KubeCon | CloudNativeCon

North America 2019

KubeCon | CloudNativeCon

North America 2019

Supercharge Your Microservices CI/CD with Service Mesh and Kubernetes

Brian Redmond, Microsoft
Technical PM – Customer Success

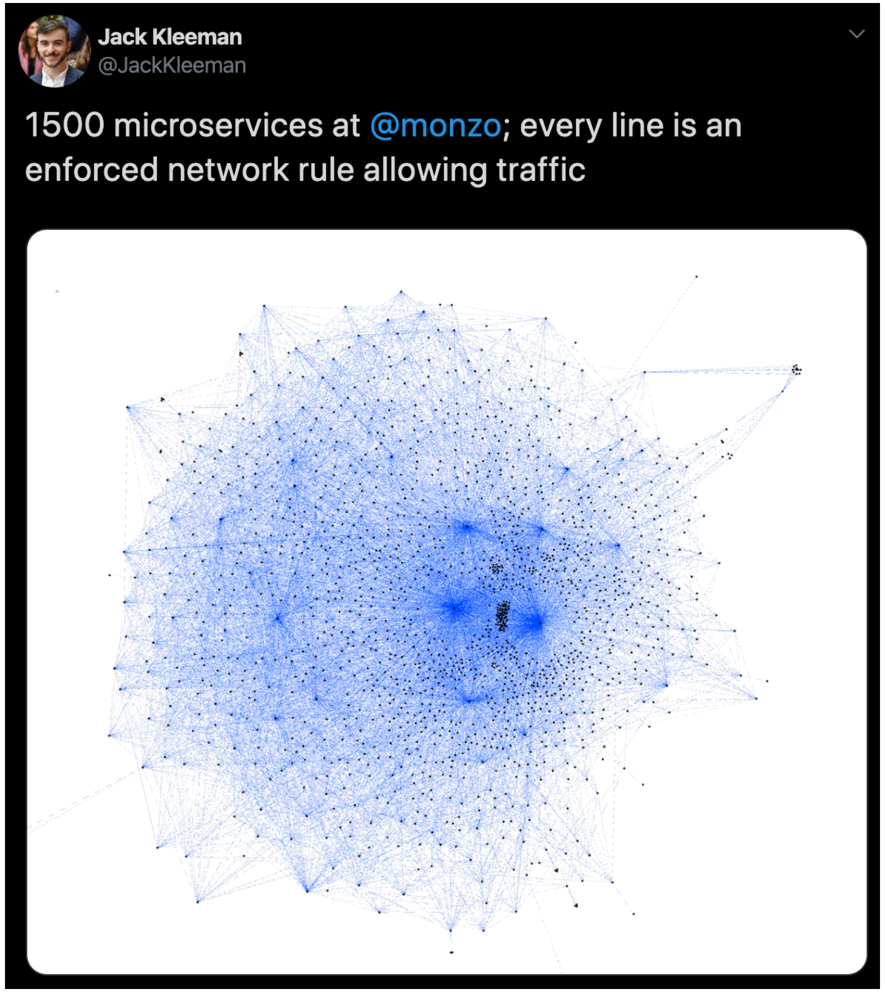
@chzbrgr71

Brian Redmond – Who am I?

- Technical Program Manager @ Microsoft (19 years)
- Global Customer Success Team
- Very recently moved to Denver, Colorado
- Avid runner, biker, and outdoors enthusiast
- Love to travel and be outside

