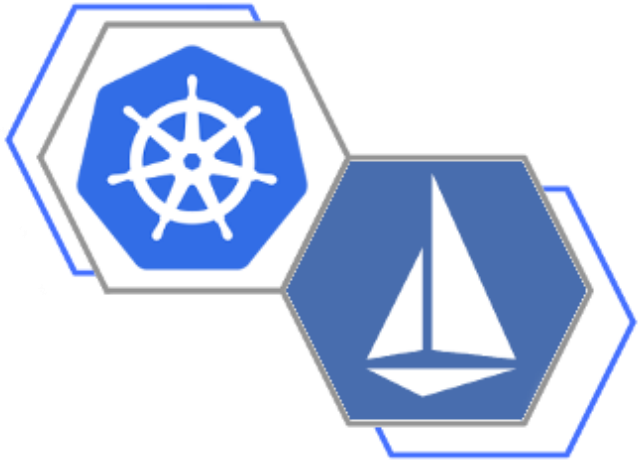
KUMULUS
TECHNOLOGIES



Service Mesh on Kubernetes With Istio

Who are we?

Robert Starmer: @rstarmer
CTO/Founder of Kumulus Technologies
OpenStack Ops contributor since 2012
Supporting Cloud enablement for Enterprise
OpenStack, Kubernetes, BareMetal to App CD

Kumulus Technologies: @kumulustech
Systems consultants supporting cloud migration & integration
Kumulus Tech Newsletter: https://kumul.us/newsletter/
Five Minutes of Cloud: https://youtube.com/fiveminutesofcloud

© 2017 Kumulus Technologies

Access Course Resources

Use the following account to create your course account:

http://bit.ly/Istio_k8s

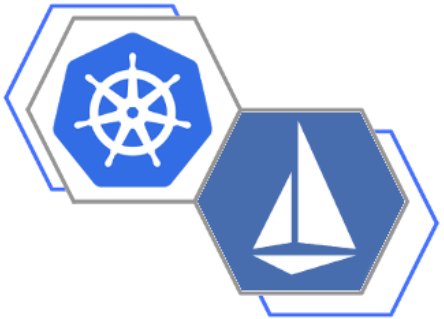
robert@kumul.us

@rstarmer

© 2017 Kumulus Technologies

Agenda

Microservices, Kubernetes and Istio



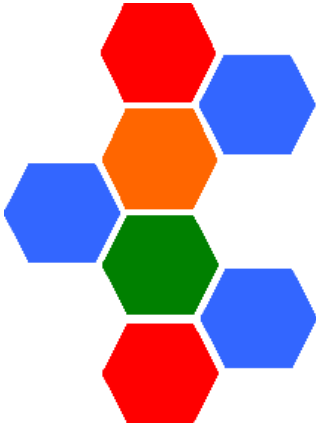
- Microservices
- Kubernetes
- Istio
    - Service Mesh
    - Mutual TLS (security)
    - Routing
    - Tracing/Metrics
    - Fault Injection
- Lab - Get Kubernetes, Istio, Launch an App
- Lab - Routing

© 2017 Kumulus Technologies

Microservices (Day 2 Operations)

Microservices are small nuggets of function, and that sounds like it could be simple, but as complexity grows, successful operations require:

- Visibility (Observability)
  - Monitoring
  - Metrics
  - Tracing
- Traffic management
- Policy Enforcement
- Security
- Resilience and efficiency

