20 Minutes to Production

Zero Downtime

Paul Payne
@paulrpayne
http://payne.io

Presentation code can be found at:
github.com/payneio/qcon-gtin.git

Photos from http://unsplash.com.

IVY SOFTWORKS

NORDSTROM

GTIN (bar code) Product Lookup Service

18 million products.

We need this change right now!

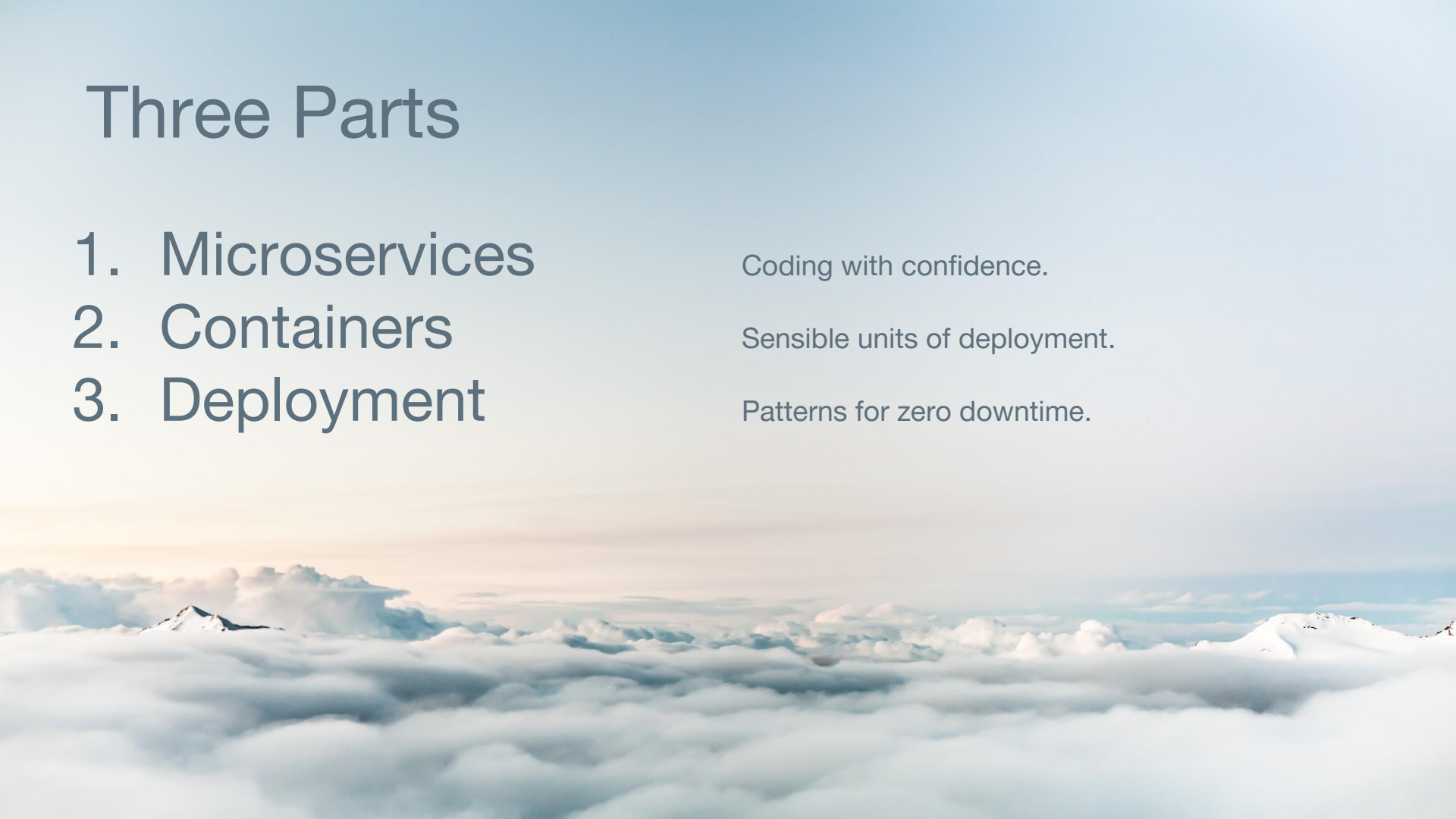
Three Parts

1. Microservices
2. Containers
3. Deployment

Coding with confidence.

Sensible units of deployment.

Patterns for zero downtime.