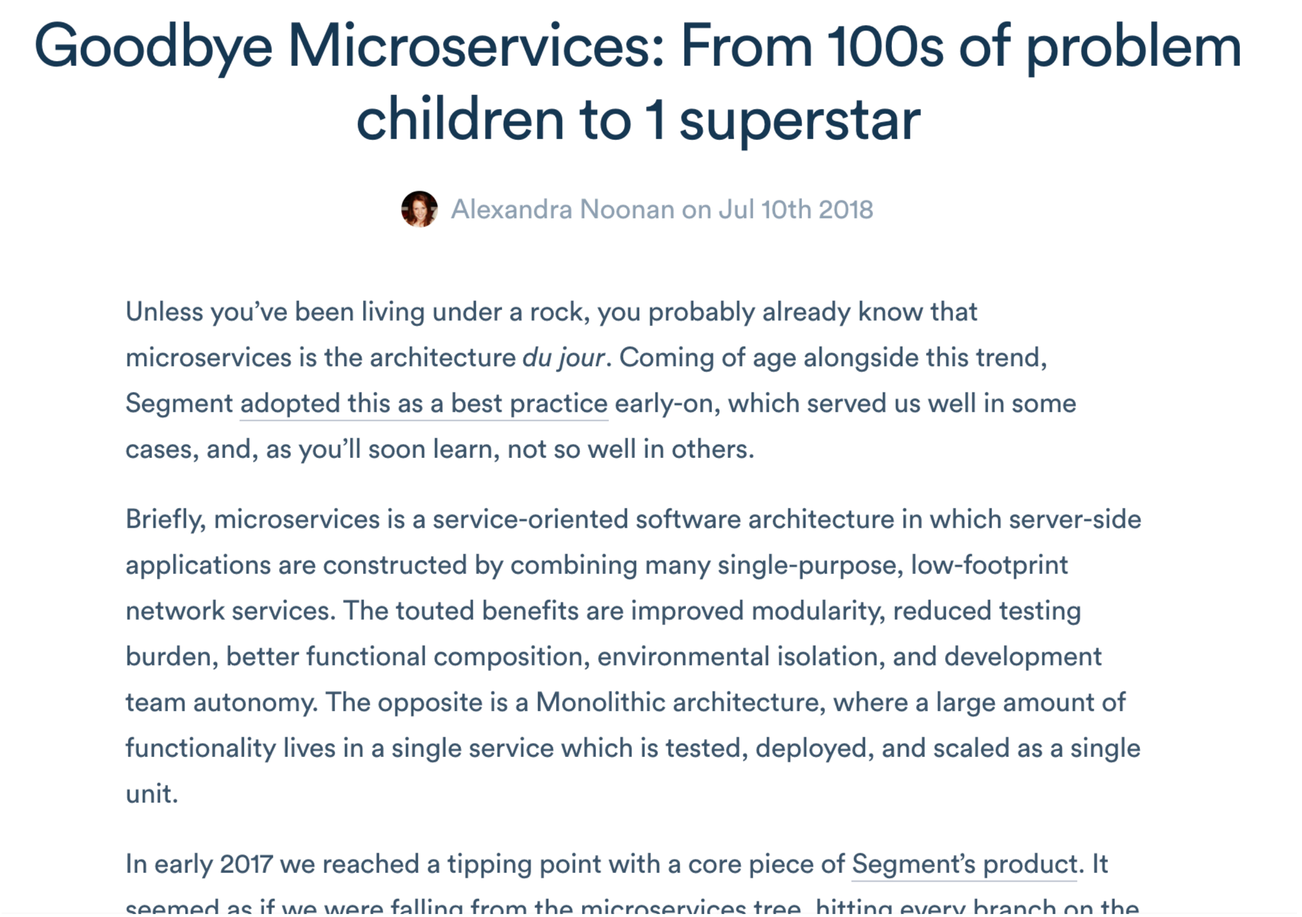
@chzbrgr71

Microservices…



Jack Kleeman
@JackKleeman

1500 microservices at @monzo; every line is an enforced network rule allowing traffic



Goodbye Microservices: From 100s of problem children to 1 superstar

Alexandra Noonan on Jul 10th 2018

Unless you've been living under a rock, you probably already know that microservices is the architecture *du jour*. Coming of age alongside this trend, Segment adopted this as a best practice early-on, which served us well in some cases, and, as you'll soon learn, not so well in others.

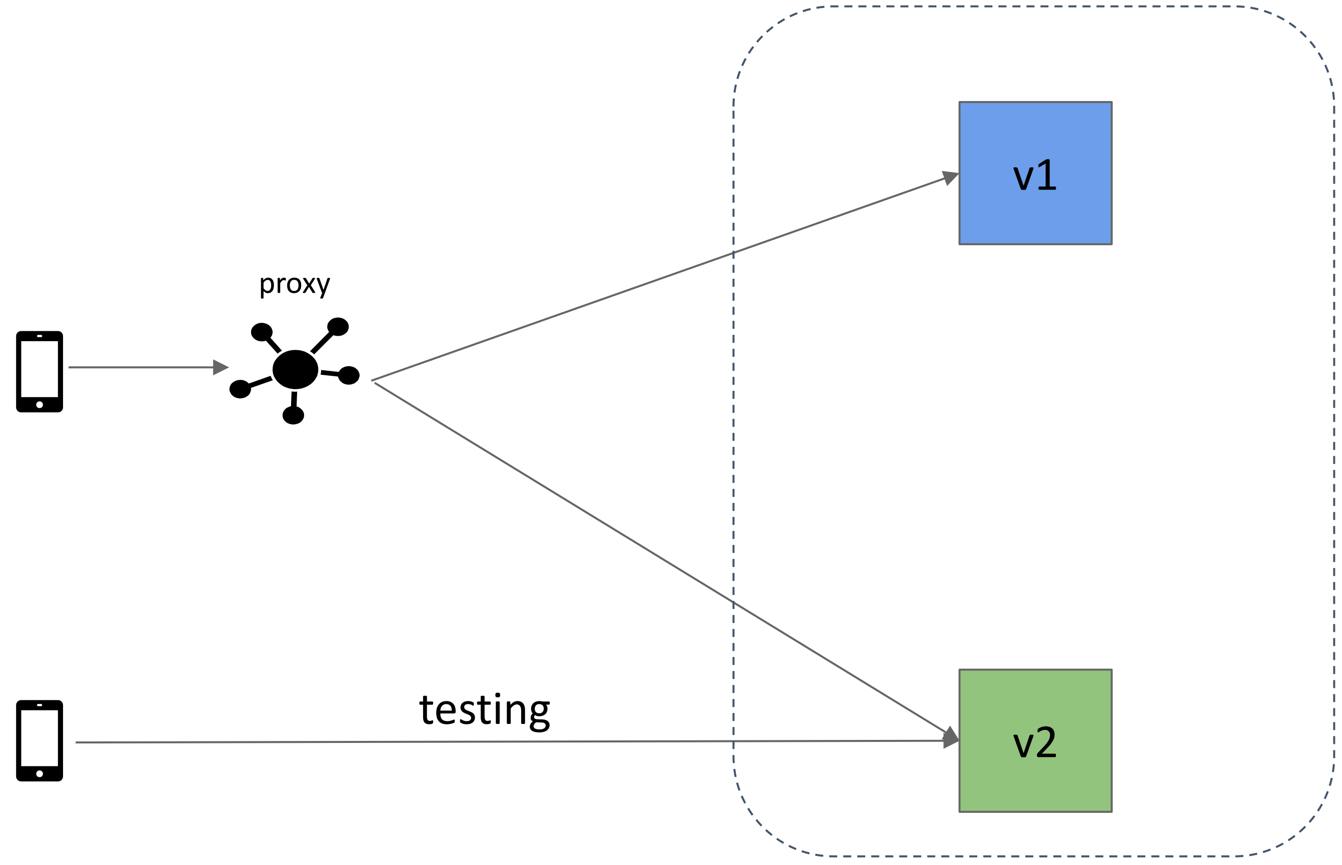
Briefly, microservices is a service-oriented software architecture in which server-side applications are constructed by combining many single-purpose, low-footprint network services. The touted benefits are improved modularity, reduced testing burden, better functional composition, environmental isolation, and development team autonomy. The opposite is a Monolithic architecture, where a large amount of functionality lives in a single service which is tested, deployed, and scaled as a single unit.

