goto; copenhagen

GOTO Copenhagen 2017
Conference Oct. 1-3, 2017

Lunar Way's journey towards Cloud Native Utopia

Speaker Kasper Nissen

Follow us @gotocph

goto; copenhagen

GOTO Copenhagen 2017

Conference Oct. 1-3, 2017

Click ‘Rate Session’ to rate session and ask questions.

Follow us @gotocph

goto; copenhagen

GOTO Copenhagen 2017

Conference Oct. 1-3, 2017

Please

Remember to rate this session

Thank you!

Follow us @gotocph

goto; copenhagen

GOTO Copenhagen 2017
Conference Oct. 1-3, 2017

Did you **remember to rate** the previous session ?

Follow us @gotocph

goto;
copenhagen

Follow us @gotocph

@phennex

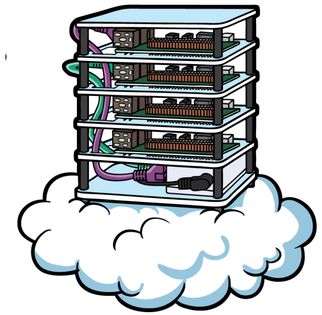
@lunarway

Kasper Nissen

DevOps & Cloud Architecture @ Lunar Way

lunar way

Cloud Native Aarhus
Cloud Native DK



kubecloud.io

lunar way

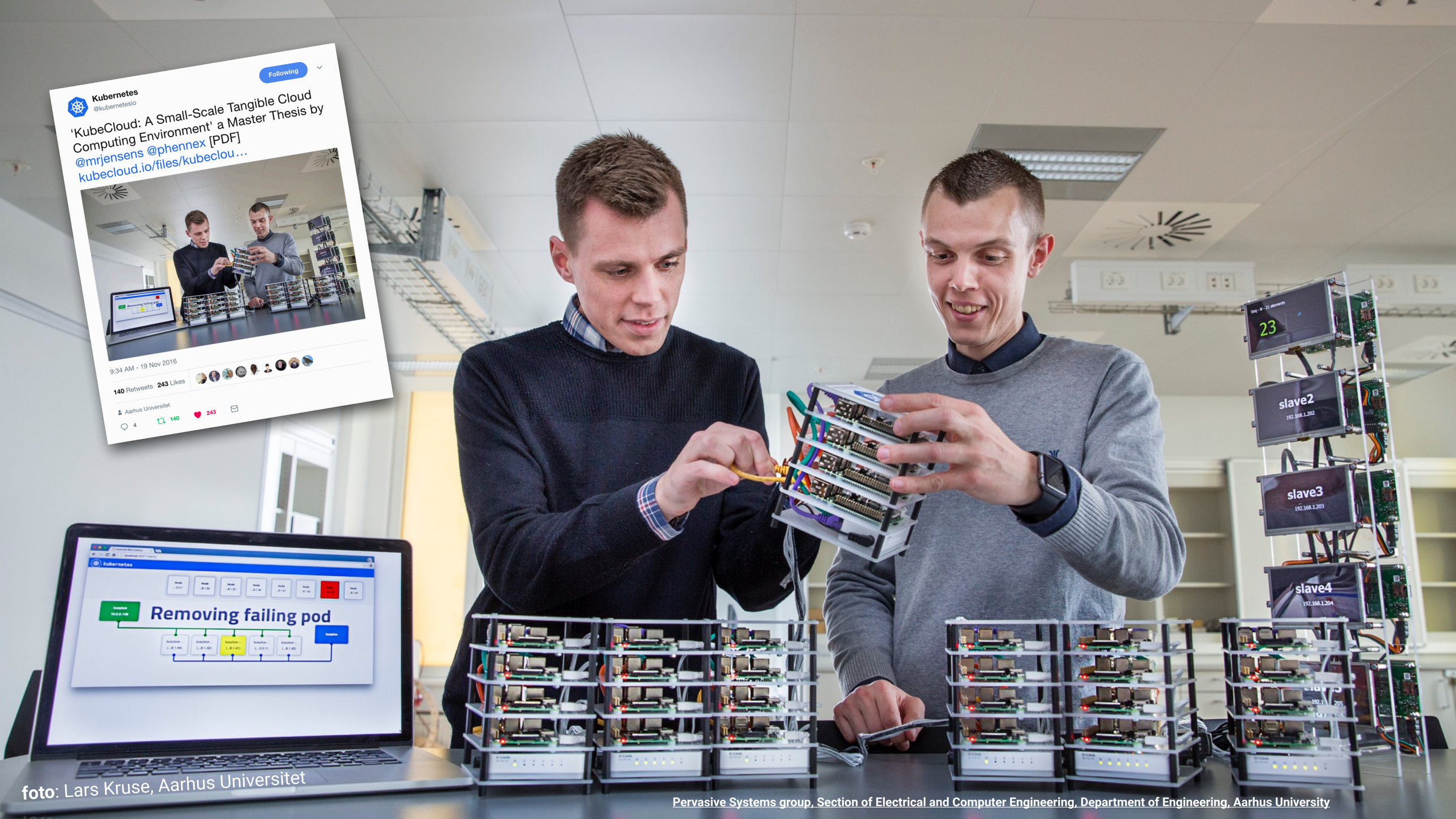


foto: Lars Kruse, Aarhus Universitet

Pervasive Systems group, Section of Electrical and Computer Engineering, Department of Engineering, Aarhus University

Opret dig og få

Opret dig og få

Vision

We're living in the era of mobile/digital only
– we believe banking and commerce should to.

Our vision is to ***rethink the interaction with money and defining a completely new category*** - by introducing a new money app.

It’s the complex coordination between banking services and commerce use:

- How I save money.
- How I get money.
- How I spend money.

Numbers

12K+
customers

1,3M+
transactions

30+
microservices

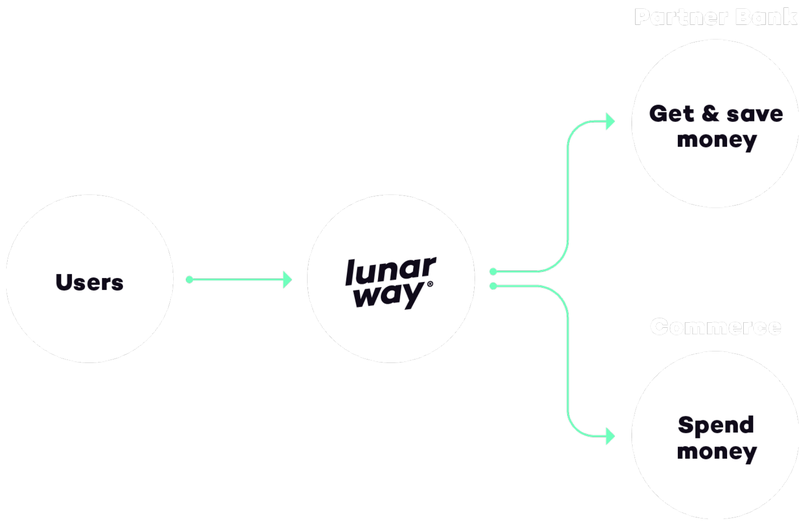
35+
employees

700M+ DKK
through our system

3
kubernetes clusters

The partner model

- All money is in the partner bank
- Leverage the partner banks' infrastructure and compliance



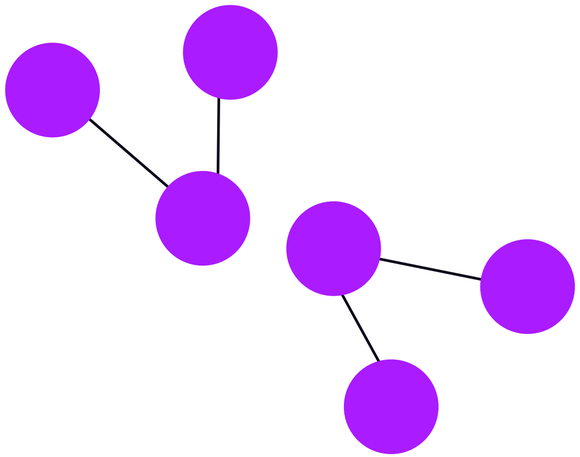
lunar way®

Lunar Way's journey towards Cloud Native Utopia

Cloud Native?

lunar way

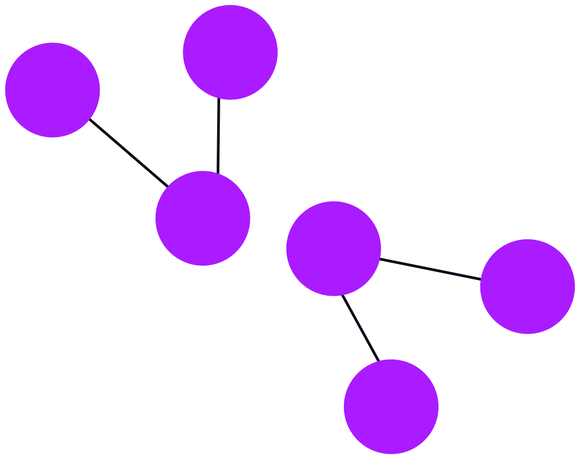
Cloud Native?

