Istio: The Packet's-Eye View

Matt Turner
@mt165
mt165.co.uk

Kubecon Seattle
December 2018

Objectives

Learn how a packet traverses an Istio/Envoy/Kubernetes system

See what control plane calls are made in that process

Build a useful mental model for reasoning about, and debugging Istio

Prerequisites

Basic networking knowledge

Intermediate Kubernetes knowledge

An understanding of what Istio is and does

Environment

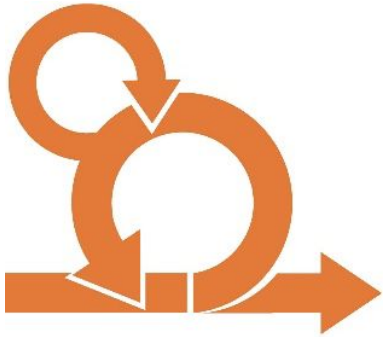
Istio 1.0.3

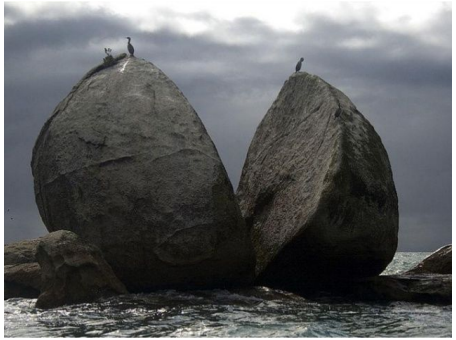
GKE Kubernetes 1.11

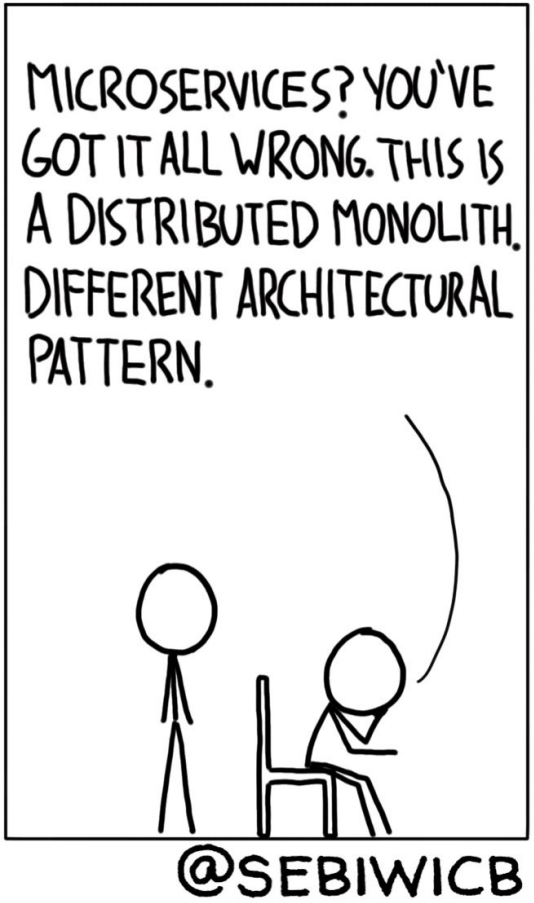
Background

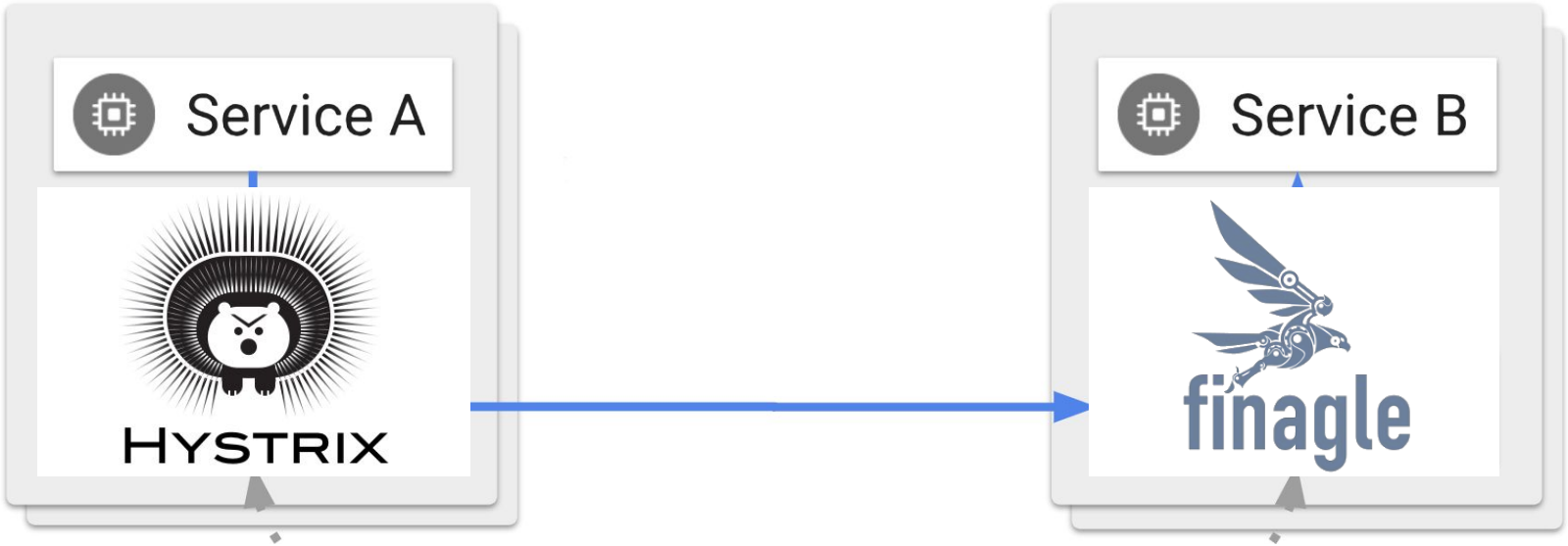
Why?

