Kubernetes Runtime Security

Google Cloud

Kubernetes Runtime Security

Google Cloud

About me

Jen Tong
Security Advocate
Google Cloud Platform

@MimmingCodes
mimming.com

Google Cloud

How many of you

- use Kubernetes in production?
- use containers?
- are security engineers?
- gotten a shell on a system?
- have ever discovered a long ago compromised system?

Agenda

Security overview

Containers & Kubernetes

Impact on security

Demo of a sad day

Fix low hanging fruit

Discuss higher up fruit

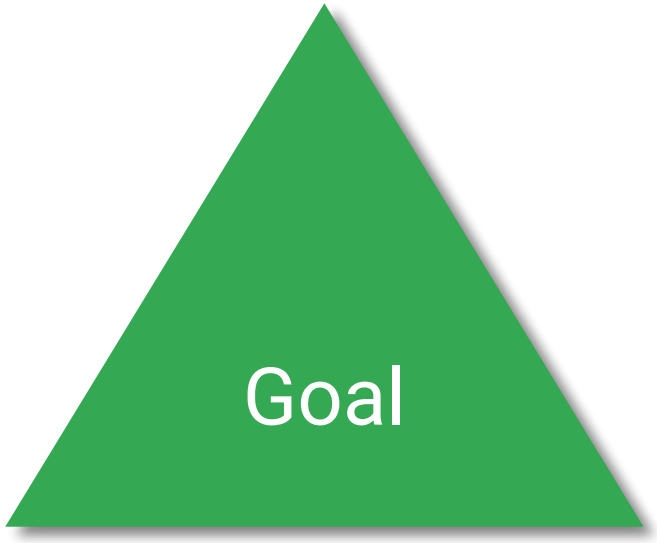
Security overview

