

Istio Service Mesh with Kubernetes and Calico



Karthik Prabhakar

kp@tiger.io



TIGERA
CLOUD NETWORKS, SECURED

Kubernetes Networking Concepts

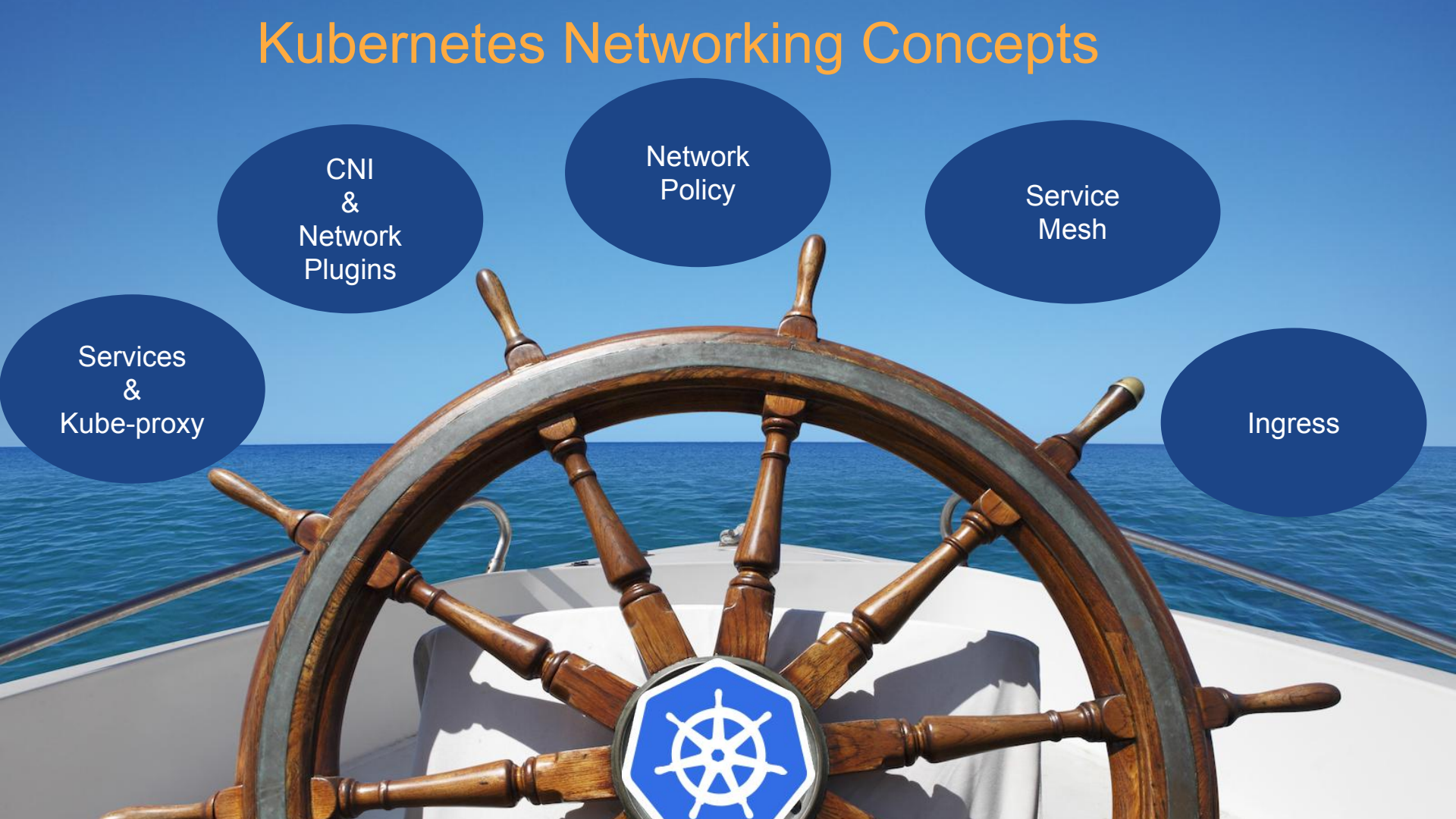
CNI
&
Network
Plugins

Network
Policy

Service
Mesh

Services
&
Kube-proxy

Ingress



Network

=>

Interconnect workloads

+

Network Policy

=>

Isolate workloads

Calico: Scalable, Secure, Cloud-native Networking and Policy

Dynamic security policy + simple networking

Intent-based policy spans

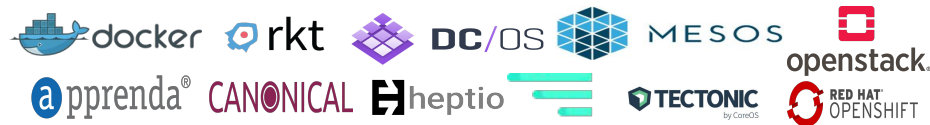
- ▶ Public, private & hybrid cloud
- ▶ Heterogeneous workloads and orchestration environments

