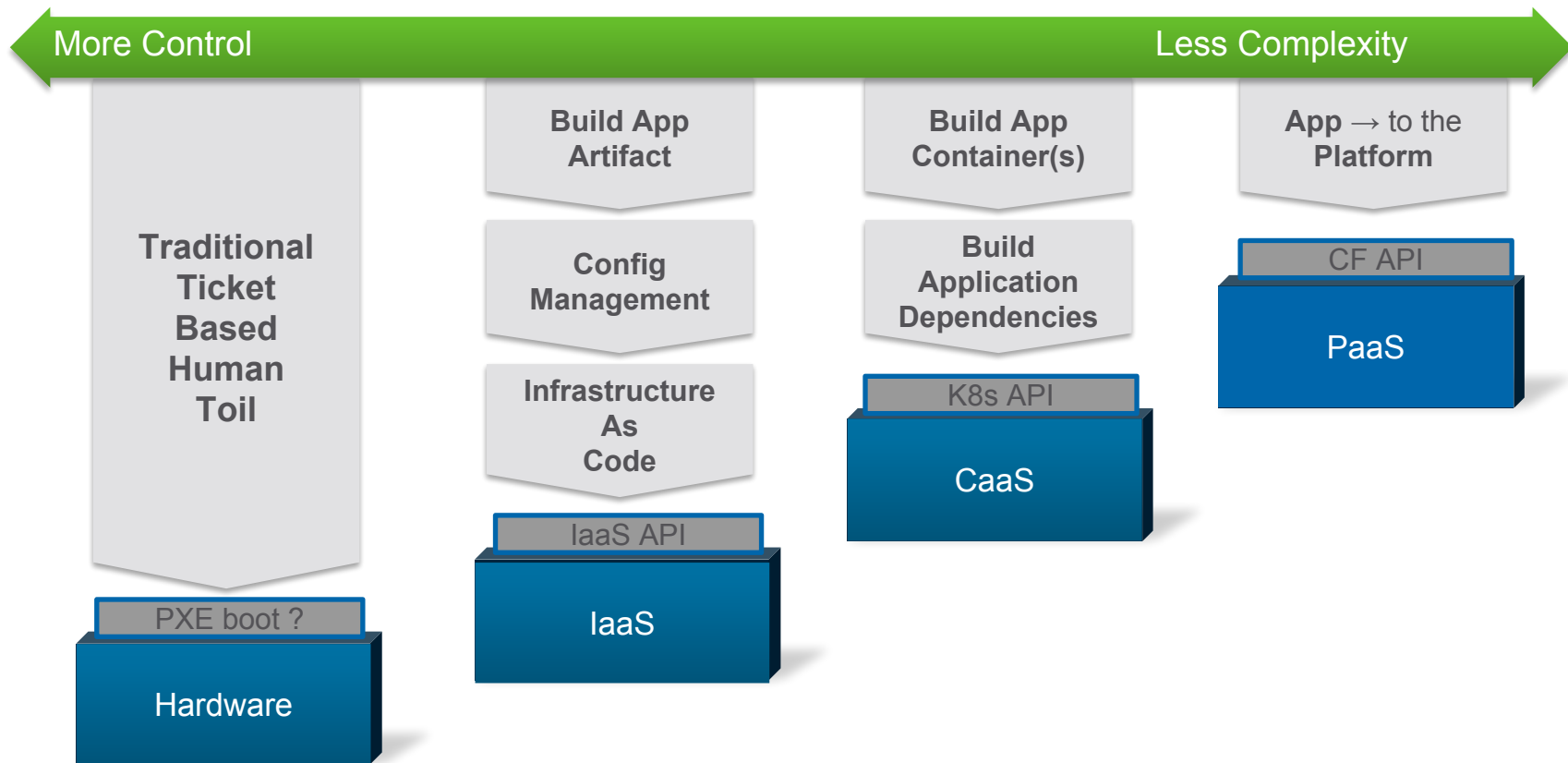


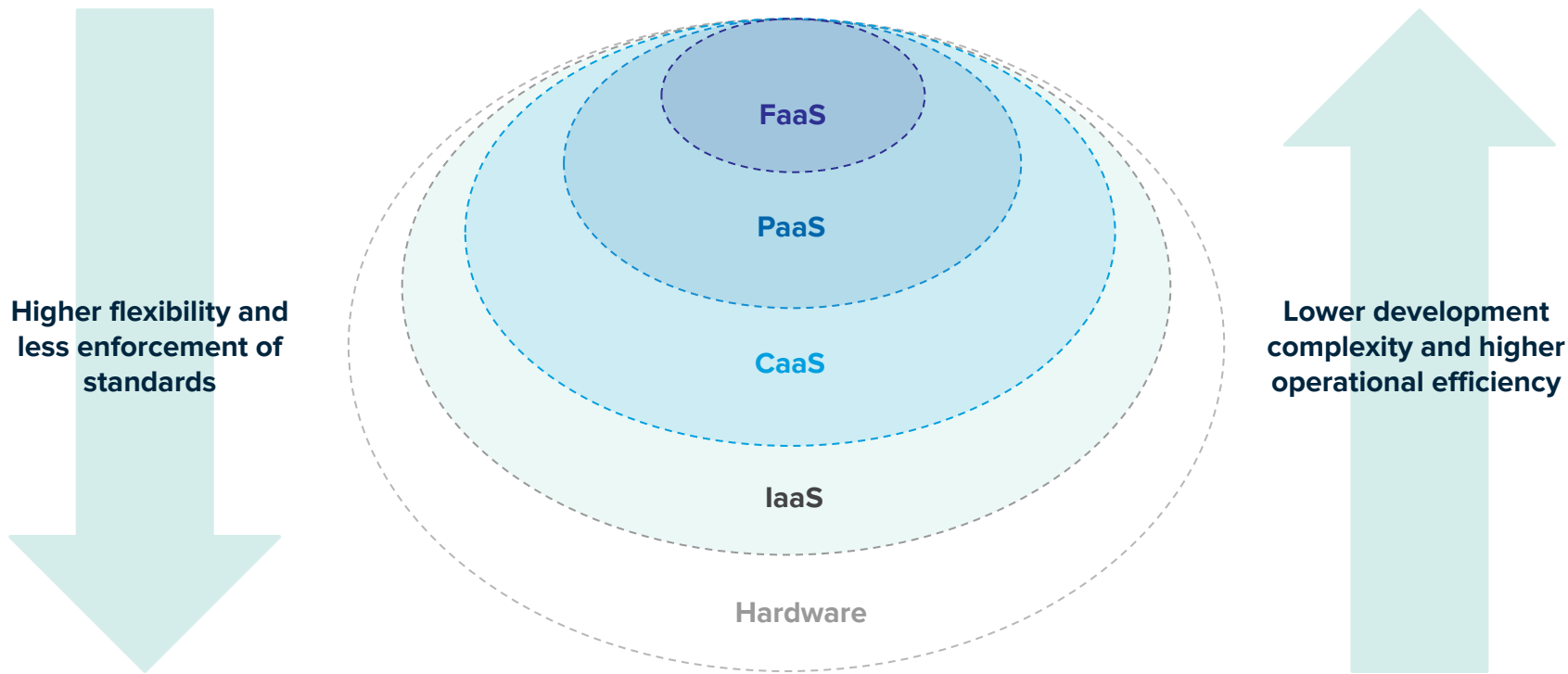


**People build Platforms**  
**People build Apps**  
**Apps run on Platforms**

People are the most  
important component  
of any platform.

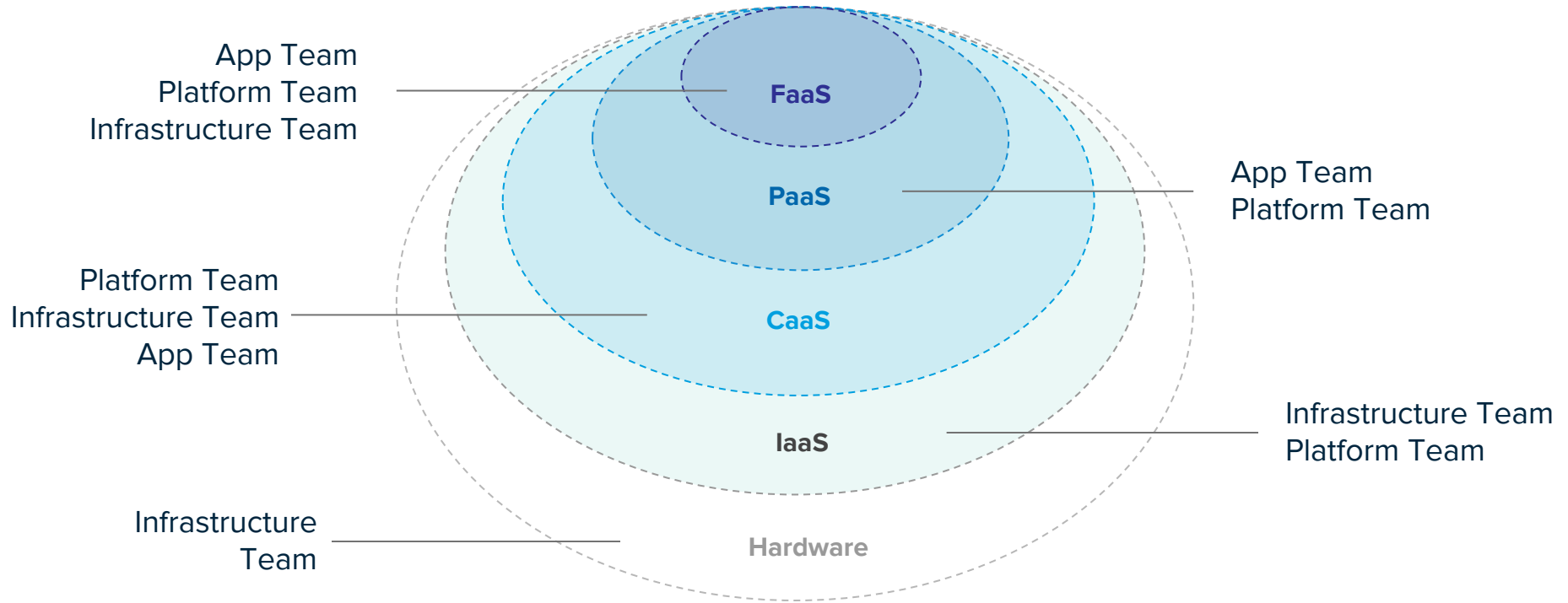
# What Abstraction is the Right One for You ?