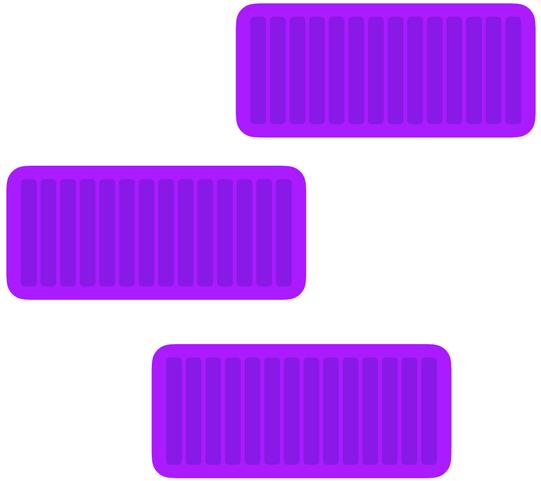


Microservice oriented

lunar way

Cloud Native?



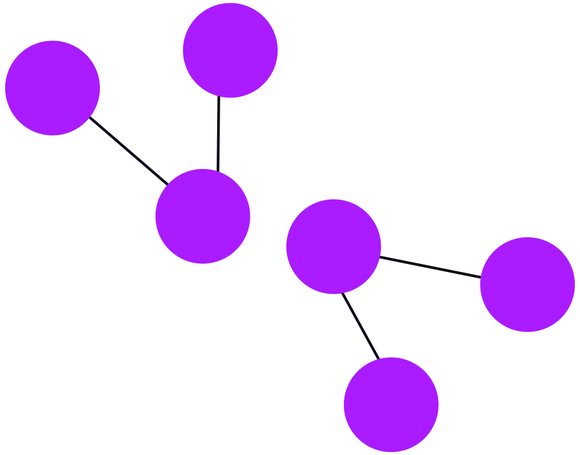


Microservice oriented

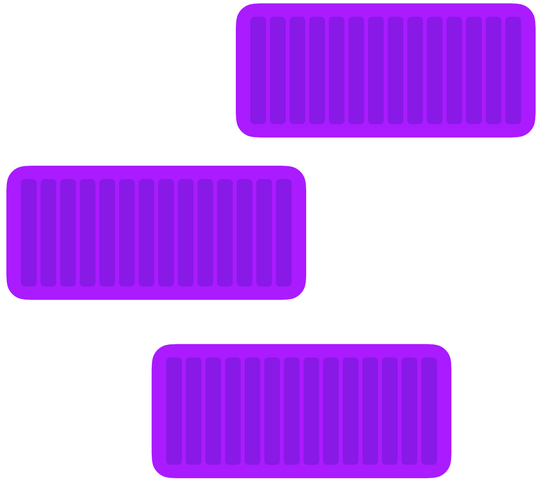
Container packaged

lunar way®

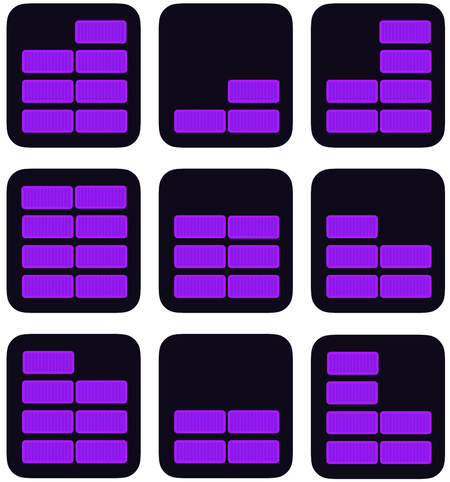
Cloud Native?



Microservice oriented



Container packaged



Dynamically scheduled

lunar way®

Cloud Native is structuring teams, culture and technology to utilize automation and architectures to manage complexity and unlock velocity.

Joe Beda, CTO at Heptio

lunar way®

Why go there?

Reduce time-to-market

Allow for continuous innovation

Ability to pivot

Our journey so far

Ruby on Rails

Strangling Rails

Kubernetes

2015

2016

2017

2018

Integrations
First services

Microservices with
monolithic deployment

The end of our
“monolith”

lunar way

Our journey so far

Ruby on Rails

Strangling Rails

Kubernetes

2015

2016

2017

2018

Integrations
First services

Microservices with
monolithic deployment

The end of our
“monolith”

lunar way

Our journey so far

Ruby on Rails

Strangling Rails

Kubernetes

2015

2016

2017

2018

Integrations
First services

Microservices with
monolithic deployment

The end of our
“monolith”

lunar way

Our journey so far

Ruby on Rails

2015

Integrations
First services

Strangling Rails

2016

Kubernetes

2017

Microservices with
monolithic deployment

2018

The end of our
“monolith”

lunar way

Our journey so far

Ruby on Rails

2015

Integrations
First services

Strangling Rails

2016

2017

Microservices with
monolithic deployment

Kubernetes

2018

The end of our
“monolith”

lunar way

Our journey so far

Ruby on Rails

2015

Integrations
First services

Strangling Rails

2016

Kubernetes

2017

Microservices with
monolithic deployment

2018

The end of our
“monolith”

lunar way

Our journey so far

Ruby on Rails

2015

Integrations
First services

Strangling Rails

2016

Kubernetes

2017

Microservices with
monolithic deployment

2018

The end of our
“monolith”

lunar way

Our journey so far

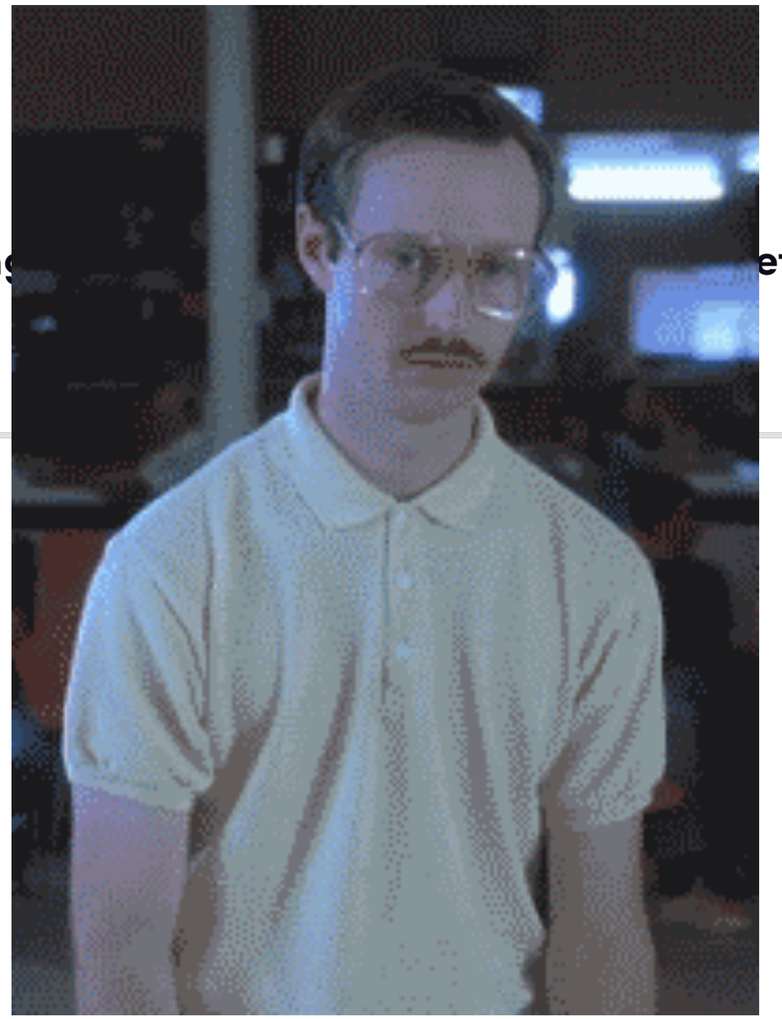
Ruby on Rails

2015

Integrations
First services

Strang

2016



etes

2018

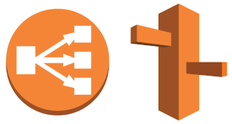
The end of our
“monolith”