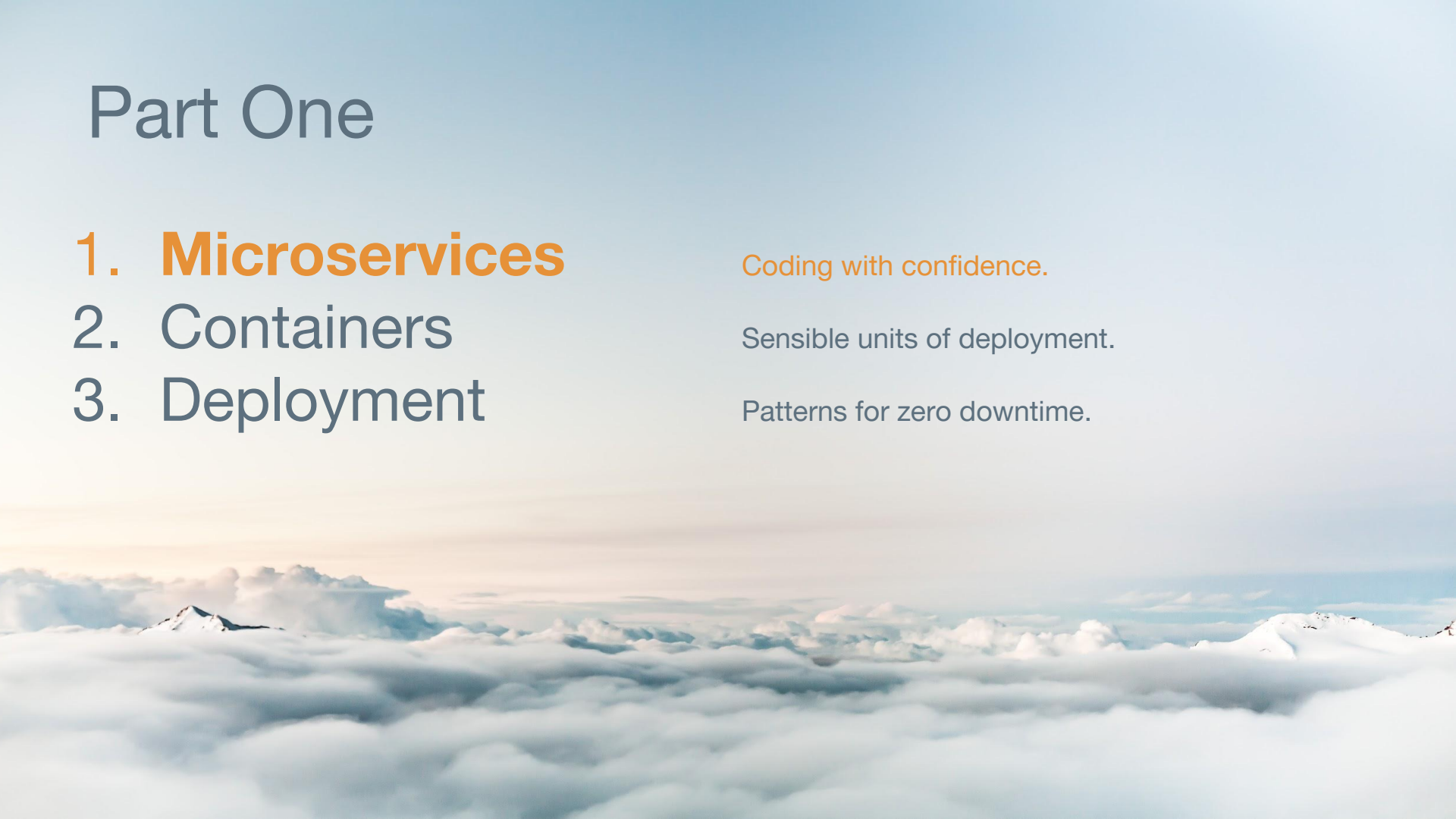
Part One

1. **Microservices**
2. Containers
3. Deployment

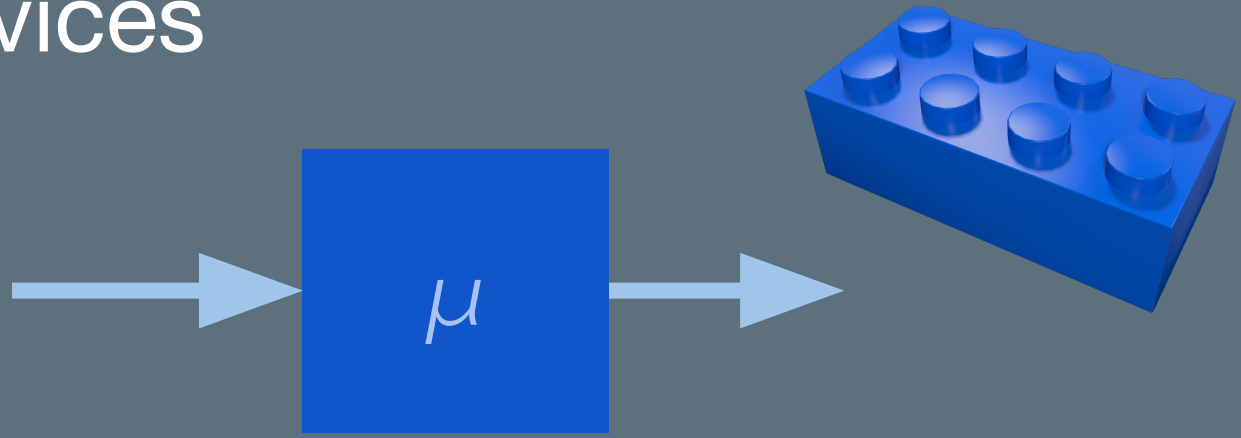
Coding with confidence.

Sensible units of deployment.

Patterns for zero downtime.



Microservices



Single responsibility.
Minimal, easily understood code.
Loosely coupled.
Modular. Composable.
Maintain their own state.

=

Independently deployable functionality

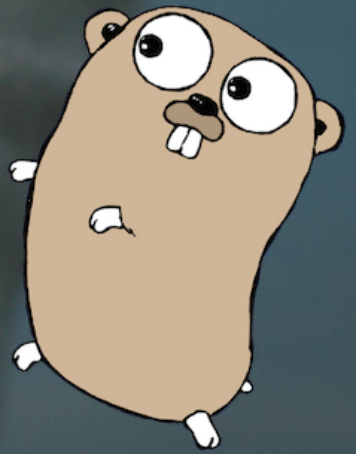
Microservice Quality Assurance

- Liberal in what you accept. Conservative in what you send (Robustness Principle).
- Fast errors (compile time vs. run time).
- Fast compile, build, test iterations.
- TDD.

Confidence.

Demo

In which the presenter demonstrates an update to the microservice with defensive coding practices and fast build and test steps with the aid of Go.