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

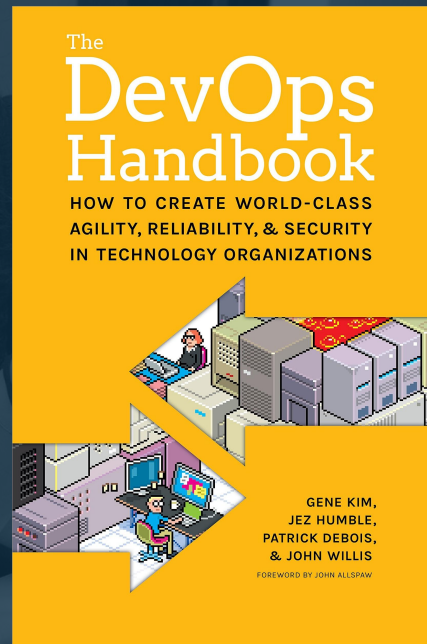
## Consumers of the abstraction

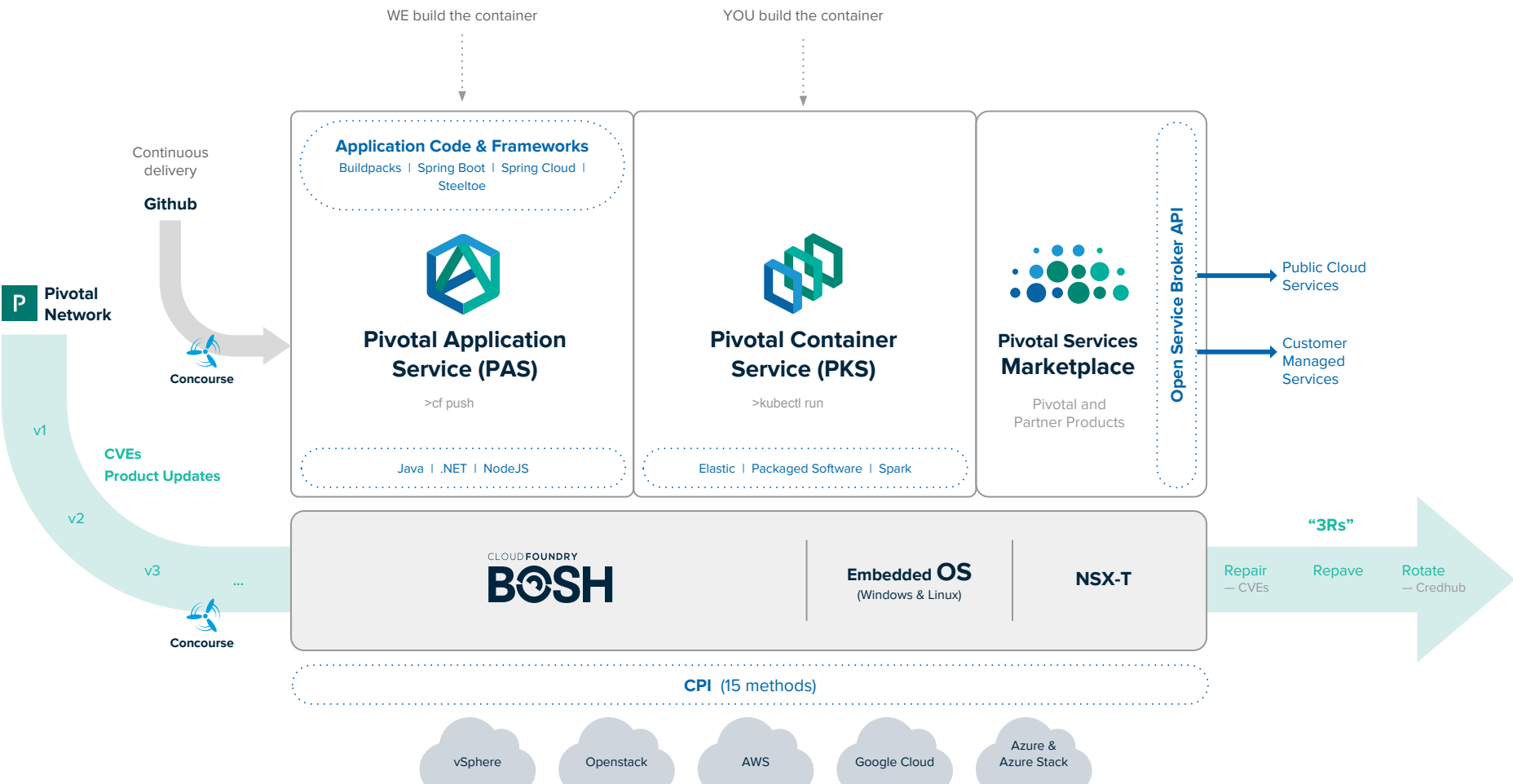


---

"In general, taking something that's already working somewhere and expanding its usage (capabilities) is far more likely to succeed than building these capabilities from scratch"

---







what is borg?



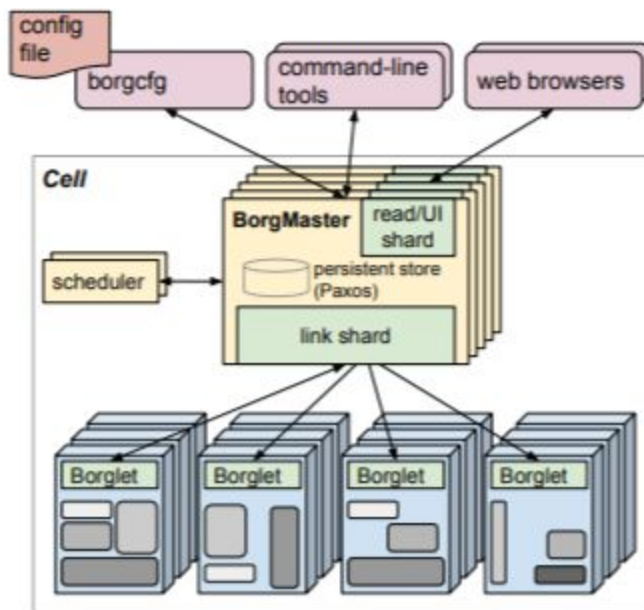
Google Search

I'm Feeling Lucky

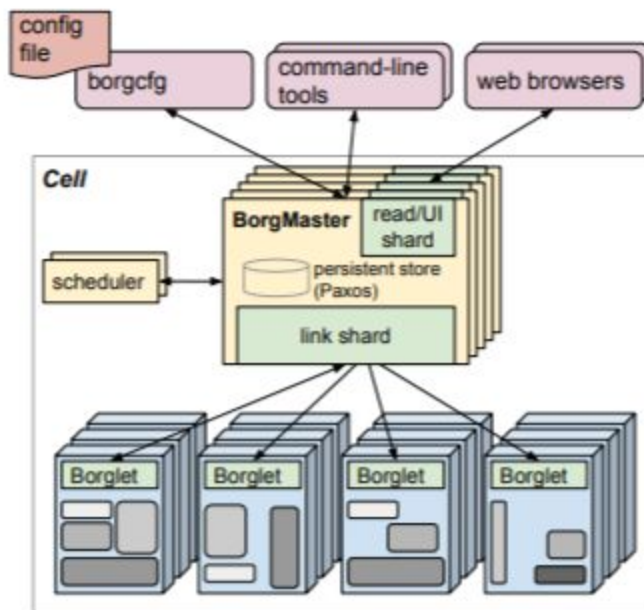
# The Google Problem



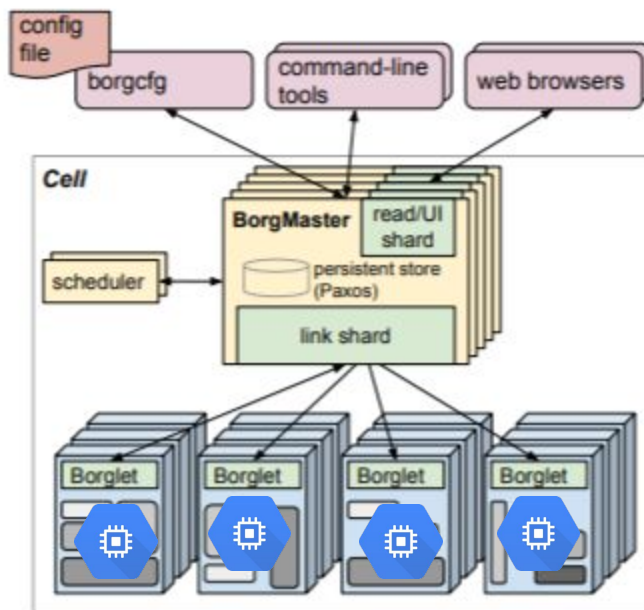
x 1,000,000



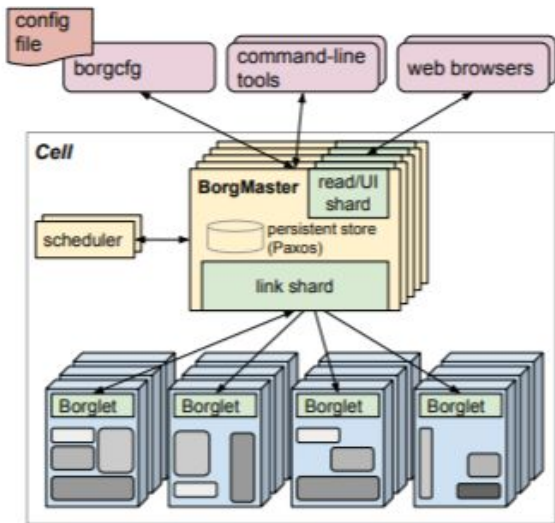
**Figure 1:** The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*