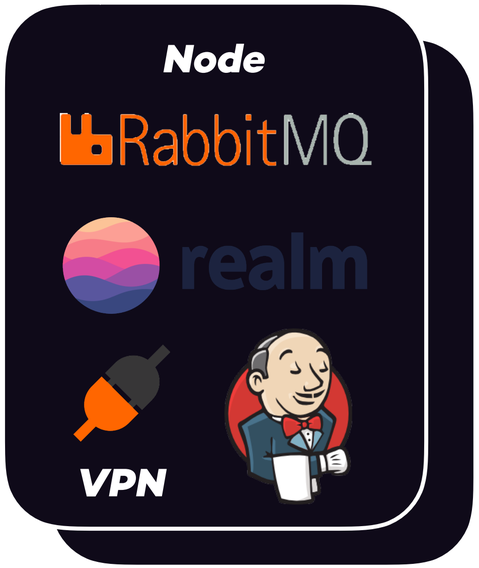
lunar way

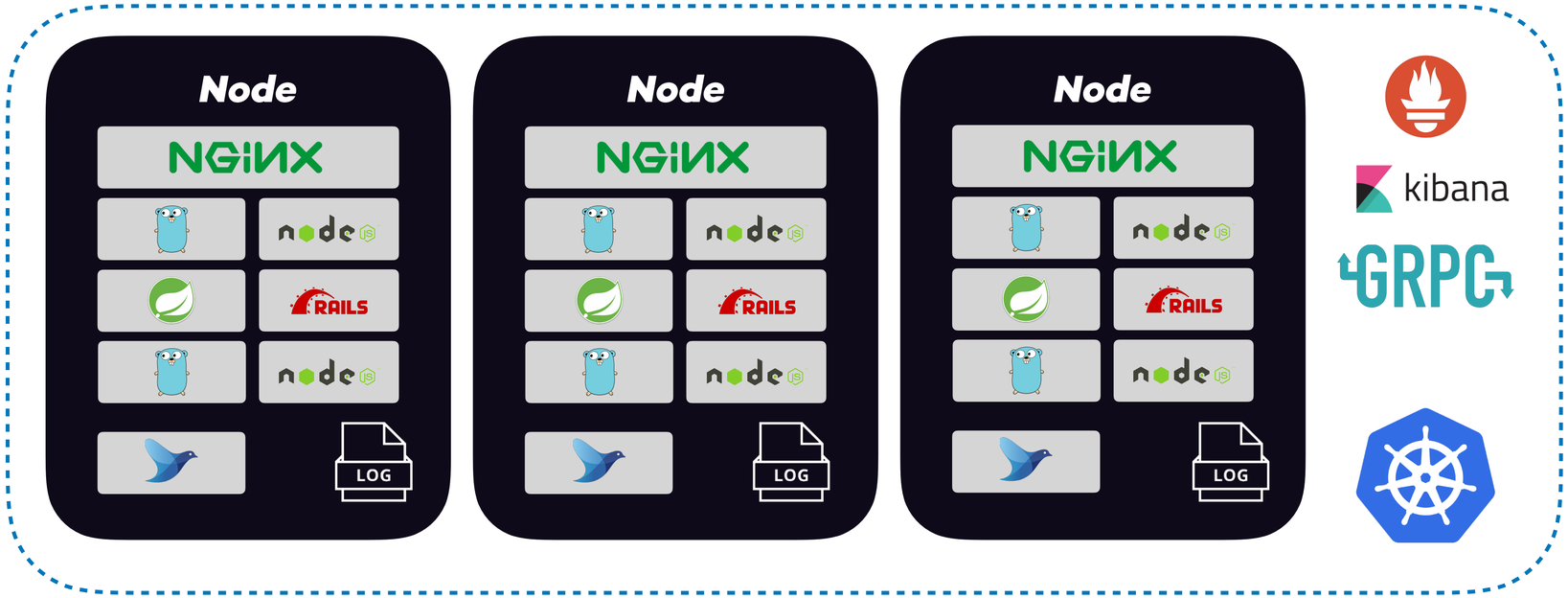
So, what do we have running?

lunar way

Highlevel overview

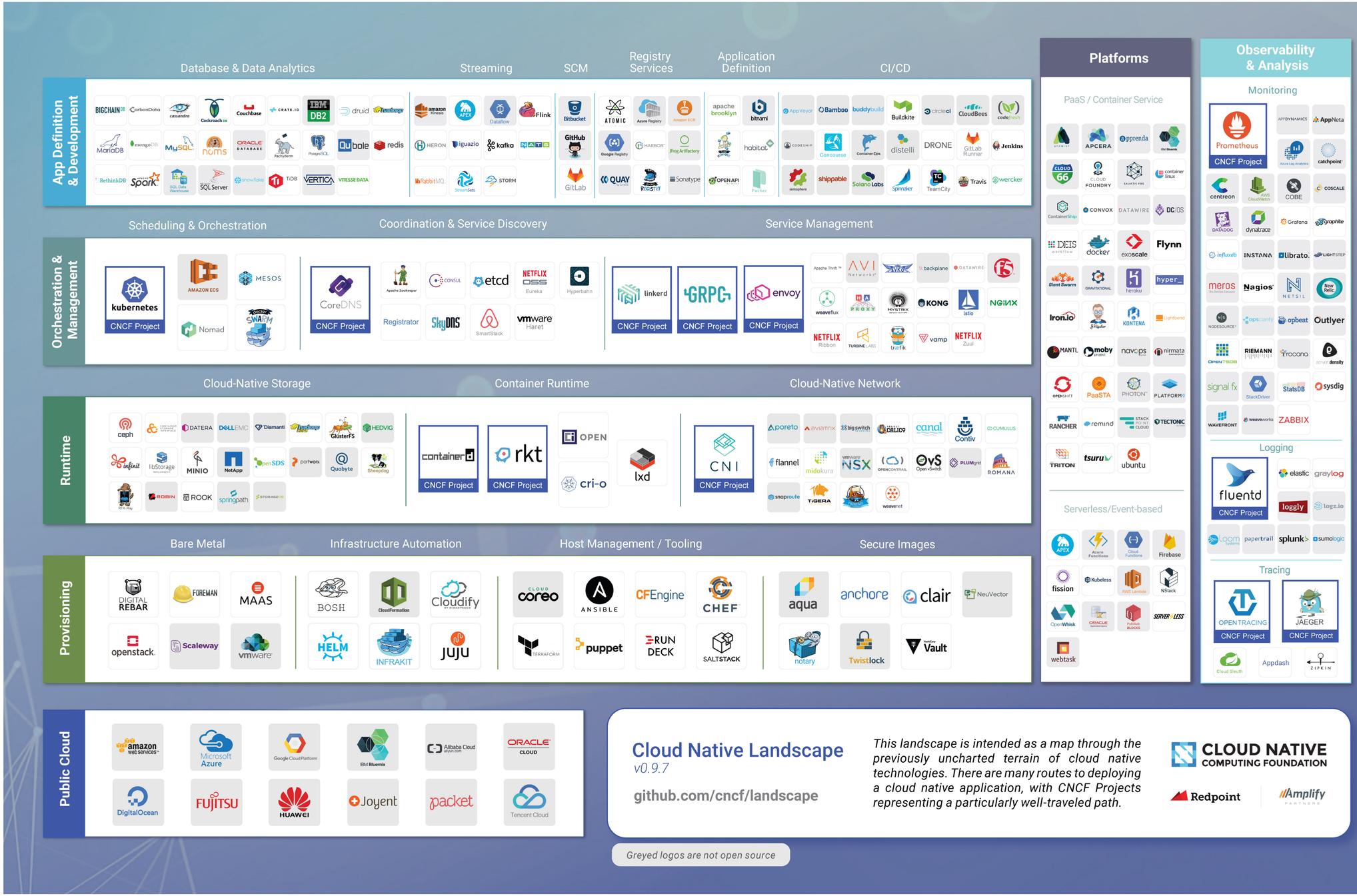






How to navigate the Cloud Native ecosystem?

lunar way


App Definition & Development
Database & Data Analytics
Streaming
SCM
Registry Services
Application Definition
CI/CD
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Service Management
kubernetes
CNCF Project
CoreDNS
CNCF Project
linkerd
CNCF Project
GRPC
CNCF Project
envoy
CNCF Project
Runtime
Cloud-Native Storage
Container Runtime
Cloud-Native Network
containerd
CNCF Project
rkt
CNCF Project
CNI
CNCF Project
Provisioning
Bare Metal
Infrastructure Automation
Host Management / Tooling
Secure Images
Public Cloud
Platforms
PaaS / Container Service
Serverless/Event-based
Observability & Analysis
Monitoring
Prometheus
CNCF Project
Logging
fluentd
CNCF Project
Tracing
OPENTRACING
CNCF Project
JAEGER
CNCF Project
Cloud Native Landscape
v0.9.7
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.
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
Greyed logos are not open source
lunar way

CLOUD NATIVE COMPUTING FOUNDATION

Currently Hosted Projects

Kubernetes

Orchestration

Prometheus

Monitoring

OpenTracing

Distributed Tracing API

Fluentd

Logging

linkerd

Service Mesh

gRPC

Remote Procedure Call

CoreDNS

Service Discovery

containerd

Container Runtime

rkt

Container Runtime

CNI

Networking API

Envoy

Service Mesh

Jaeger

Distributed Tracing

lunar way

Dynamic Scheduling with Kubernetes

lunar way

Why do we need an orchestration tool?