In early 2017 we reached a tipping point with a core piece of Segment's product. It seemed as if we were falling from the microservices tree, hitting every branch on the

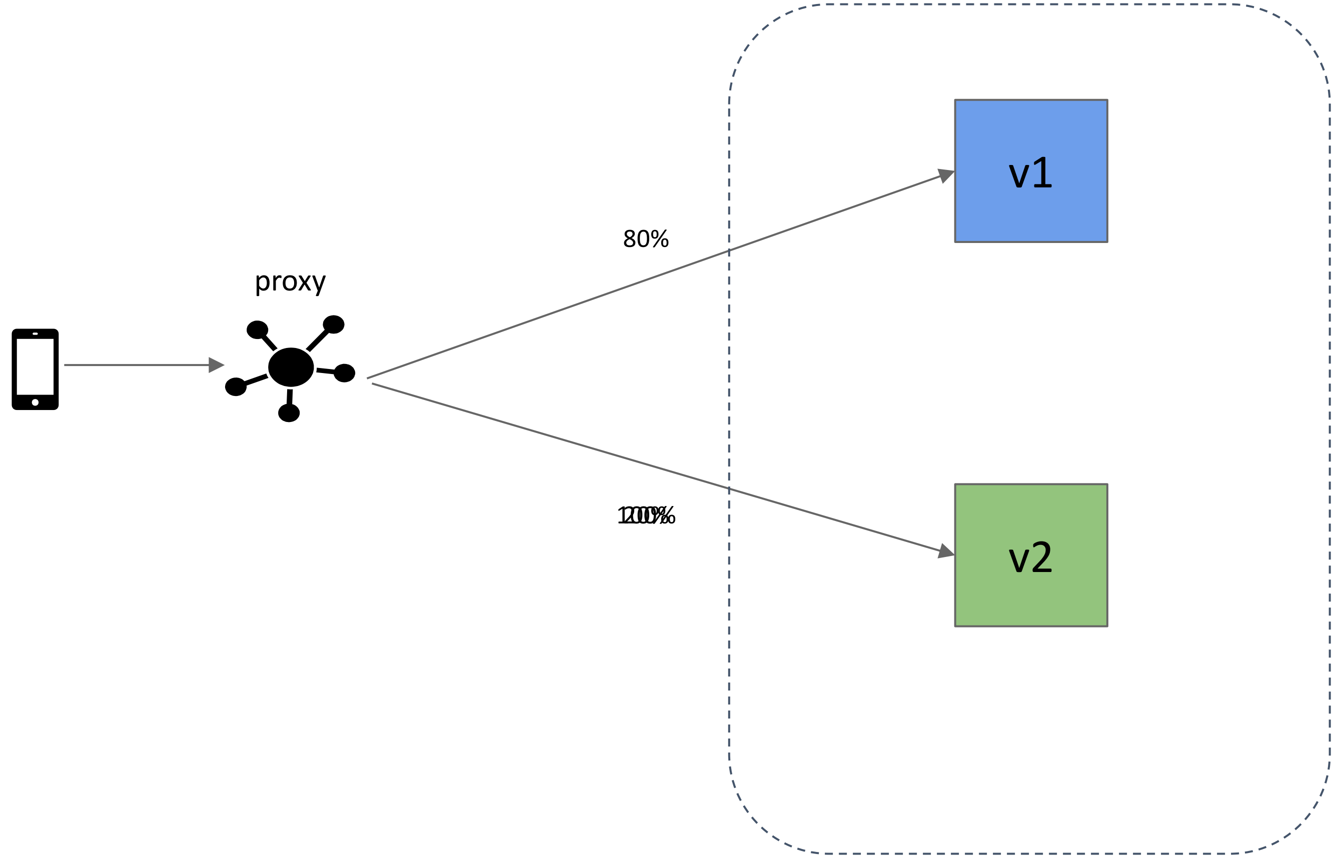
Testing is important

I DON'T ALWAYS TEST MY CODE
BUT WHEN I DO, I DO IT IN PRODUCTION
makeameme.org

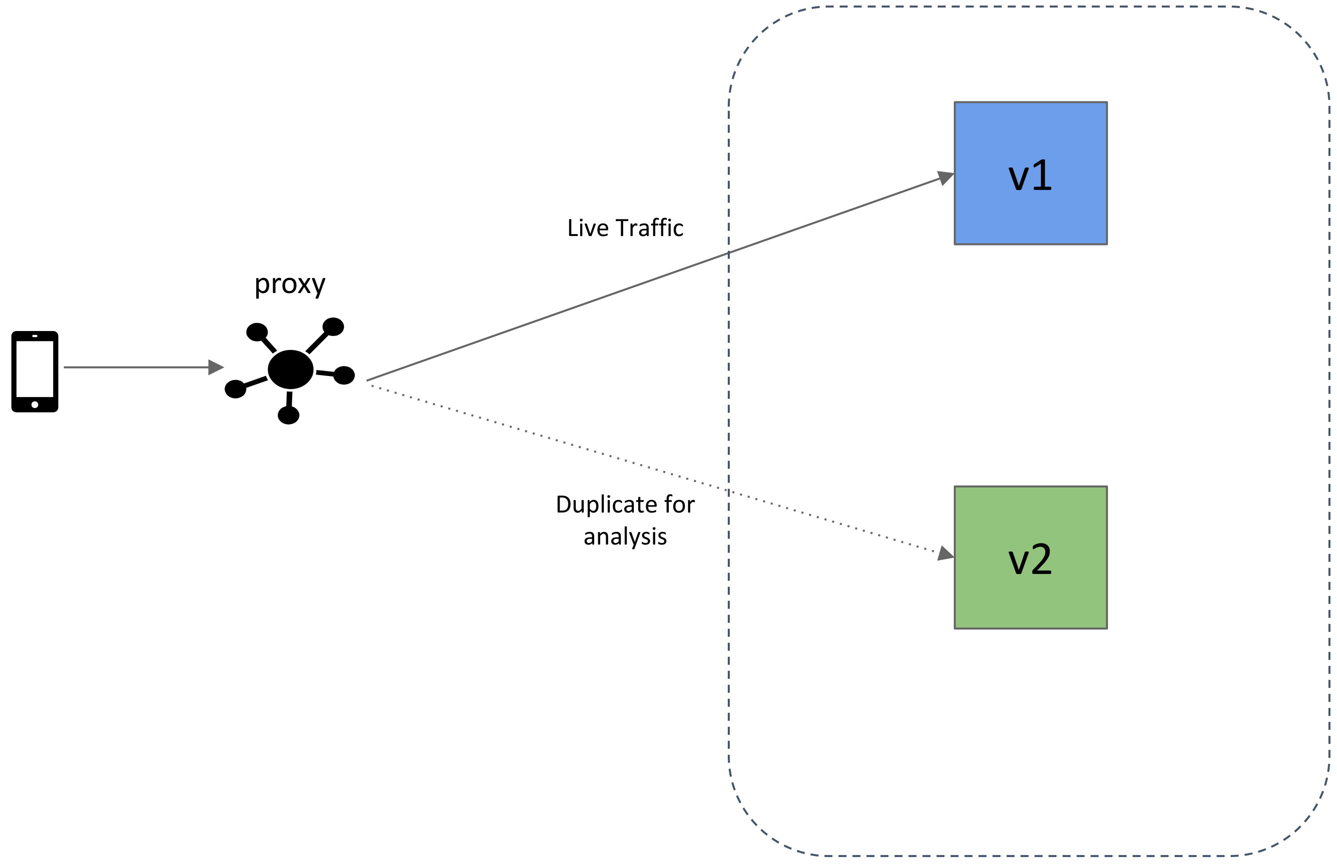
Blue / Green Testing



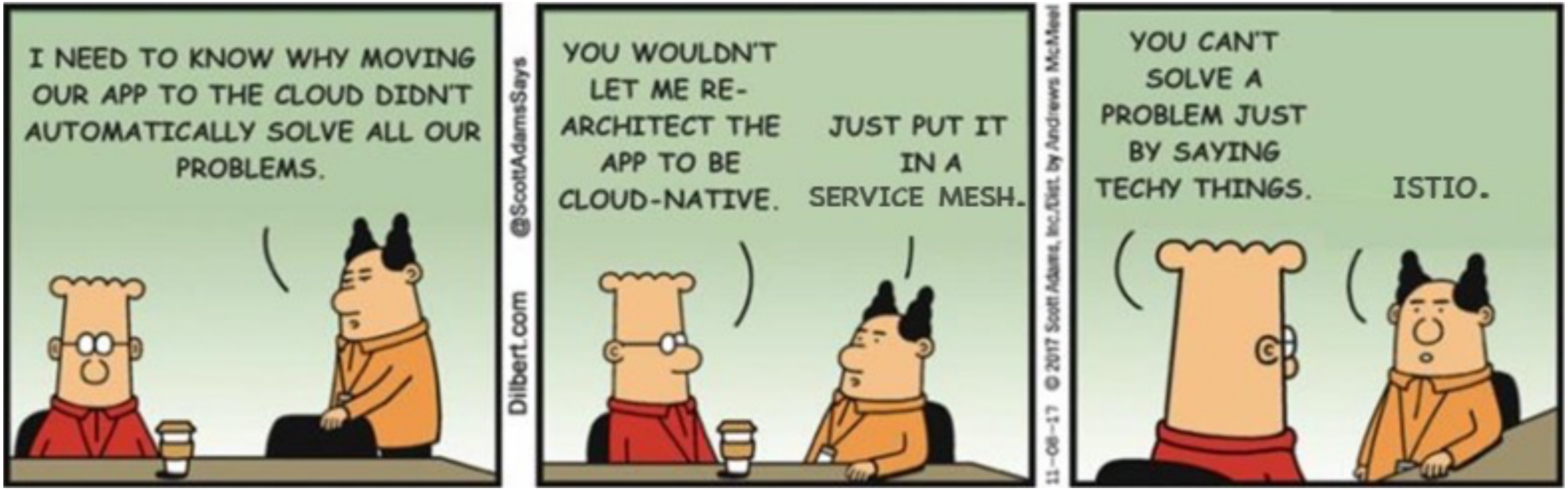
Canary Testing



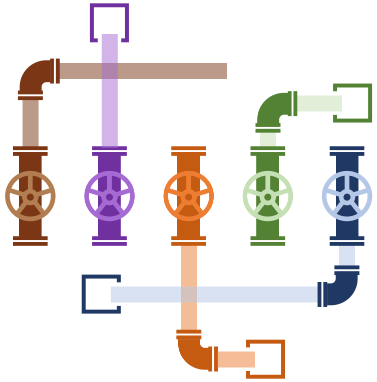
A/B Testing



Service Mesh solves everything


I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.
YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE.
JUST PUT IT IN A SERVICE MESH.
YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS.
ISTIO.
Dilbert.com @ScottAdamsSays
11-08-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel

Smart endpoints, dumb pipes



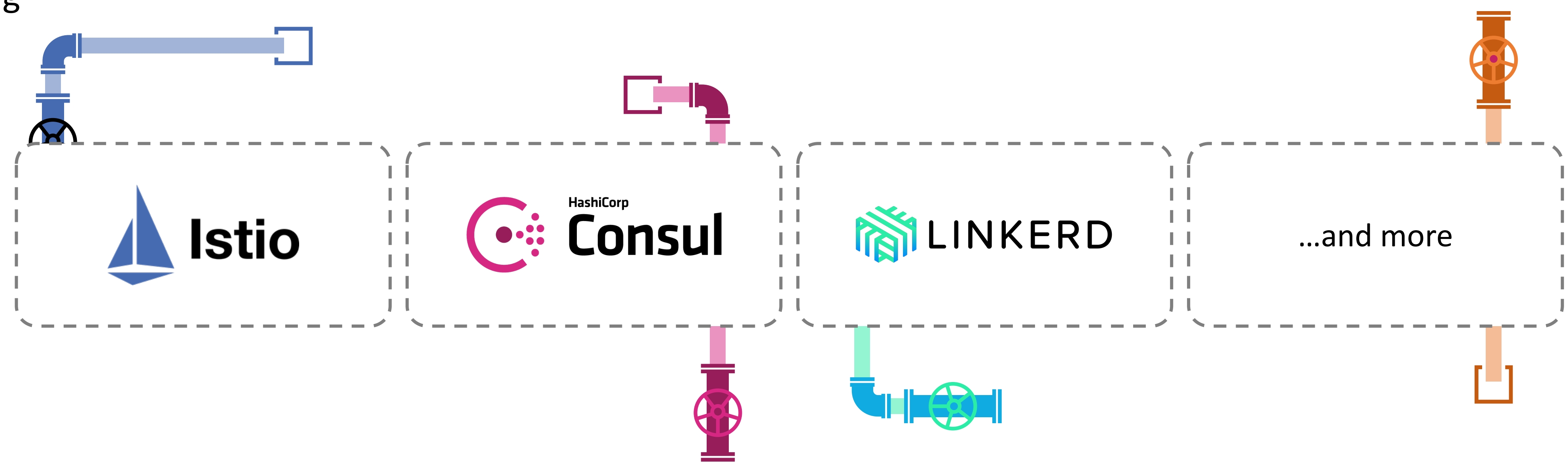
This has worked for the past 25 years

But with so many endpoints today, how do you manage

Service Mesh – Smarter Pipes

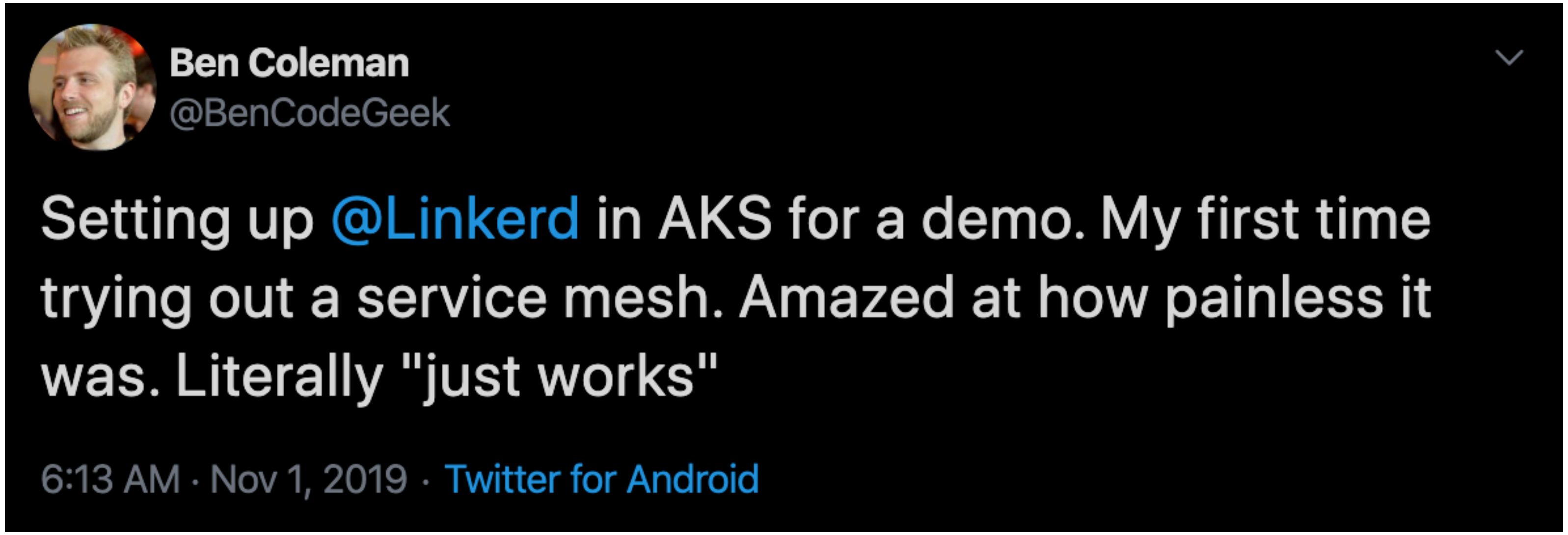
- *“A service mesh is a dedicated infrastructure layer for making service-to-service communication safe, fast, and reliable”* William Morgan (Buoyant)

- What you get:
  - Observability
  - Latency aware load balancing
  - Traffic shaping
  - Security (authorization, encryption, etc.)
  - Retries & circuit breaking
  - Distributed tracing