offense vs defense

01

Google Cloud

Offensive Security

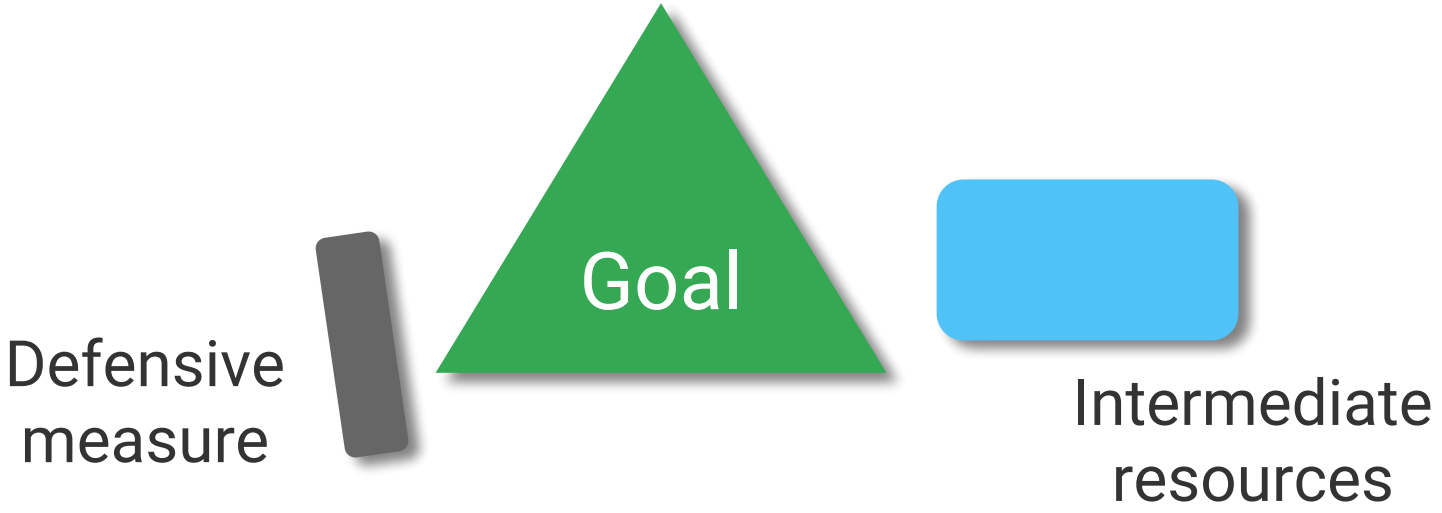


Offensive Security

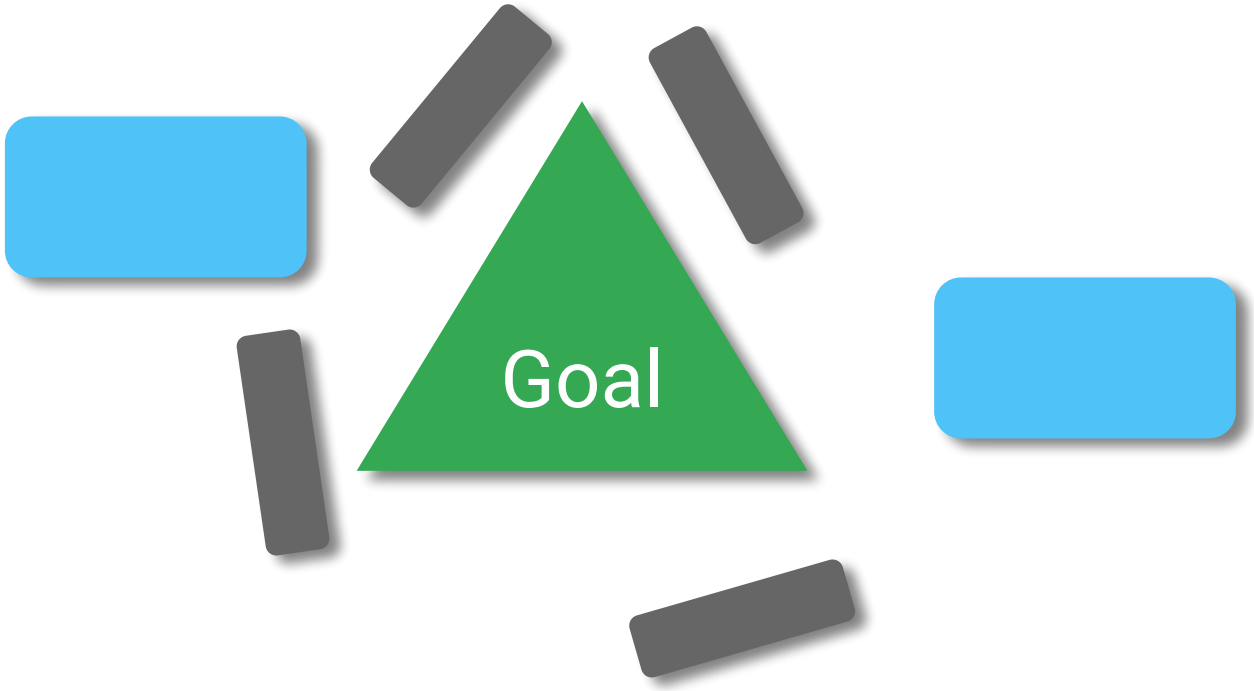
Offensive Security

Offensive Security

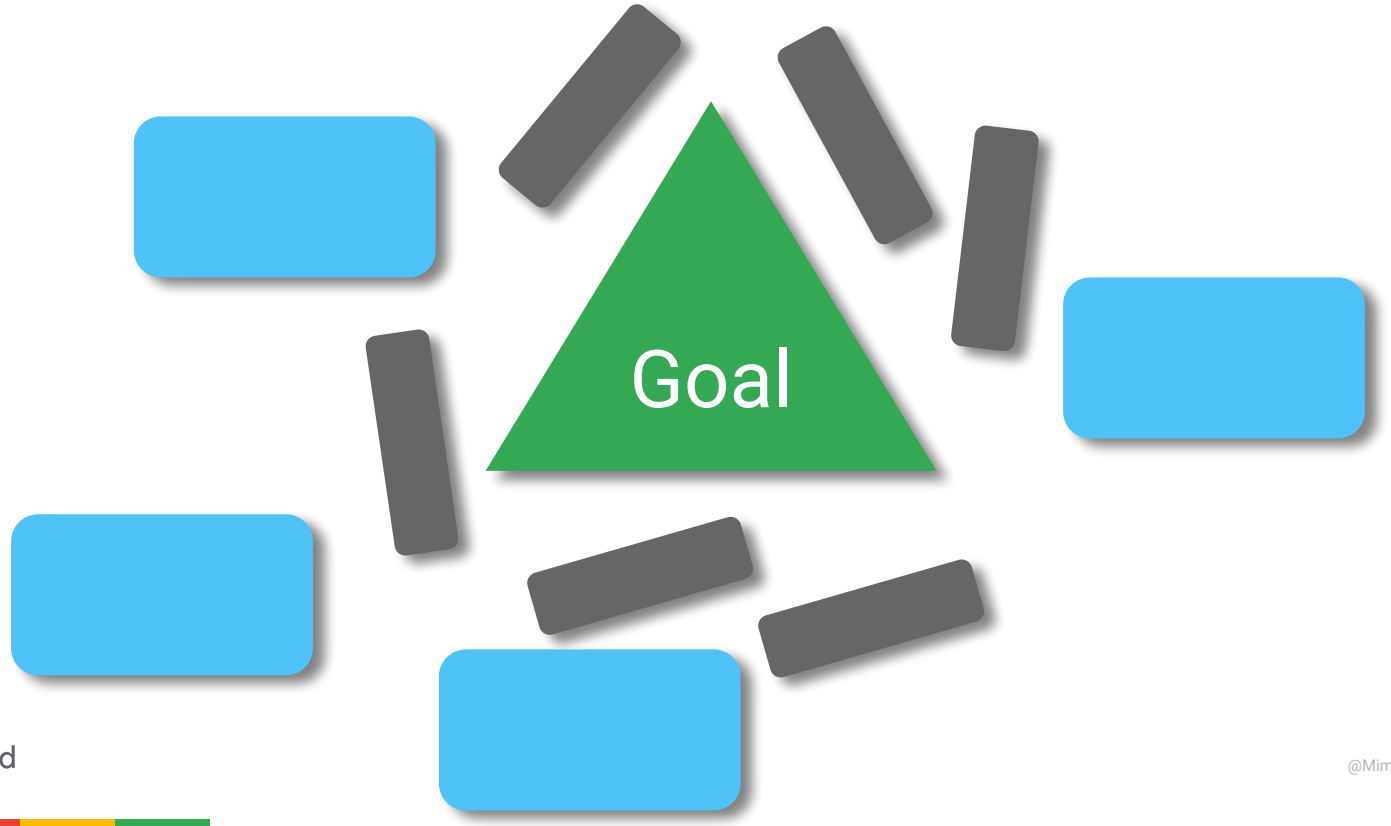
Offensive Security



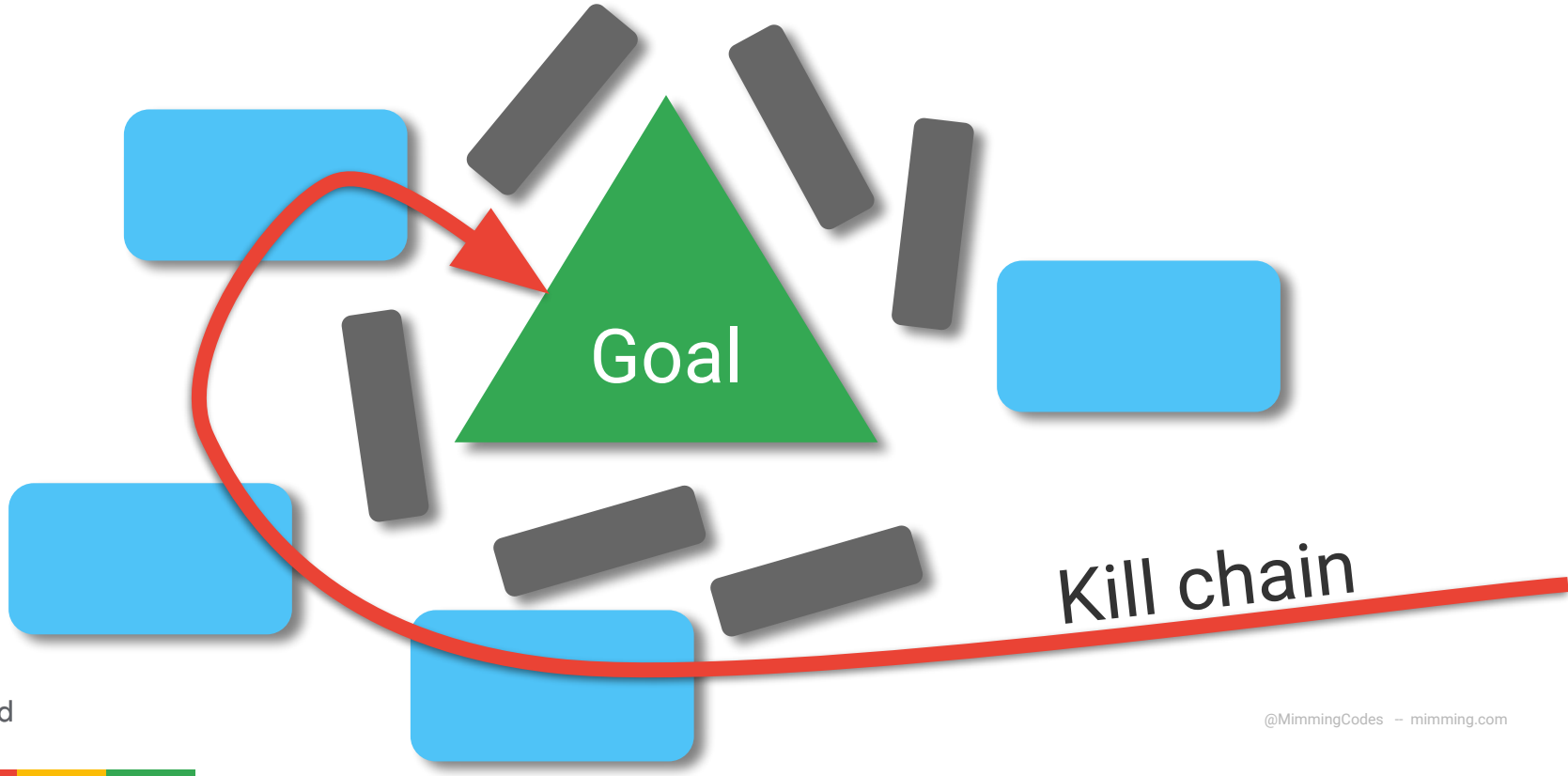
Offensive Security



Offensive Security

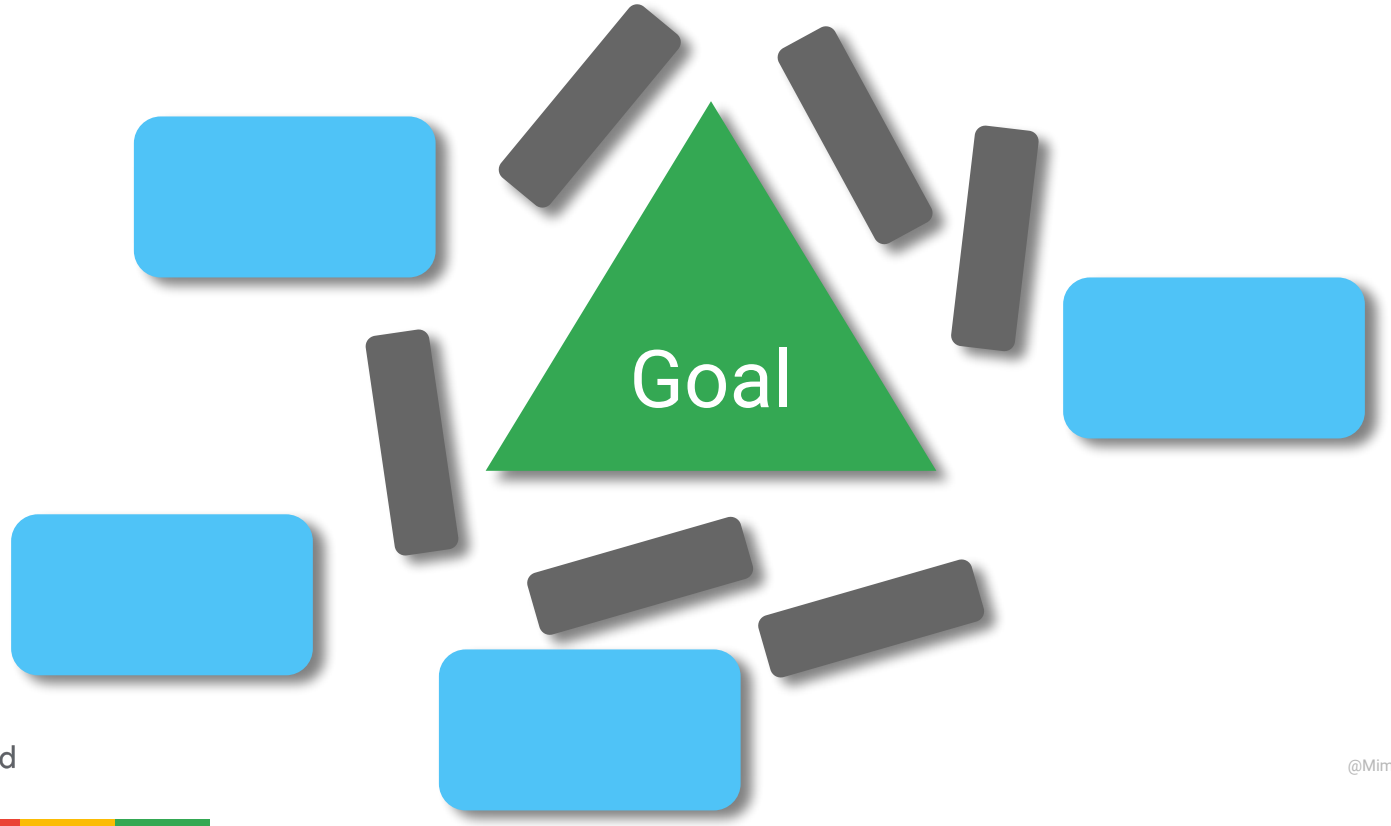


Offensive Security

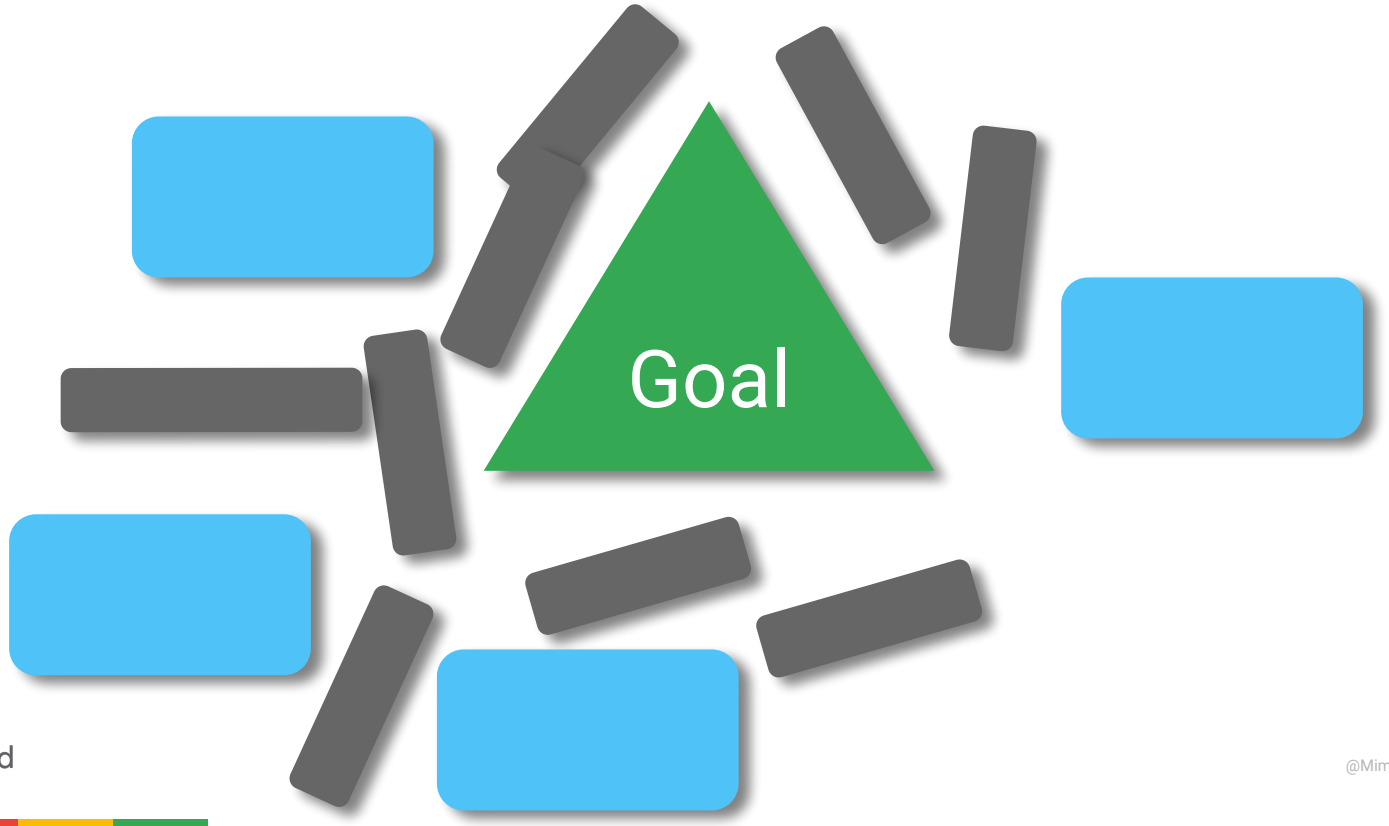


It feels like development on a terrible API

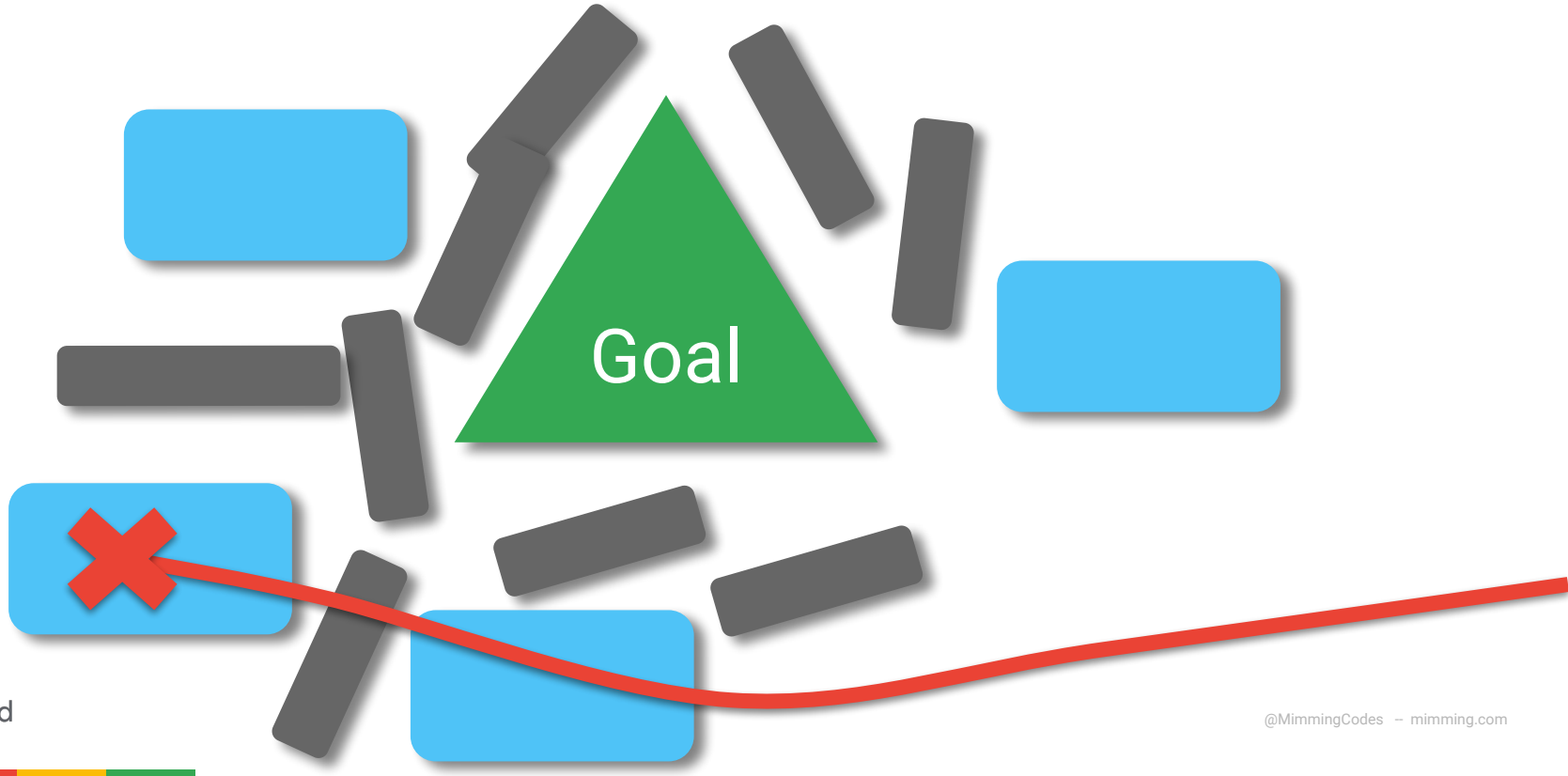
Defensive security