- **Scheduling** - where are our containers going to run?
- **Availability** - scale to a desired state
- **Resilience** - if a container dies, we need a new one to spin up
- **Storage** - where do we store our data?
- **Deployments** - we want a way to canary deployments
- **Updates** - how can we update our containers without downtime?
- **Networking** - how are our containers going to communicate?
- **Service Discovery** - how will they find each other?

lunar way®

What is Kubernetes?

Kubernetes is an open-source platform designed to automate deploying, scaling and operating application containers.

- **Portable:** public, private, hybrid, multi-cloud
- **Extensible:** modular, pluggable, hookable, composable
- **Self-healing:** auto-placement, auto-restart, auto-replication, auto-scaling

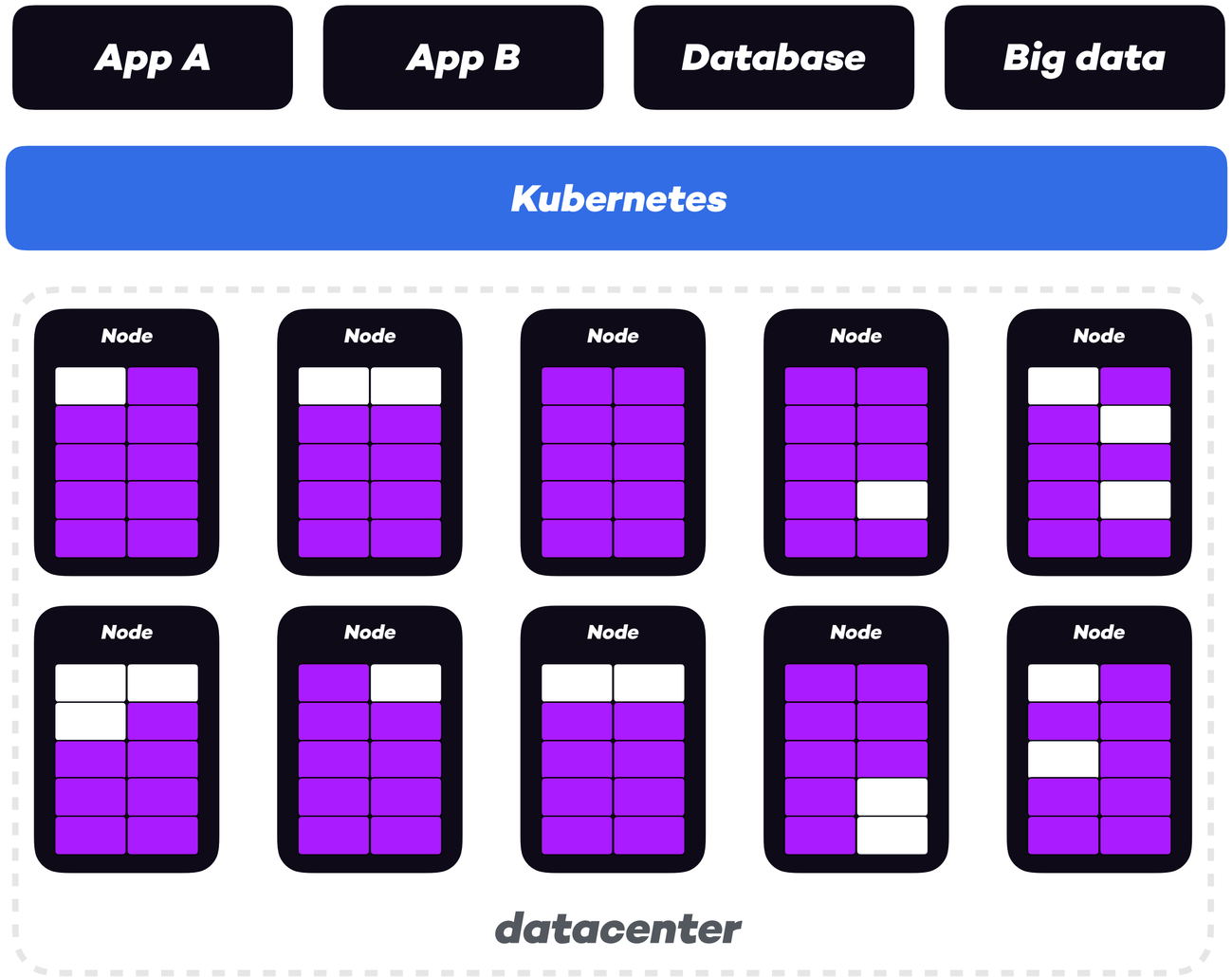
Google started the Kubernetes project in 2014.

Kubernetes builds upon a decade and a half of experience that Google has with running production workloads at scale, combined with best-of-breed ideas and practices from the community.

Source: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

lunar way

What does it do?



lunar way

Kubernetes at Lunar Way

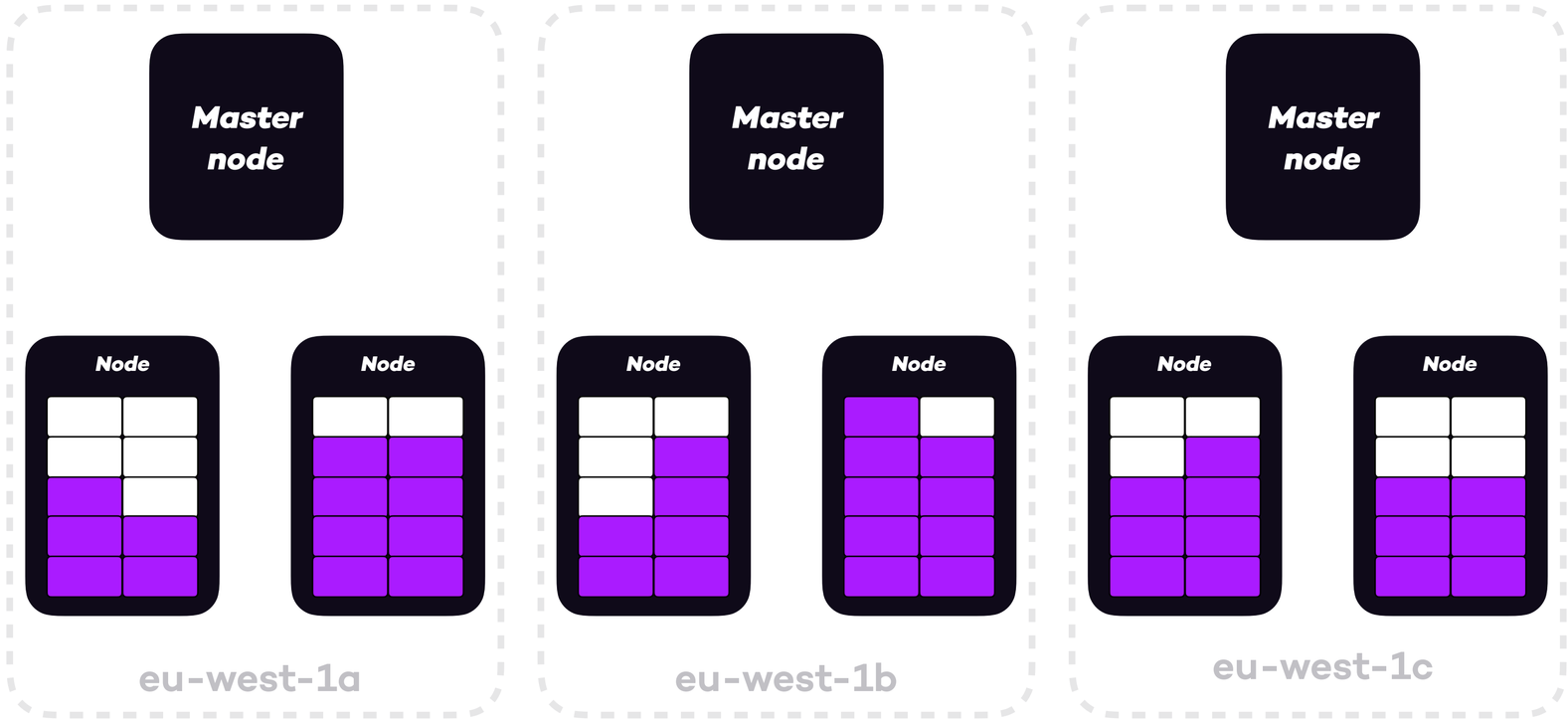
3
Clusters
in AWS

Minikube
for local
development

KOPS
for
maintaining
cluster

lunar way

Highly available



lunar way

What do we think of it?