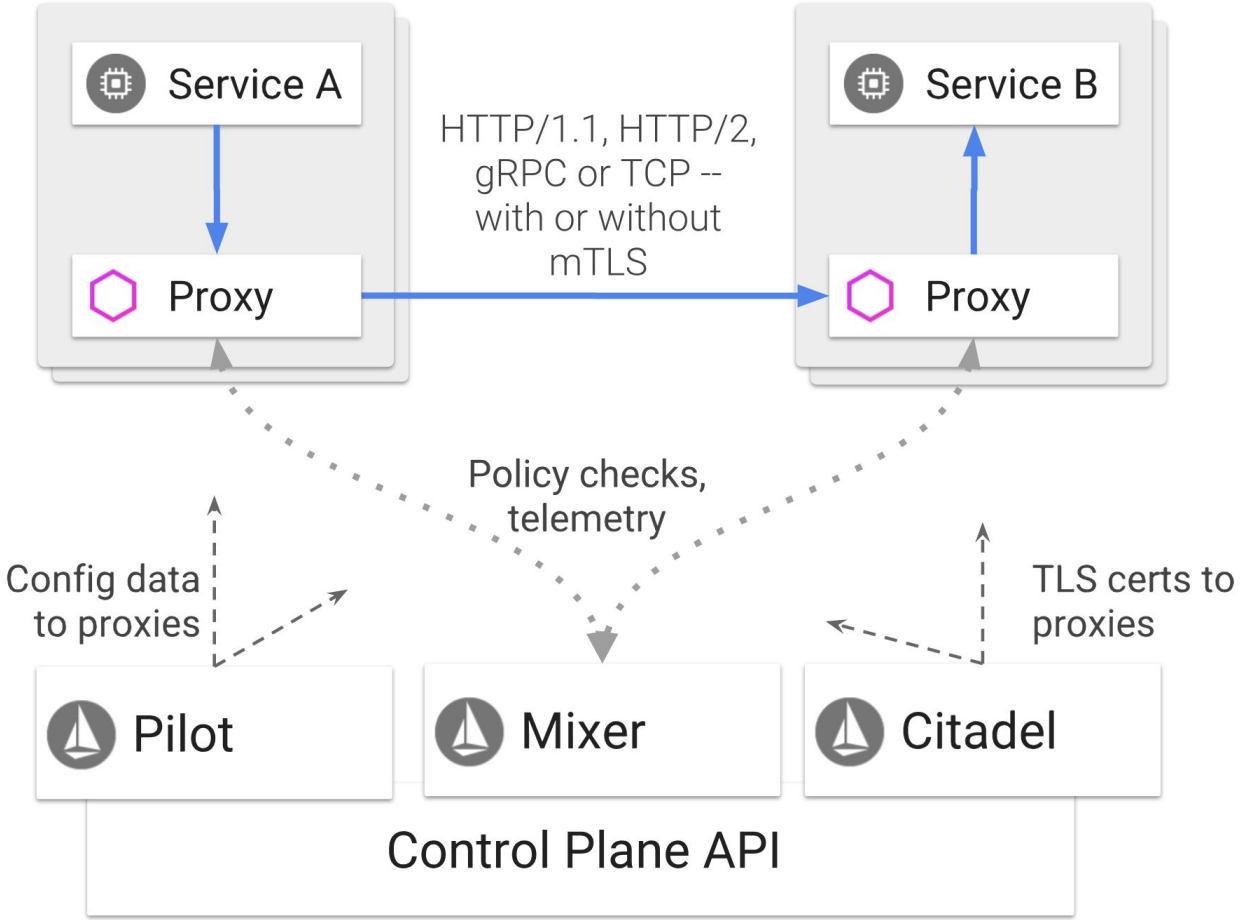

NEW
RELEASE!