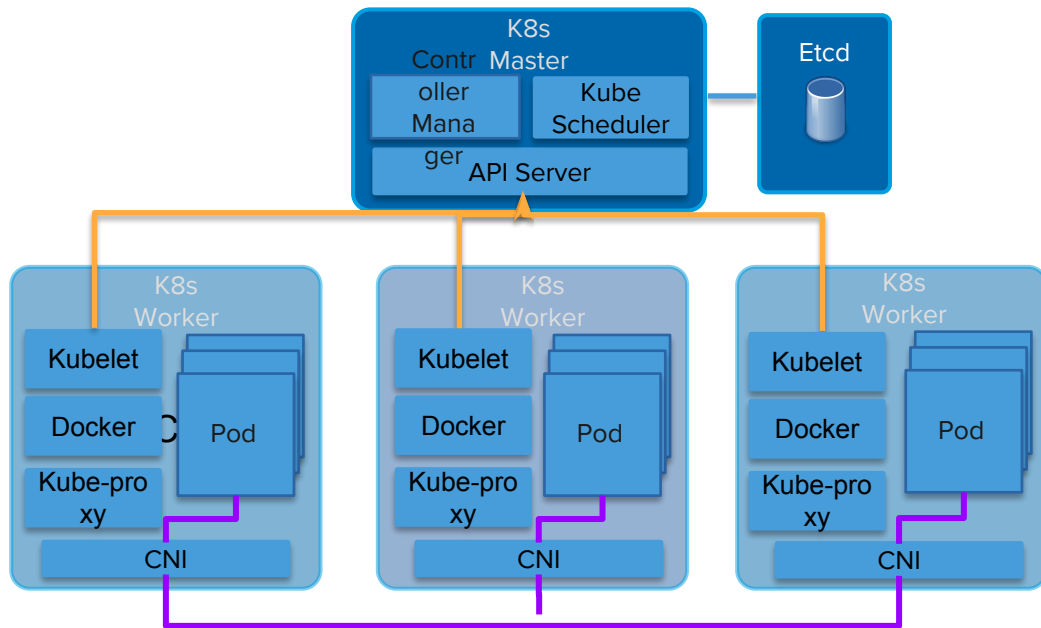
**Figure 1:** The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*






**Figure 1:** The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

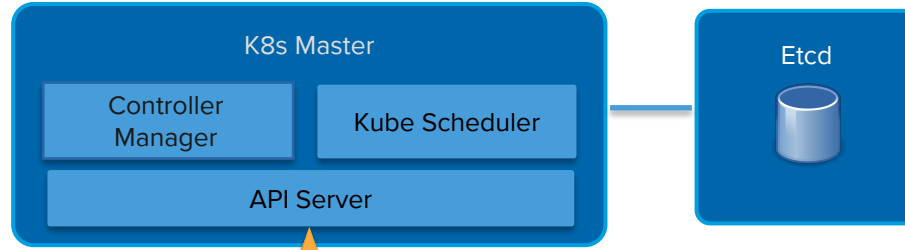


**Figure 1:** The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

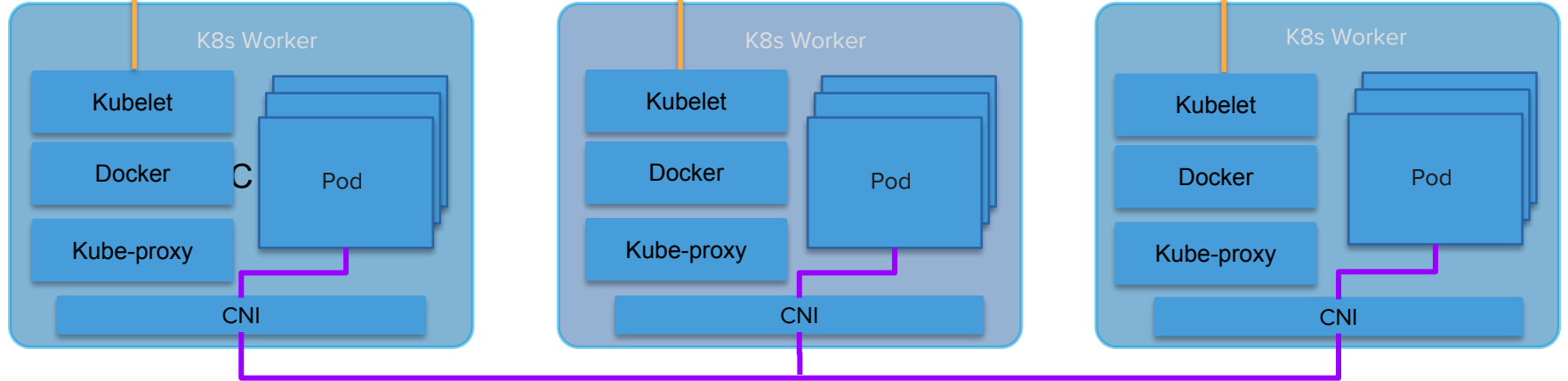


<input type="checkbox"/>	Name ^	Zone
<input type="checkbox"/>	 gke-example-gke-cluster-default-pool-19d0ce49-nn28	us-central1-a
<input type="checkbox"/>	 gke-example-gke-cluster-default-pool-19d0ce49-xkt3	us-central1-a
<input type="checkbox"/>	 gke-example-gke-cluster-default-pool-19d0ce49-zp7l	us-central1-a