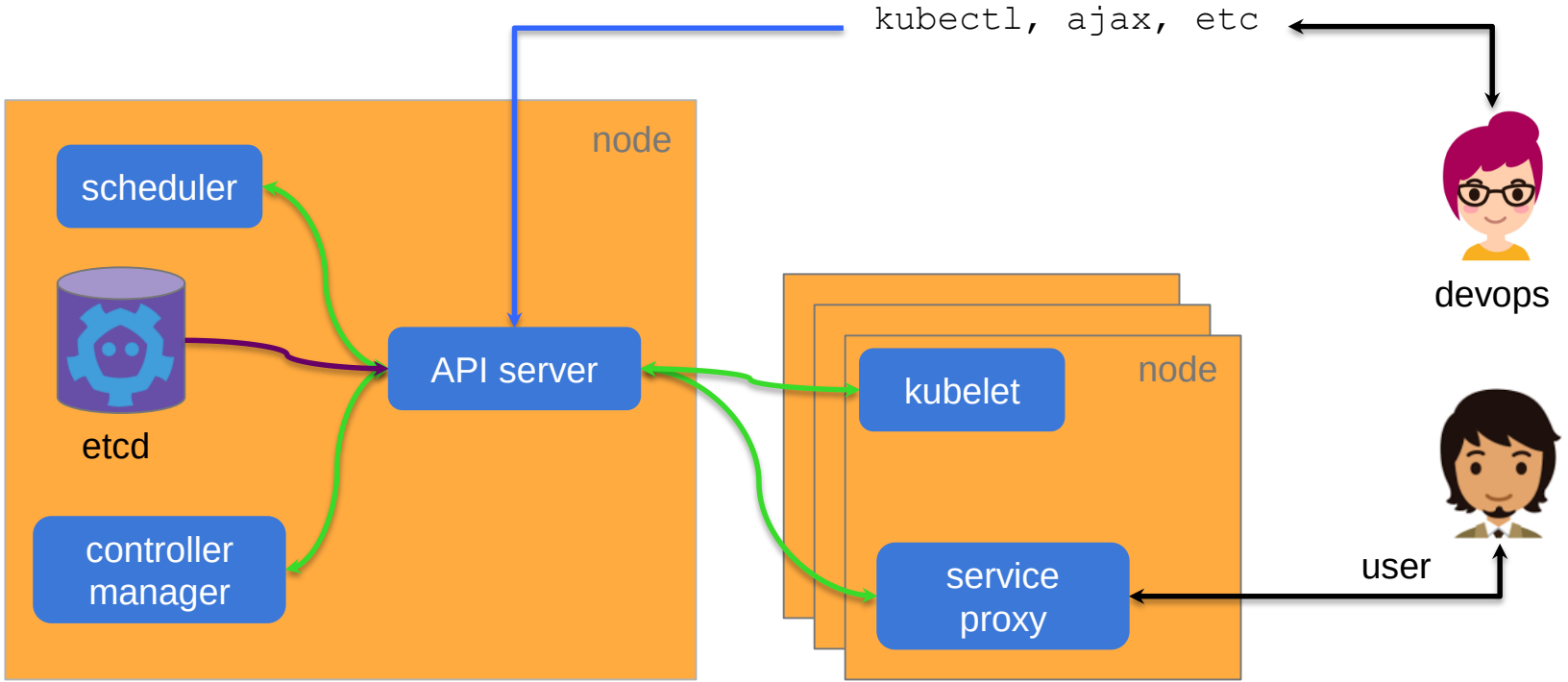


A service mesh (an application network for services) can provide the above.