…and more

Linkerd – Just so easy



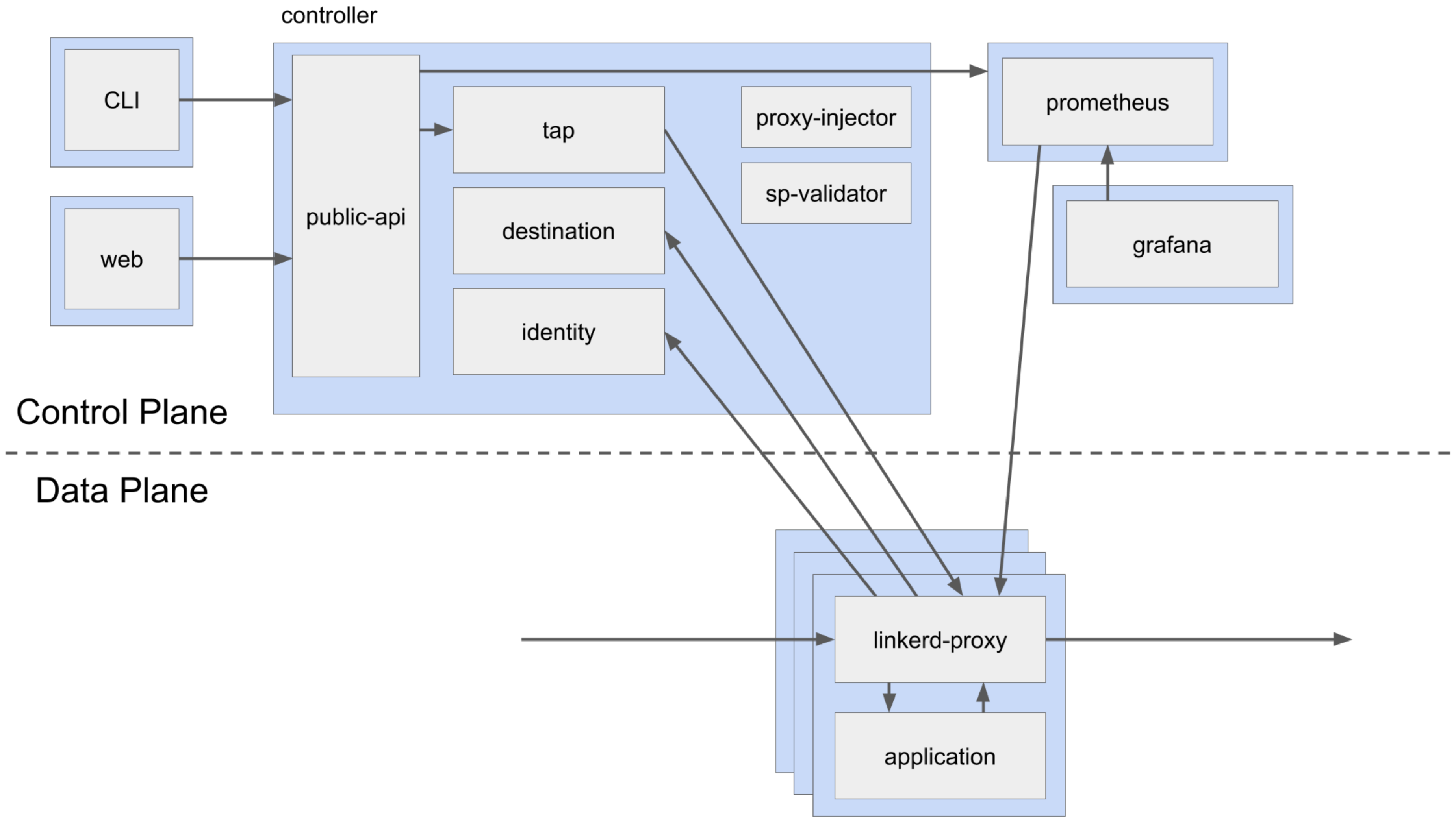
Ben Coleman
@BenCodeGeek

Setting up @Linkerd in AKS for a demo. My first time trying out a service mesh. Amazed at how painless it was. Literally "just works"

6:13 AM · Nov 1, 2019 · Twitter for Android

“I thought I'd done something wrong when I set it up the first time it was so easy”
@justindavies

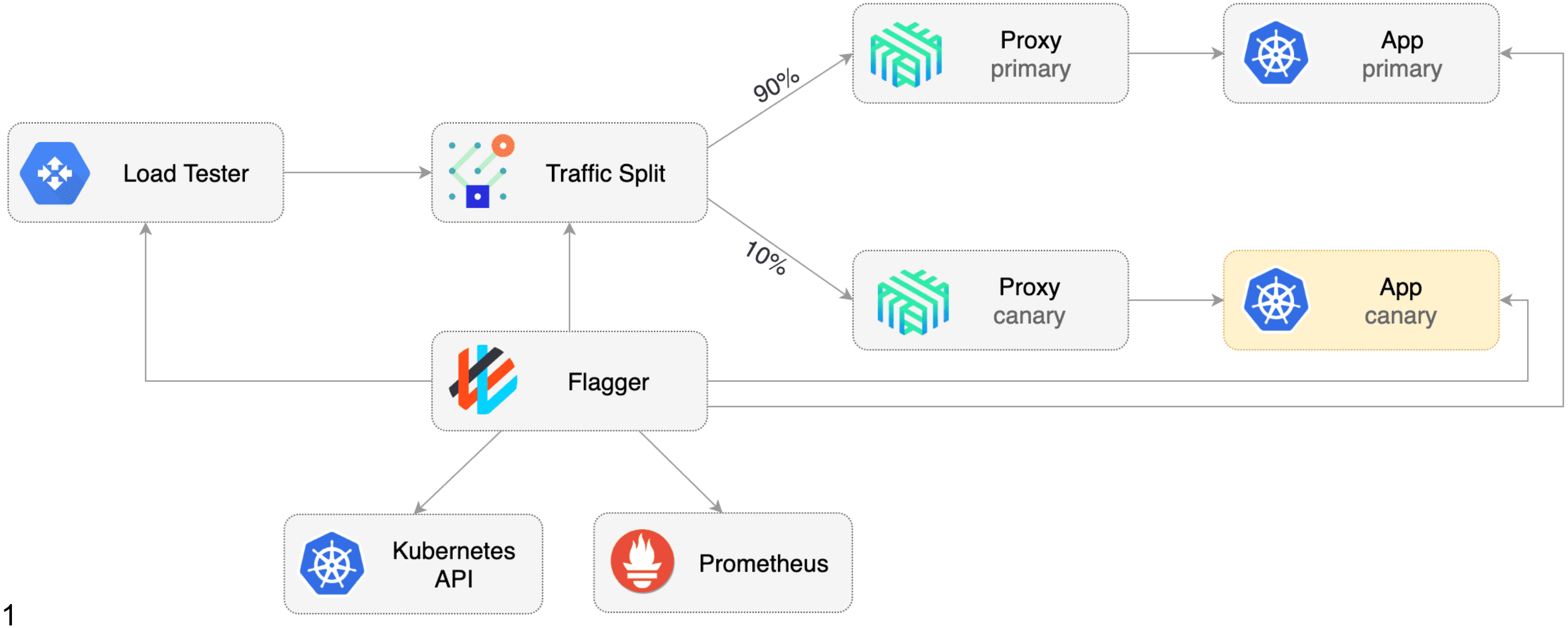
Linkerd Architecture



Flagger

- Flagger is a Kubernetes operator automating canary deploys using service mesh and metrics
  - Supports Istio, Linkerd, AppMesh, NGINX, and Gloo
  - Measures key performance indicators while gradually shifting traffic
  - Integrates with Service Mesh Interface (SMI)

weaveworks



Service Mesh Interface (SMI) for Kubernetes

A Kubernetes interface that provides traffic routing, traffic telemetry, and traffic policy