Google Compute Engine  
*(hidden from user)*



Google Compute Engine

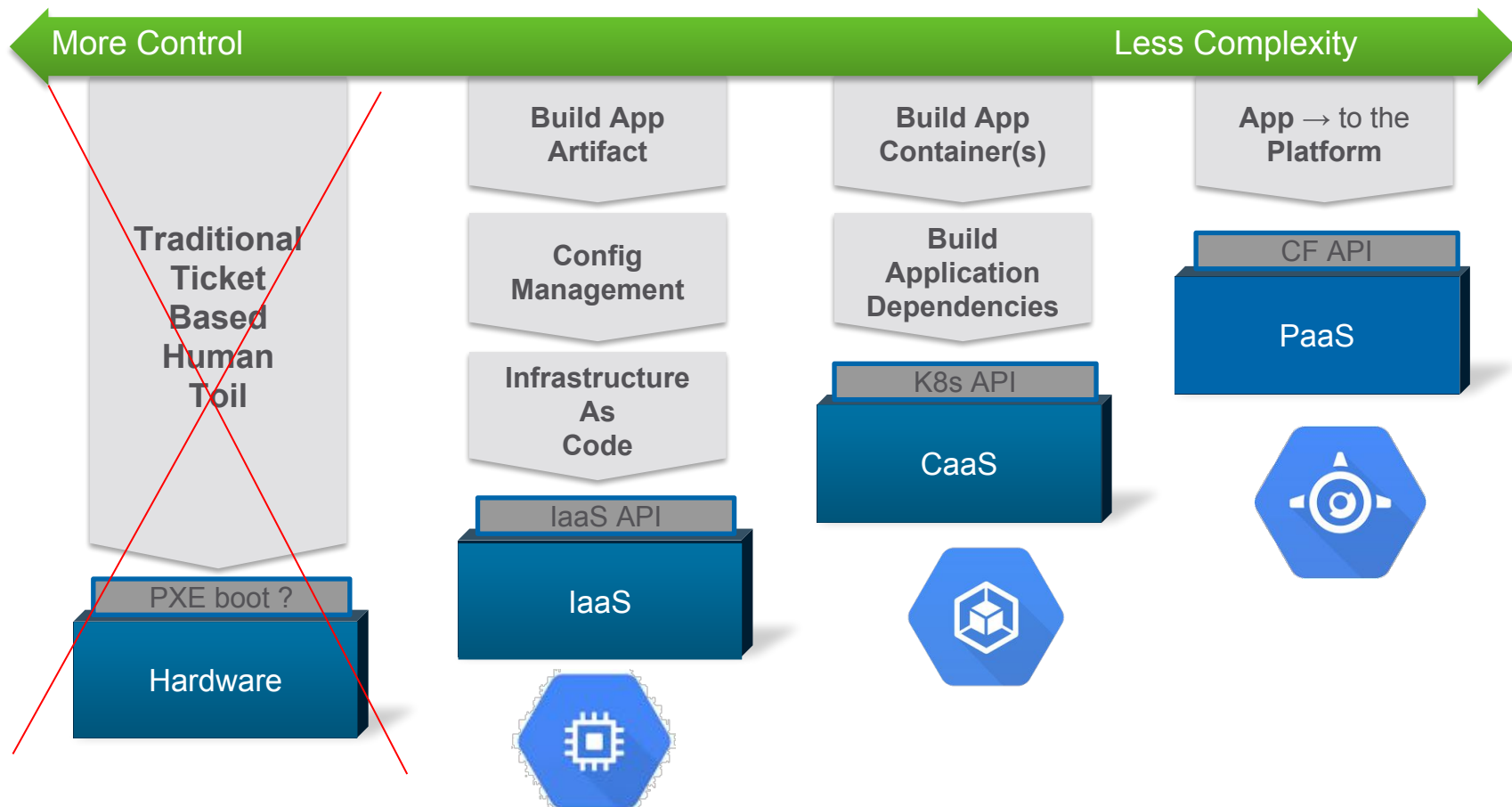


**I HEARD YOU LIKE PLATFORMS**

**SO I BUILT A PLATFORM ON A PLATFORM  
SO I COULD RUN A PLATFORM.**

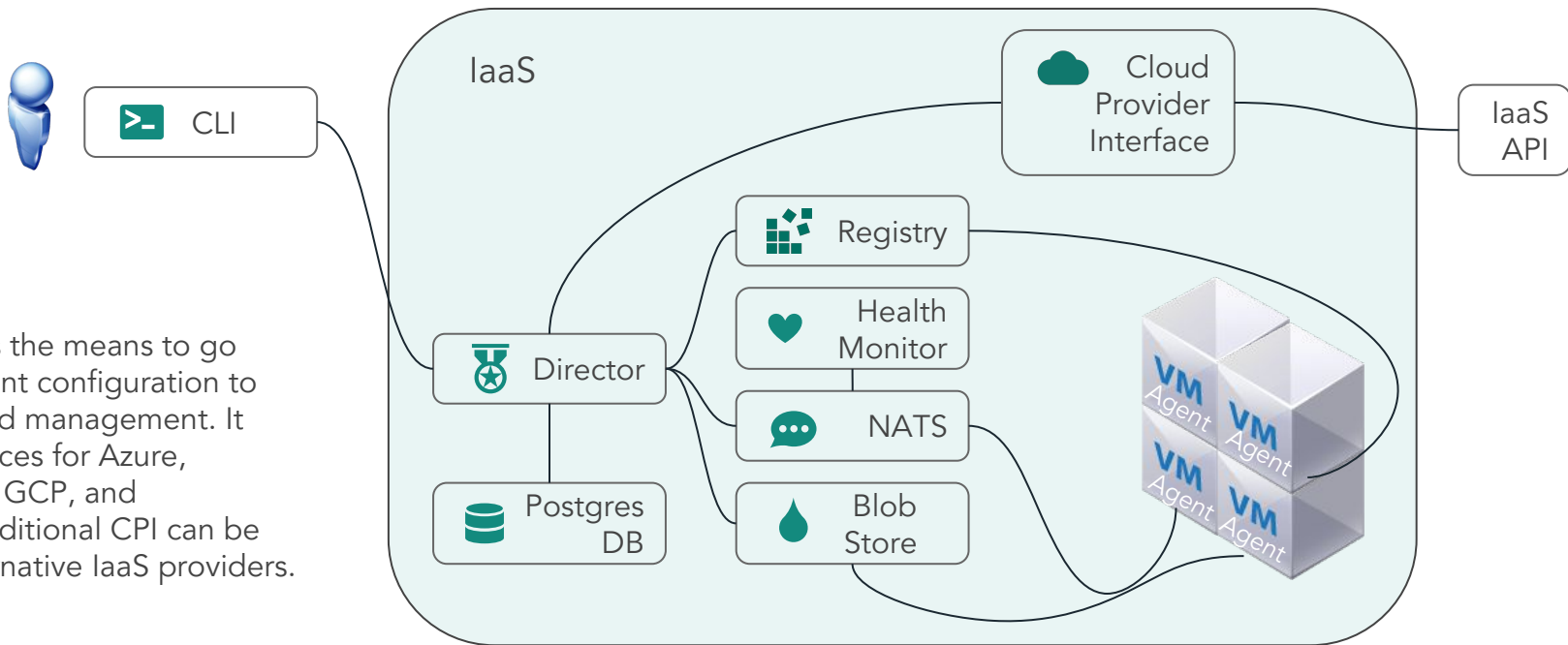
imgflip.com

# What Abstraction is the Right One for You ?



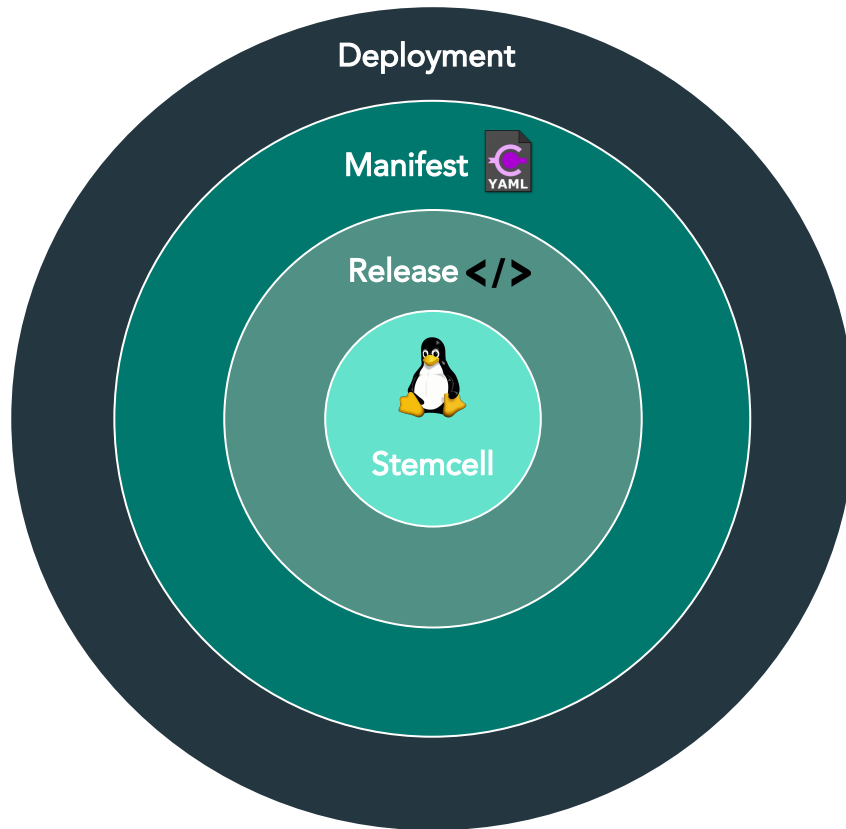


# BOSH - Component Architecture

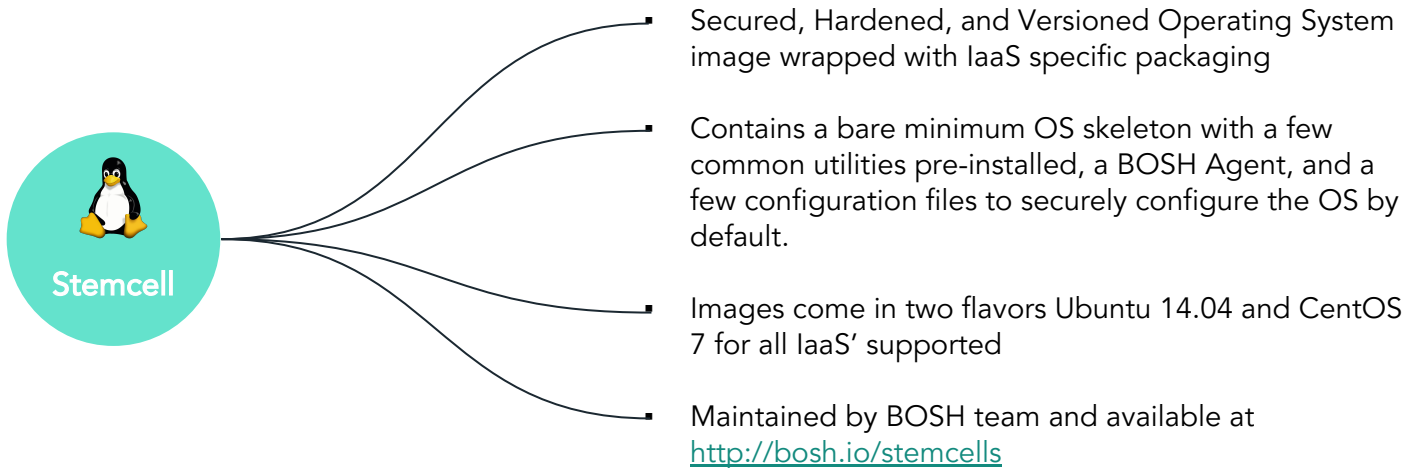


**BOSH** provides the means to go from deployment configuration to VM creation and management. It includes interfaces for Azure, vSphere, AWS, GCP, and OpenStack. Additional CPI can be written for alternative IaaS providers.