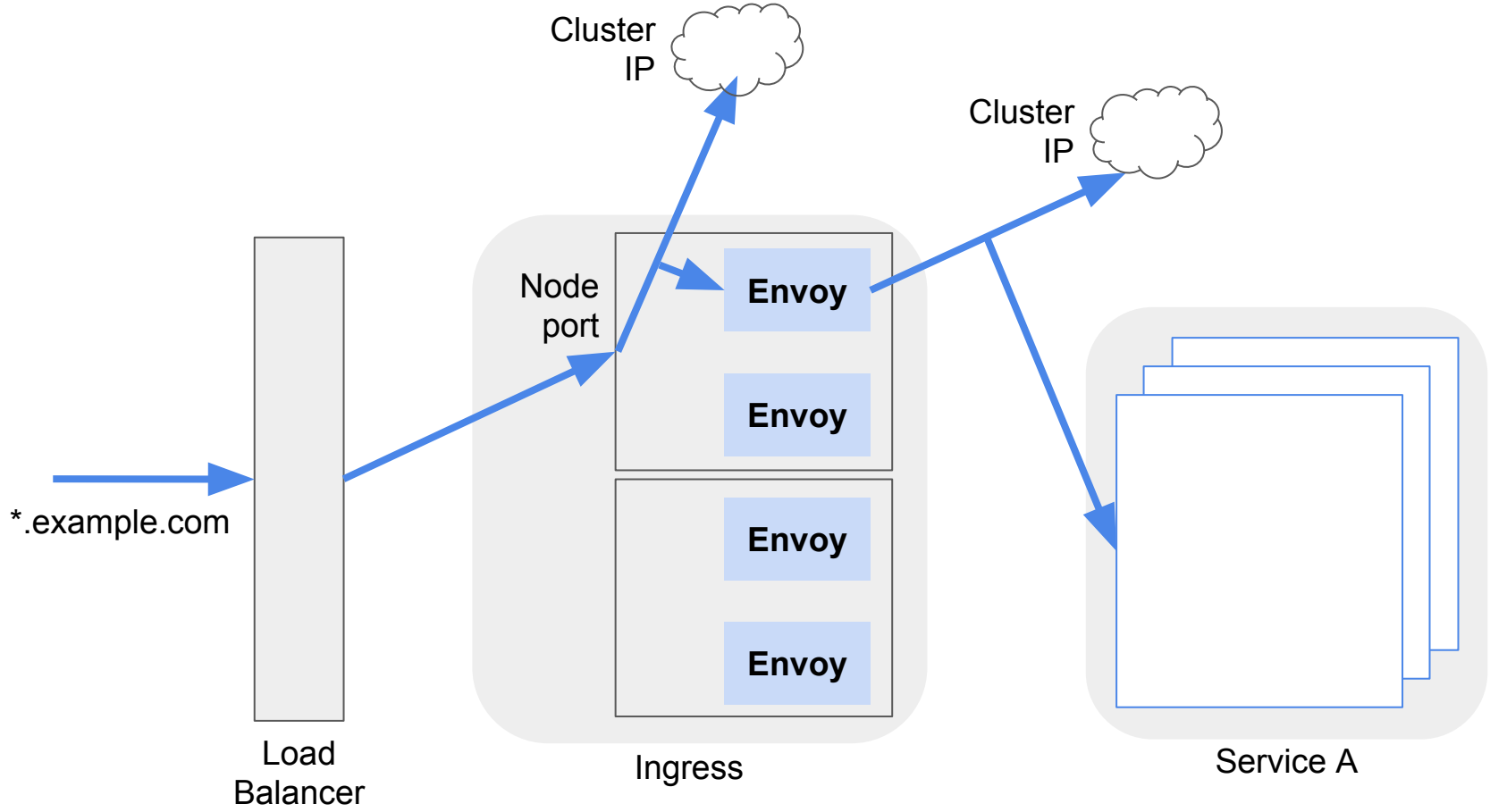


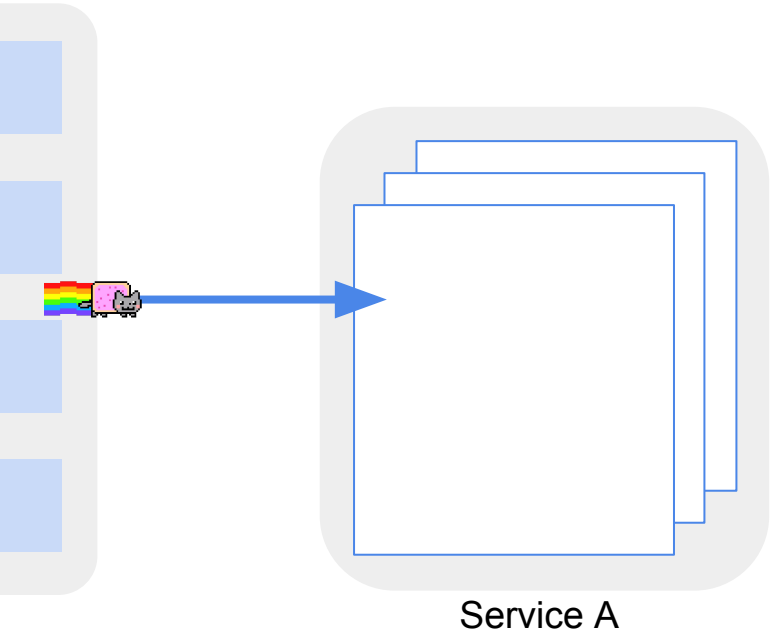

MICROSERVICES? YOU'VE GOT IT ALL WRONG. THIS IS A DISTRIBUTED MONOLITH. DIFFERENT ARCHITECTURAL PATTERN.
@SEBIWICB

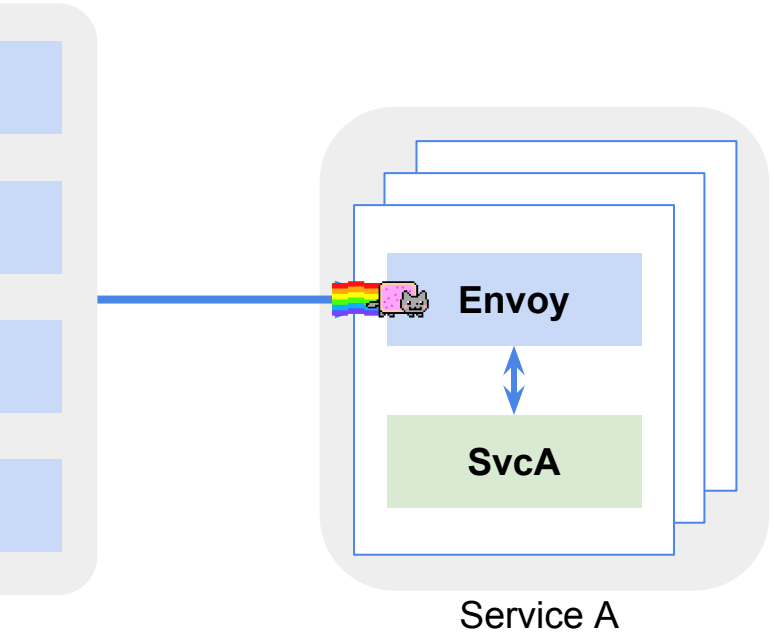

Service A
HYSTRIX
Service B
finagle


Service A
Service B
HTTP/1.1, HTTP/2,
gRPC or TCP --
with or without
mTLS
Proxy
Proxy
Policy checks,
telemetry