Standardized

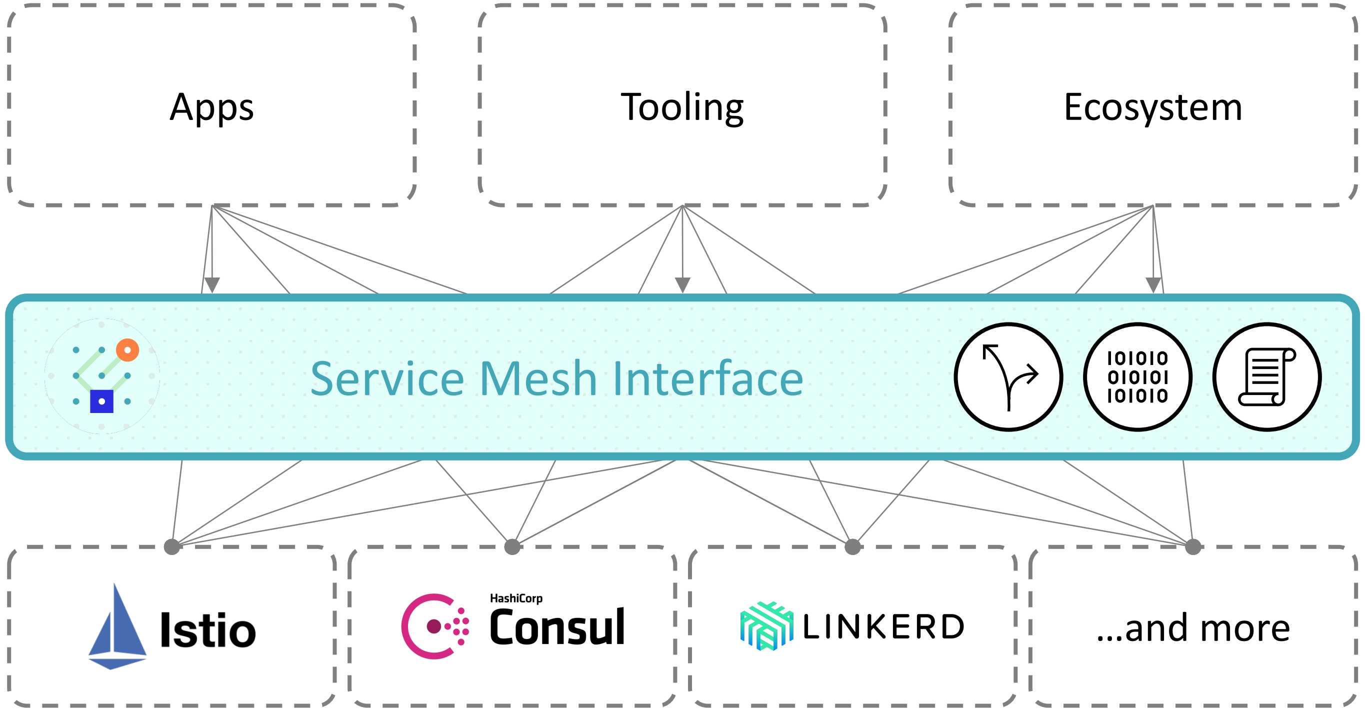
Standard interface for service mesh on Kubernetes

