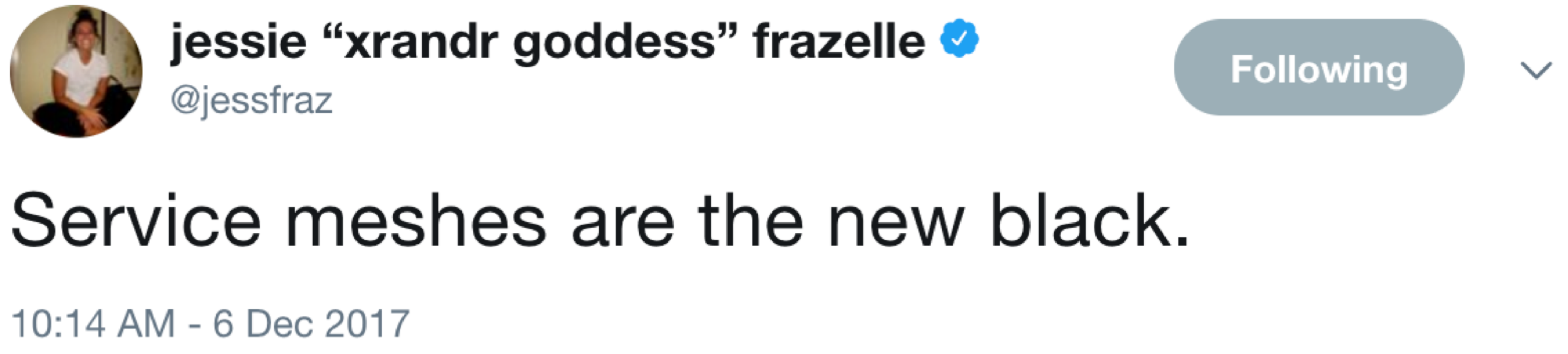
KubeCon

North America 2017

Setting sail with Istio

Lachlan Evenson - Principal Program Manager – AKS/ACS, Microsoft
@LachlanEvenson

But first. Some context

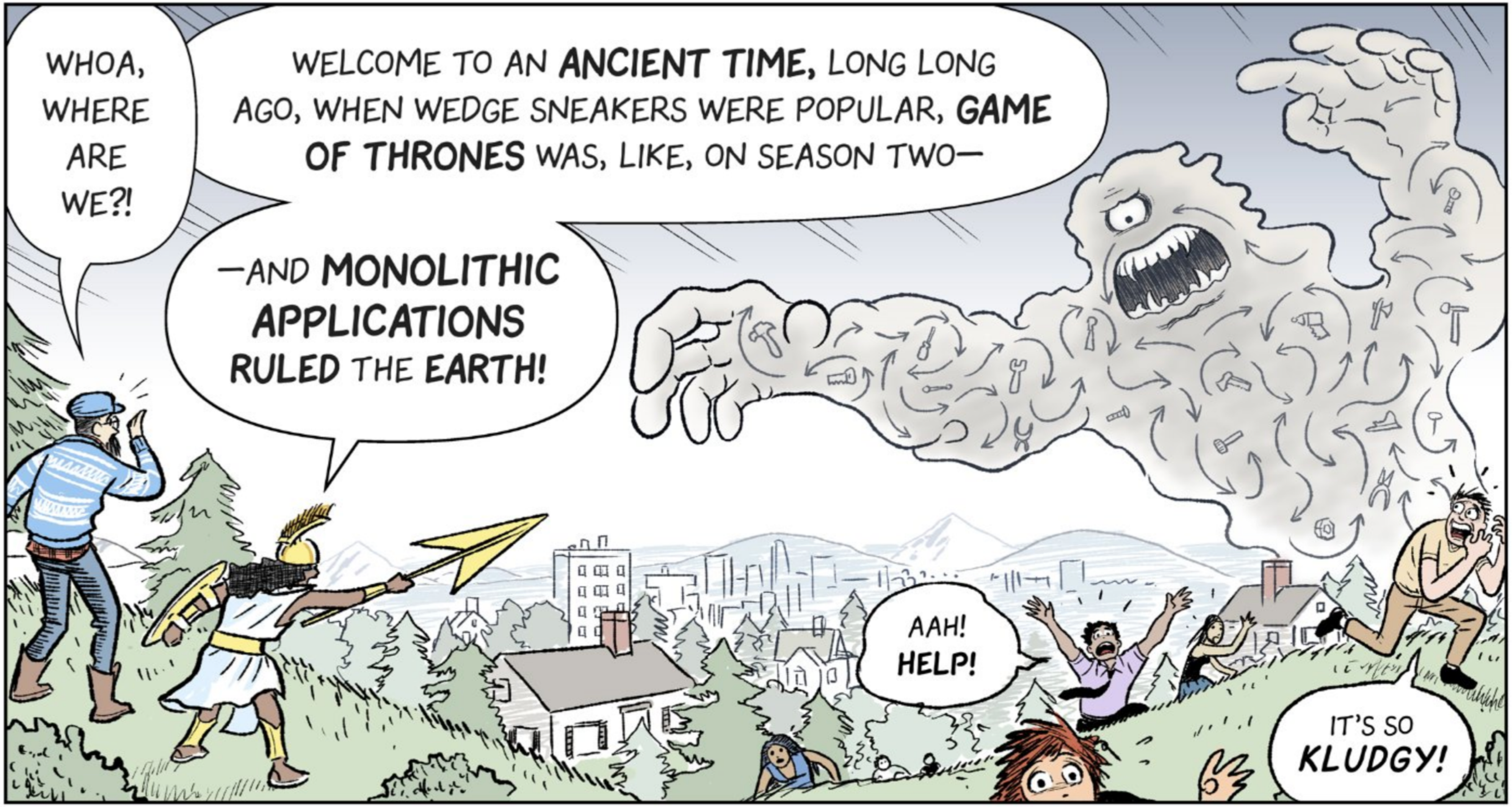


jessie “xrandr goddess” frazelle
@jessfraz

Service meshes are the new black.

10:14 AM - 6 Dec 2017