© 2017 Kumulus Technologies

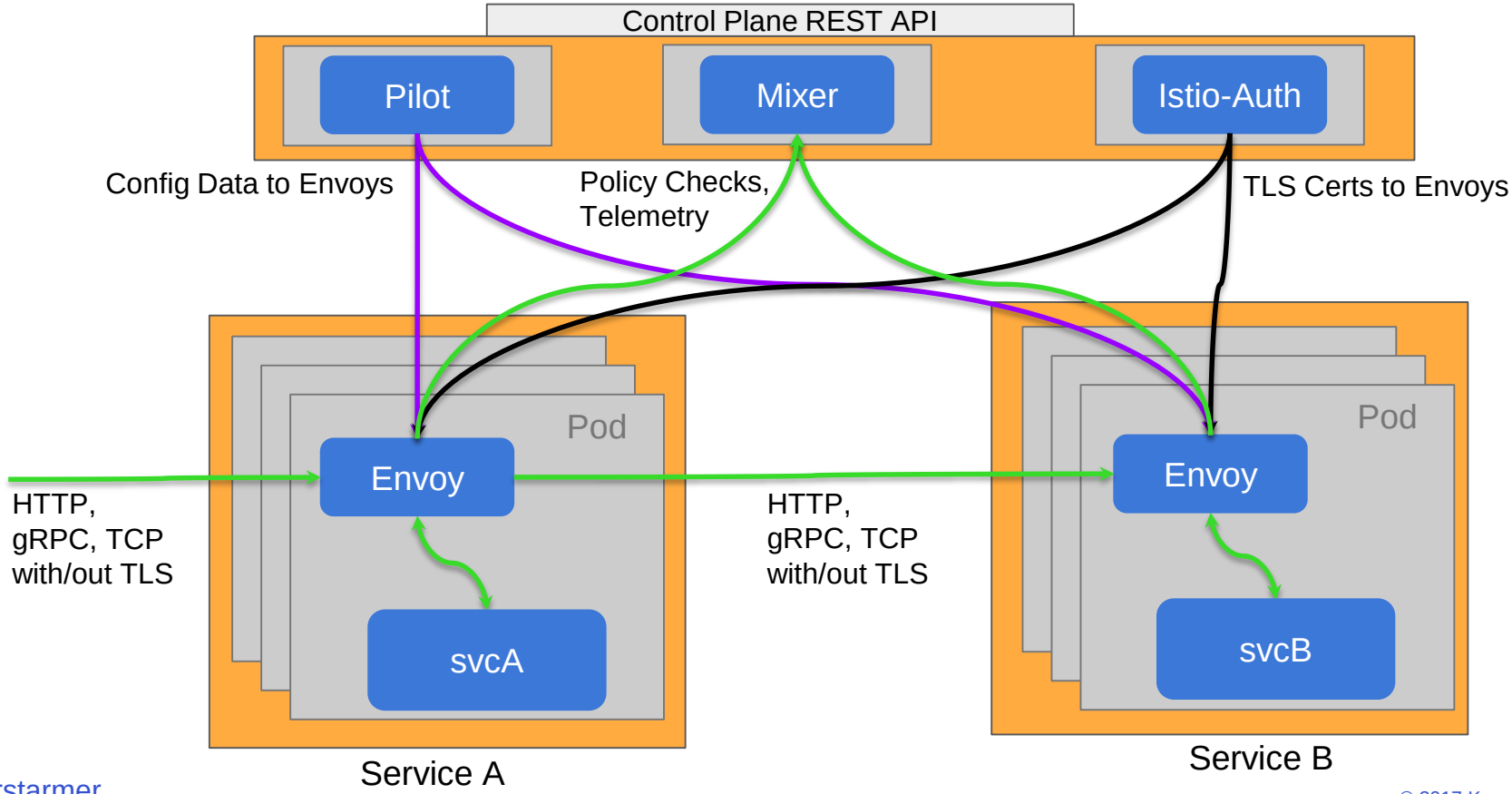
Kubernetes

Kubernetes provides an infrastructure management service



© 2017 Kumulus Technologies

Istio Architecture



© 2017 Kumulus Technologies

Istio

Istio is a service mesh (microservices platform) providing:

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

Kubernetes “native” via platform adapter plugins - also plugs into Mesos, Cloud Foundry, …

© 2017 Kumulus Technologies

Istio - Pilot

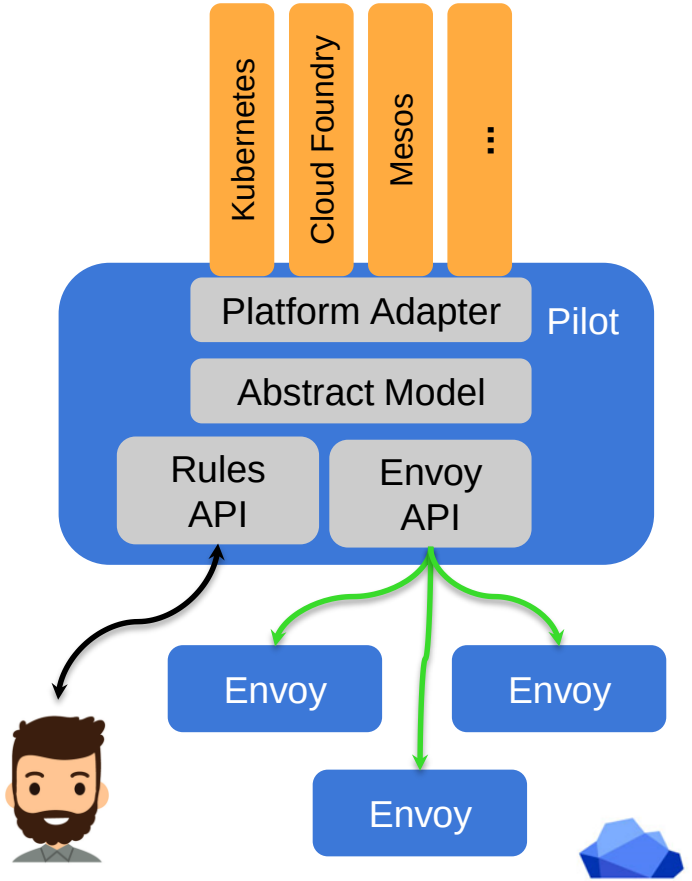
Control plane for distributed Envoy instances