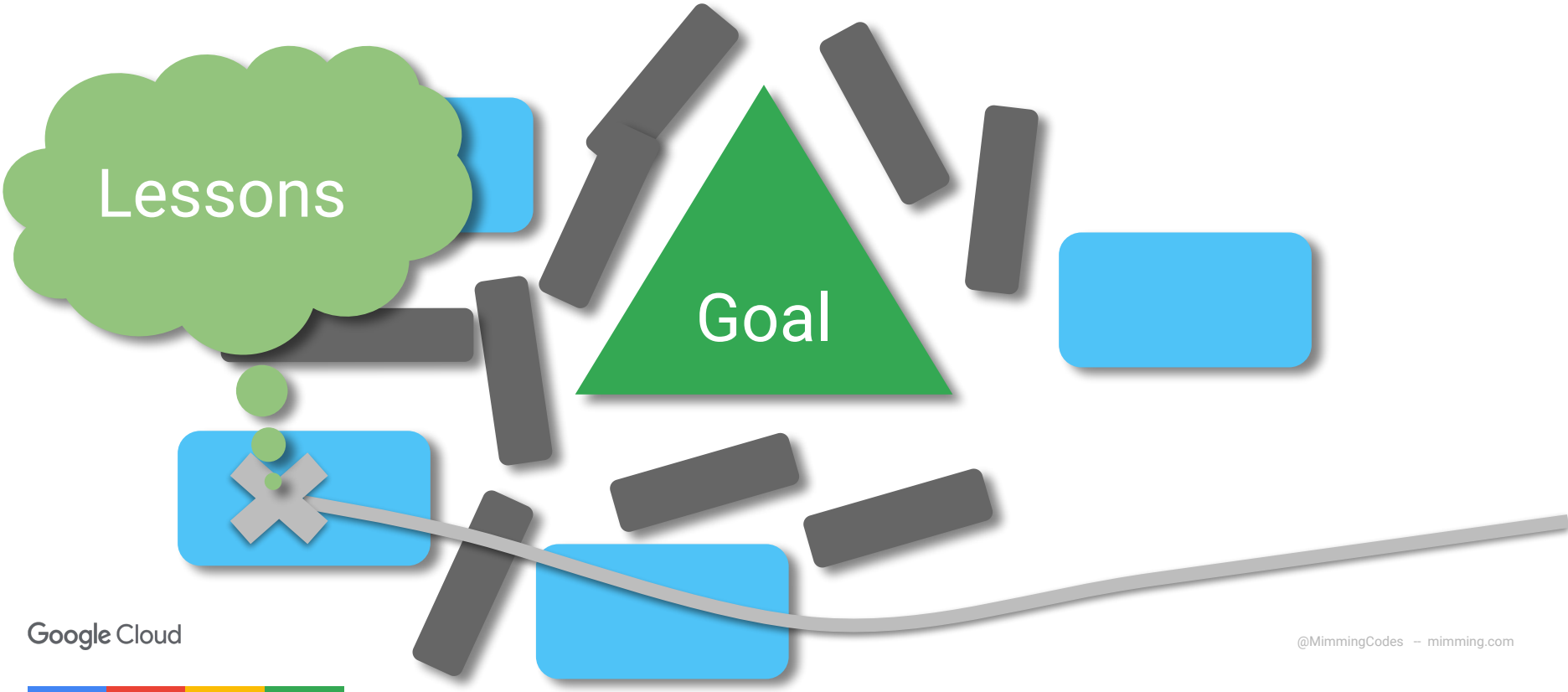
Defensive security



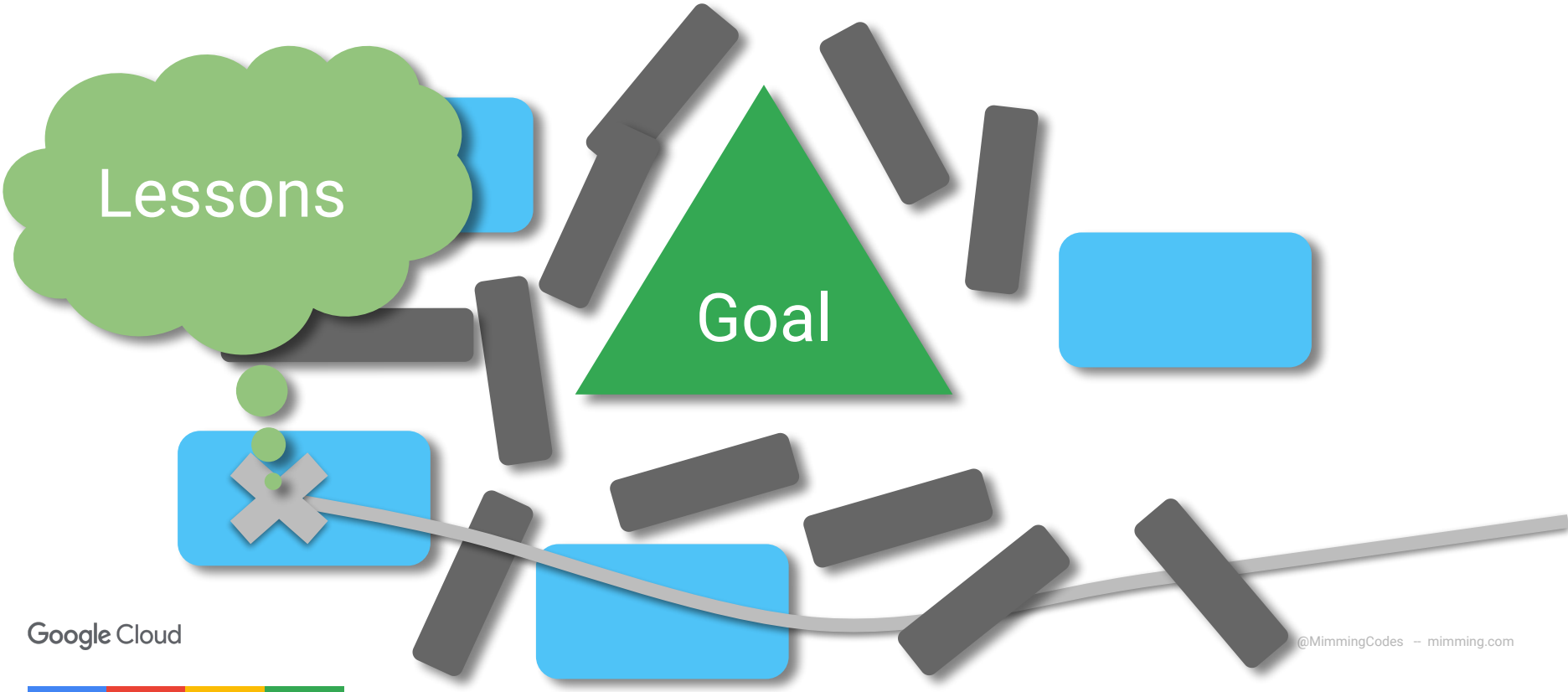
Defensive security



Defensive security



Defensive security



Containers & Kubernetes

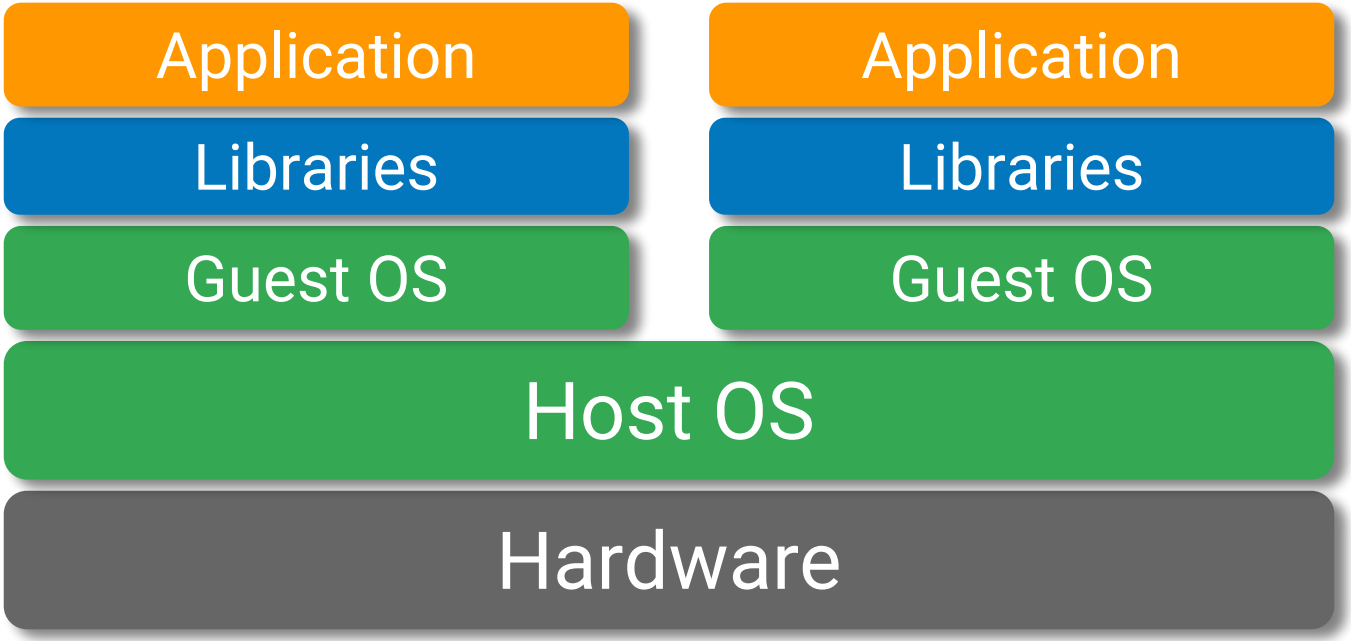
… or as much as I can cover in 5 min

02

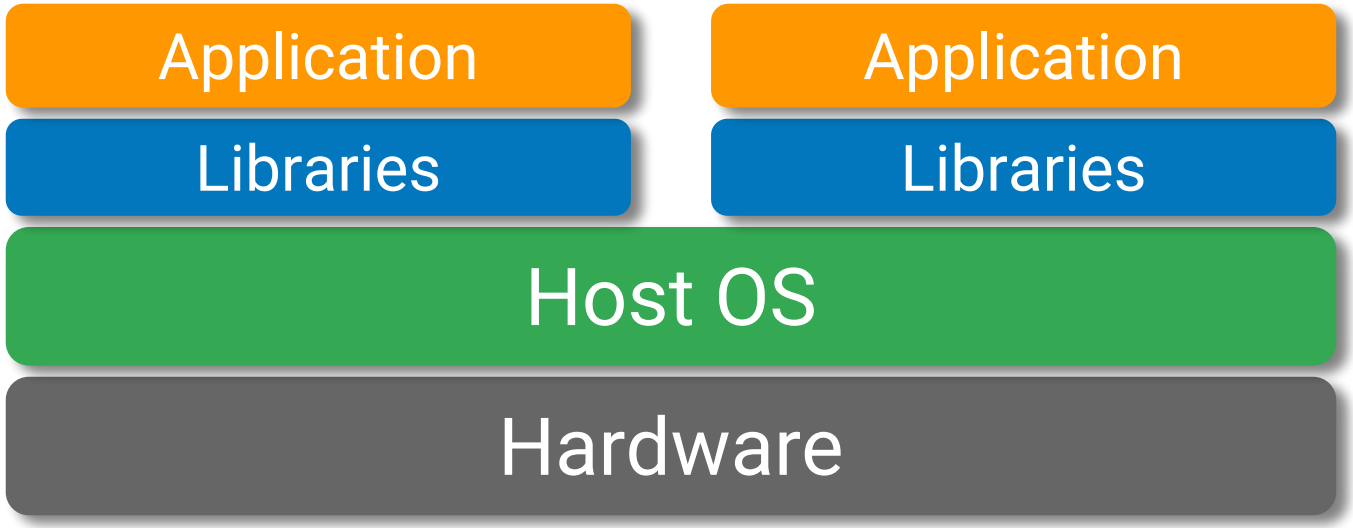
Google Cloud

The promises of virtualization, but it actually works

Virtualization



Containers


Application
Application
Libraries
Libraries
Host OS
Hardware

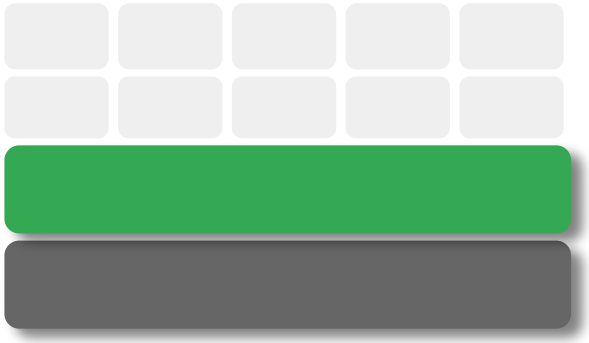
Lots of containers