Simple, highly scalable IP networking model

Distributed, Dynamic Policy Enforcement

Open source

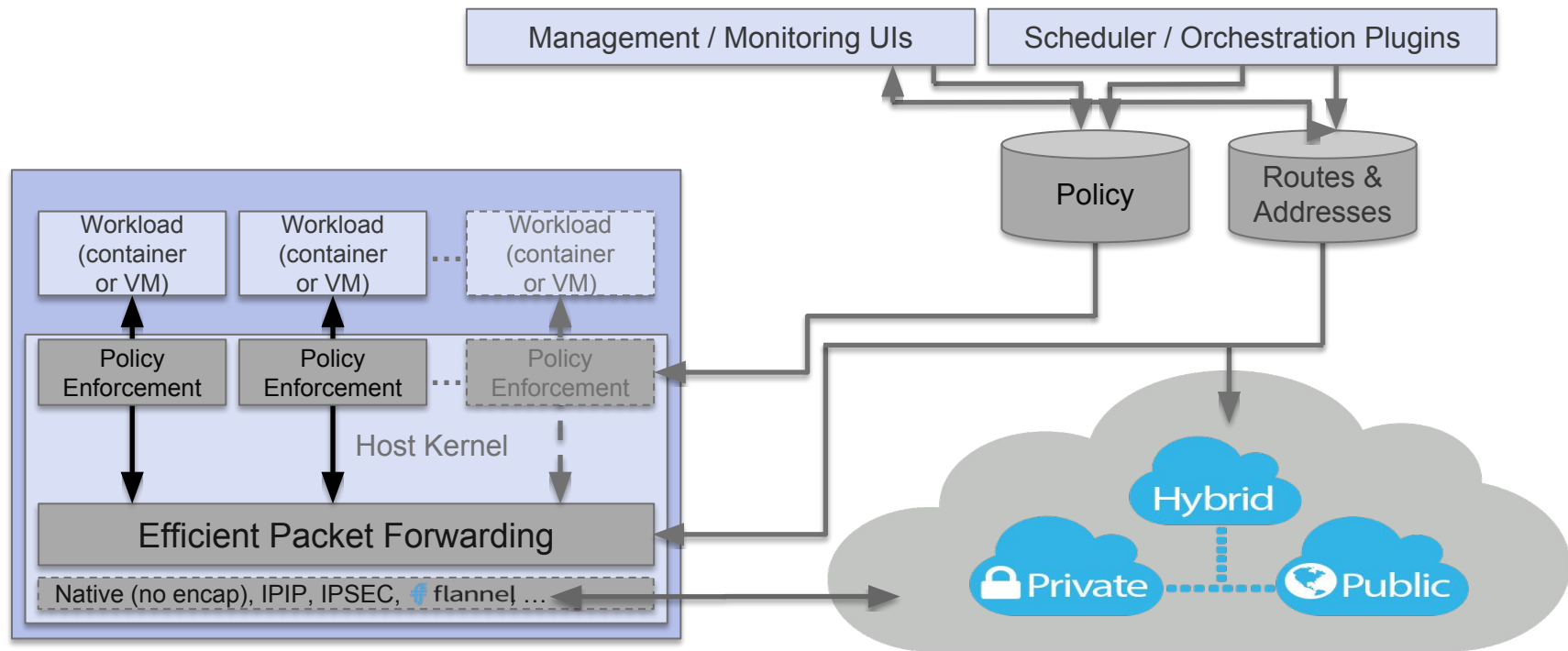
Apache license



kubernetes



Calico Networking and Network Policy: Functional Architecture