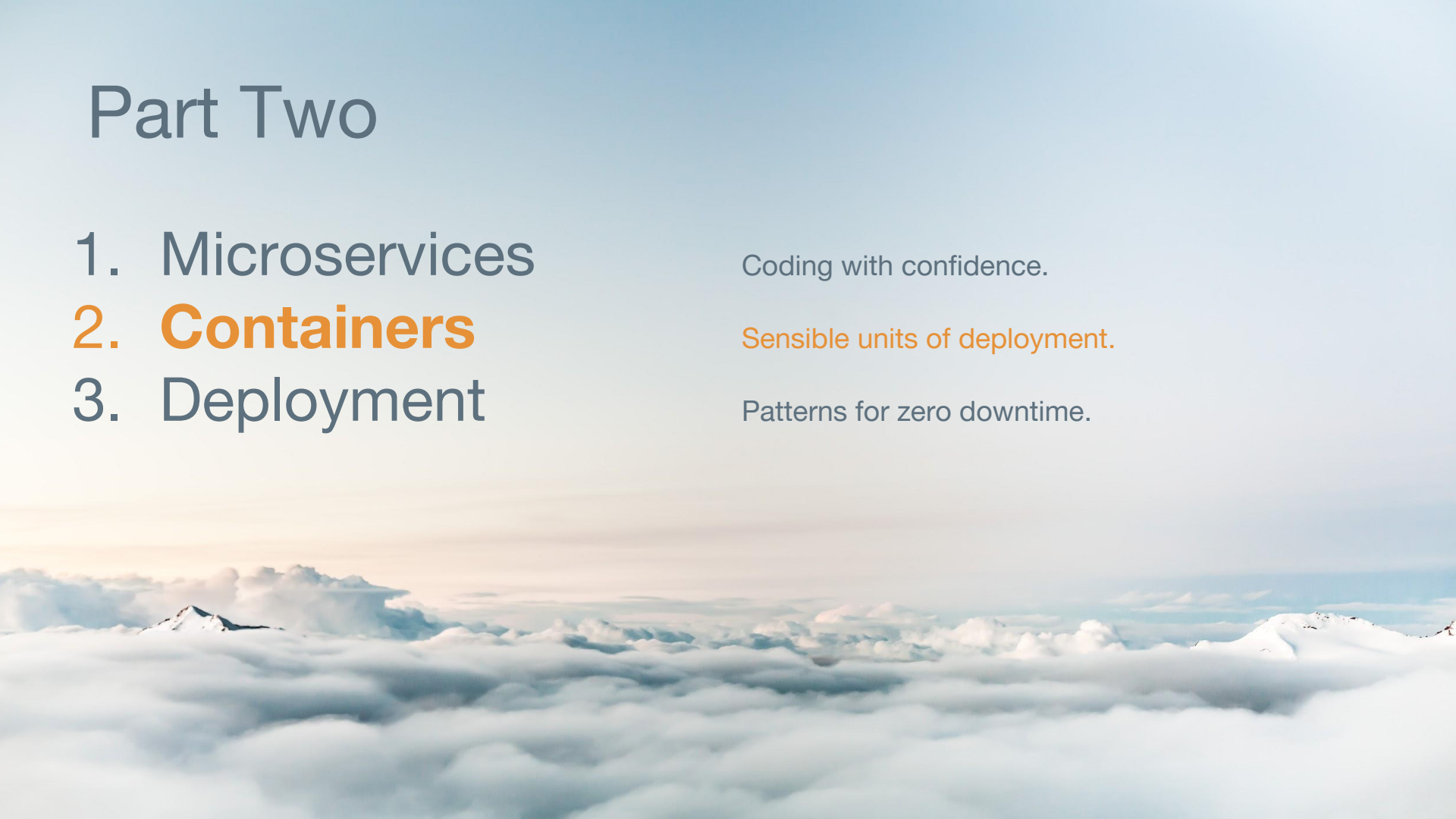
Part Two

1. Microservices
2. **Containers**
3. Deployment

Coding with confidence.

Sensible units of deployment.

Patterns for zero downtime.



Application Containers

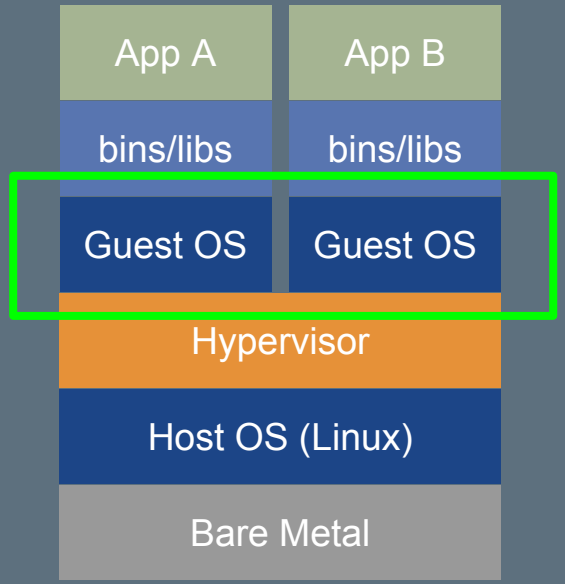
“An application container is a way of packaging and executing processes on a computer system that isolates the application from the underlying host operating system” —https://github.com/appc/spec, 2015.

Various projects...
chroot (1979)
jail
Linux-VServer
OpenVZ
...

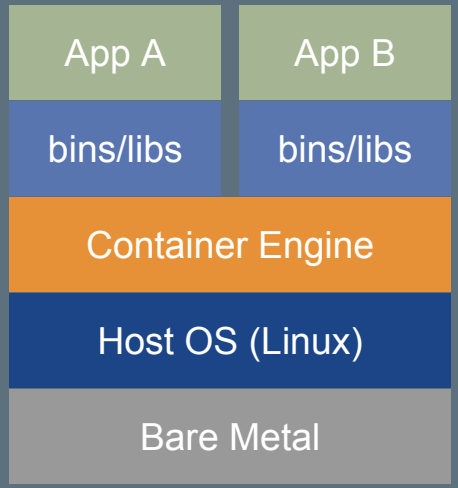
brought into the kernel...
namespaces
cgroups
SELinux
AppArmor
btrs/aufs/device mapper/etc
...

and packaged up.
systemd-nspawn
LXC
lmctfy
libvirt-lxc
Docker / libcontainer
rkt / appc
...