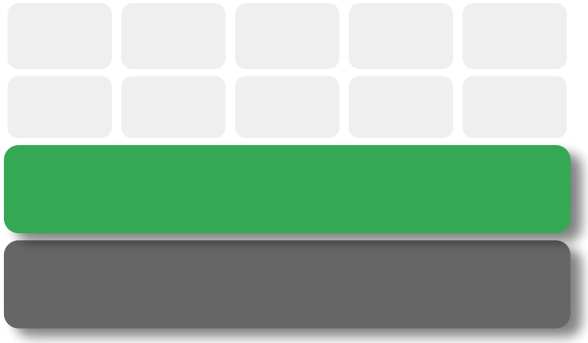
Google Cloud

kubernetes

Google Cloud

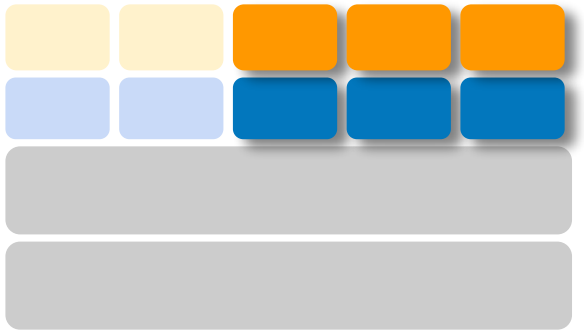
Nodes





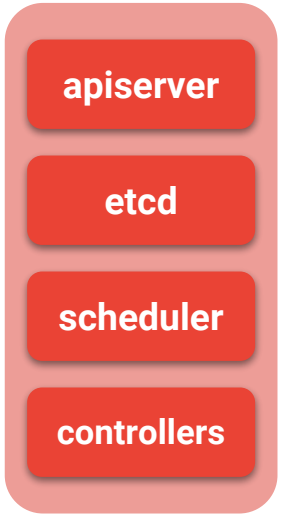
Google Cloud

Pods



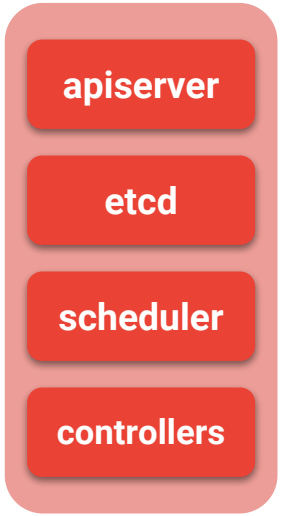
Pods

Management infrastructure

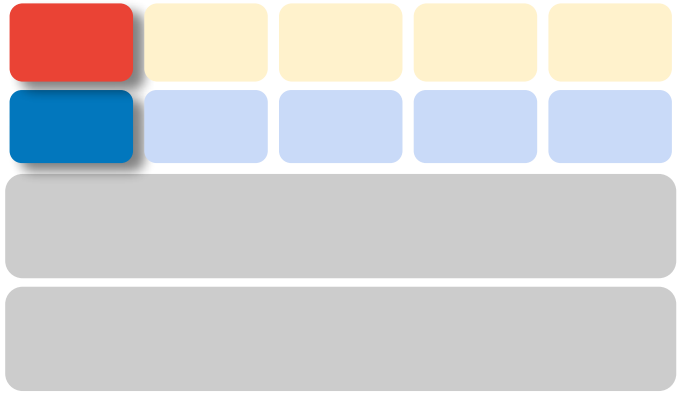


Master

Management infrastructure

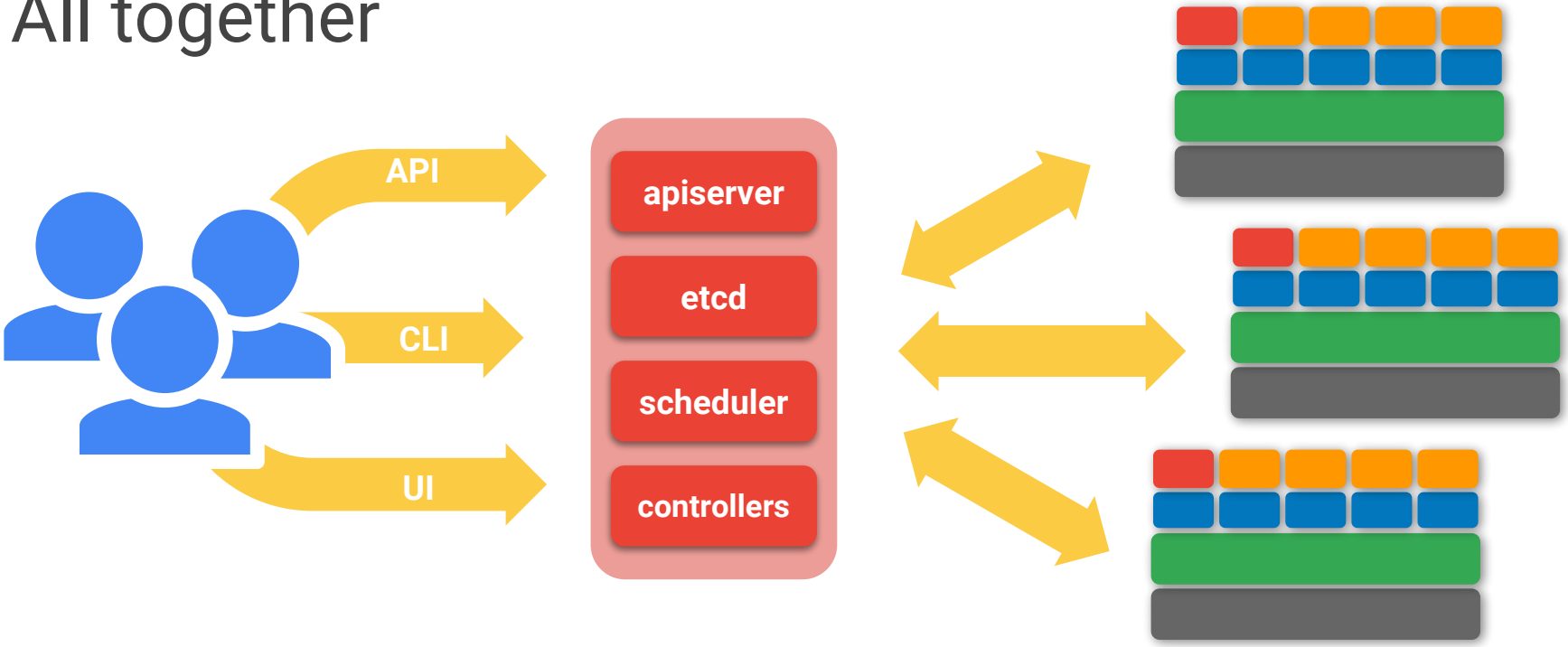


Master



kubelet

All together



Users Master Nodes

Impact on security

Containerization changes some stuff

Dynamic

Dynamic

Dynamic

Dynamic

Some things are harder for both sides