Config data
to proxies
TLS certs to
proxies
Pilot
Mixer
Citadel
Control Plane API

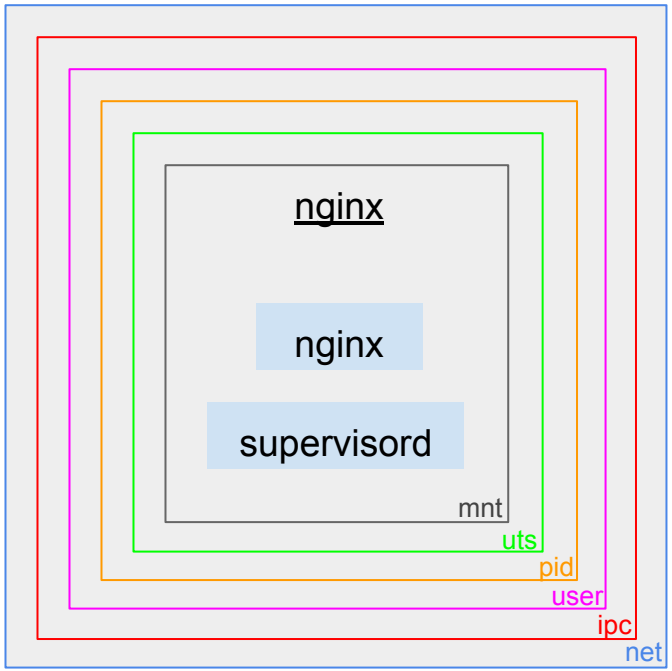
Networking and Containers


Cluster
IP
Cluster
IP
Node
port
Envoy
Envoy
Envoy
Envoy
*.example.com
Load
Balancer
Ingress
Service A

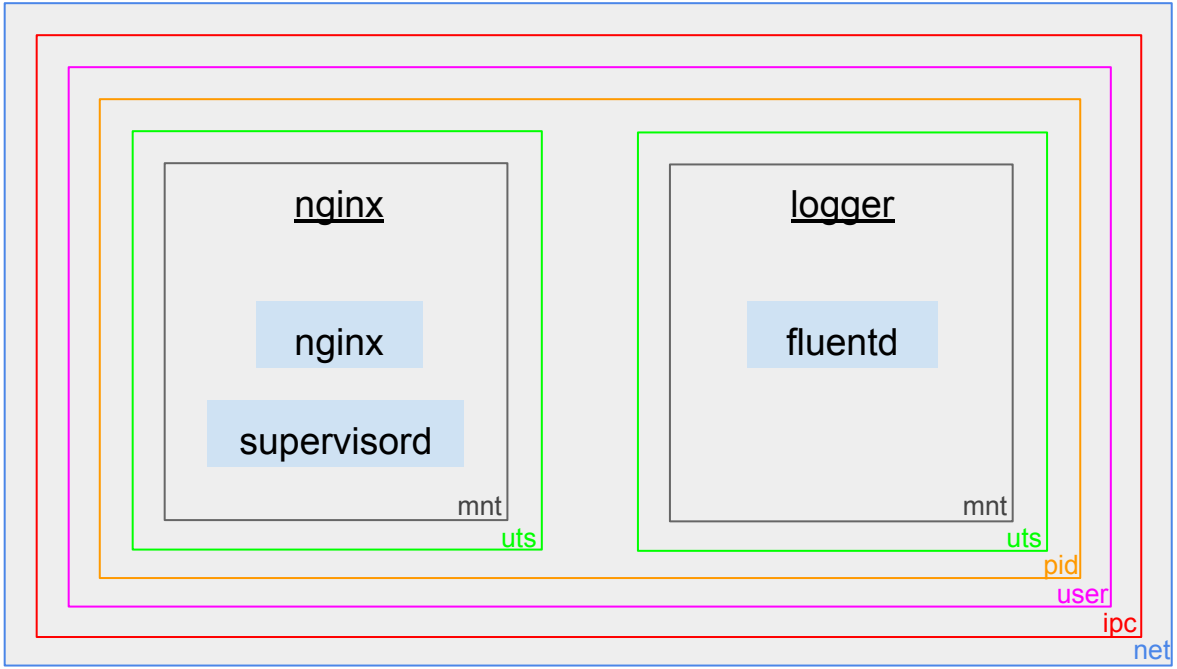

Service A


Envoy
SvcA
Service A

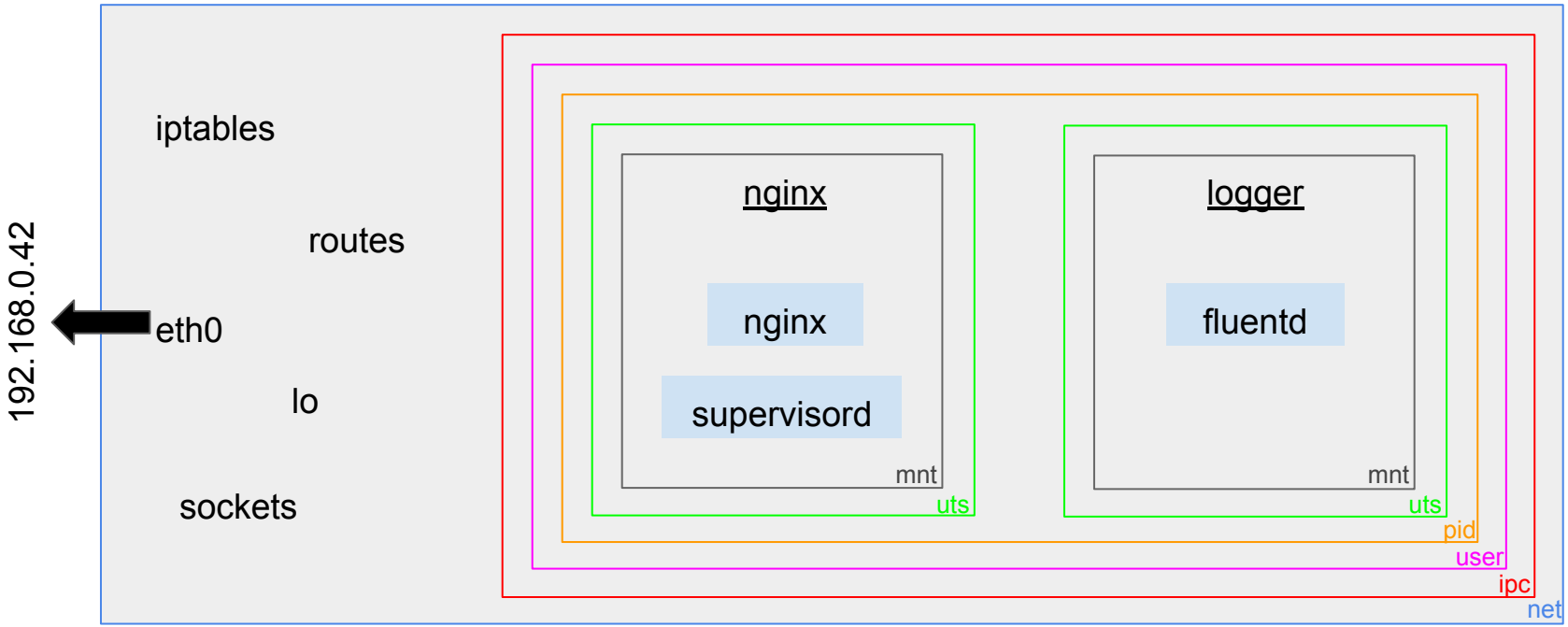
"Containers"