Virtualization Overhead

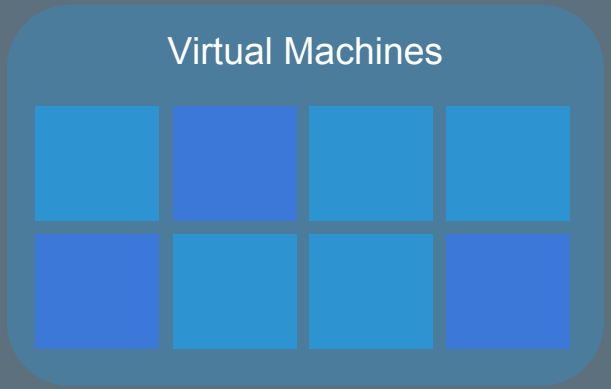


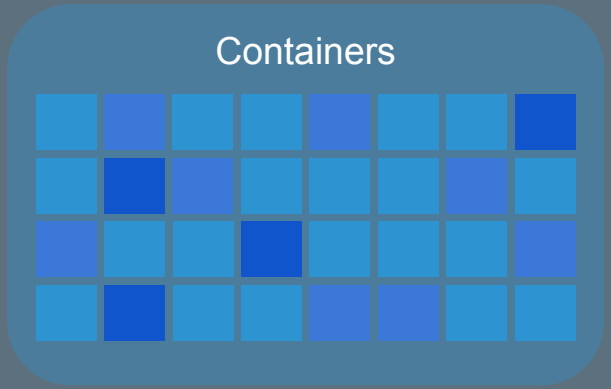
VM



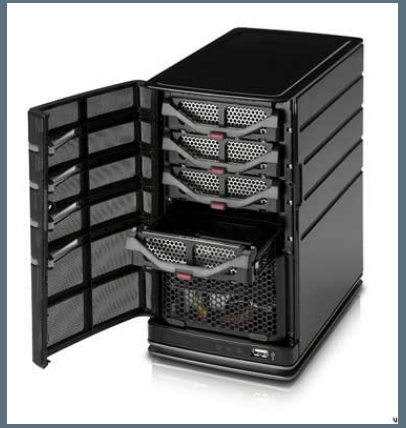
Container

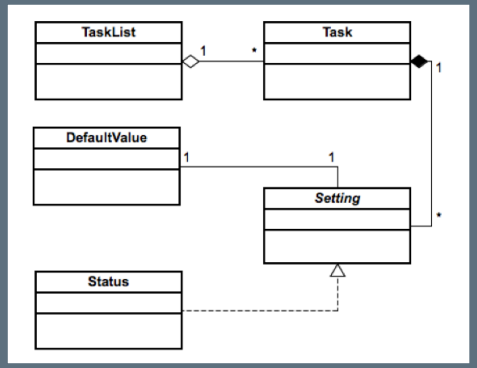
Better Hardware Utilization





The Right Abstraction for Software





Immutable ~~Infrastructure~~ Microservices

Analogy from
Bill Baker, Microsoft

Clean Org Interface (Reverse Conway's Law)

The DevOPS continuum

The mythical DevOPS unicorn

VMs

Installing and configuring VMs.
Configuring app environments.
Creating, building and installing apps.
Making VM snapshots.
Installing OS updates.

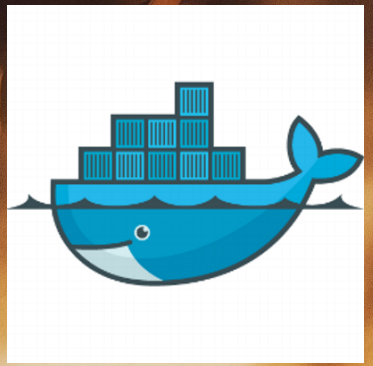
Installing and configuring hardware.
Distributing VMs.
Monitoring and fixing hardware.
Working **with** devs on bad VMs.

Containers

Configuring app environments.
Creating, building and installing apps.

Installing and configuring hardware.
Distributing containers.
Monitoring and fixing hardware.
Rejecting bad containers.

Demo



```
$ make
```

In which the presenter demonstrates the simple containerization of the microservice using Docker and make.