# BOSH - Service Deployment



# BOSH - Stemcell



# BOSH - Release

## Elements:

**Jobs** - Pieces of the service or application you are releasing, including how to compile & run them

**Packages** - Provide source code and dependencies to jobs

**Src** - Non-binary files which is provided to packages

**Blobs** - Provide binary files (other than those checked into a source code repository) to packages

**Monit** - Script utilized to start/stop/restart the job

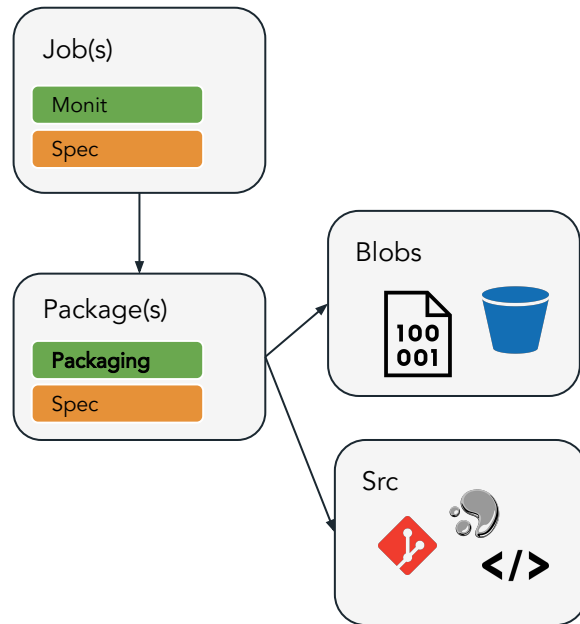
**Packaging** - Script utilized to compile the source needed by a job

**Spec** - Key/Value file which stores all configuration properties which can be set externally

## Organization:

```
$ tree .
.
├── blobs
├── config
│   └── blobs.yml
├── jobs
│   ├── bg_worker
│   │   ├── monit
│   │   ├── spec
│   │   └── templates
│   └── web_ui
│       ├── monit
│       ├── spec
│       └── templates
├── packages
└── src
9 directories, 5 files
```

## Dependency Tree:



Green - Script (erb or bash)

Orange - Properties (yaml)

# BOSH - Manifest

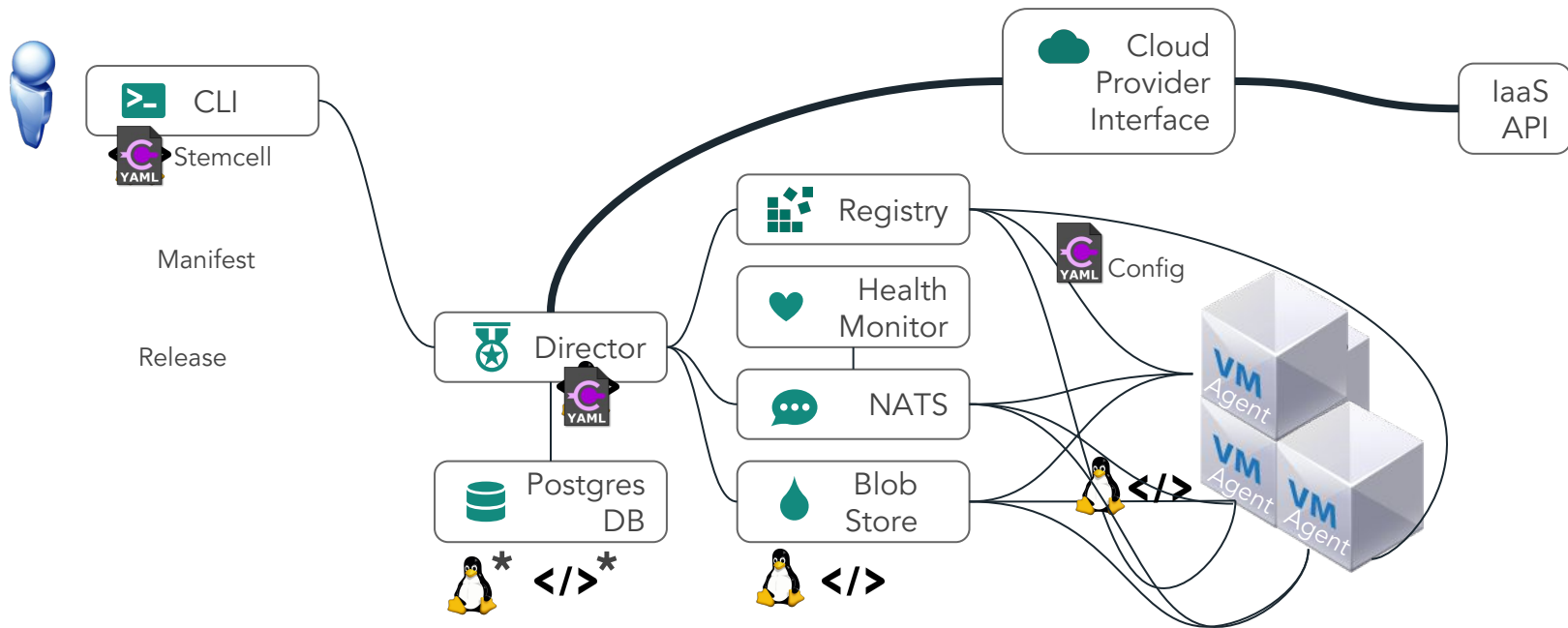
Provides the ability to customize BOSH releases (your service)

YAML - Primer found at end of presentation

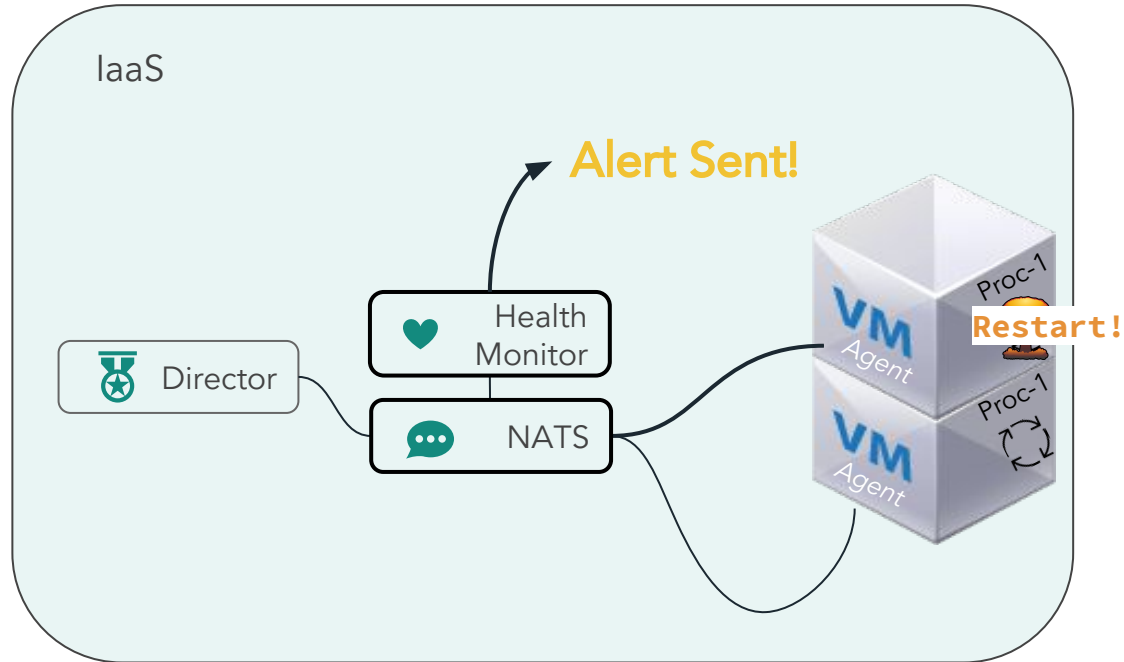
Required Blocks:

- Deployment Identification: A name for the deployment and the UUID of the Director managing the deployment
- Releases Block: Name and version of each release in a deployment
- Stemcells Block: Name and version of each stemcell in a deployment
- Update Block: Defines how BOSH updates instances during deployment
- Instance Groups Block: Configuration and resource information for instance groups (Jobs)
- Properties Block: Describes global properties and generalized configuration information