Configures Istio deployment and pushes out configuration to other system components

System of Record for Service Mesh

Routing and resiliency rules

Exposes API for service discovery, load balancing, routing tables



© 2017 Kumulus Technologies

Envoy Proxy

Out of process load balancer:
- High performance server/small memory footprint

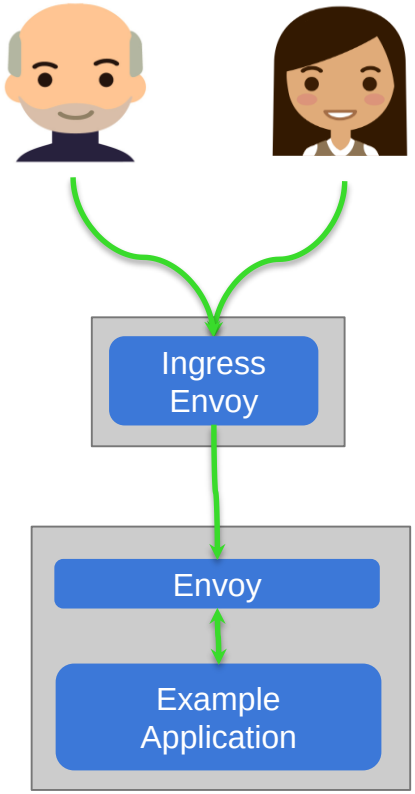
HTTP/2 and GRCP support:
- Transparent HTTP/1.1 to HTTP/2 proxy.

APIs for Config Management:
- Configuration management via API alone

Advanced Load Balancing:
- Retries, Circuit Breaking, Health Checks, Rate Limits

Observability
- L7 visibility, distributed flow tracing

In Istio:
- Envoy container is injected with istioctl kube-inject or kubernetes initializer
- Controls pod ingress/egress routing
- Config is via API from Pilot



© 2017 Kumulus Technologies

Istio - Mixer

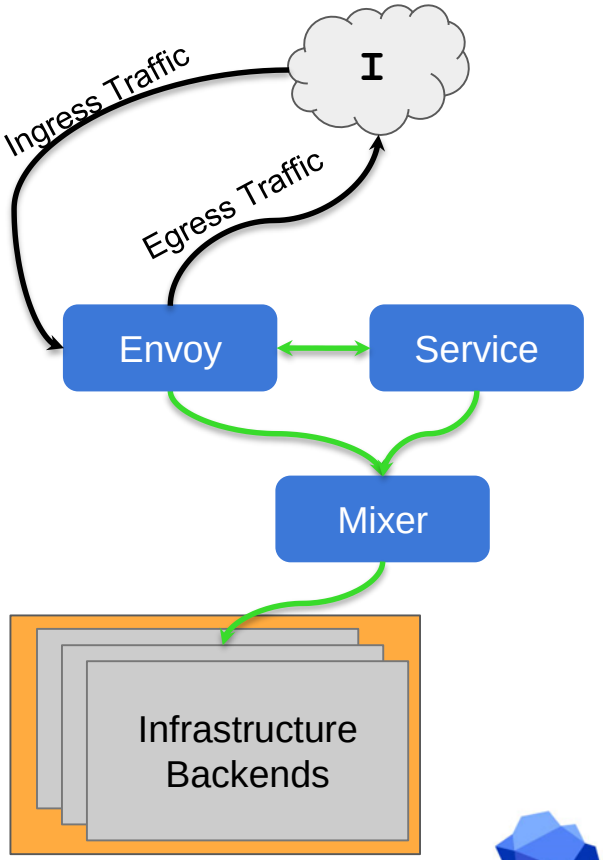
Attribute processor that controls the runtime behavior of mesh-attached services