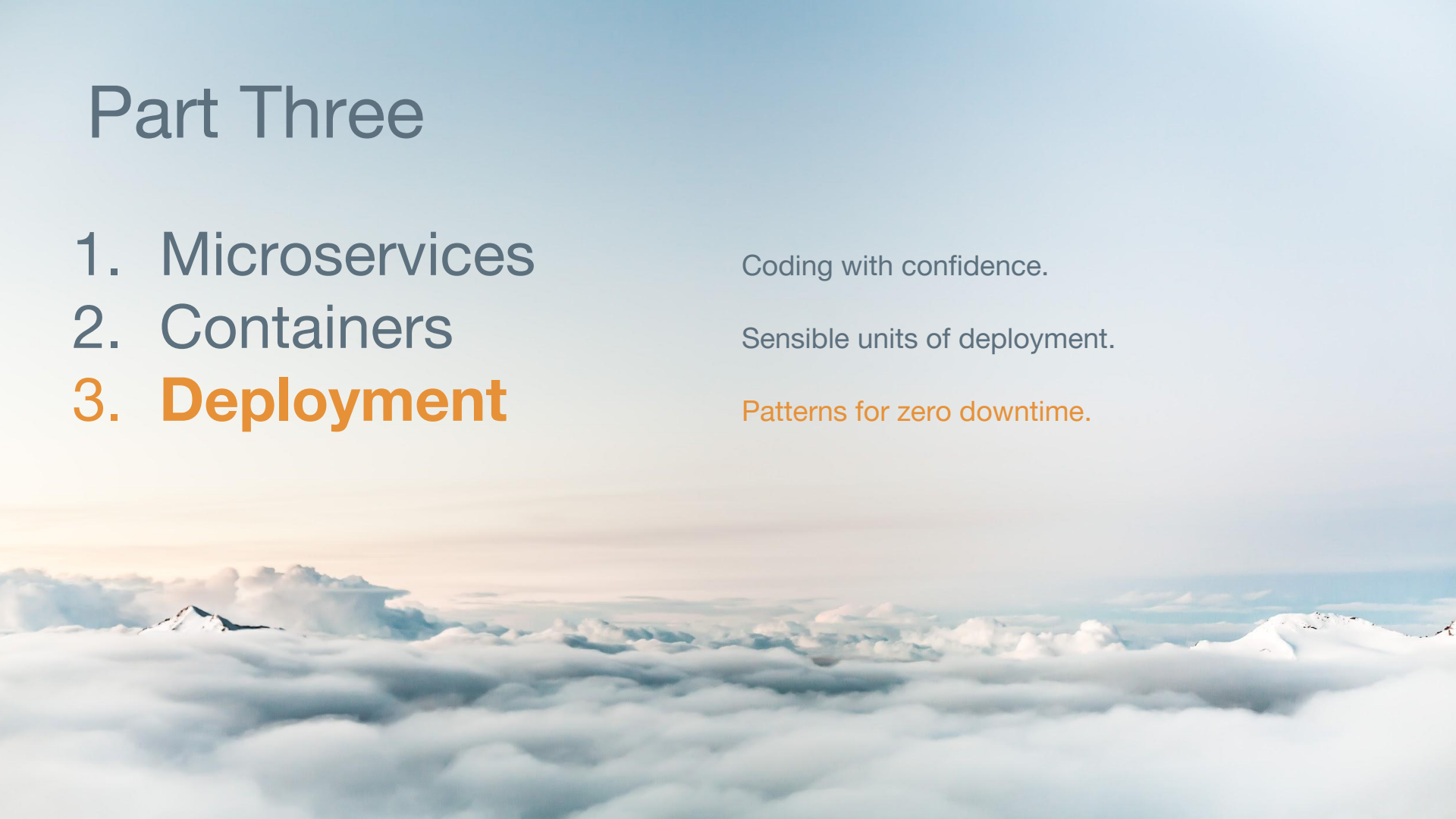
Part Three

1. Microservices
2. Containers
3. **Deployment**

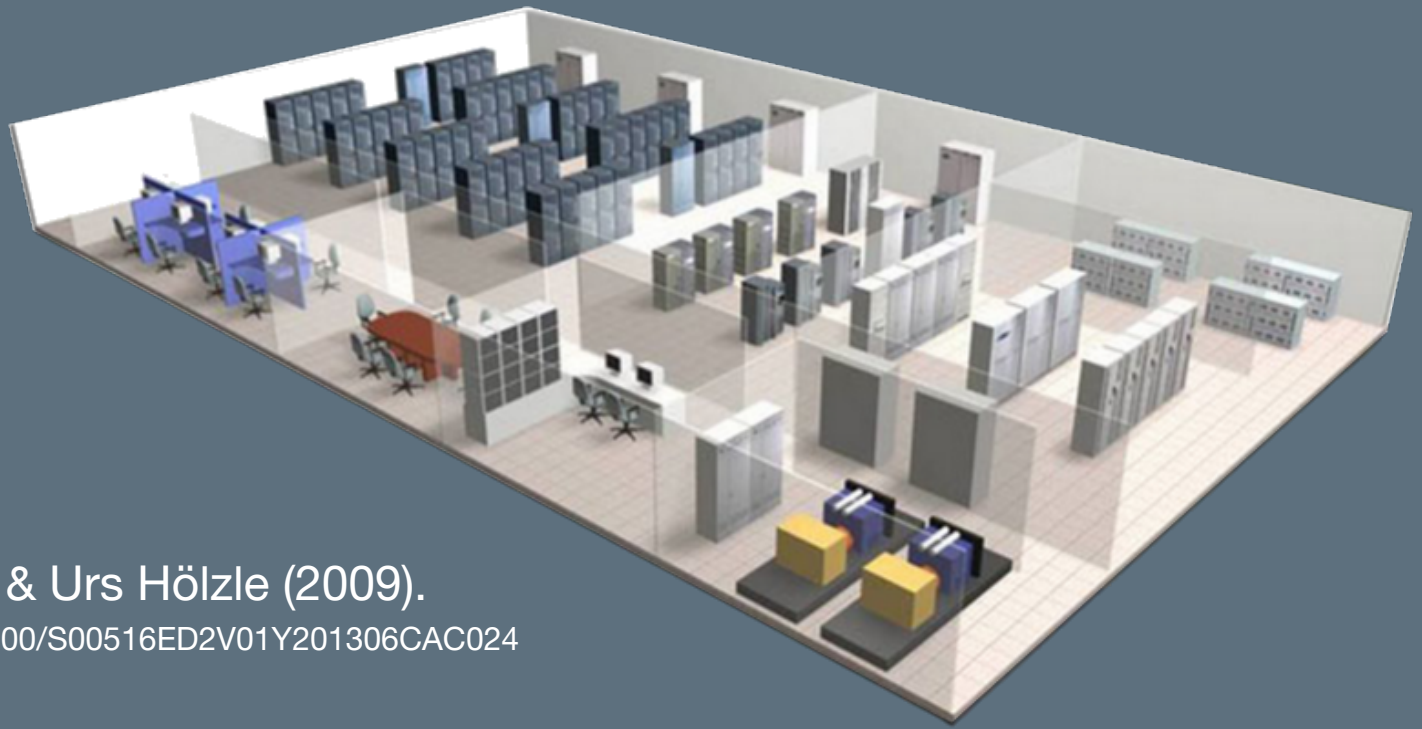
Coding with confidence.

Sensible units of deployment.

Patterns for zero downtime.