Simple

Deploy: *kubectl apply -f calico.yaml*

Troubleshoot: *ping, traceroute, ip route show, iptables -L*

Scalable

Widespread deployment, scales to very large clusters

Seamless

Integration across the Kubernetes ecosystem

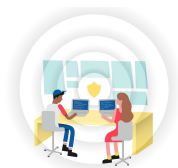
Secure

Dynamic, Distributed Network Policy Enforcement

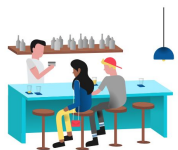
Demo:
Deploying Kubernetes with Calico



Traffic Management, Resilience and Load Balancing



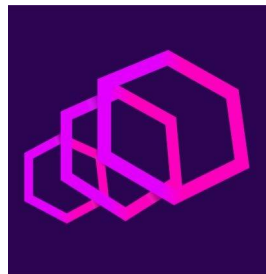
Security, encryption and SPIFFE



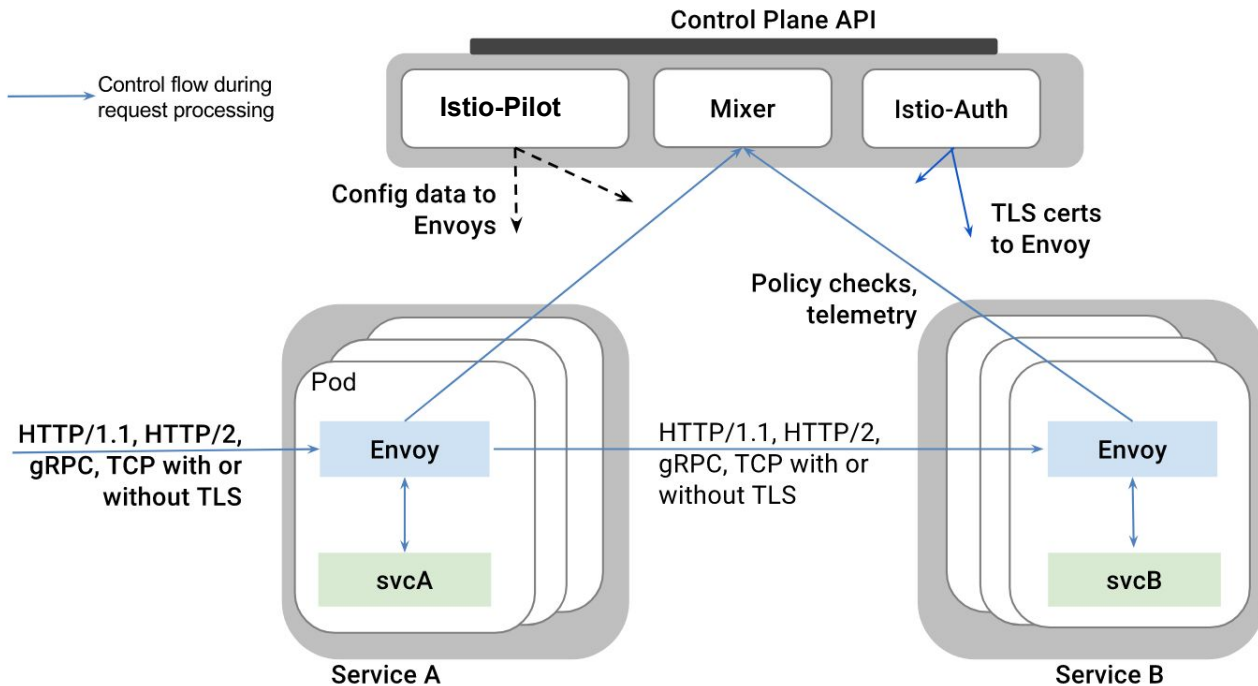
Fleet-wide Policy enforcement

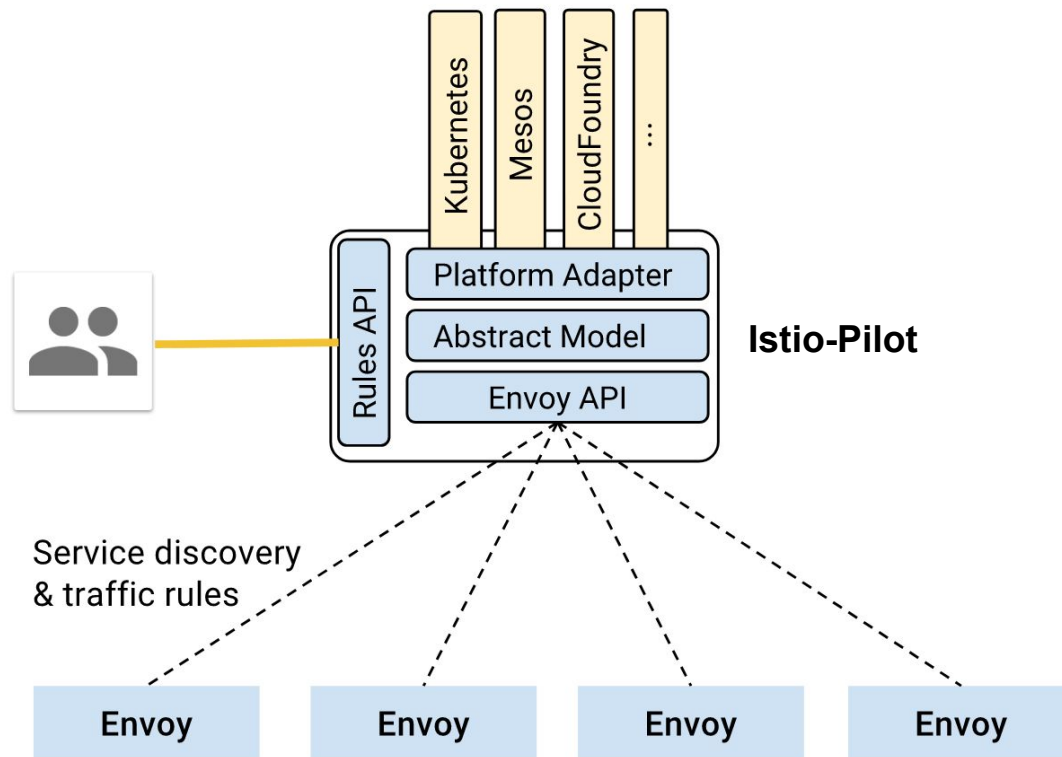