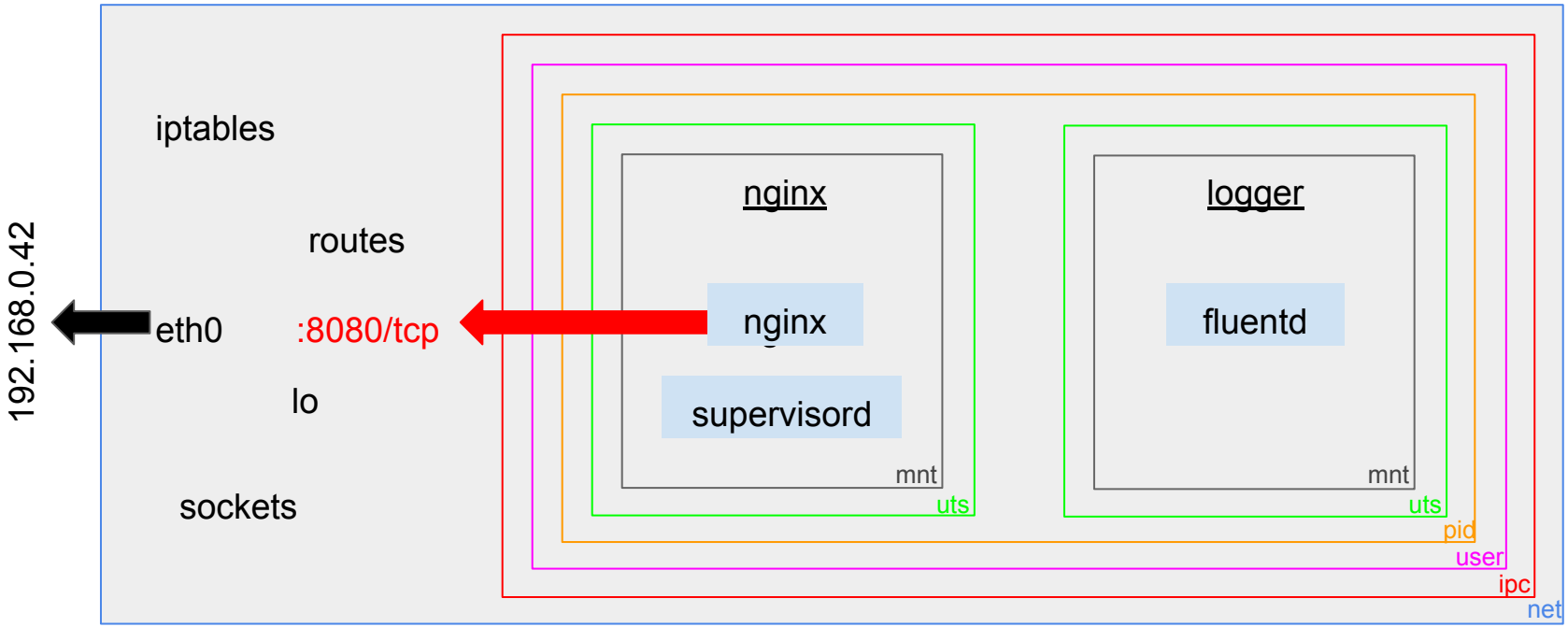
Kubernetes Pods



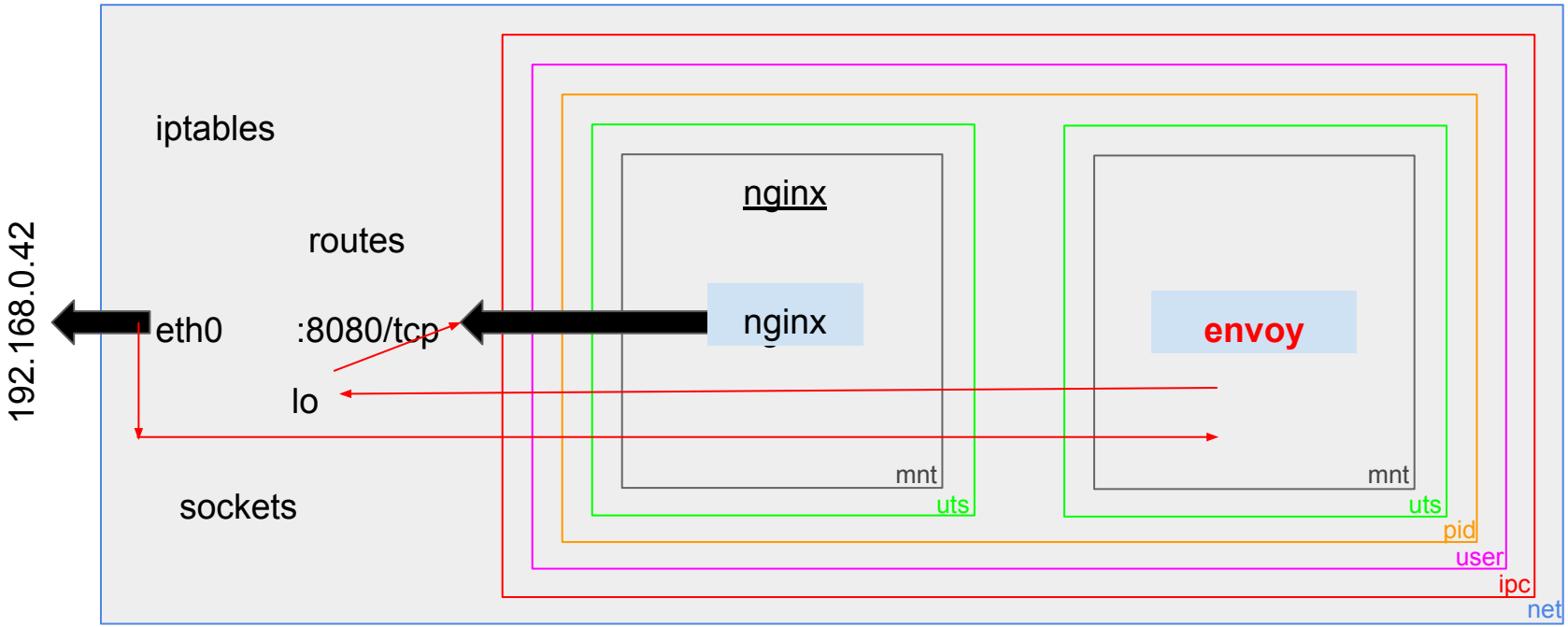
Kubernetes Pods



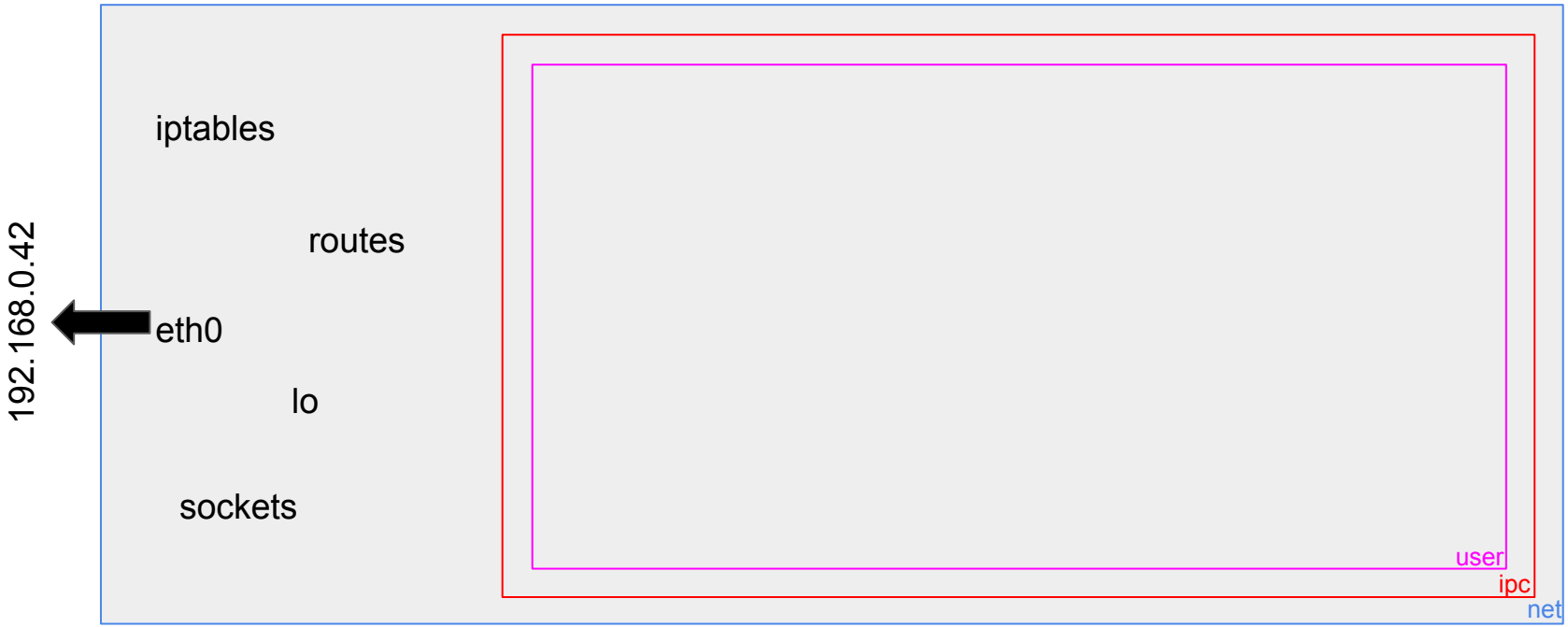
Kubernetes Pods



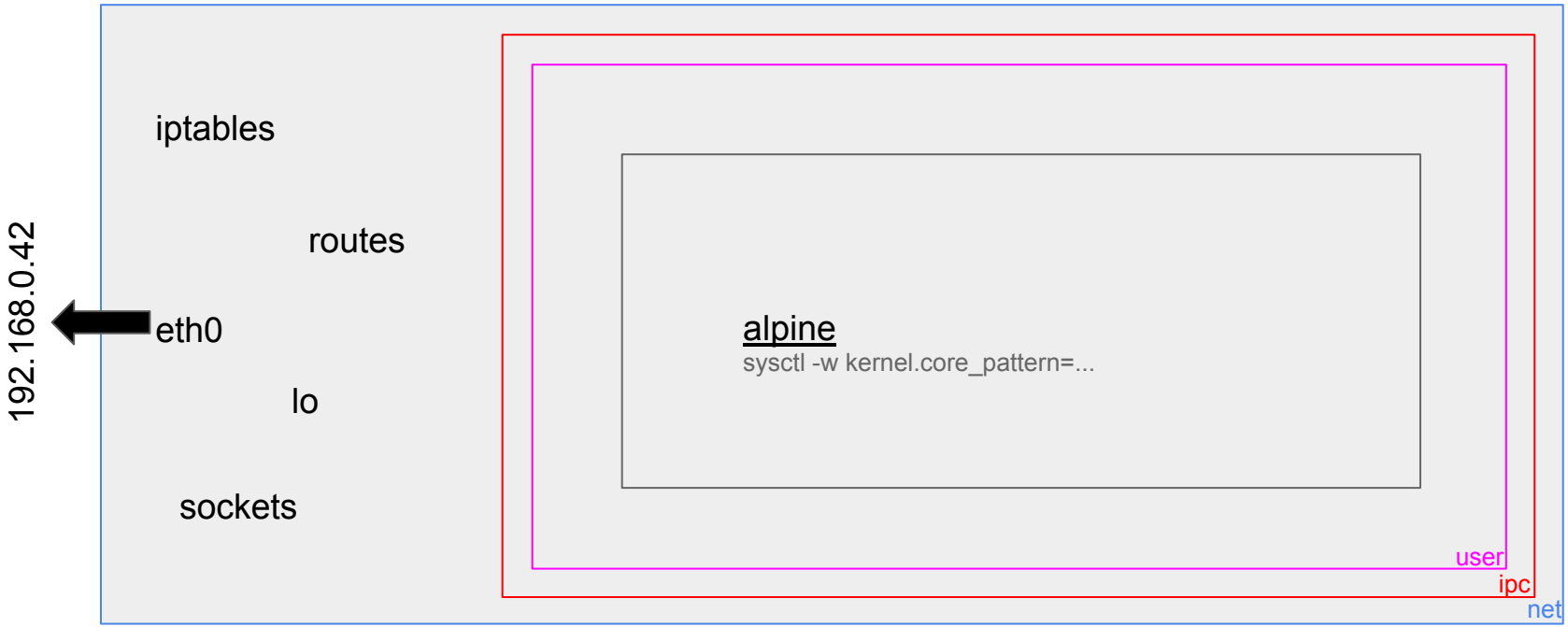
Kubernetes Pods



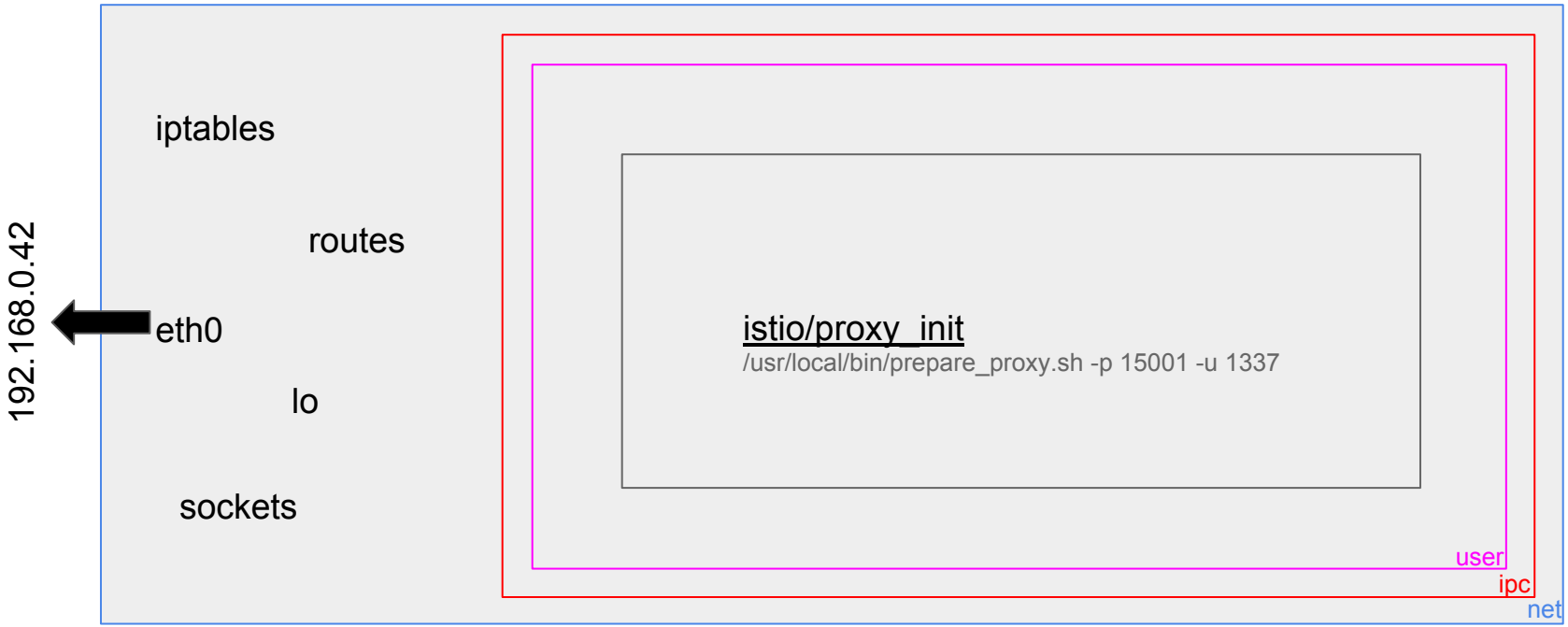
Sidecar Injection



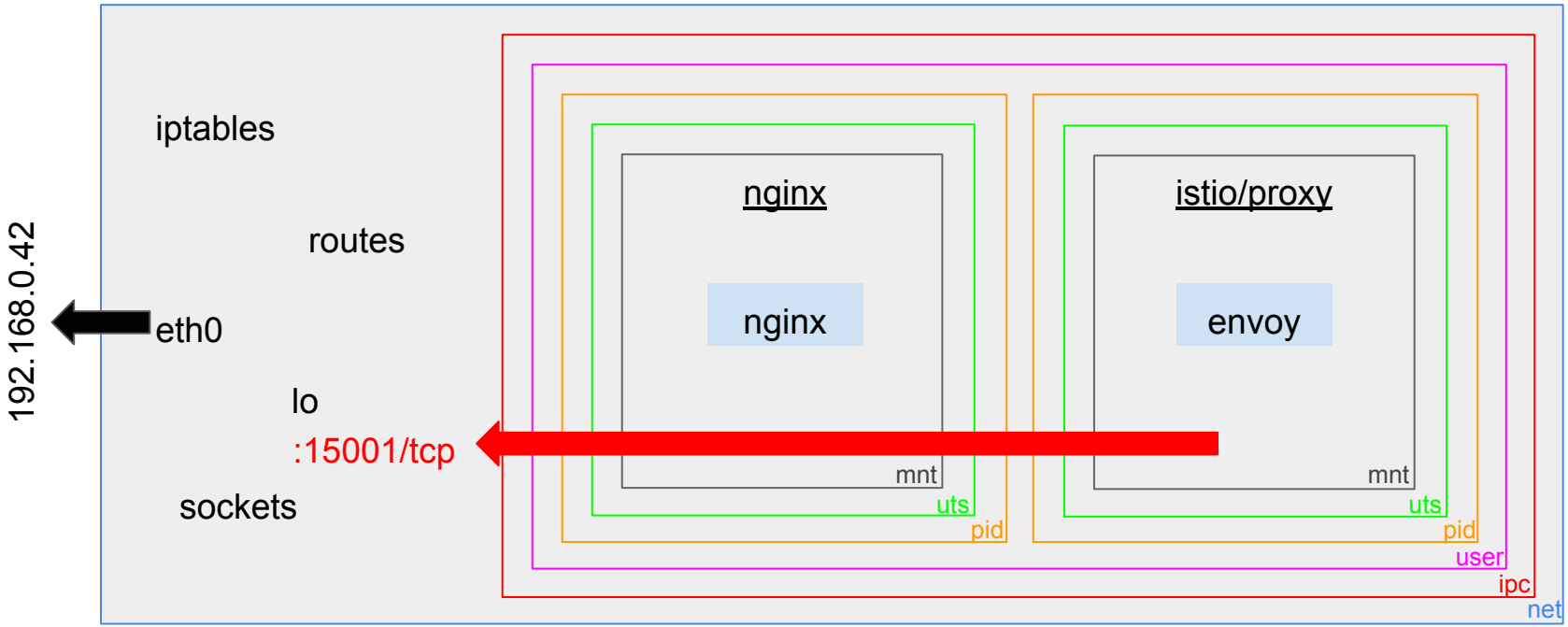
Sidecar Injection