Autonomous services
Squads can work independent of other squads.

High availability
Kubernetes takes care of container failures.

Easy maintenance
KOPS to spin up our clusters, and maintain them.

Freedom & Flexibility
We run many different type of workloads in the cluster.
Gives us mobility to become cloud agnostic.

Easy independent deployment
Kubernetes allows us to deploy multiple times a day.

Scalable infrastructure
Scaling the infrastructure is easy, both on node and container level.

lunar way

*Containerization transforms the data center from being **machine-oriented** to being **application-oriented***

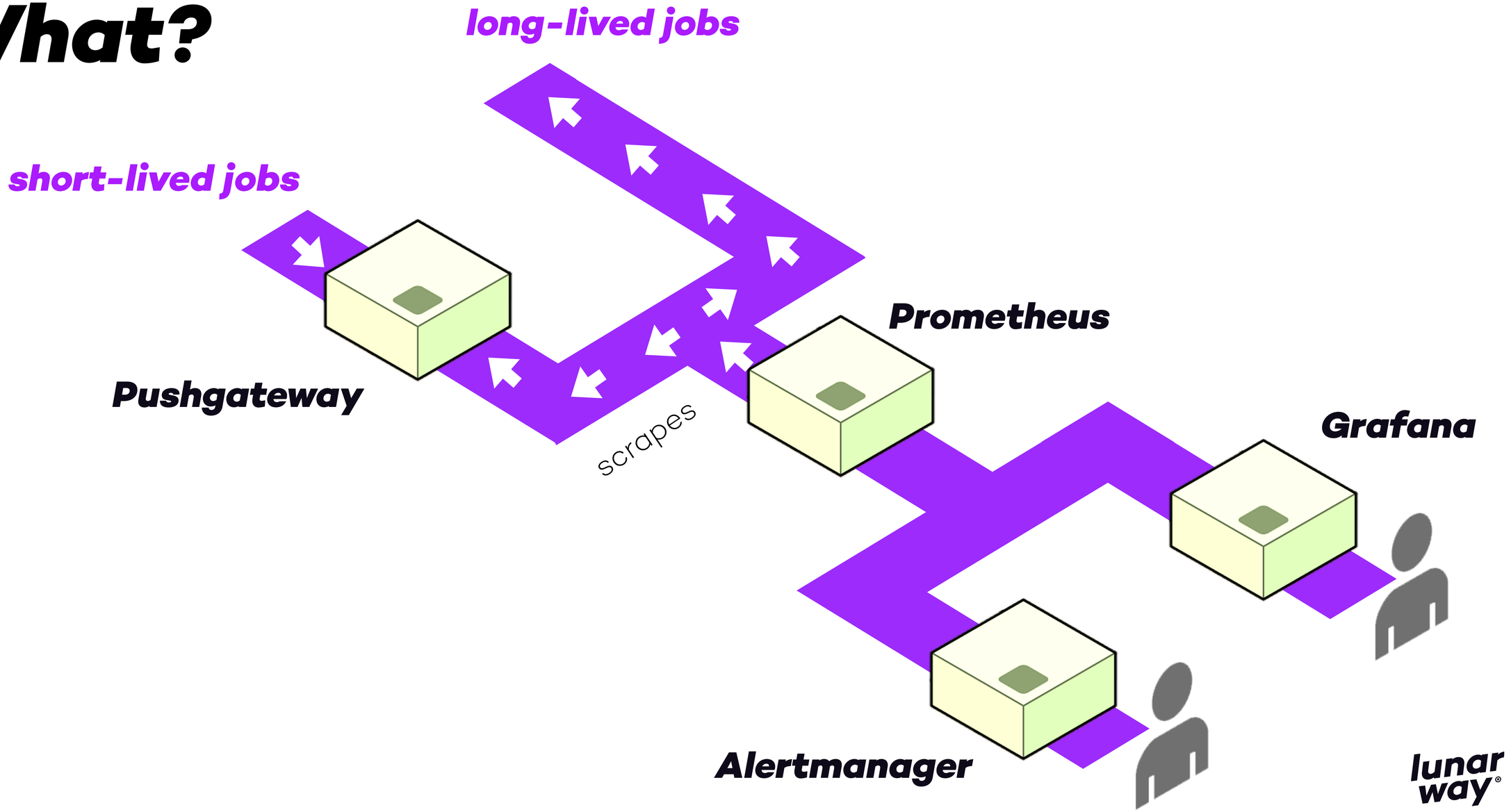
Burns et al., Borg, Omega, and Kubernetes, 2016

lunar way®

Monitoring with Prometheus

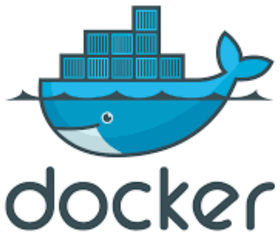
lunar way

What?



lunar way

What metrics are we collecting?



RabbitMQ

NGINX

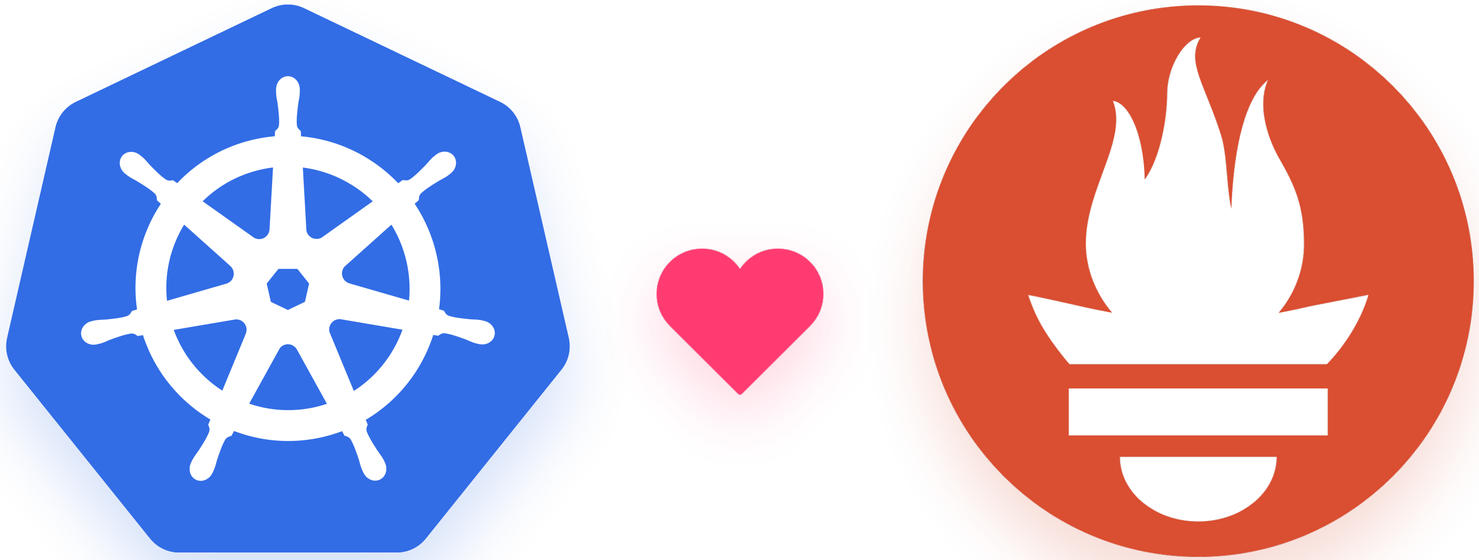
Amazon EC2

elasticsearch

Custom Metrics

node.js

lunar way



```
annotations:
  prometheus.io/scrape: 'true'
```

lunar way

What do we think of it?