Simplified

Basic feature set to address most common scenarios

Extensible

Support for new features as they become widely available

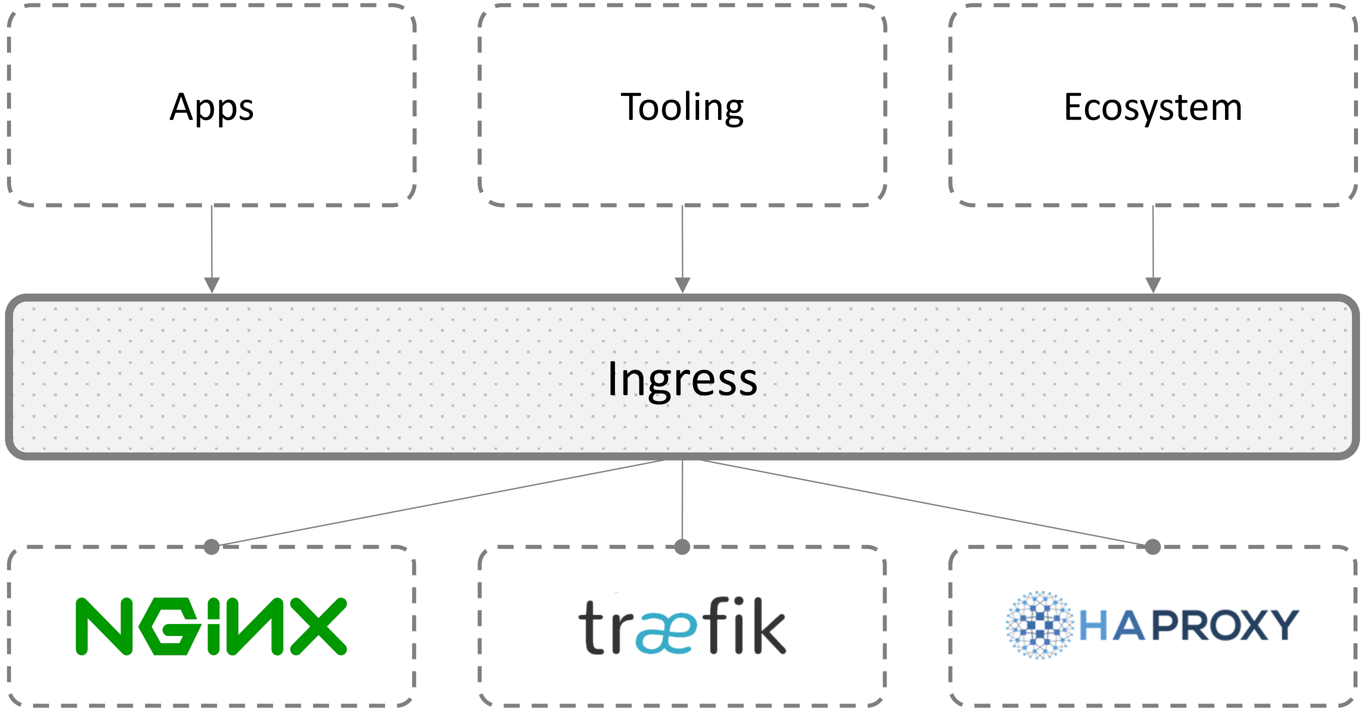


SMI Initial Focus

- **Traffic Split**
  - Shift traffic between different services
- **Traffic Metrics**
  - Capture key metrics like error rate and latency between services
- **Traffic Access**
  - Apply policies like identity and transport encryption across services

This isn't a new concept

If the SMI concept sounds familiar, that's because it is



Service Mesh Interface (SMI) for Kubernetes

In partnership with:

Microsoft

LINKERD

HashiCorp

solo.io

Red Hat

RANCHER

docker

weaveworks

ASPEN MESH

Pivotal

vmware

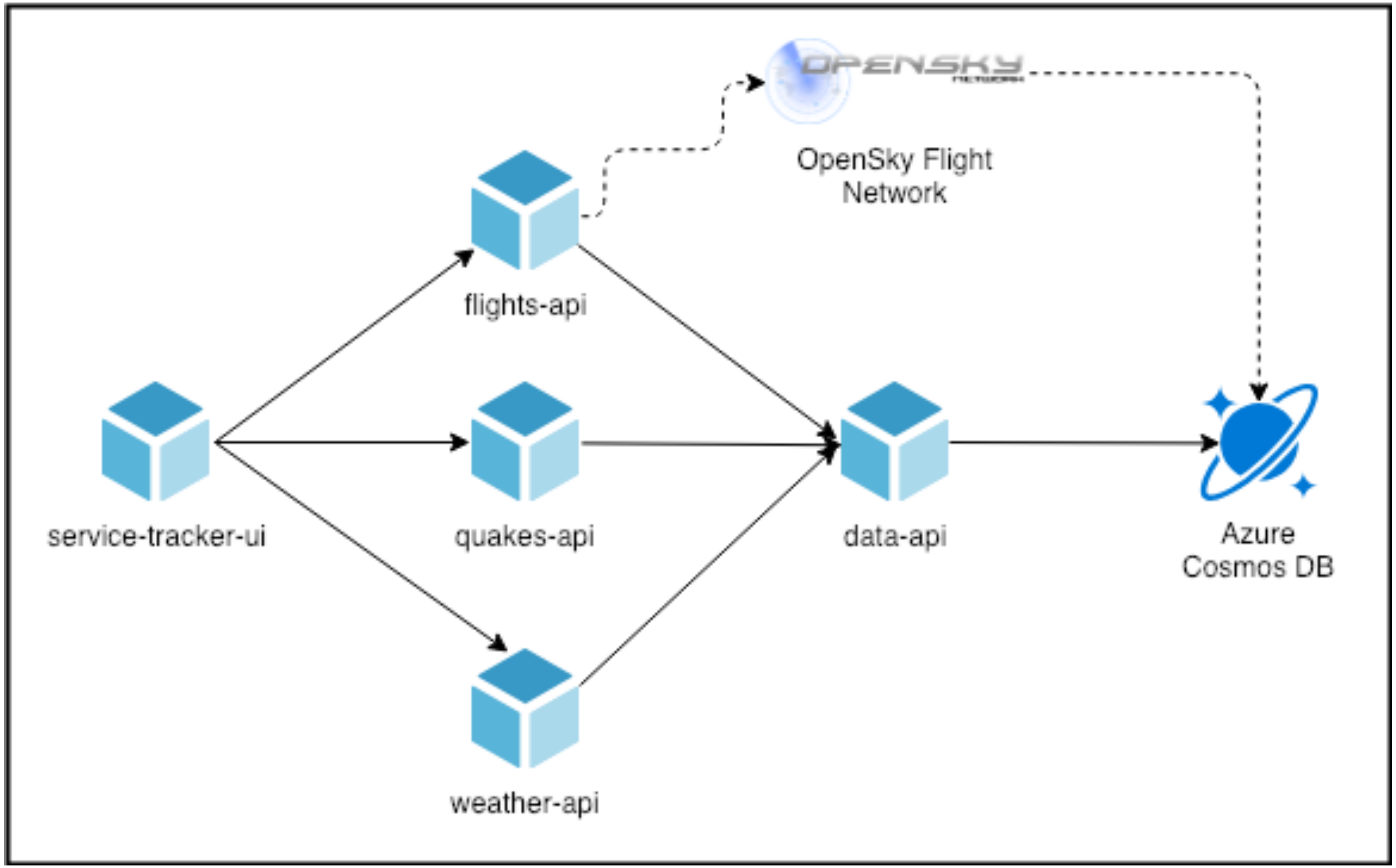
kinvolk

CANONICAL

kubecost

Demo

Demo architecture



Questions?

Find me at @chzbrgr71

Source for demos: https://github.com/chzbrgr71/smi-demos