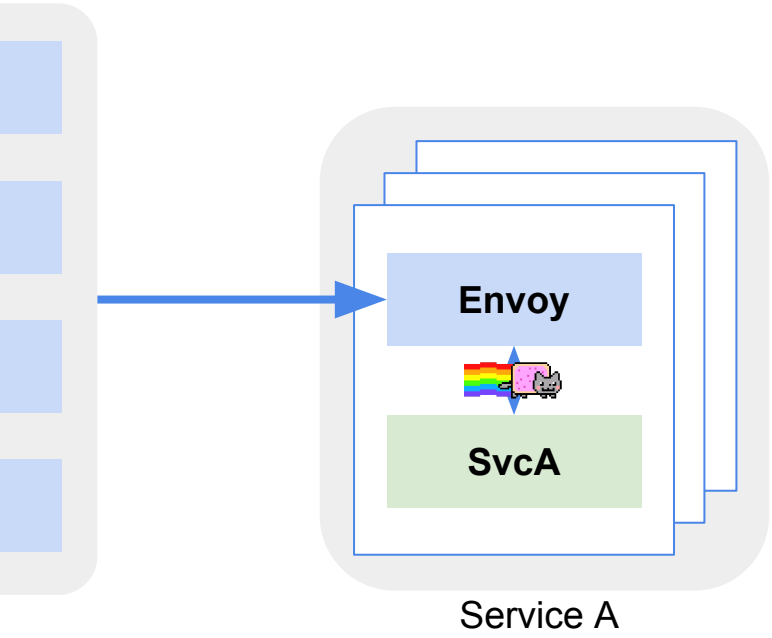


Sidecar Injection

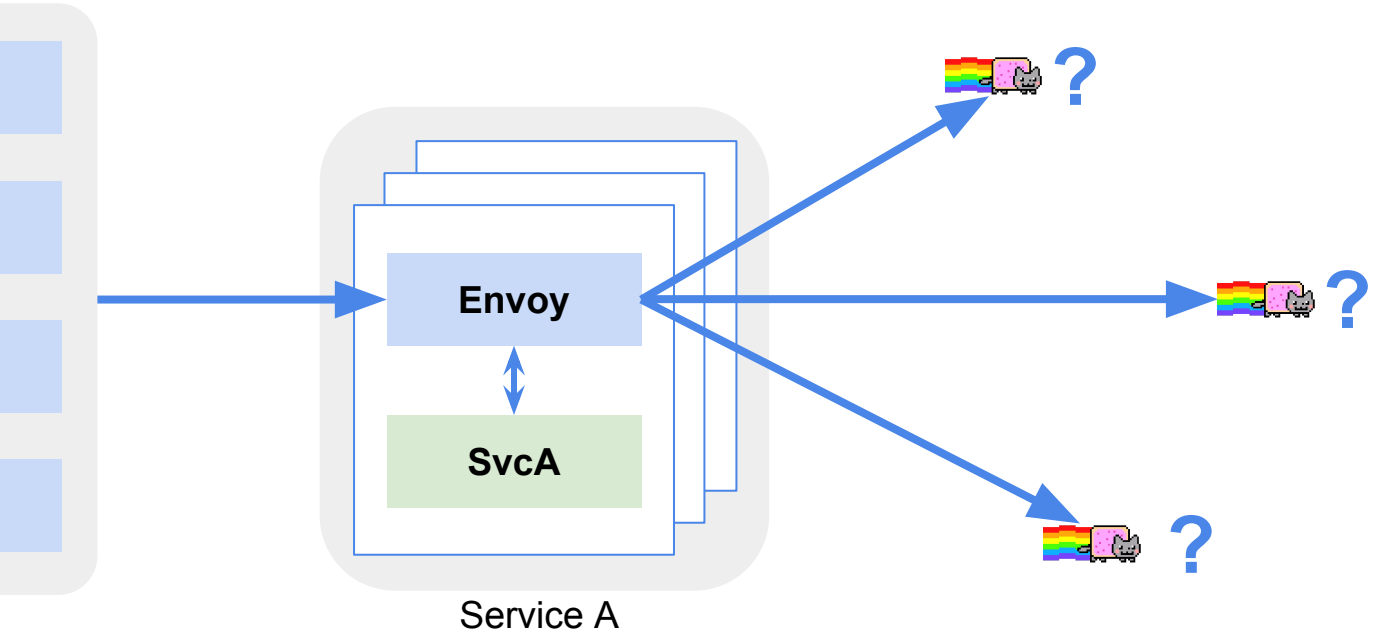


Sidecar Injection




Envoy
SvcA
Service A

Pilot and Routing


Envoy
SvcA
Service A
?
?
?

Services

```
$ kubectl get services  -o wide | grep httpbin

httpbin      NodePort    10.0.0.244   <none>        80:30082/TCP   16m       app=httpbin
```

Service DNS exposure

```
# dig httpbin.default.svc.cluster.local

    httpbin.default.svc.cluster.local. 23 IN A  10.0.0.244
```

Pods

$ kubectl get pods -o wide | grep httpbin

```
httpbin-76ddd74666-2m6ds   1/1   Running   0   16m   172.17.0.13   minikube
httpbin-76ddd74666-ls66n   1/1   Running   0   16m   172.17.0.12   minikube
httpbin-76ddd74666-x5ql2   1/1   Running   0   16m   172.17.0.5    minikube
```