Source: Twitter


WHOA, WHERE ARE WE?!
WELCOME TO AN ANCIENT TIME, LONG LONG AGO, WHEN WEDGE SNEAKERS WERE POPULAR, GAME OF THRONES WAS, LIKE, ON SEASON TWO—
—AND MONOLITHIC APPLICATIONS RULED THE EARTH!
AAH! HELP!
IT'S SO KLUDGY!

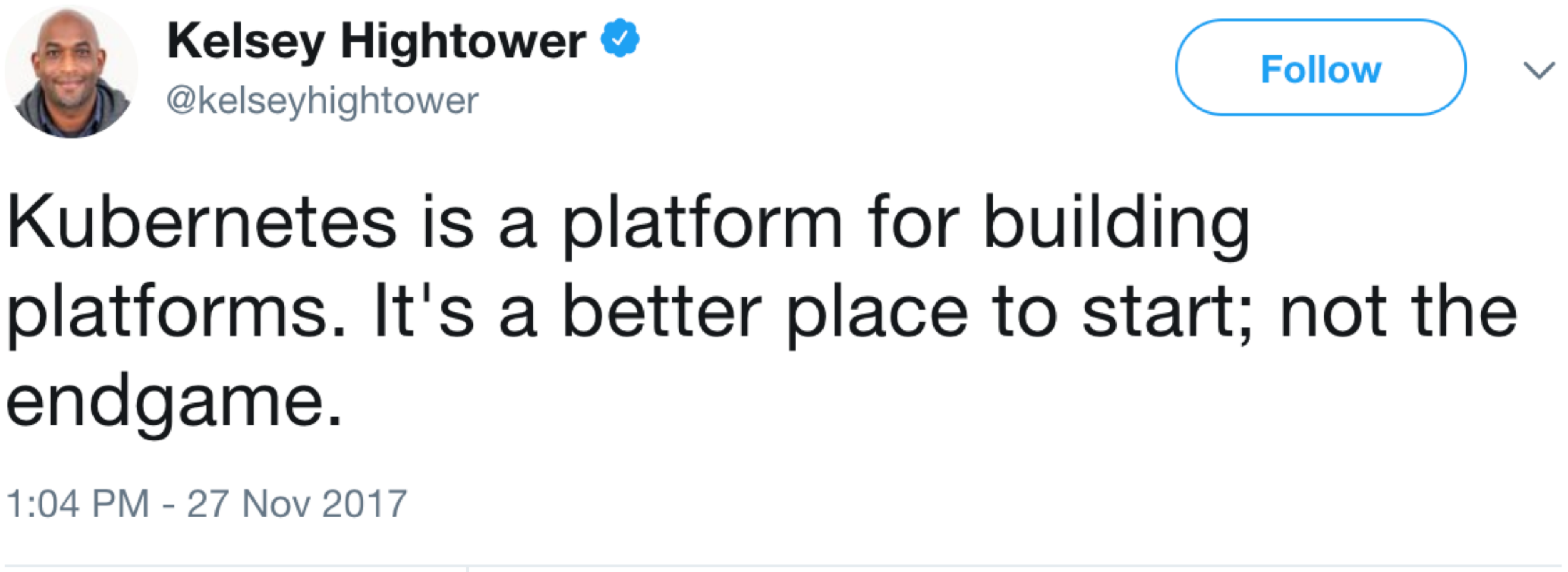
Source: https://cloud.google.com/kubernetes-engine/kubernetes-comic/

Who am I?