- Offense
  - Kill chains have less time to execute
  - More layers to break out of

- Defence
  - Old tricks don't work as well
  - More complexity -- bigger attack surface

Development
Deployment
Runtime

During development

Tools for securely building containerized services

- Identity, RBAC (role based access control)
- Secure inter-service communication
- Secret access control & rotation

During deployment

Secure supply chain to prevent threats from entering

- Detect known vulnerable in dependencies
- Add metadata to images
- Verify the build pipeline

During runtime

Detect and respond to threats in running containers

- Proper configuration
- Security context
- Security centric monitoring

Demo

Of a really bad day :(

04

Google Cloud

Low hanging fruit

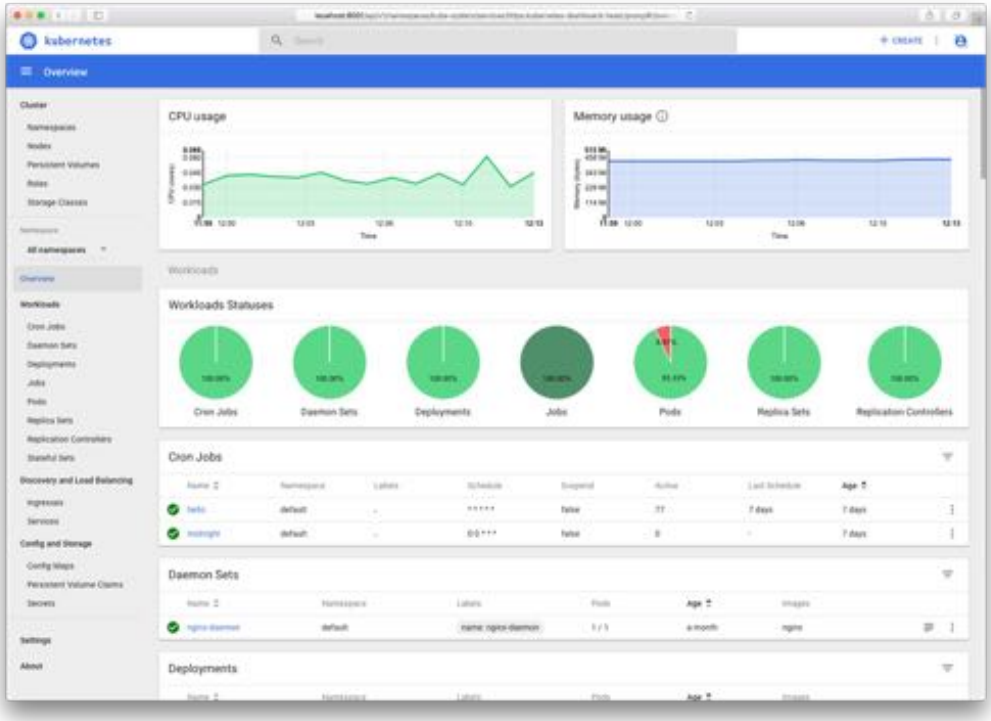
05

Google Cloud

Never do this

```
$ kubectl create -f
https://foo.com/bar.yml
```

Disable the Kubernetes Dashboard



Restrict the GCP service account

- Currently has *project editor* permission
- Only need a few narrow permissions
  - monitoring.viewer
  - monitoring.metricWriter
  - logging.logWriter