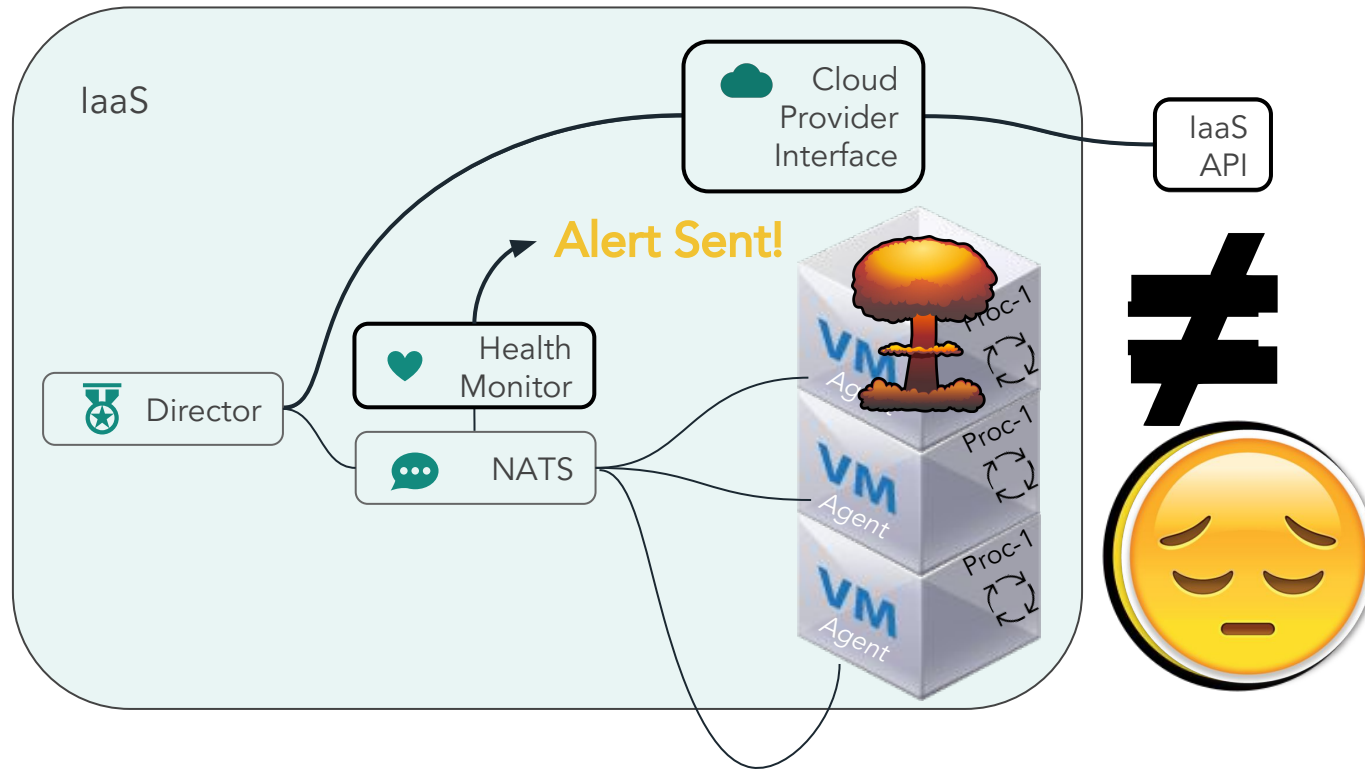
# BOSH - Service Deployment



# BOSH - Process High Availability



# BOSH - VM High Availability



Manifest - Desired State



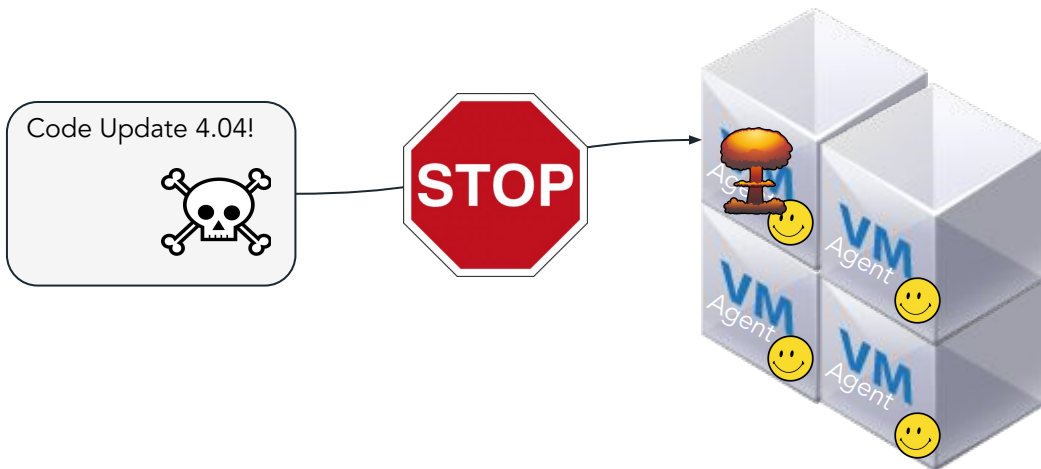
# BOSH - Canary Upgrades

Manifest -



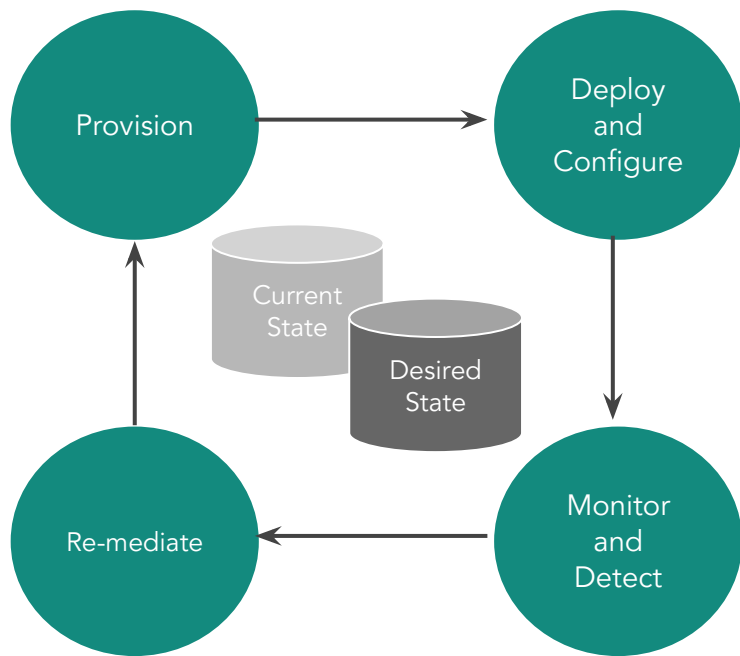
```
33 update:
34   canaries: 1
35   canary_watch_time: 30000-300000
36   max_errors: 2
37   max_in_flight: 1
38   update_watch_time: 30000-300000
```

Any update error causes the deployment to stop. Since only canaries are affected before an update stops, problem jobs and packages are prevented from taking over all instances.



# BOSH - Day 2 Ops

Consistent, Reliable, Scalable, Secure

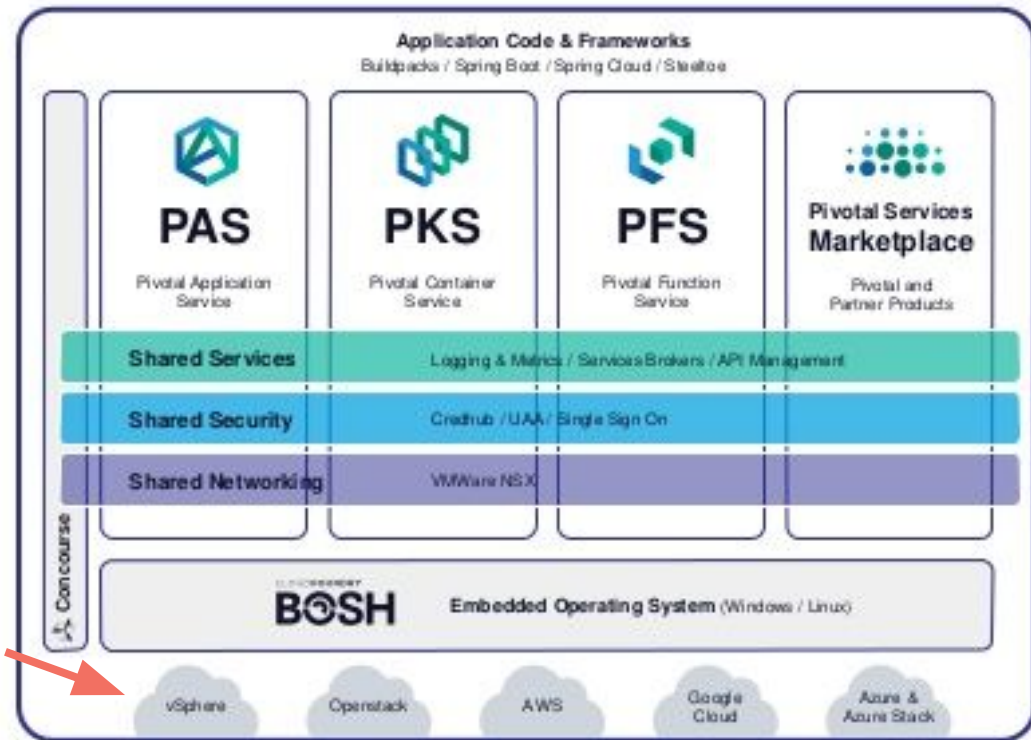
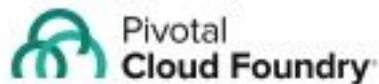


- Checks against "desired state to return consistency
- No ad hoc automation burden
- Manage services, not servers
- 4 layers of Self Healing