- Run microservices in production in Kubernetes since 2015
- Built early incarnations of “Istio” like platforms
- Helping customers be successful on Kubernetes
- Built the upstream Istio Helm chart

Doing Microservices is Hard!

Kubernetes is not the endgame



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

1:04 PM - 27 Nov 2017

Source: Twitter

Why are microservices hard?

- Generally not operating in green-field environments
- Microservices command an overhaul of
  - People
  - Tooling
  - Processes
- These all take time to change
- Microservices expose all the cracks in architectures
- (Once) well understood behaviors change

Why are microservices hard?

- The first few services are relatively easy
- Contract points, SLAs and responsibilities
- Tooling is nascent and bespoke
- We’re not yet equipped for the change over time (or maybe you are)

What do we need?

- Observability
- Monitoring
- Metrics
- Tracing
- Traffic Management
- Policy
- Security
- Service Mesh

But what are the expectations

- Should developers be implementing all that list on their own?
- Should the platform provide an abstraction?

Enter Istio

- Istio is a microservice platform that provides all of the aforementioned features
- Istio plugins into Kubernetes natively via platform adapters
- Istio isn’t a silver bullet. It’s the next level platform.

Istio Platform features

- Traffic Management
- Policy Enforcement
- Metrics, Logs and Traces
- Security

Istio for Operators

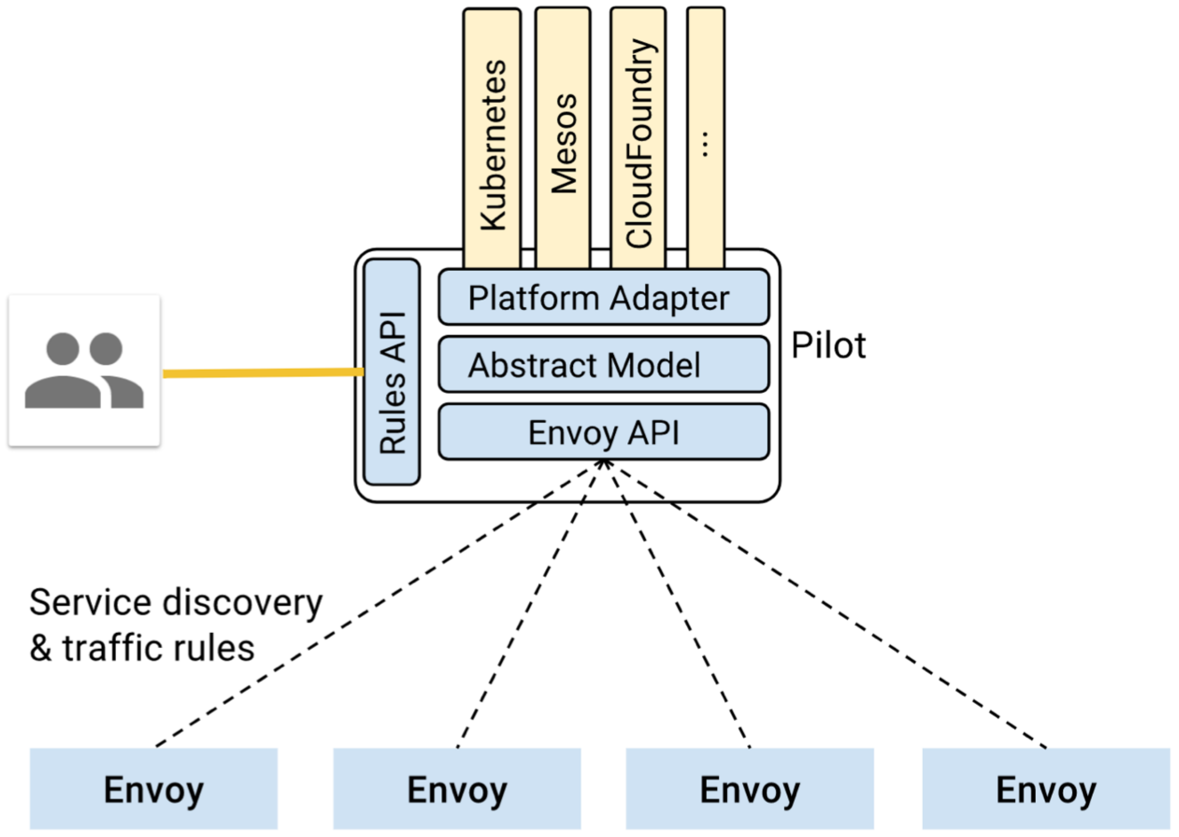
- Istio comprises of several microservices interacting with Kubernetes.