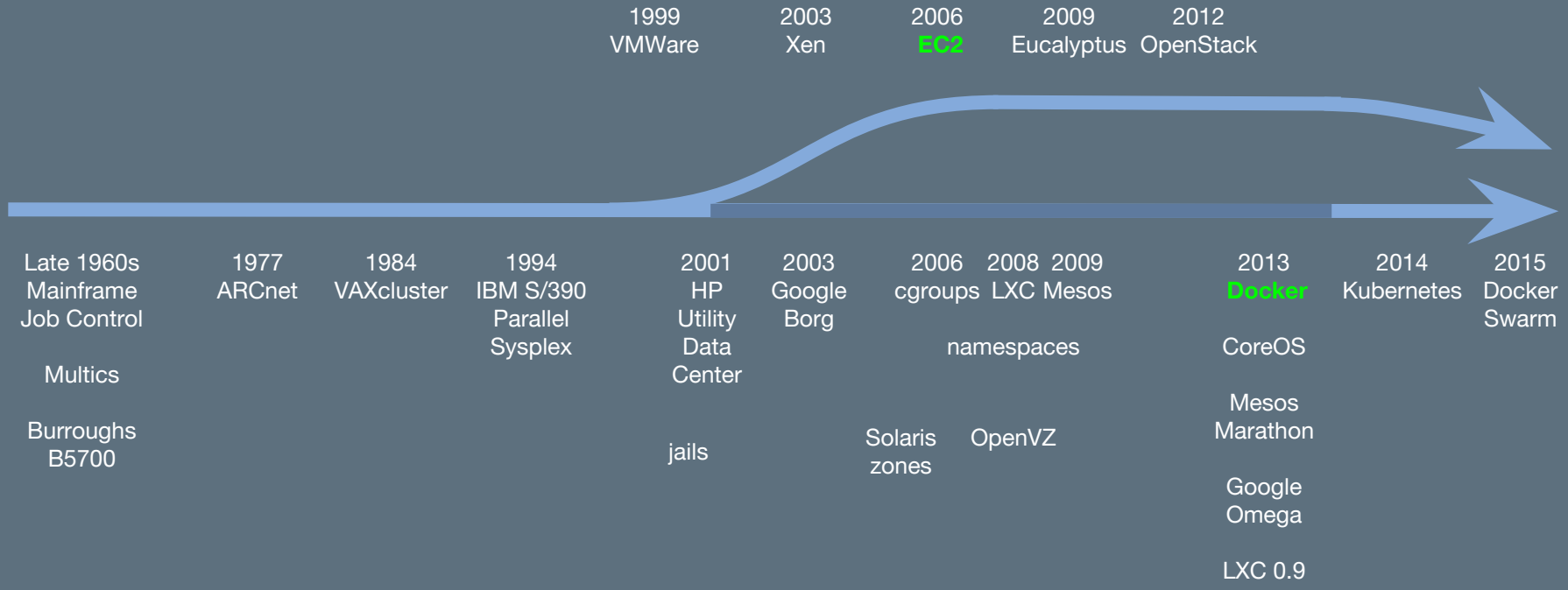
The Datacenter as a Computer



Luiz Barroso & Urs Hölzle (2009).

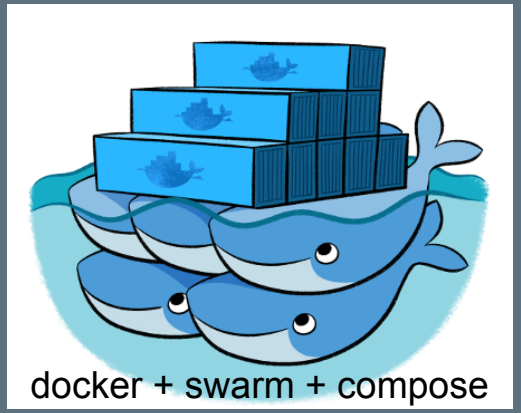
http://doi.org/10.2200/S00516ED2V01Y201306CAC024

A Virtual Machine Detour on the Cluster Scheduling Timeline

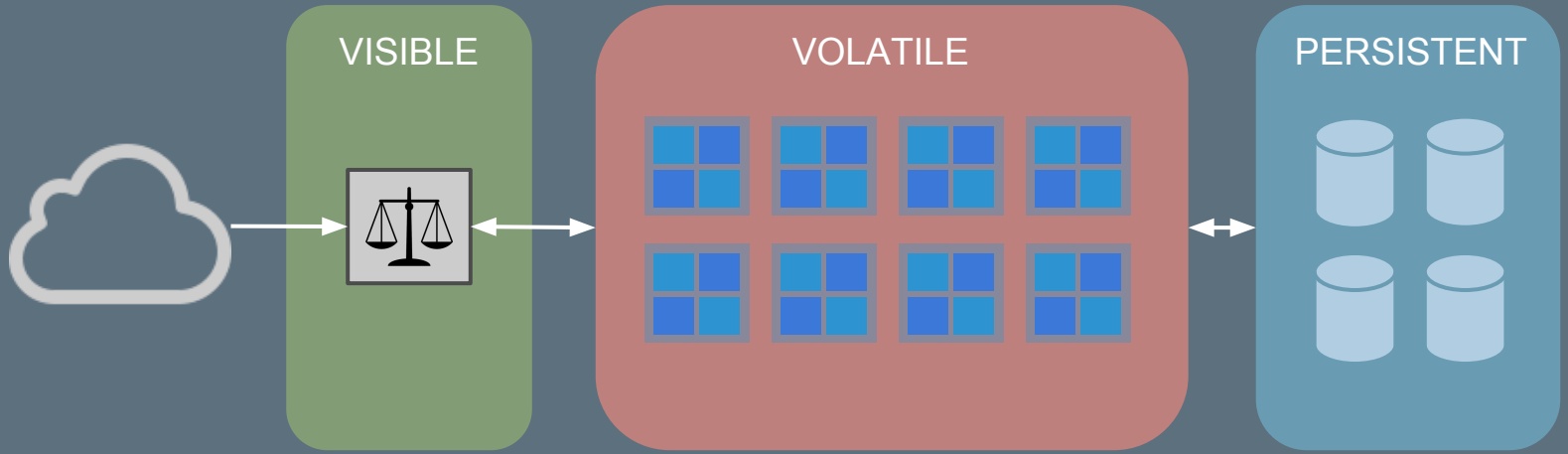


Cluster Schedulers, aka Microservice Platforms

DCOS



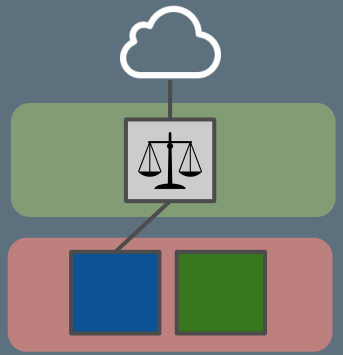
A Microservice Platform Pattern



Visible/Volatile/Persistent — Peter Gillardmoss, 2012
Phoenix servers — Martin Fowler, 2012