Network policies

So an attacker can't hop between pods

Great list of examples:
github.com/ahmetb/kubernetes-network-policy-recipes

Demo 2.0

Google Cloud

Higher up fruit

If you have more time

Google Cloud

Security context

Further restrict permissions with

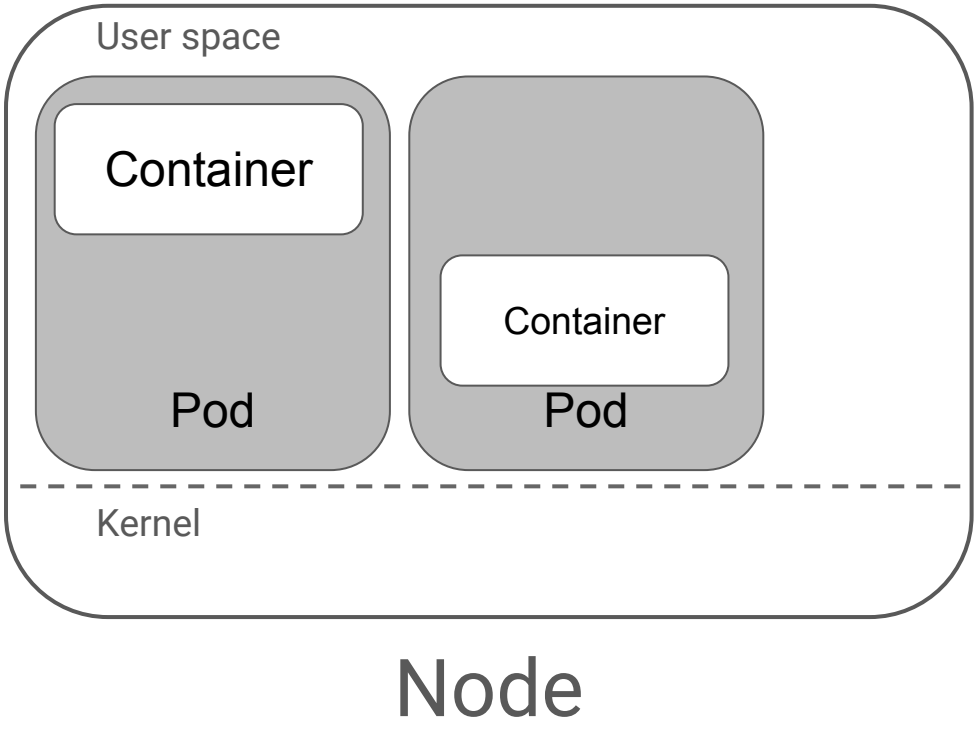
- AppArmor
- SELinux
- Seccomp

kubernetes.io/docs/tasks/configure-pod-container/security-context/

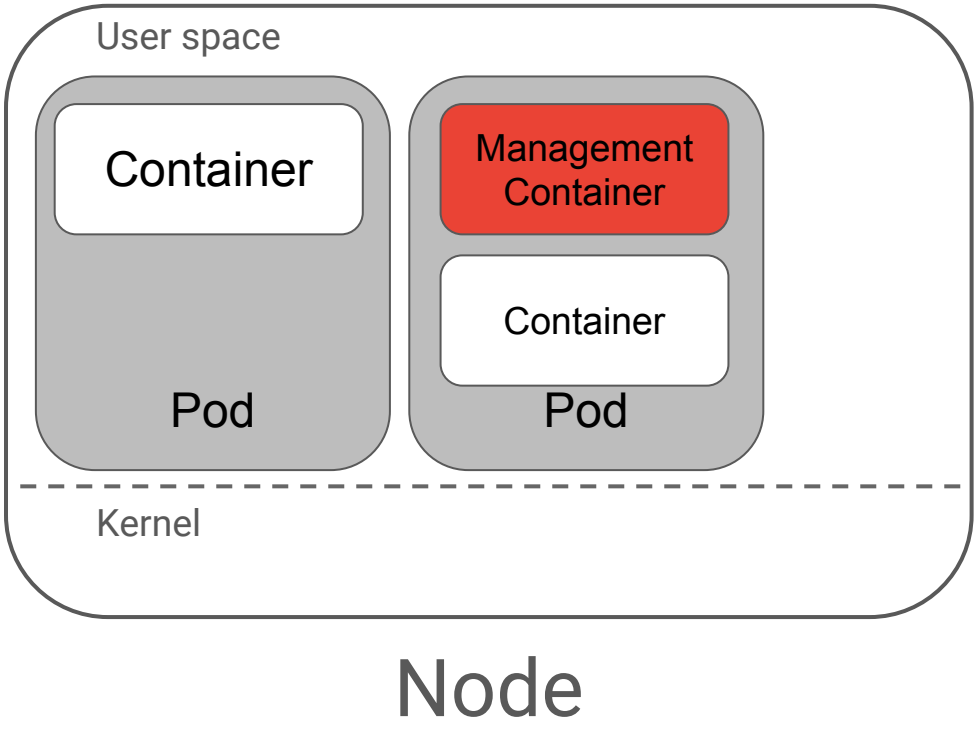
Security monitoring

- Hook into your cluster
- Log a bunch of stuff
- More policies
  - alerts
  - automatic remediation
  - forensics
- Mostly commercial products… for now