Istio for Operators (continued)

- Pilot
  - Control-plane for the distributed Envoy instances
  - System of record for service mesh
  - Abstracted from underlying platform (Kubernetes, Mesos, CF)
  - Adapters manage this representation on the underlying platform
  - Kubernetes Adapter manages controllers and resources
    - Ingresses, CRDs, etc…. (system state)
  - Exposes API for Service Discovery, LoadBalancing and Routing Tables
  - These directly translate to Envoy config

Istio for Operators (continued)

- Pilot



Pilot Architecture

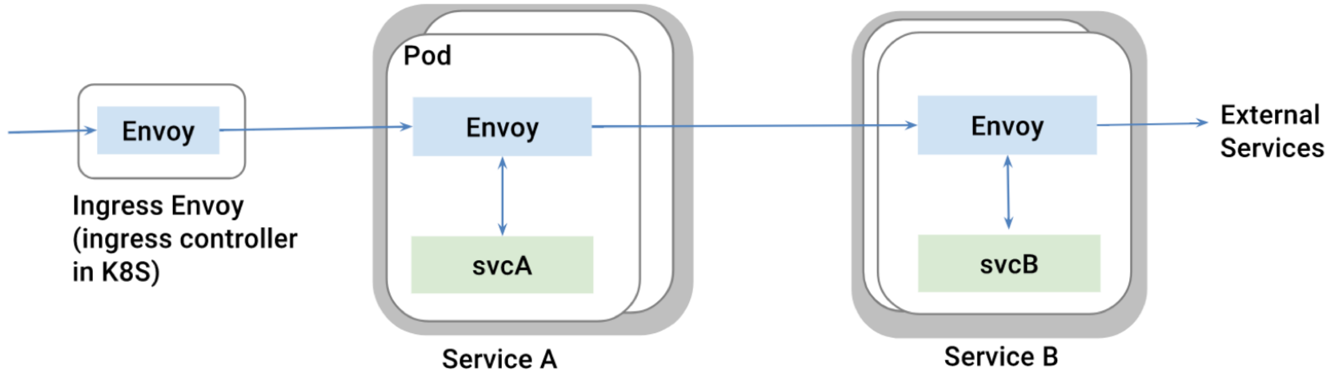
Source: https://istio.io/docs/concepts/traffic-management/pilot.html

Istio for Operators (continued)

- Envoy
  - The data-plane component that lives as a container in each pod deployed by Istioctl
  - All ingress/egress traffic from/to this pod is routed via the Envoy container
  - Serves as an in/off ramp to the service mesh
  - Envoy config is distributed by Pilot
  - Envoy container injected via istioctl kube-inject OR Kubernetes initializer

Istio for Operators (continued)

- Ingress/Egress
  - All traffic entering/leaving the service mesh is routed via an Ingress/Egress router
  - Envoy proxy
  - Enables static egress routing



Request Flow

Source: https://istio.io/docs/concepts/traffic-management/request-routing.html

Istio for Operators (continued)

- Mutual TLS
  - May be enabled
  - Enables service to service encryption without user intervention
  - Istio ships with a CA
  - This CA watches for Kubernetes service accounts and creates corresponding cert keypair secrets in Kubernetes
  - When a pod is created these secrets are mounted
  - Pilot generates the appropriate Envoy config and ships it
  - e2e mTLS established for each connection

Istio for Operators (continued)

- Mixer
  - Policy engine that comprises all the tools needed to run microservices
    - Access control
    - Telemetry
    - Quota
    - Billing
    - Tracing
  - Generic underlying platform independent abstraction
  - Pluggable adapters
  - Information is passed from Istio to Mixer via ”Attributes”

Istio for Operators (continued)

- Mixer (continued)
  - Attribute processing machine that controls the runtime behavior of services running in the mesh
  - Attributes are generated by Envoy
  - Mixer then generates calls to infrastructure backends via Adapters
    - Eg. Rate limits
  - Handlers
  - Instances
  - Rules
  - These are all expressed at CRDs

Istio for Operators (continued)

- Demo!

Istio for Developers

- Istio allows the developer to effectively deploy and utilize microservices without deep knowledge of the underlying infrastructure.

Istio for Developers

- Demo!
- Deploying your application
- DotViz
- Zipkin
- Grafana/Prometheus

The Istio Community

- Istio has a vibrant community
- https://istio.io/

Resources

- Documentation
  - https://istio.io/docs/
  - https://github.com/lukebond/walk-run-fly-istio-kubernetes-talk
- Helm Chart
  - https://github.com/kubernetes/charts/tree/master/incubator/istio
  - https://kubeapps.com/charts/incubator/istio
- Twitter
  - @LachlanEvenson
  - Setting Sail with Istio – YouTube channel

Thanks!