Live Deployment

Blue-Green Deploy

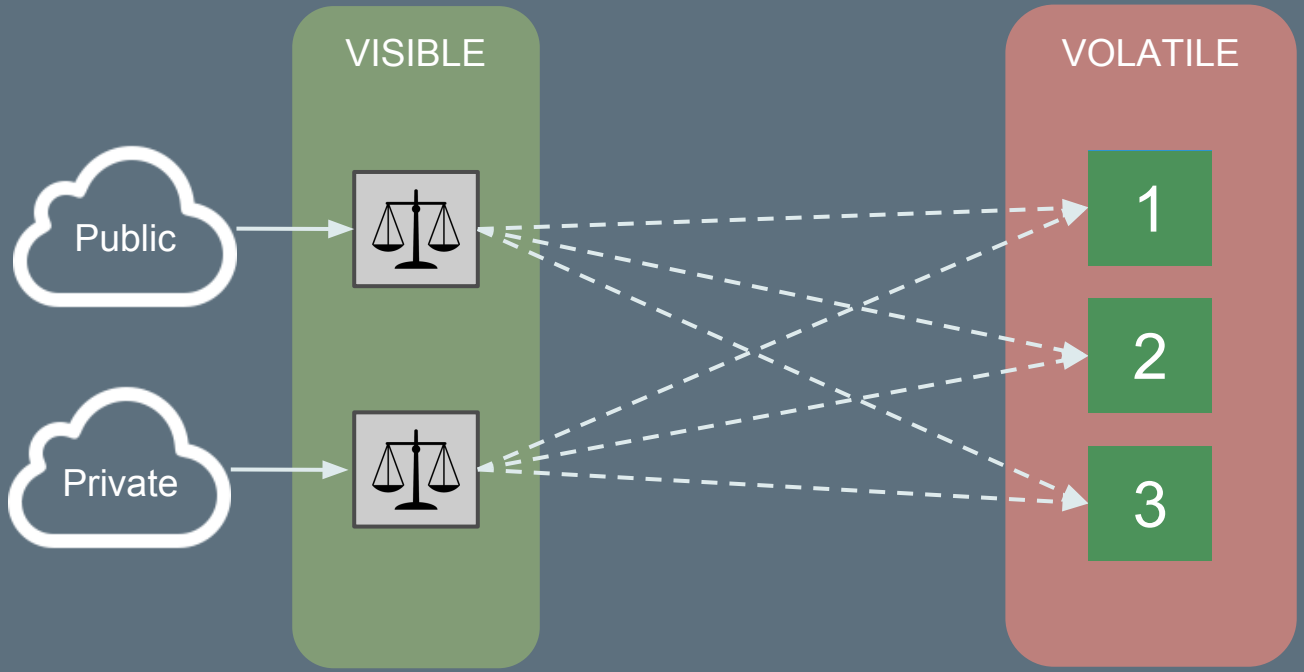


Canary Deploy

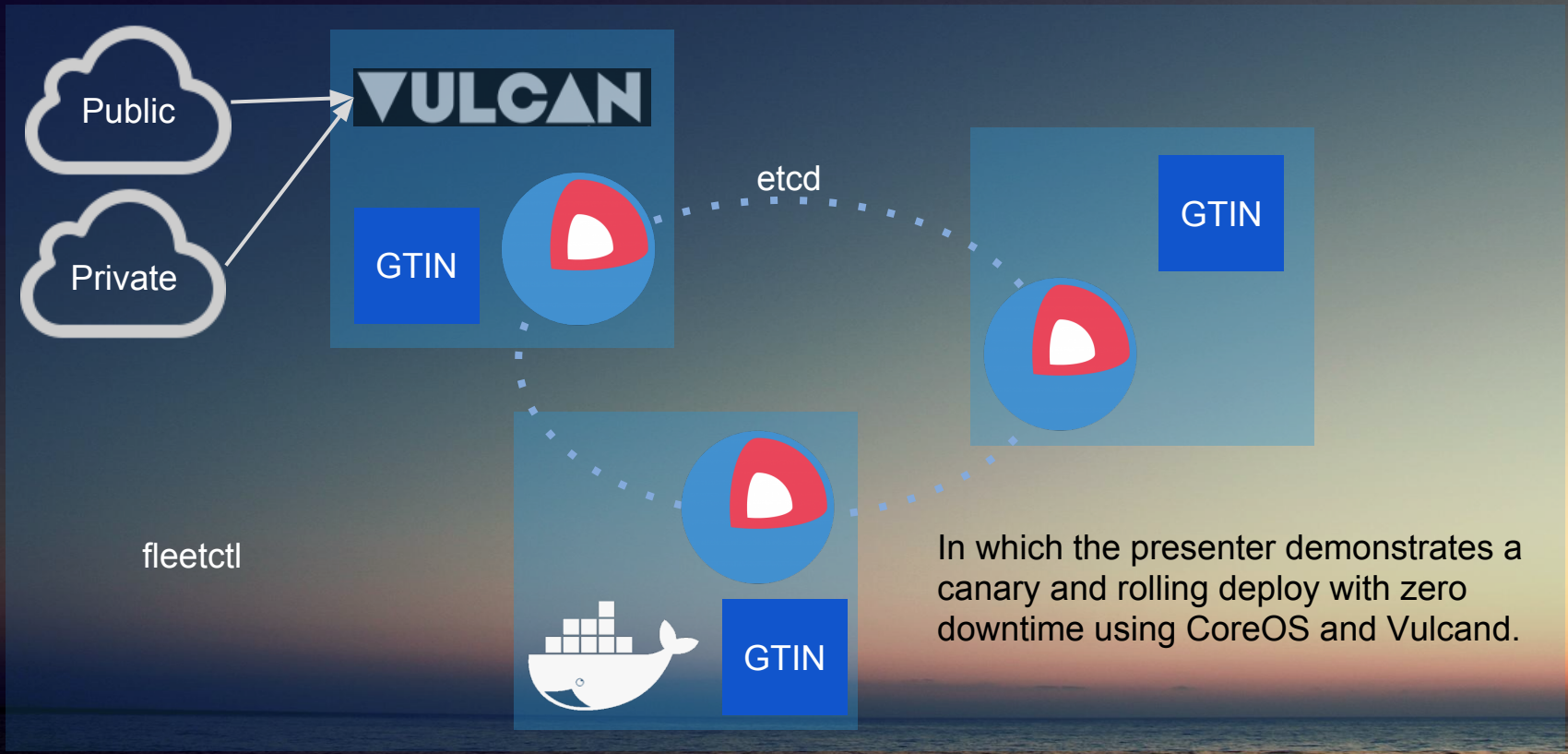
Rolling Deploy

Jez Humble and David Farley (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley.