Envoy generates *attributes*

Mixer then generates calls to backend infrastructure through adapters

Handlers provide integration for 3rd party tools (Prometheus, Grafana, custom tools, …)

All of these “Istio” pieces are expressed as Kubernetes custom resources (CRDs)



© 2017 Kumulus Technologies

Mutual TLS

Available by default, but not required

When enabled, provides automatic service-to-service encryption

Istio has a built in CA that watches for k8s service accounts and creates certificate keypair secrets in k8s

Secrets are automatically mounted when pod is created

Pilot generates appropriate Envoy config and deploys it

End-to-end mTLS session generated for each connection.

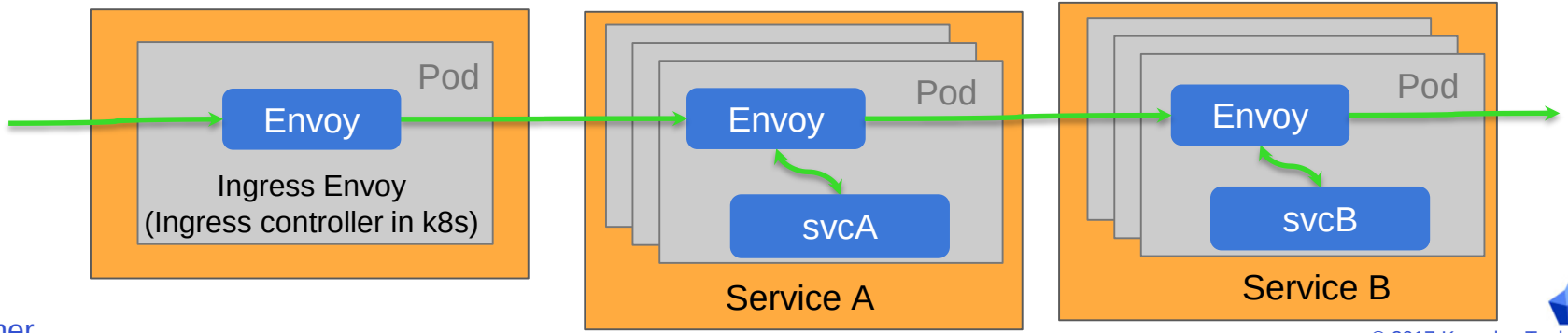
© 2017 Kumulus Technologies

Ingress/Egress

Istio assumes that all traffic entering/exiting the service mesh transits through Envoy proxies.