Provides great insights to all of our services

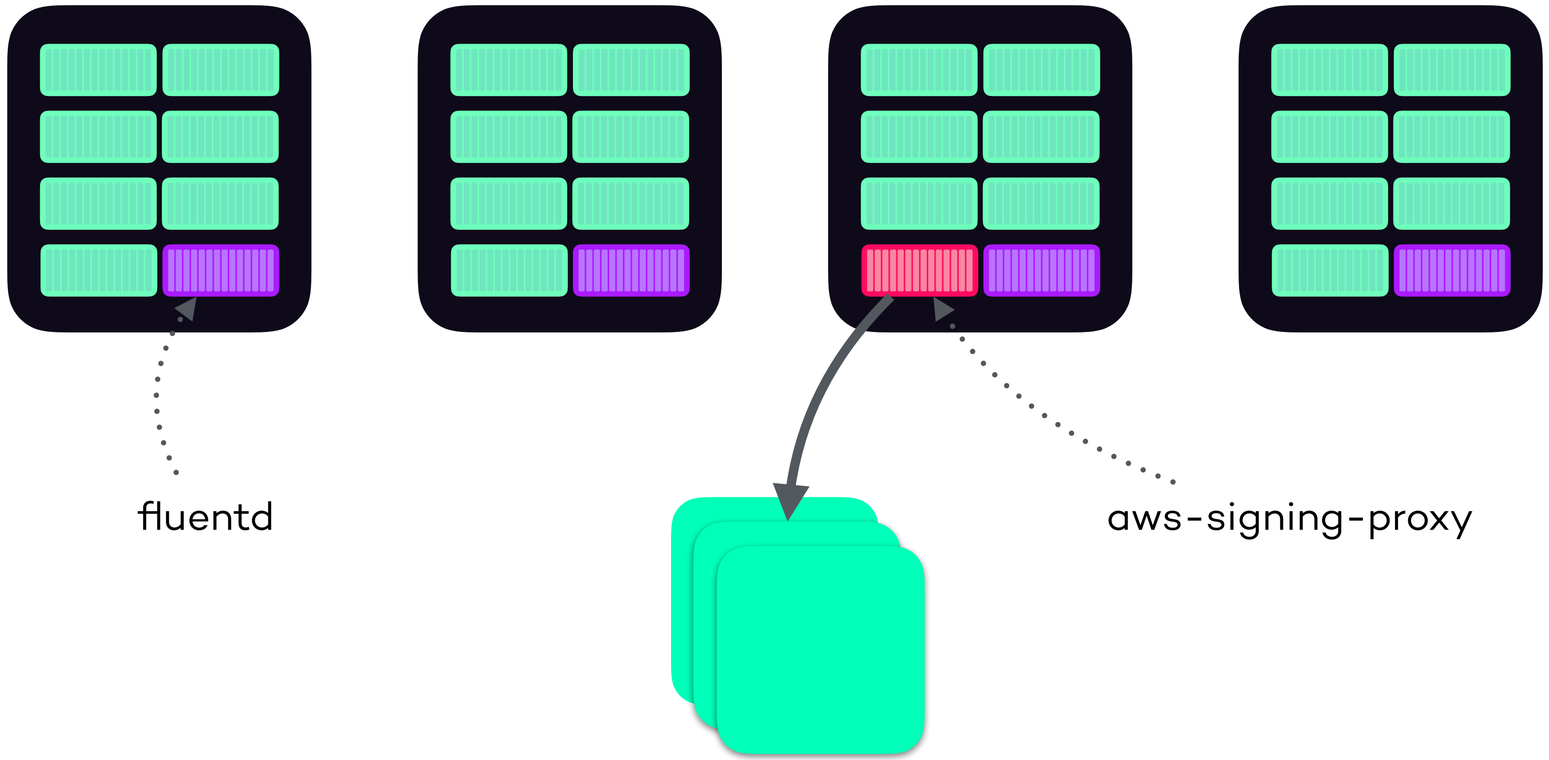
Makes it easy for developers to instrument their services

Integrates well with many different services

Log Collection with Fluentd

lunar way

Logging setup



AWS Elasticsearch Cluster

lunar way

What do we think of it?

Works great with Kubernetes.
Deployed as a DaemonSet

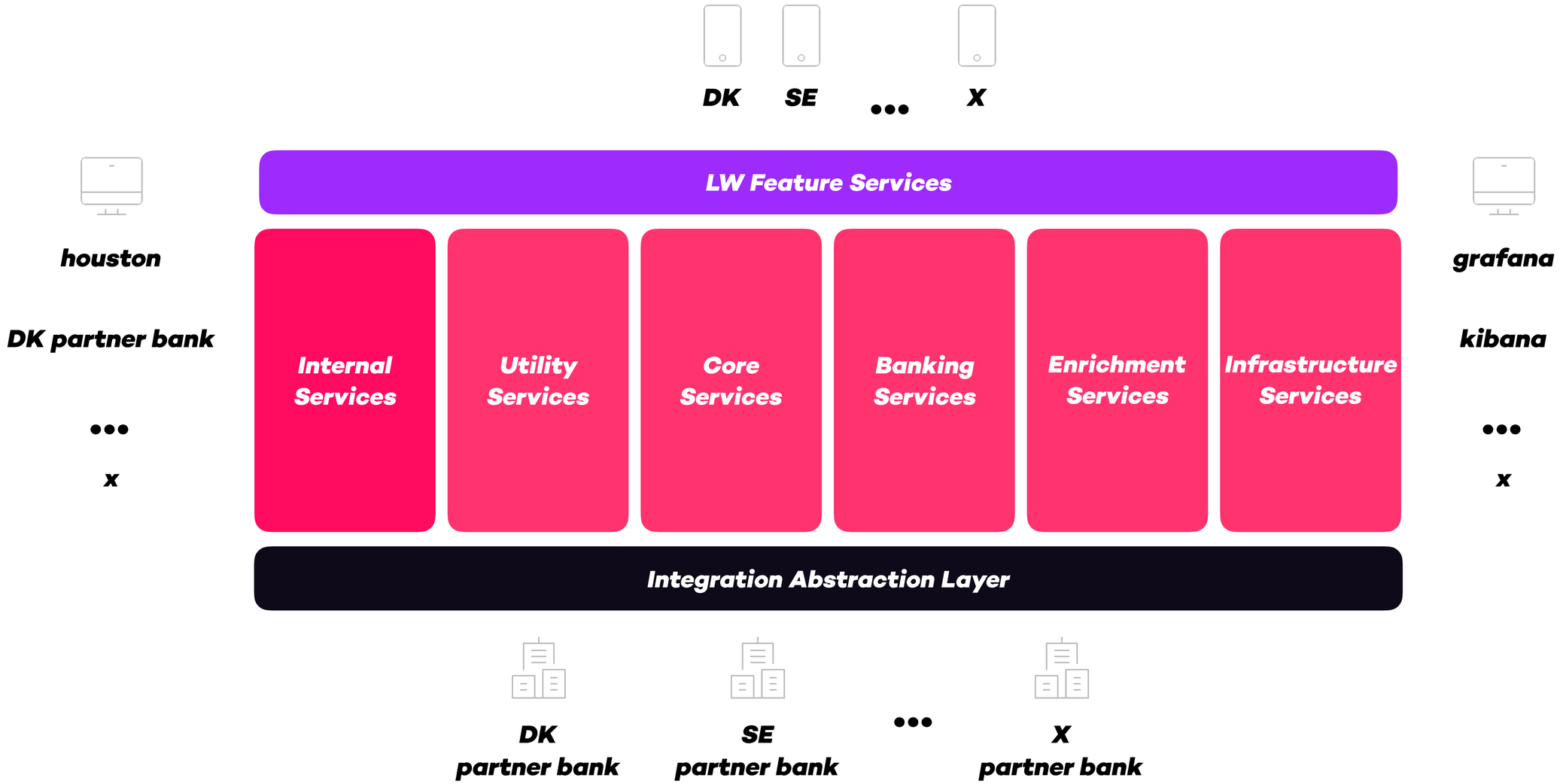
Small memory footprint

Proven reliability and
performance.

Back to what it is we are trying to do...

lunar way®

Building a scalable architecture

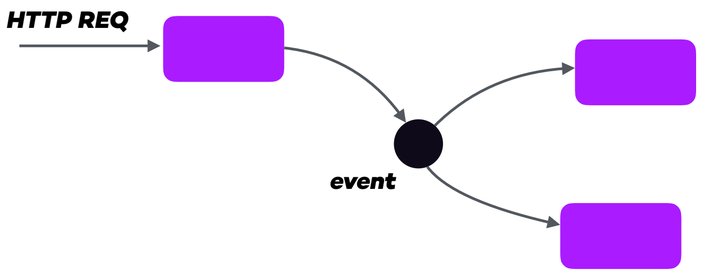


lunar way

How are we building our services?

lunar way

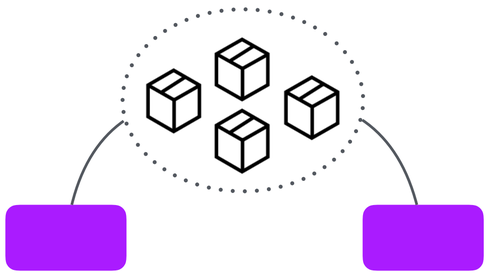
How are we building our services?