# Any App Every Cloud One Platform

PCF 2.0 — for everything  
that matters

Pivotal





Pivotal  
**Container Service™**



**BEEES!!!**

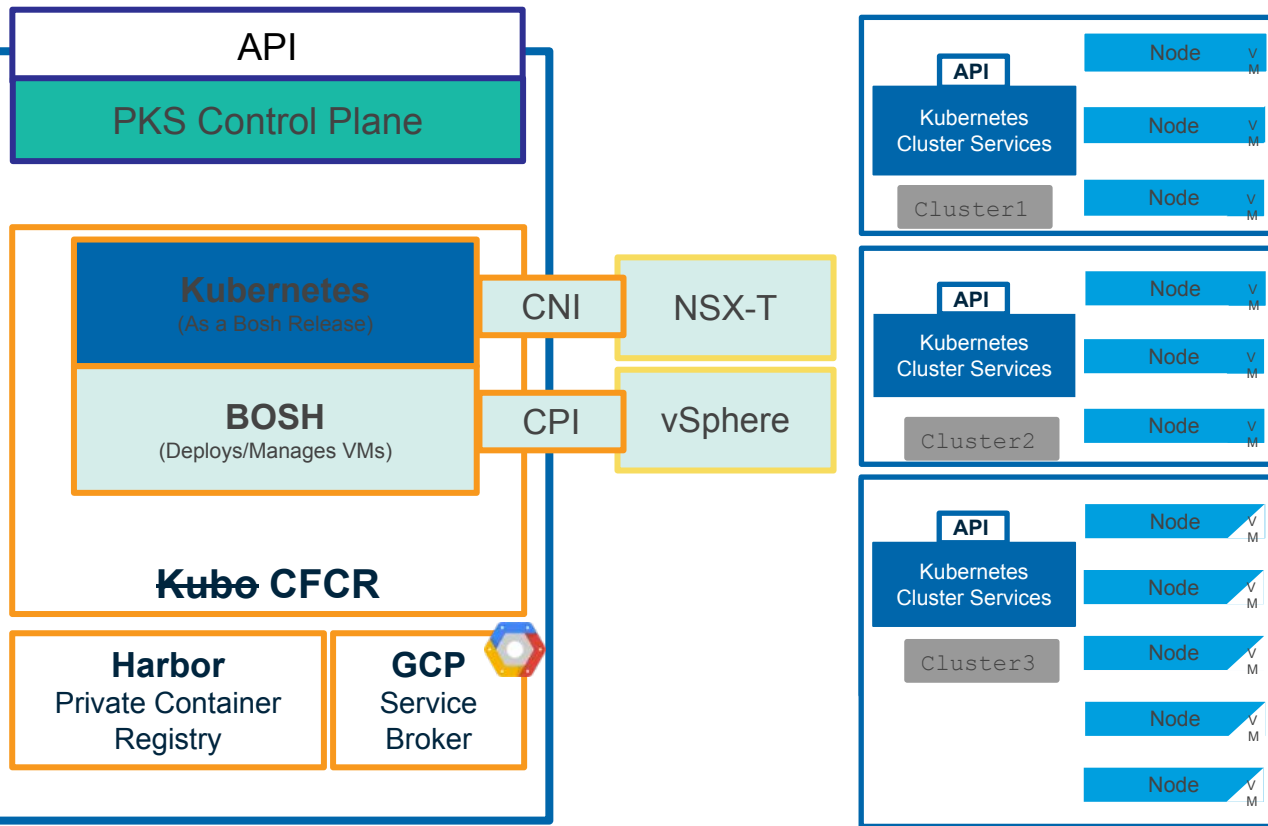
# PKS “How it Works”

## The value of BOSH

```
# pks create-cluster --plan small Cluster3
```

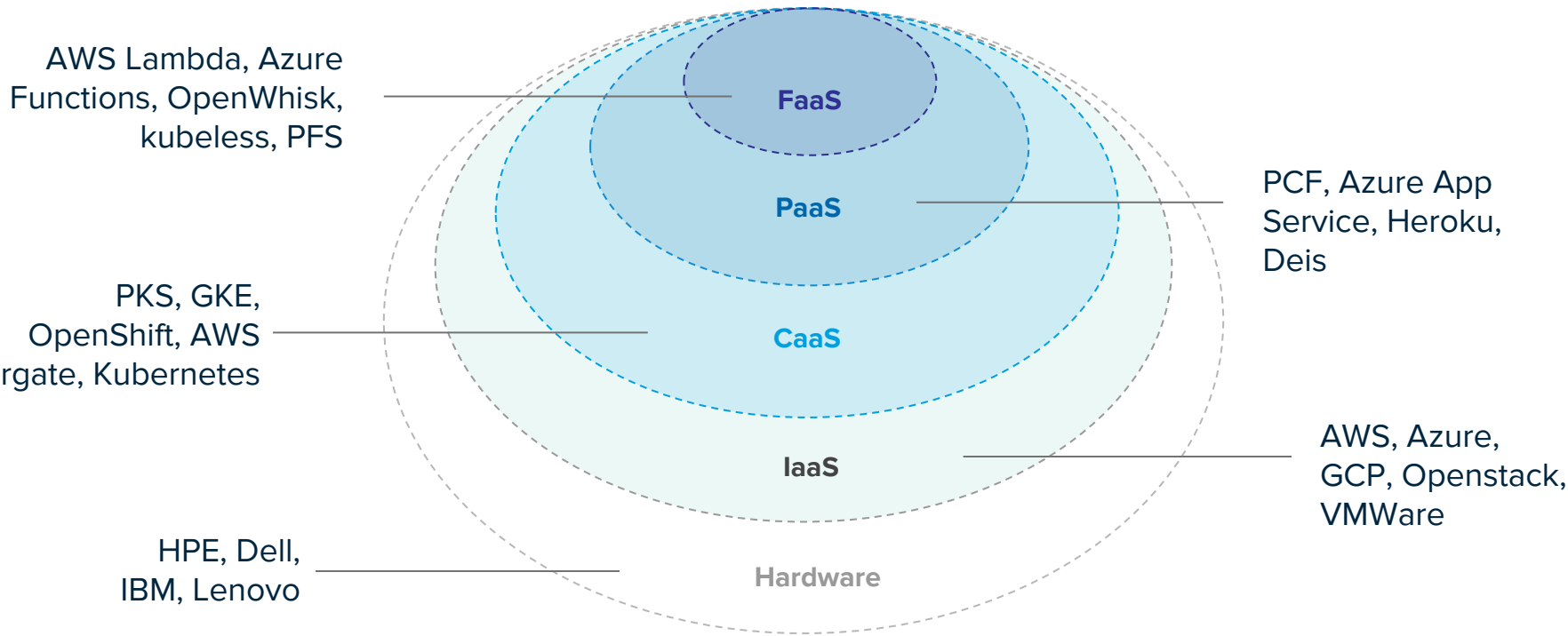
### PKS includes:

- PKS Control Plane, CFCR
- NSX-T, Harbor, GCP Broker
- BOSH Release for Kubernetes
- Configures **Day 1** of
  - CFCR
  - vSphere
  - NSX Integration
  - Harbor
- Manages **Day 2** of Kubernetes Clusters
  - Scaling
  - Patching
  - Upgrades
  - Failures



IT'S DANGEROUS TO GO  
ALONE! TAKE THIS.







Czarcloudski

@pczarkowski



Cant tell if london underground map or  
openstack architecture diagram.