The General Deployment Story



Demo



In which the presenter demonstrates a canary and rolling deploy with zero downtime using CoreOS and Vulcand.


12
1
2
3
4
TIMEX
QUARTZ

The Result

Other Reactions

“Wait! You’re not deploying to production NOW are you??!”

“I think we found a bug in the system.”

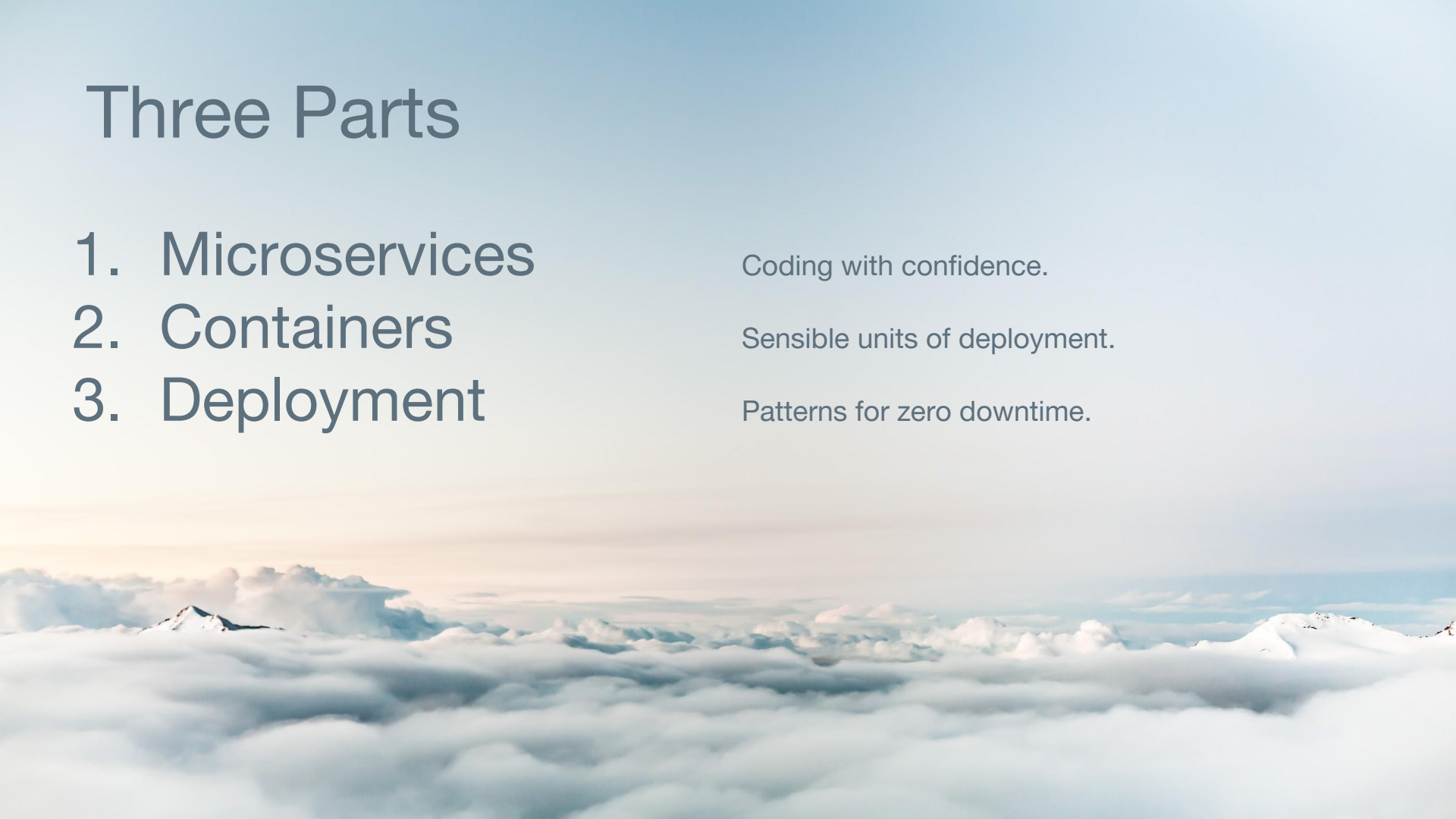
Three Parts

1. Microservices
2. Containers
3. Deployment

Coding with confidence.

Sensible units of deployment.

Patterns for zero downtime.




Getting There
Part Four
(surprise!)

Infrastructure as Code

Quick reproducible infrastructure.

Entire Blue/Green stacks.

Cattle, not pets!

Google Cloud Platform

Deployment Manager

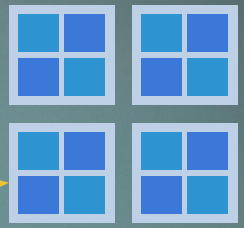
Amazon Web Services

CloudFormation

Cross-platform

Terraform

Anti-Corruption Layers



Each service manages its own data; ie. maintains its own table.

Integrate through interfaces, not data stores!

Our canonical GTIN data is kept in a commercial enterprise product.

We copy the data we need for our service to Dynamo, syncing daily.

Application Strangler

Wrap a proxy layer around the entire enterprise application.

Route requests to your microservices as you bring them online.

Someday, none of your legacy app will remain.

http://martinfowler.com/bliki/StranglerApplication.html.
Image by http://journals.worldnomads.com/beckandphil.

Build to Learn

20 Minutes to Production

Zero Downtime

Paul Payne
@paulrpayne
http://payne.io

Presentation code can be found at:
github.com/payneio/qcon-gtin.git

Photos from http://unsplash.com.