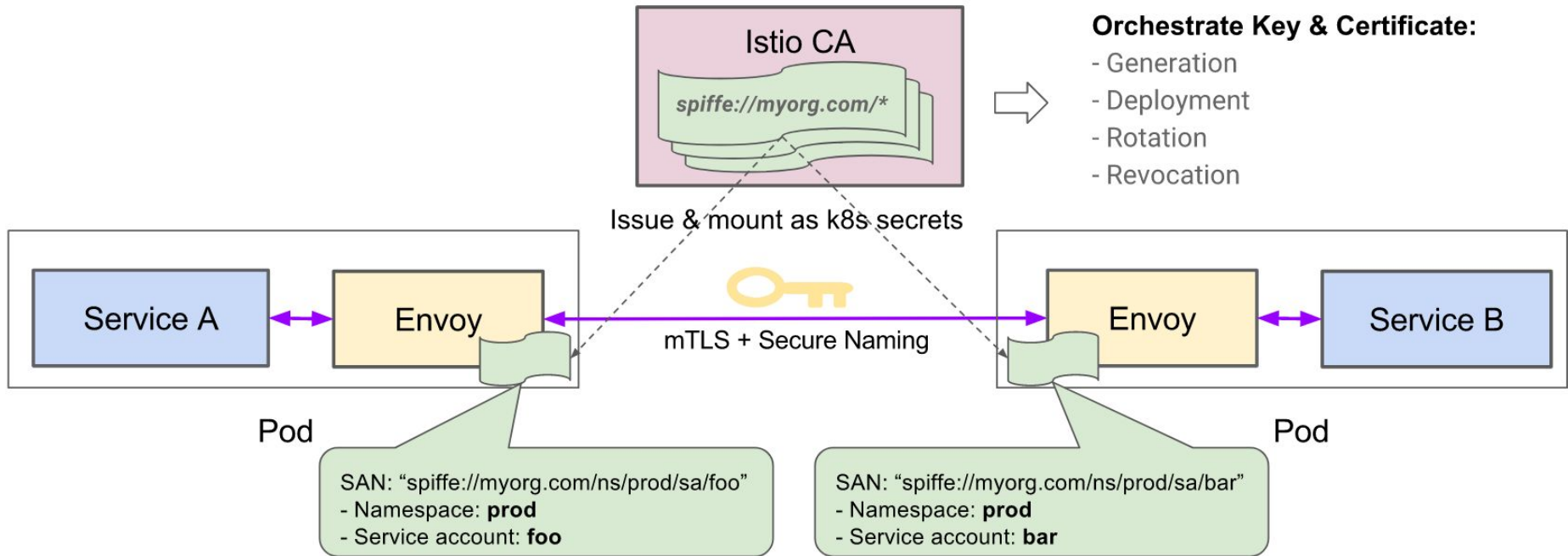
Telemetry and Reporting

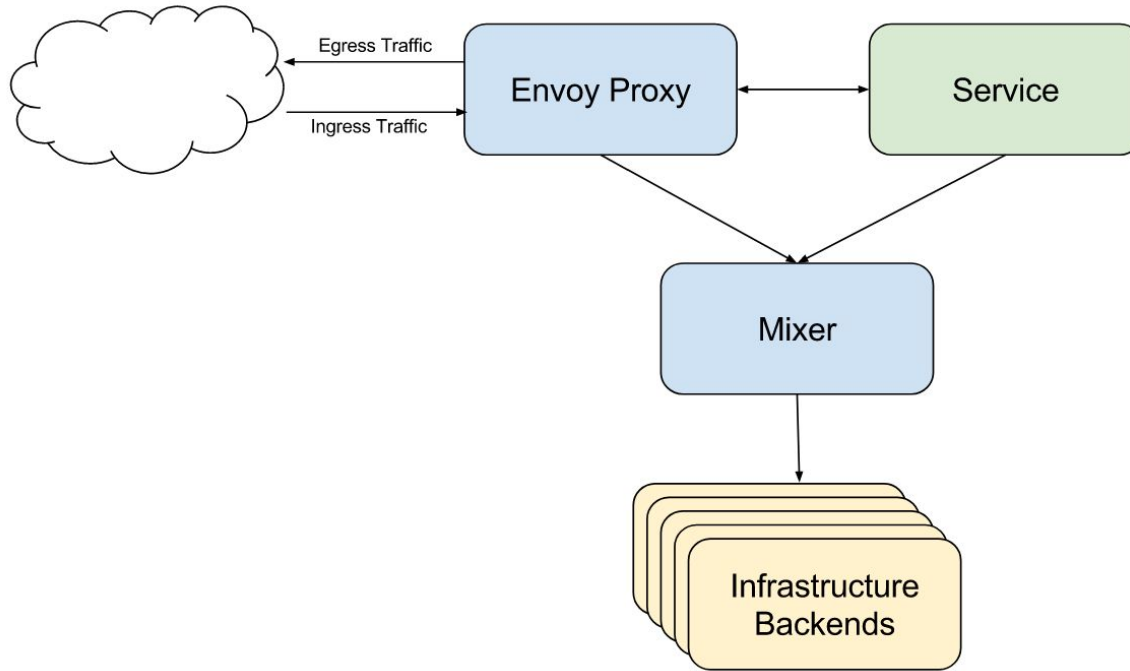


Istio Architecture

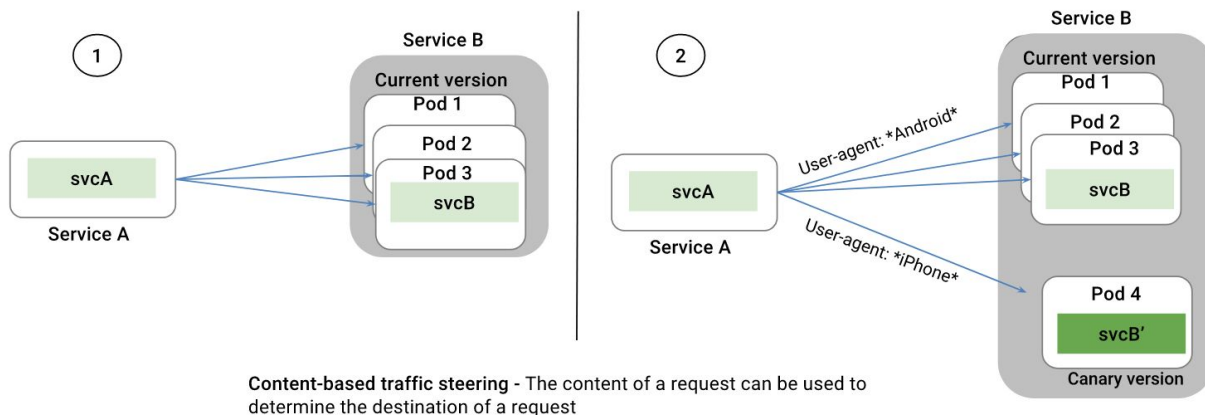
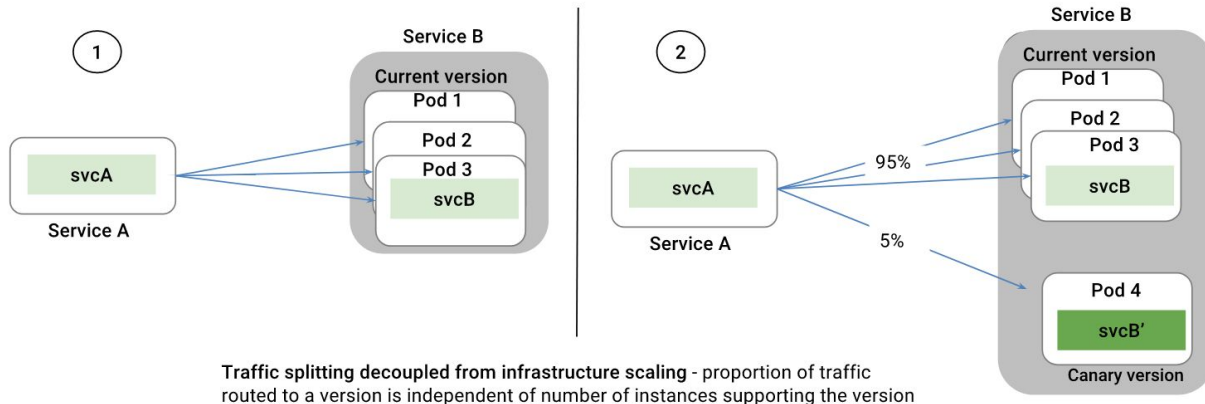