7:43 AM - 19 Feb 2016

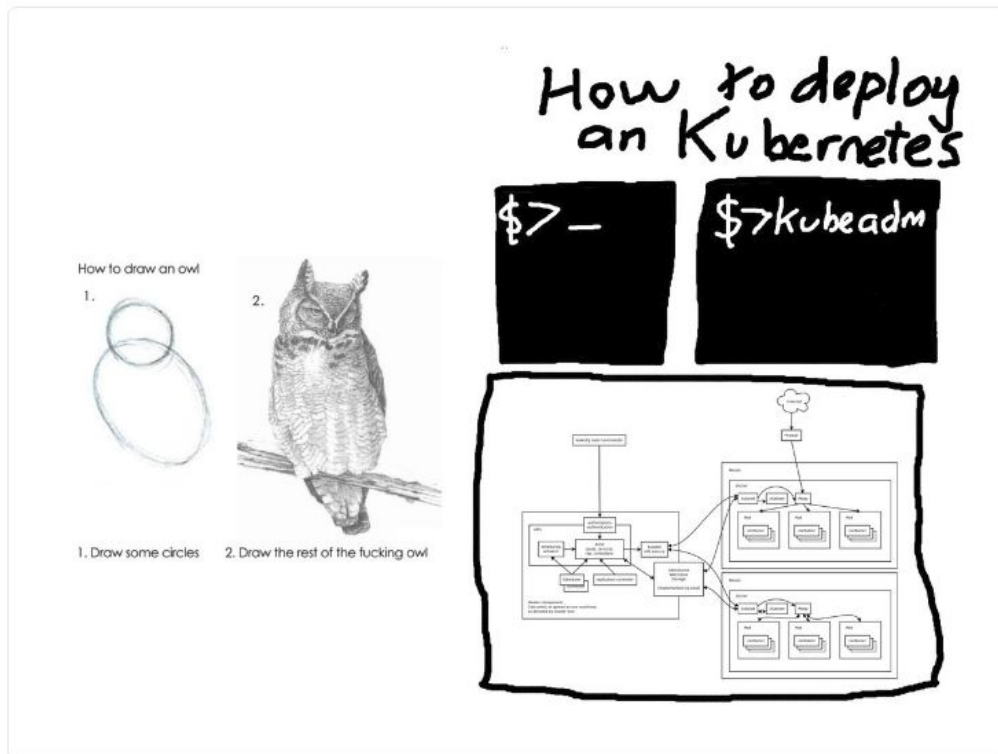


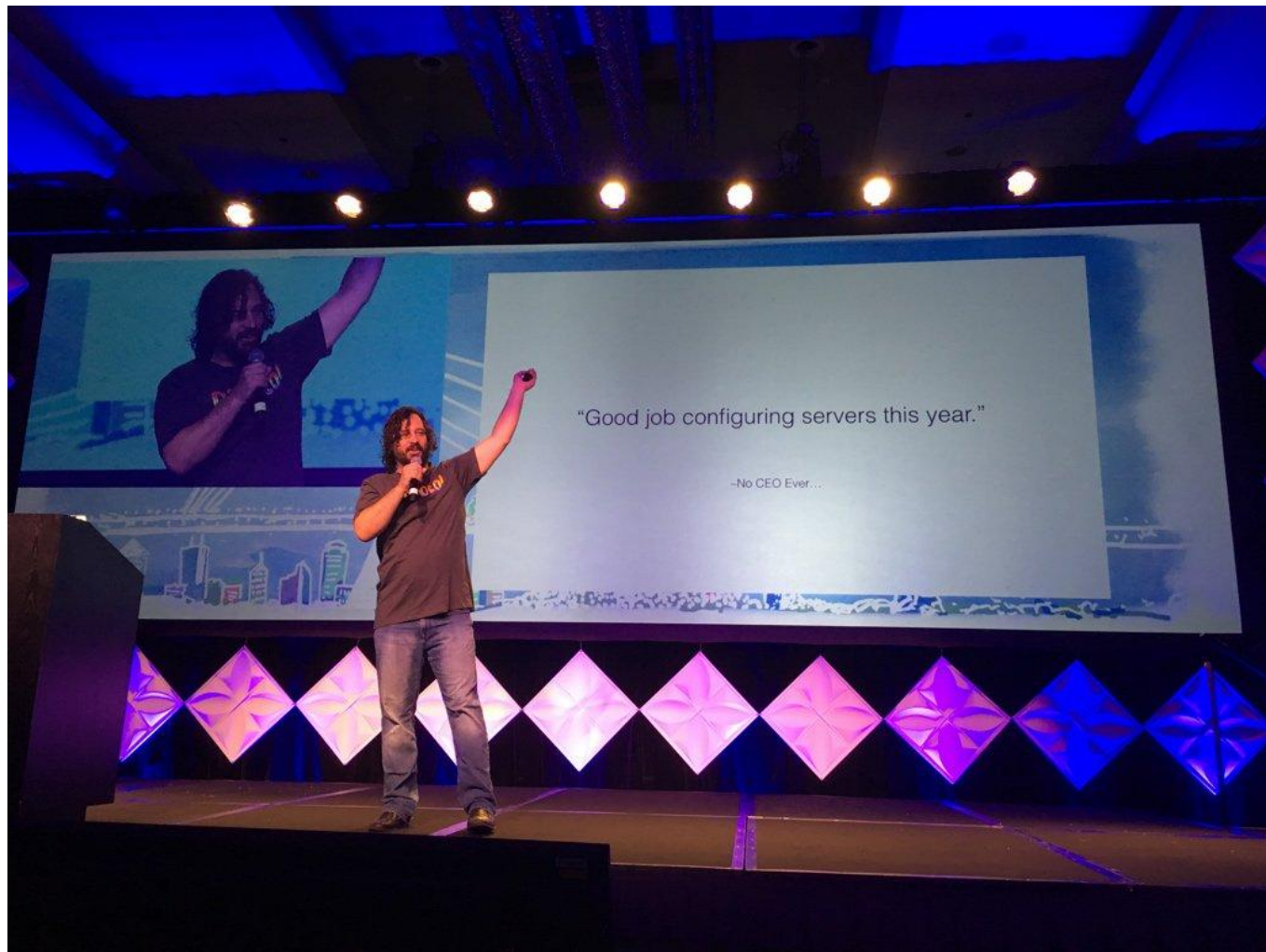
Czarcloudski

@pczarkowski



Did you know that Deploying an Kubernetes  
is as simple as drawing an Owl ?







<http://www.bsielearning.com.au/keep-simple-stupid/>

# Full Opensource DIY

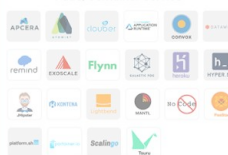
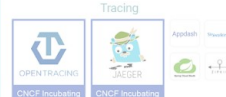
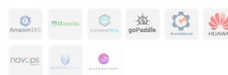
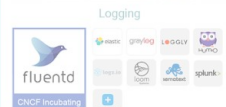
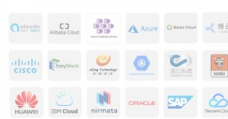
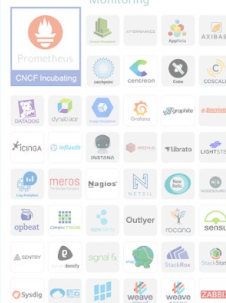
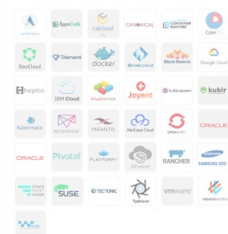
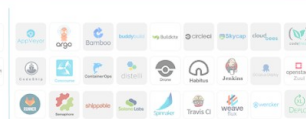
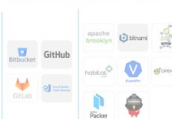
- ??? pxe
- Openstack Ansible
- Kubespray
- Ansible Hardening

<https://github.com/openstack/openstack-ansible>

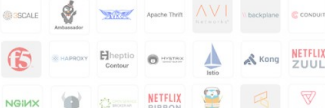
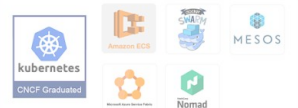
<https://github.com/openstack/ansible-hardening>

<https://github.com/kubernetes-incubator/kubespray>

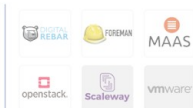
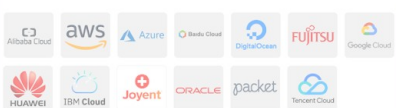
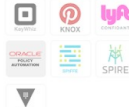
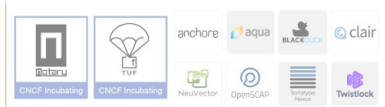
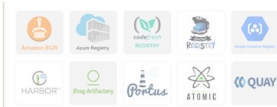
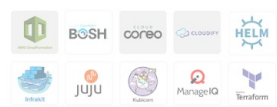
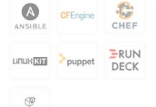
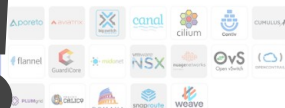
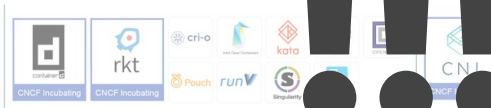
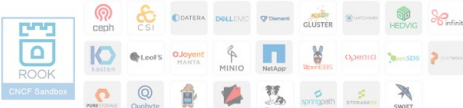




See the separate serverless landscape



## Cloud-Native Network

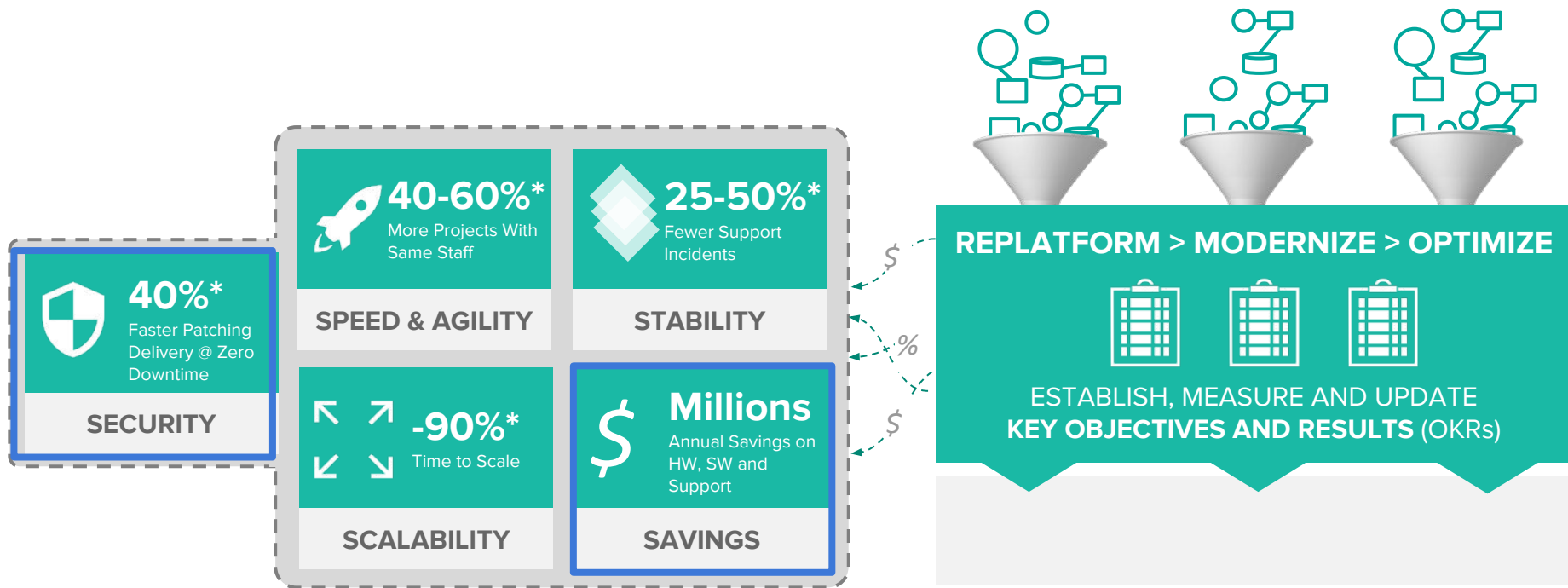


[github.com/cncf/landscape](https://github.com/cncf/landscape)

*This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.*



# How We Think about the Business Case



## PLATFORM VALUE STREAM AND METRICS

THE END

---

**Questions ?**