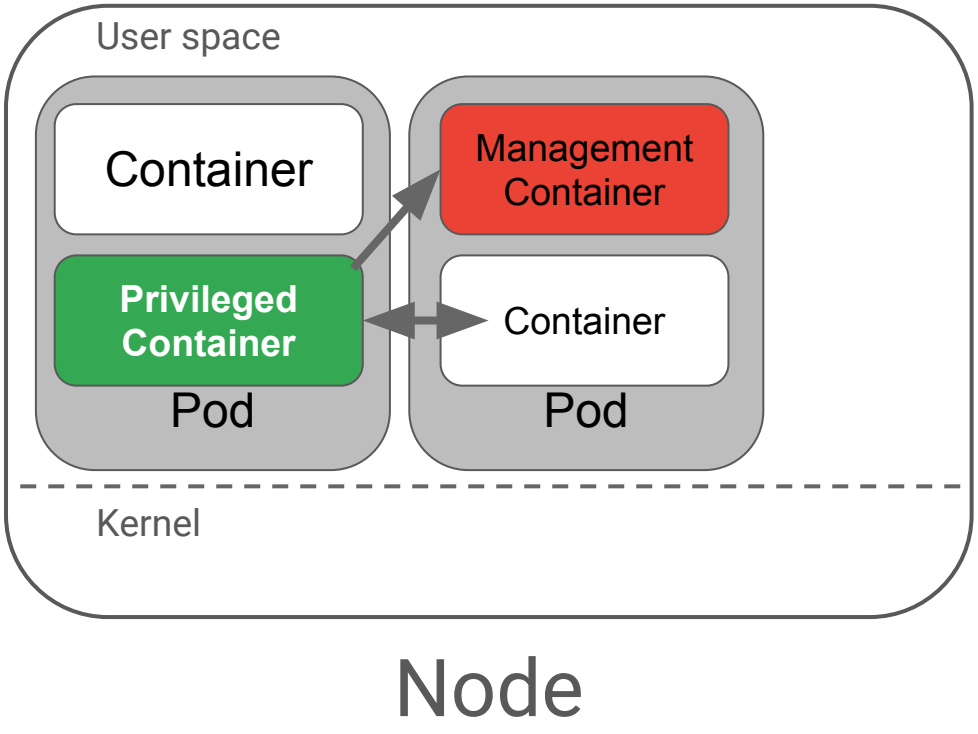
Deployment models



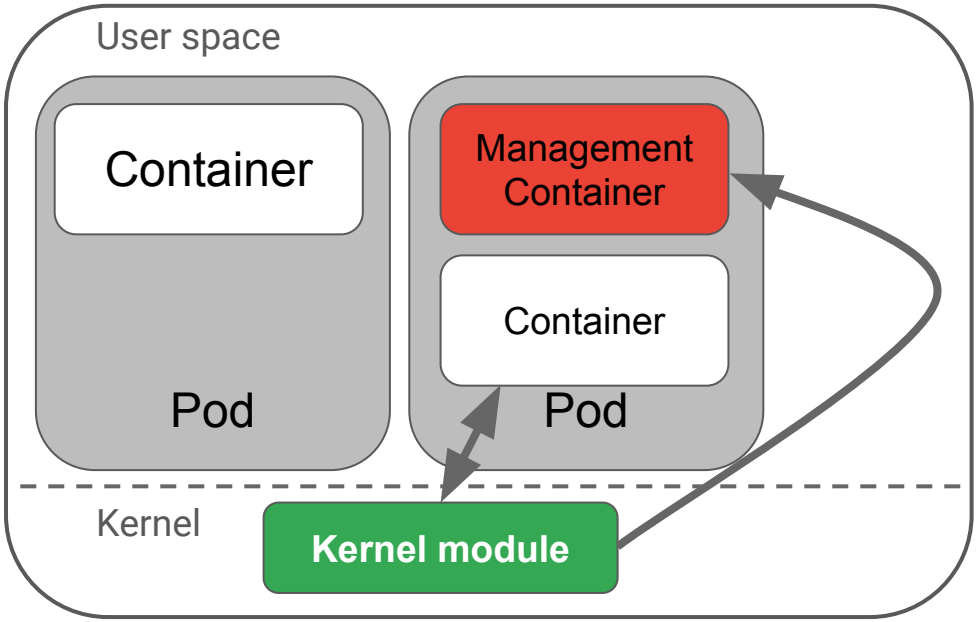
Deployment models



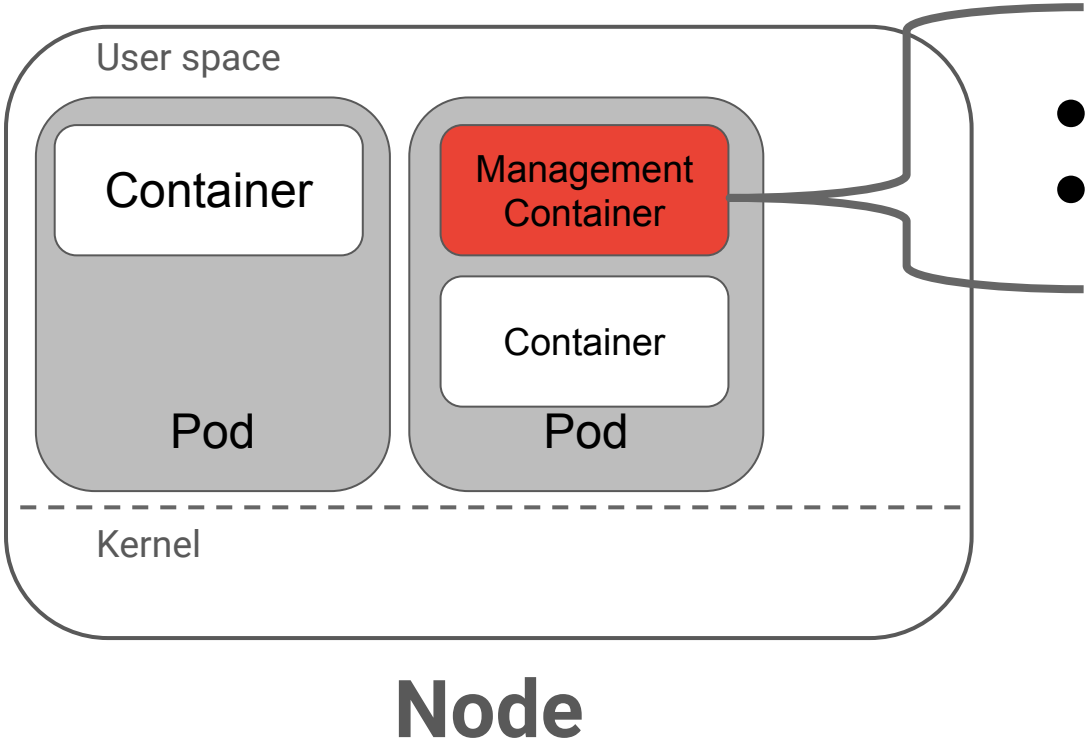
Deployment models



Deployment models

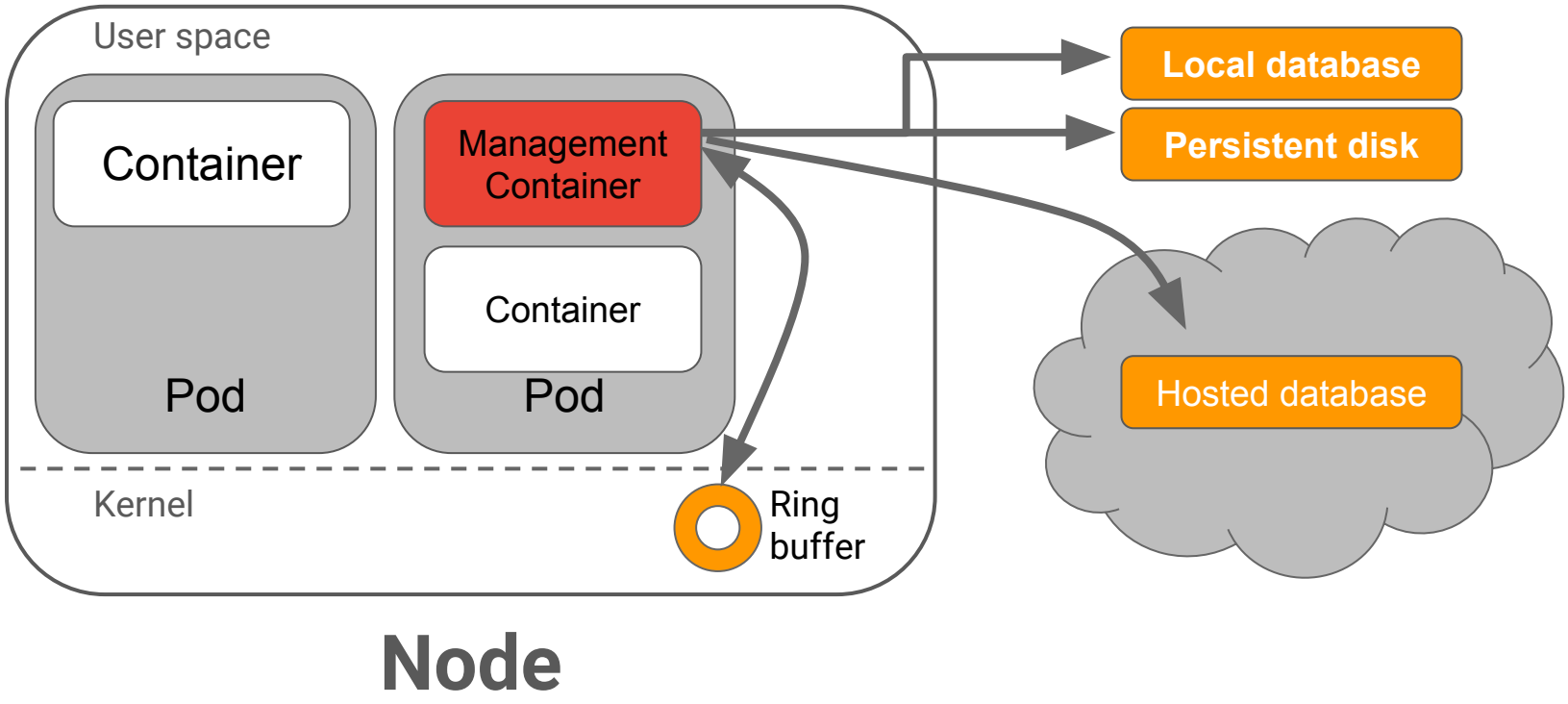


Deployment models



- Network events
- System calls

Deployment models



Open source options

- Sysdig
  - sysdig
  - Inspect
  - Falco
- Cilium
- Capsule8

What we discussed

Security overview

Containers & Kubernetes

Impact on security

Low hanging fruit

Higher up fruit

Thank you!