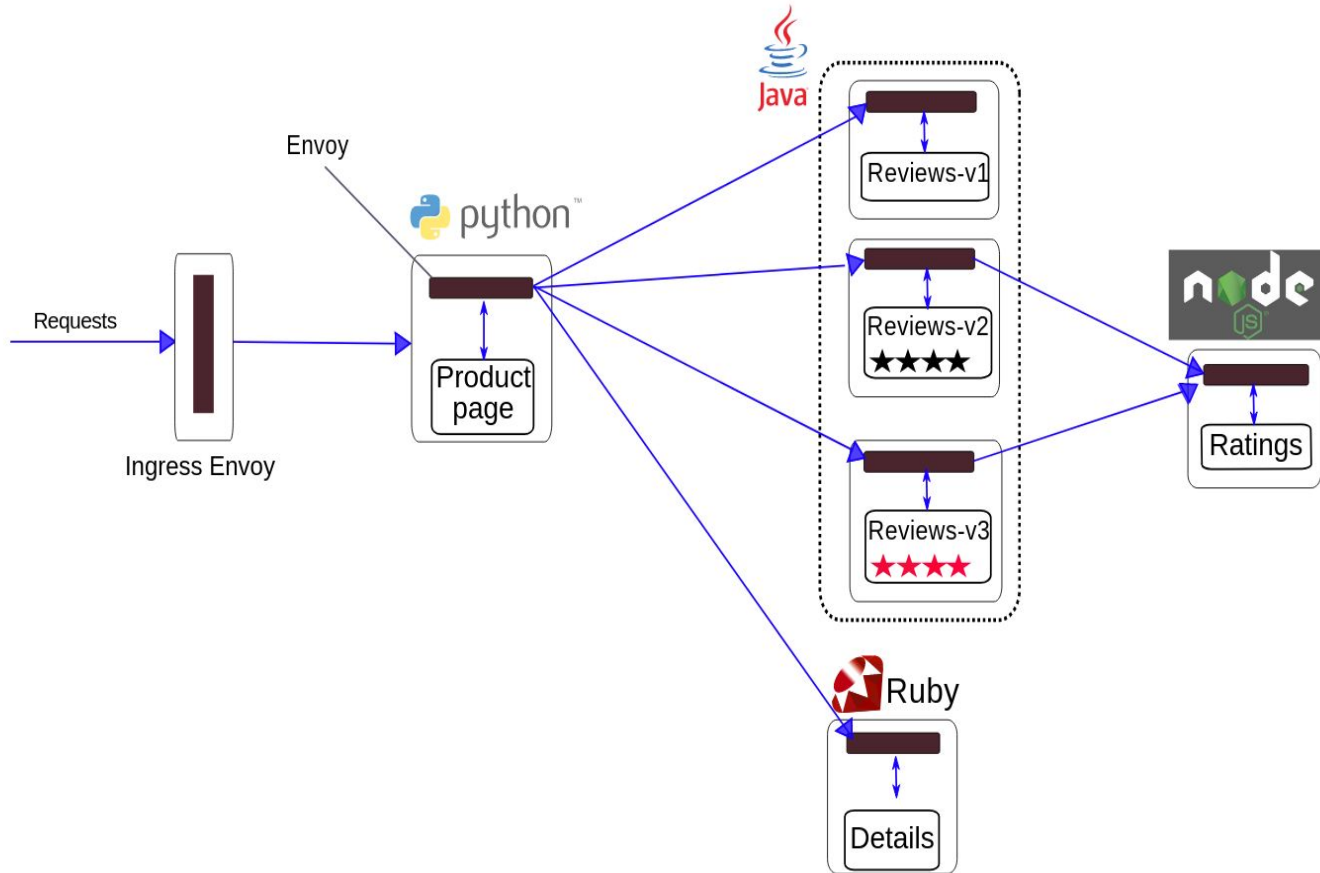




Traffic Management Examples with a Service Mesh



Istio: Microservices Example - Bookinfo application



Demo:

Deploying Istio+Envoy Service Mesh

Deploying & Managing a Microservices App



**KEEP
CALM
and
kubectl
ON**



KEEP
CALM
and
calicoctl
ON



**KEEP
CALM
and
istioctl
ON**



TIGERA
CLOUD NETWORKS, SECURED



slack.projectcalico.org



istio.io