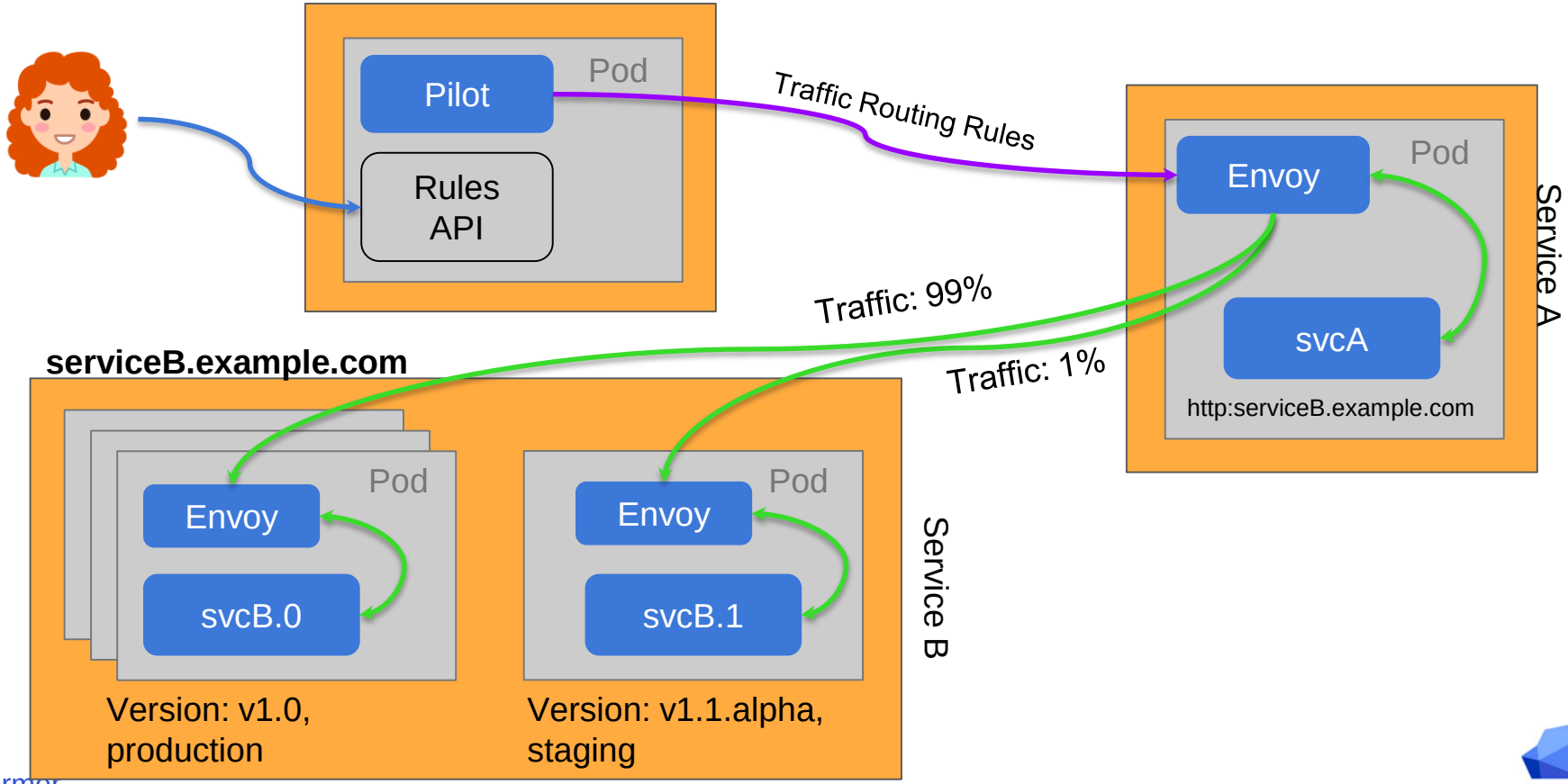
Deploying the Envoy proxy in front of services, operators can conduct A/B testing, deploy canary services, etc. for user-facing services.

Routing traffic to external web services (e.g video service API) via the sidecar Envoy allows operators to add failure recovery features (e.g.timeouts, retries, circuit breakers, etc.) and obtain detailed metrics on the connections to these services.



© 2017 Kumulus Technologies

Request Routing - Service Versions



© 2017 Kumulus Technologies

Service Observability/Visibility

Monitoring & tracing should not be an afterthought

Ideally a monitoring/tracing system should provide:

- Metrics without instrumenting apps
- Consistent metrics across fleet
- Trace flow of requests across services
- Portable across metric backend providers

Istio adapters seamlessly integrate a number of tools:
Prometheus - gathers metrics from Istio Mixer
Grafana - produces dashboards from Prometheus metrics
Service Graph - generates visualizations of dependencies between services.
Zipkin - distributed tracing

© 2017 Kumulus Technologies

Application/service Resilience with Istio

As the number of microservices increase, failure is expected (inevitable?). Fault-tolerance is applications is (should be) a requirement.

Istio provides fault tolerance/resilience with no impact on application code.