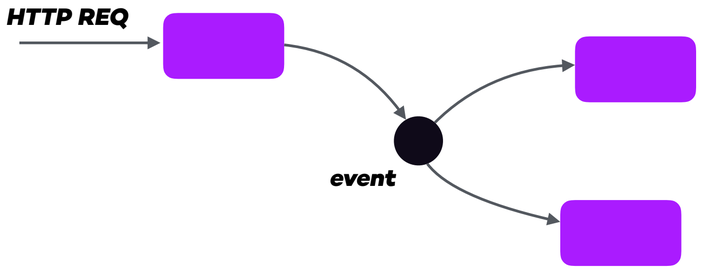
Asynchronous first

lunar way

How are we building our services?



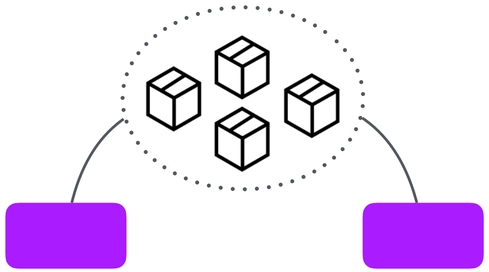
Shared Core Dependencies



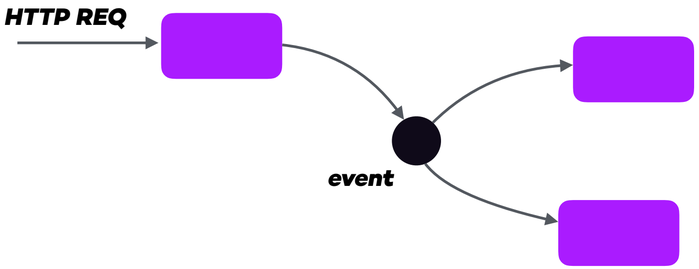
Asynchronous first

lunar way

How are we building our services?



Shared Core Dependencies

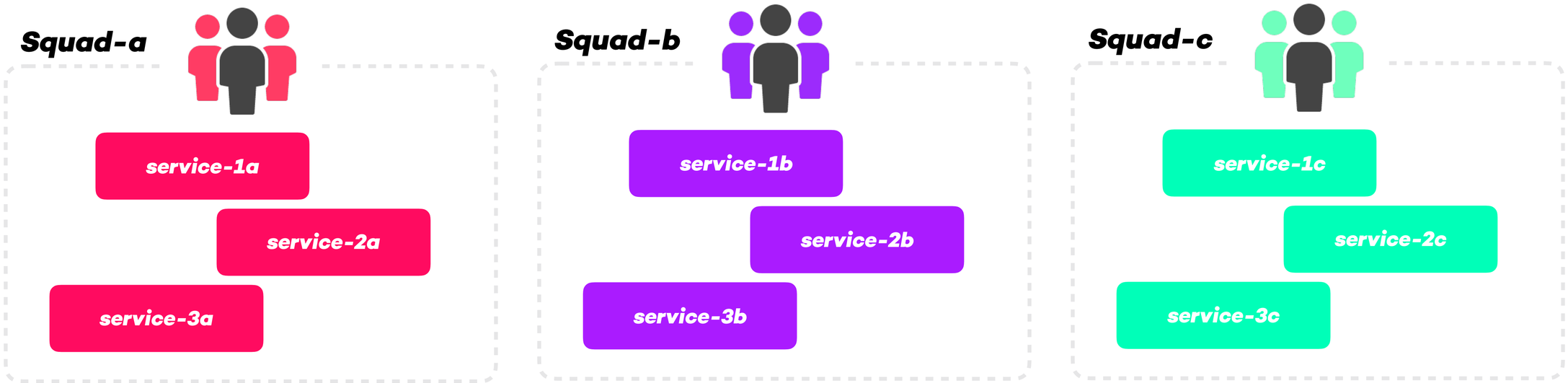


Asynchronous first

Self-contained

lunar way

Organization & Culture

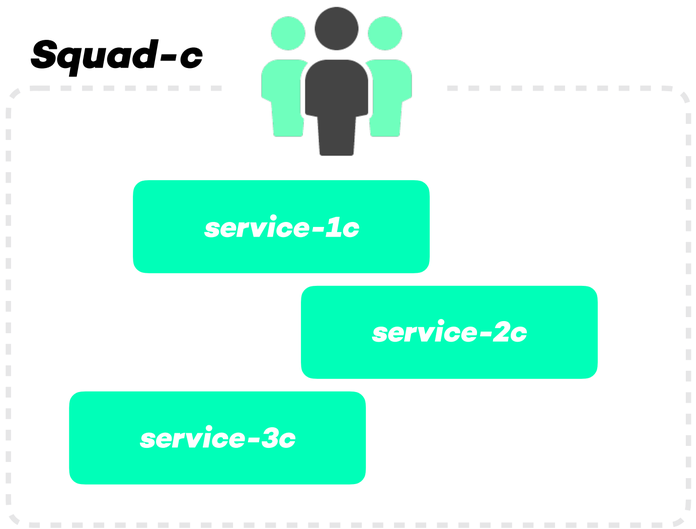


lunar way

#squad-core

Environments

Deployment



Logging

Pipeline

Monitoring

Alerting

Communication

lunar way

#squad-core

Provide feature teams with tools and services, that allows them to move faster and build more quality in.

lunar way®

Did we find Cloud Native utopia?

lunar way

We are on the right path!

We are doing microservices.
We package services in containers.
We deploy these in a dynamically scheduled environment.

But, there's still room for improvement...

lunar way

Challenges

lunar way

Challenges

lunar way

Architecture

Avoid building a distributed monolith

Use bounded context pattern to avoid cross cutting concerns.

Strangler Application Pattern

Building new functionality as new services

Asynchronous vs Synchronous

Problems when integrating with external partners. Synchronous calls from app.

Architecture

Avoid building a distributed monolith

Use bounded context pattern to avoid cross cutting concerns.

Strangler Application Pattern

Building new functionality as new services

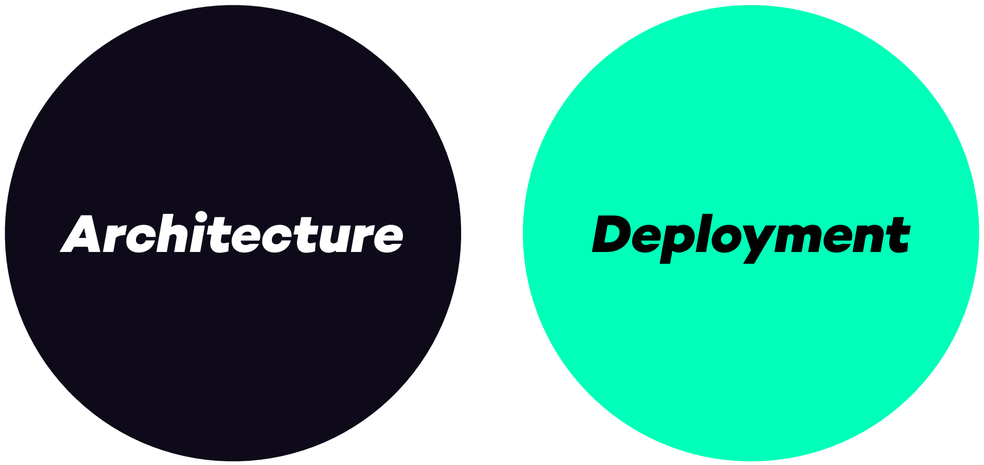
Asynchronous vs Synchronous

Problems when integrating with external partners. Synchronous calls from app.

Challenges

lunar way

Challenges



lunar way®

Deployment

Monolithic deployments

A lot of risk involved, and less frequent deployment.

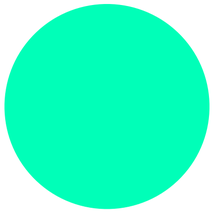
Cloud Native maturity of CI/CD

Good old Jenkins and scripts to the rescue

Configuration follows Image

Container registry just stores the image.

Deployment



Monolithic deployments

A lot of risk involved, and less frequent deployment.