Istio provides multiple, built-in features to provide fault tolerance:

Timeouts, Retries with timeout budget, Circuit breakers, Health checks

AZ-aware load balancing w/ automatic failover

Control connection pool size and request load

Systematic fault injection

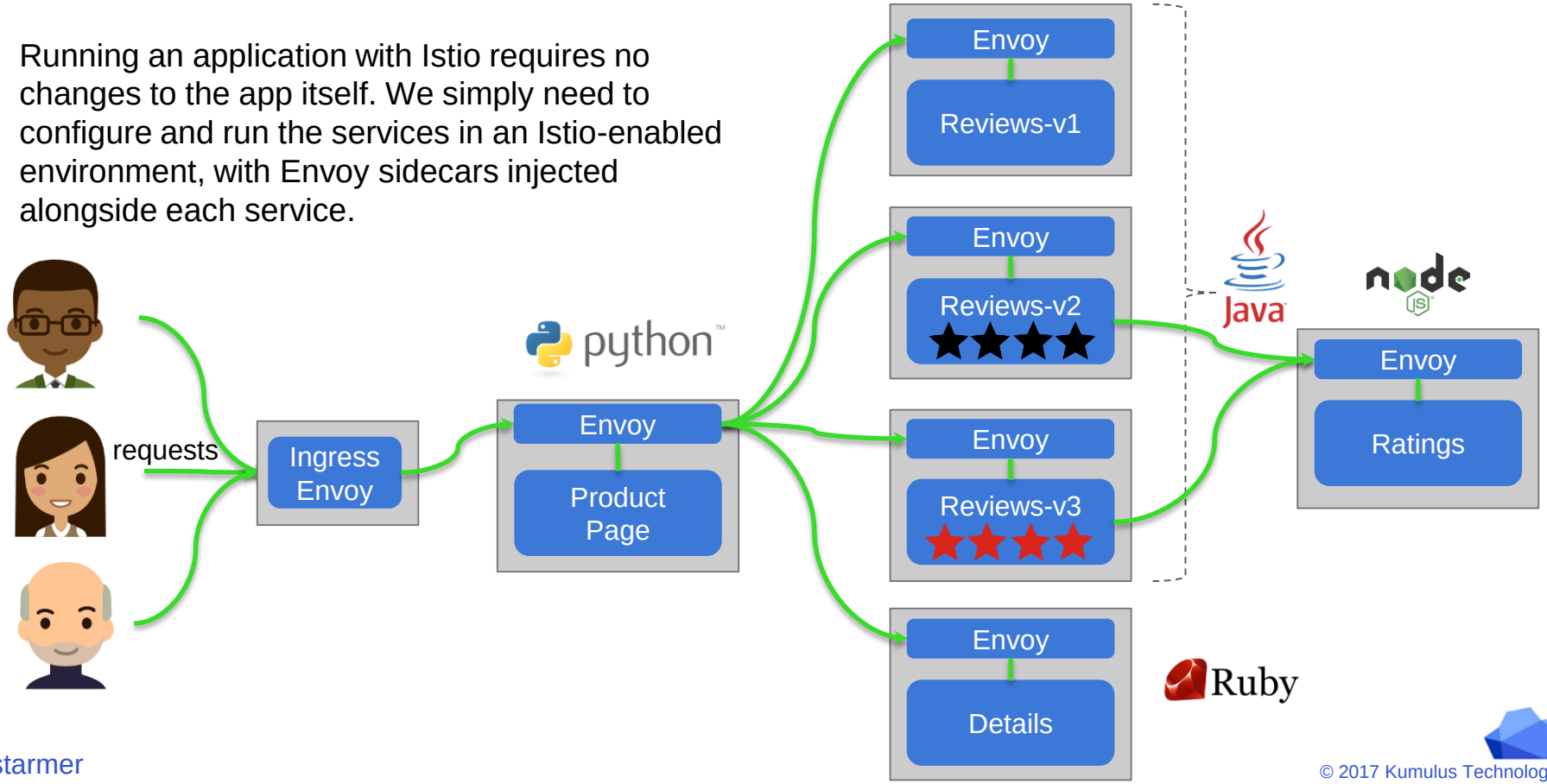
© 2017 Kumulus Technologies

Istio Lab

Istio on Kubernetes

Example Microservice Application with Istio

Running an application with Istio requires no changes to the app itself. We simply need to configure and run the services in an Istio-enabled environment, with Envoy sidecars injected alongside each service.



© 2017 Kumulus Technologies

Get Started - Deploy Kubernetes

Easiest approach: Launch in the cloud

- GKE
- Azure
- AWS with Kops

Or, launch on your own hardware

- Vagrant/Ansible (kubespray)
- Kubeadm/Minikube

© 2017 Kumulus Technologies