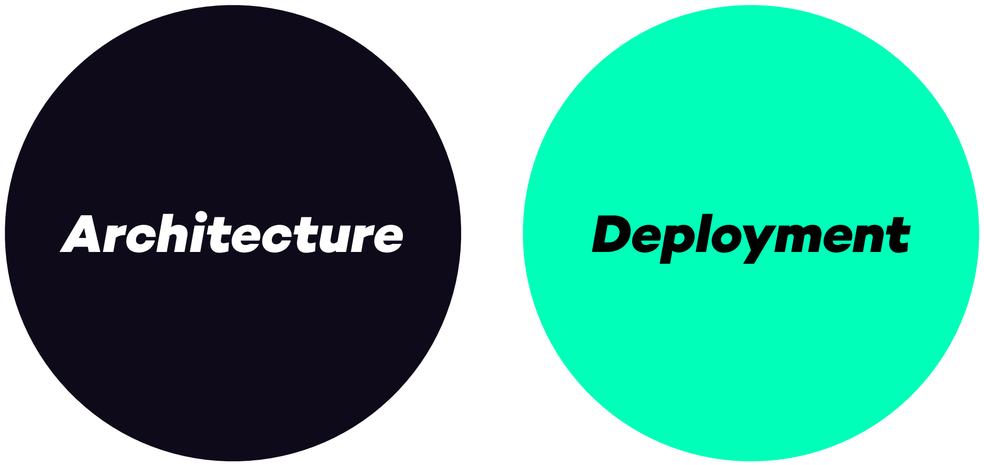
Cloud Native maturity of CI/CD

Good old Jenkins and scripts to the rescue

Configuration follows Image

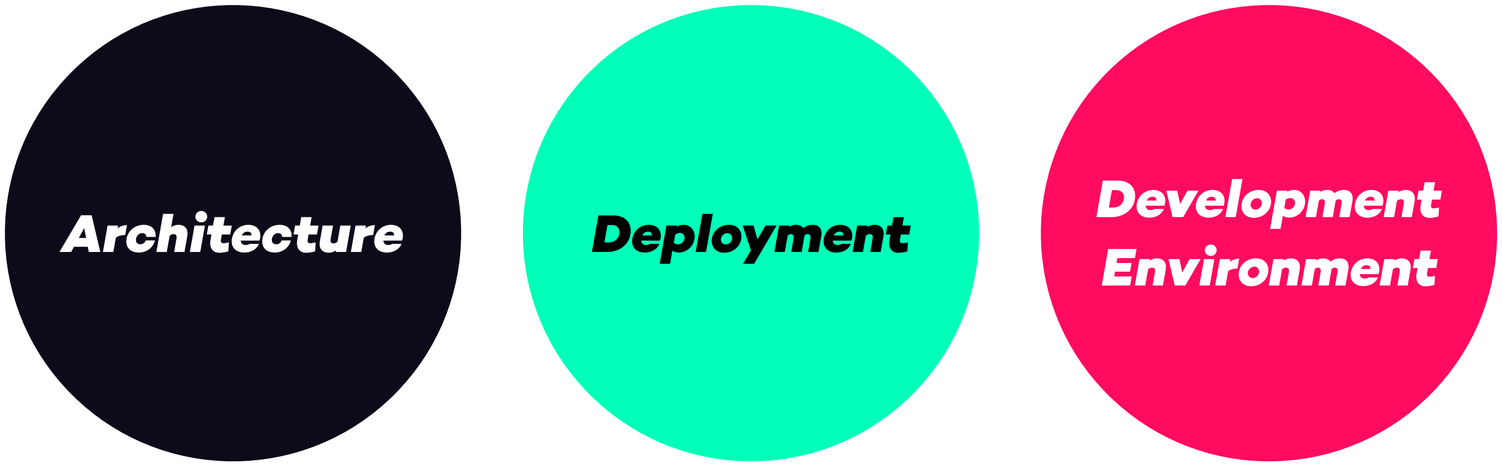
Container registry just stores the image.

Challenges



lunar way®

Challenges



lunar way®

Development Environment

Minikube is great in the beginning

What about when running 30 services?

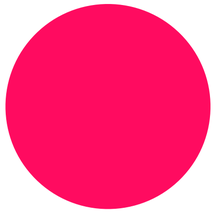
Local cluster boot time is a pain

Fetching services over the internet everytime is slow.

Proxy into a cloud environment instead?

We are looking at a project called telepresence.io

Development Environment



Minikube is great in the beginning

What about when running 30 services?

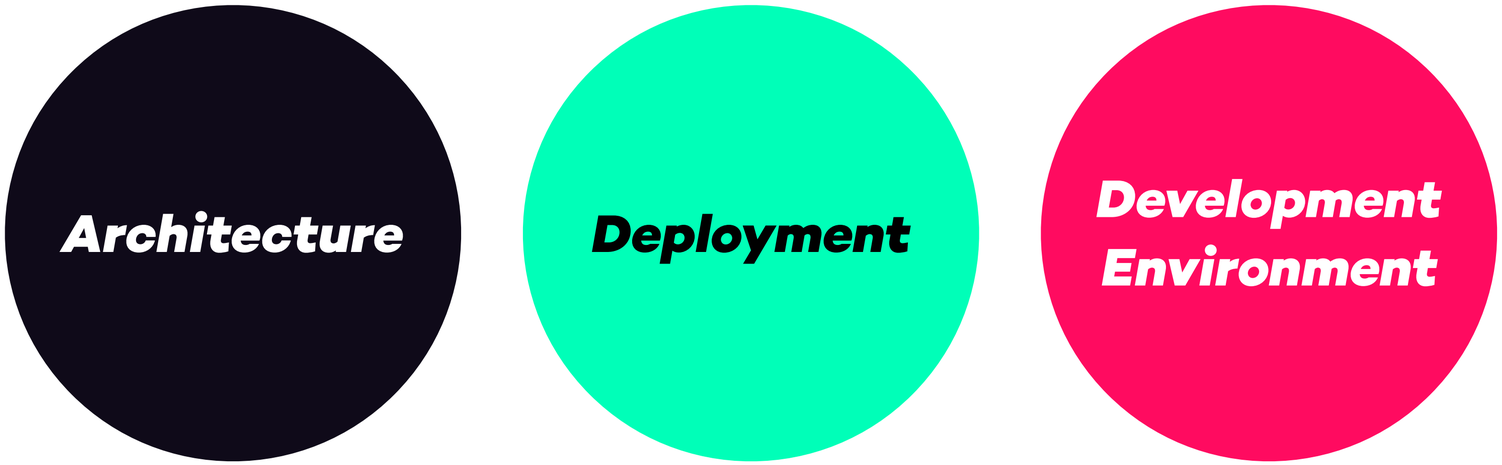
Local cluster boot time is a pain

Fetching services over the internet everytime is slow.

Proxy into a cloud environment instead?

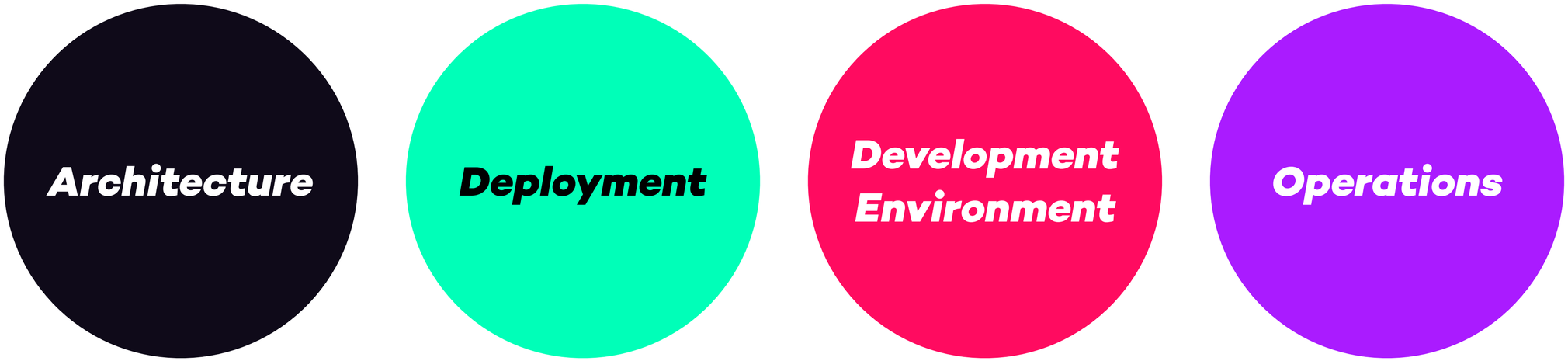
We are looking at a project called telepresence.io

Challenges



lunar way®

Challenges



lunar way®

Operations

No more SSH'ing into machines

We use kubectl for management.

Kubernetes is moving fast

Keeping up is time-consuming

Kops helps us manage our clusters

Kops makes it fairly easy to update and maintain.

Operations

No more SSH'ing into machines

We use kubectl for management.

Kubernetes is moving fast

Keeping up is time-consuming

Kops helps us manage our clusters

Kops makes it fairly easy to update and maintain.

What else are we looking at

lunar way

lunar way®

THANK YOU!

We are hiring!

kni@lunarway.com
@phennex

goto;
copenhagen

Follow us @gotocph