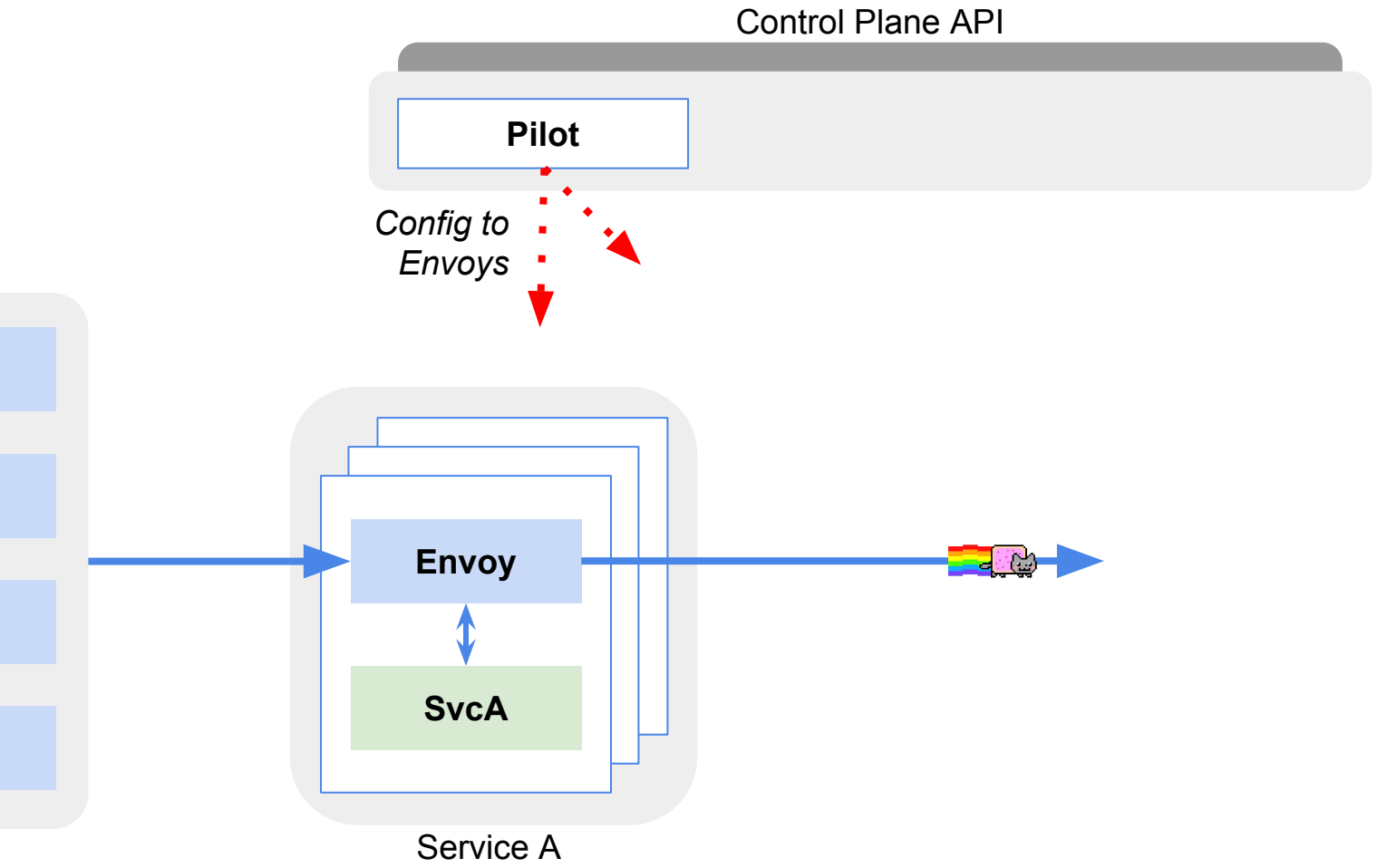
Endpoints

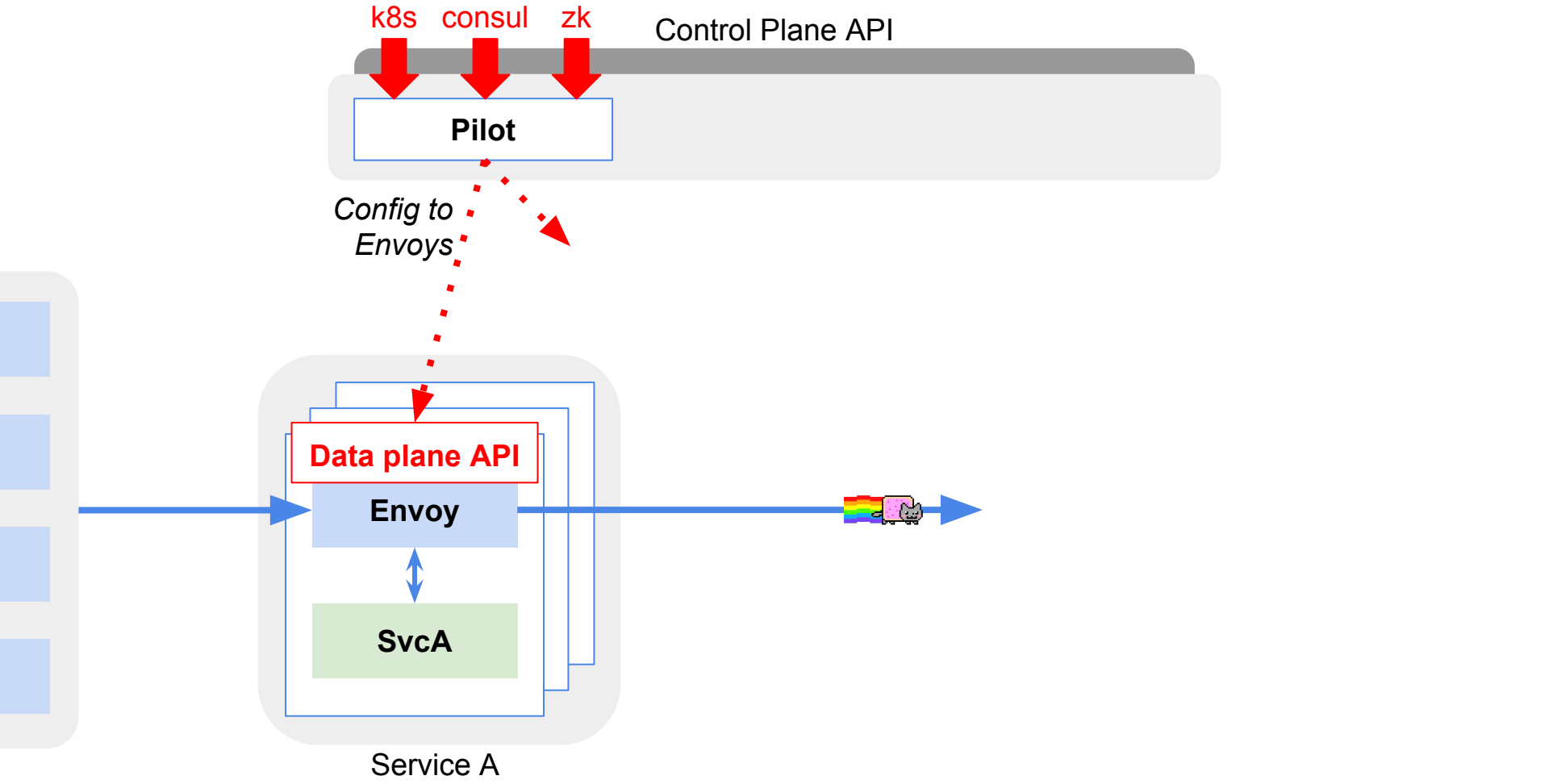
```
$ kubectl get endpoints | grep httpbin

httpbin      172.17.0.12:8000,172.17.0.13:8000,172.17.0.5:8000   21m
```

Endpoints

```
$ kubectl get endpoints httpbin -o yaml
apiVersion: v1
kind: Endpoints
…
subsets:
- addresses:
  - ip: 172.17.0.12
    nodeName: minikube
    targetRef:
      kind: Pod
…
  ports:
  - name: http
    port: 8000
    protocol: TCP
```

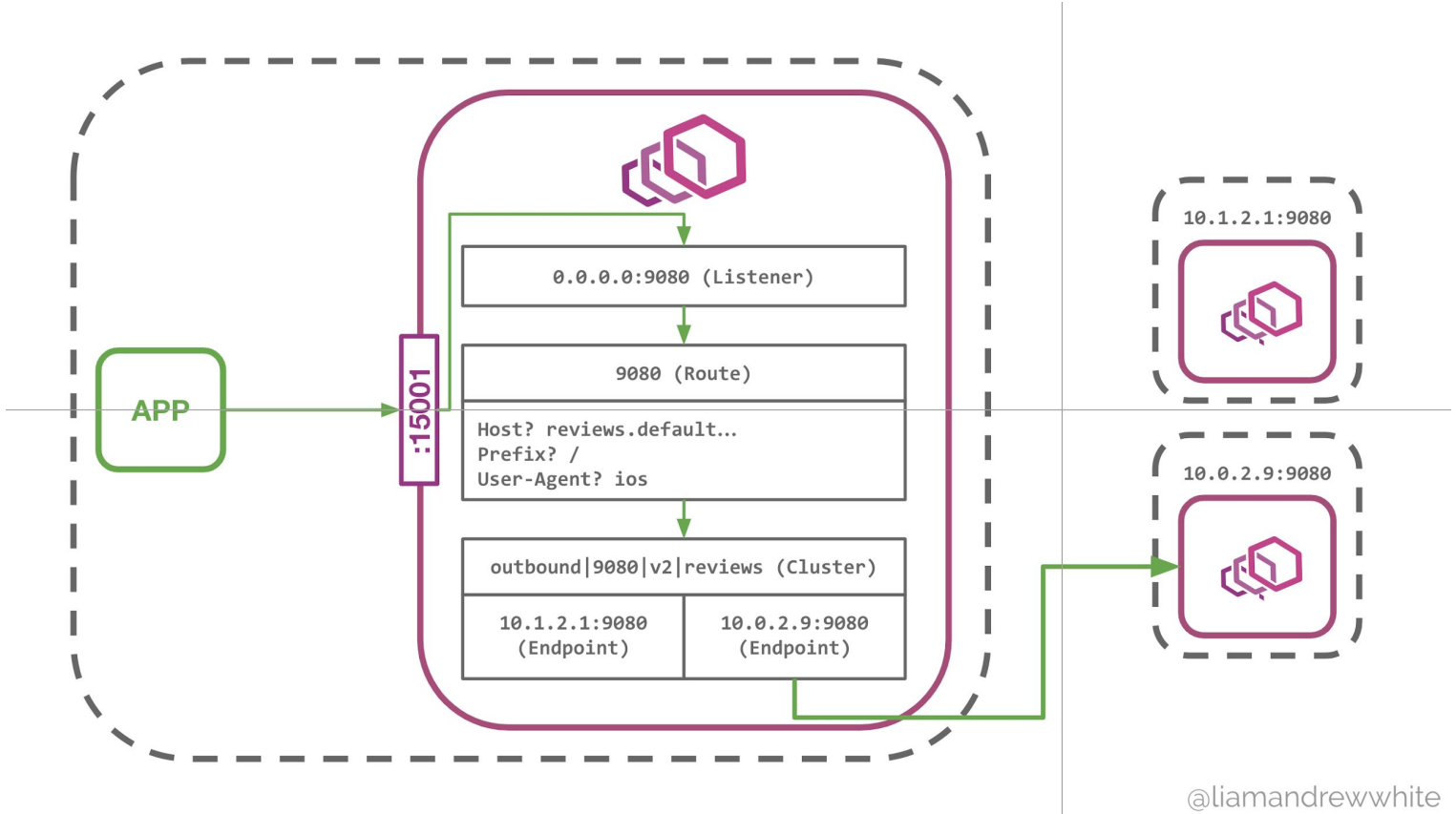

Control Plane API
Pilot
Config to Envoys
Envoy
SvcA
Service A


k8s
consul
zk
Control Plane API
Pilot
Config to Envoys
Data plane API
Envoy
SvcA
Service A

Demo: set up

Demo: proxy-config

Envoy


APP
:15001
0.0.0.0:9080 (Listener)
9080 (Route)
Host? reviews.default...
Prefix? /
User-Agent? ios
outbound|9080|v2|reviews (Cluster)
10.1.2.1:9080 (Endpoint)
10.0.2.9:9080 (Endpoint)
10.1.2.1:9080
10.0.2.9:9080
@liamandrewwhite

Ingress Routing

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: bookinfo-gateway
spec:
  selector:
    istio: ingressgateway # use istio default controller
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
    - "*"
```

Ingress Routing

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bookinfo
spec:
  hosts:
  - "*"
  gateways:
  - bookinfo-gateway
  http:
  - match:
    - uri:
        exact: /productpage
    - uri:
        exact: /login
    - uri:
        exact: /logout
    - uri:
        prefix: /api/v1/products
    route:
    - destination:
        host: productpage
        port:
          number: 9080
```

Traffic Mirroring

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: httpbin
spec:
  hosts:
    - httpbin
  http:
  - route:
    - destination:
        host: httpbin
        subset: v1
      weight: 100
    mirror:
      host: httpbin
      subset: v2
```

Traffic Shifting

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
  - route:
    - destination:
        host: reviews
        subset: v1
      weight: 50
    - destination:
        host: reviews
        subset: v3
      weight: 50
```

Canary Deployments

- Send a small amount of traffic
- Test on traffic with specific headers / cookies / user-agents / etc

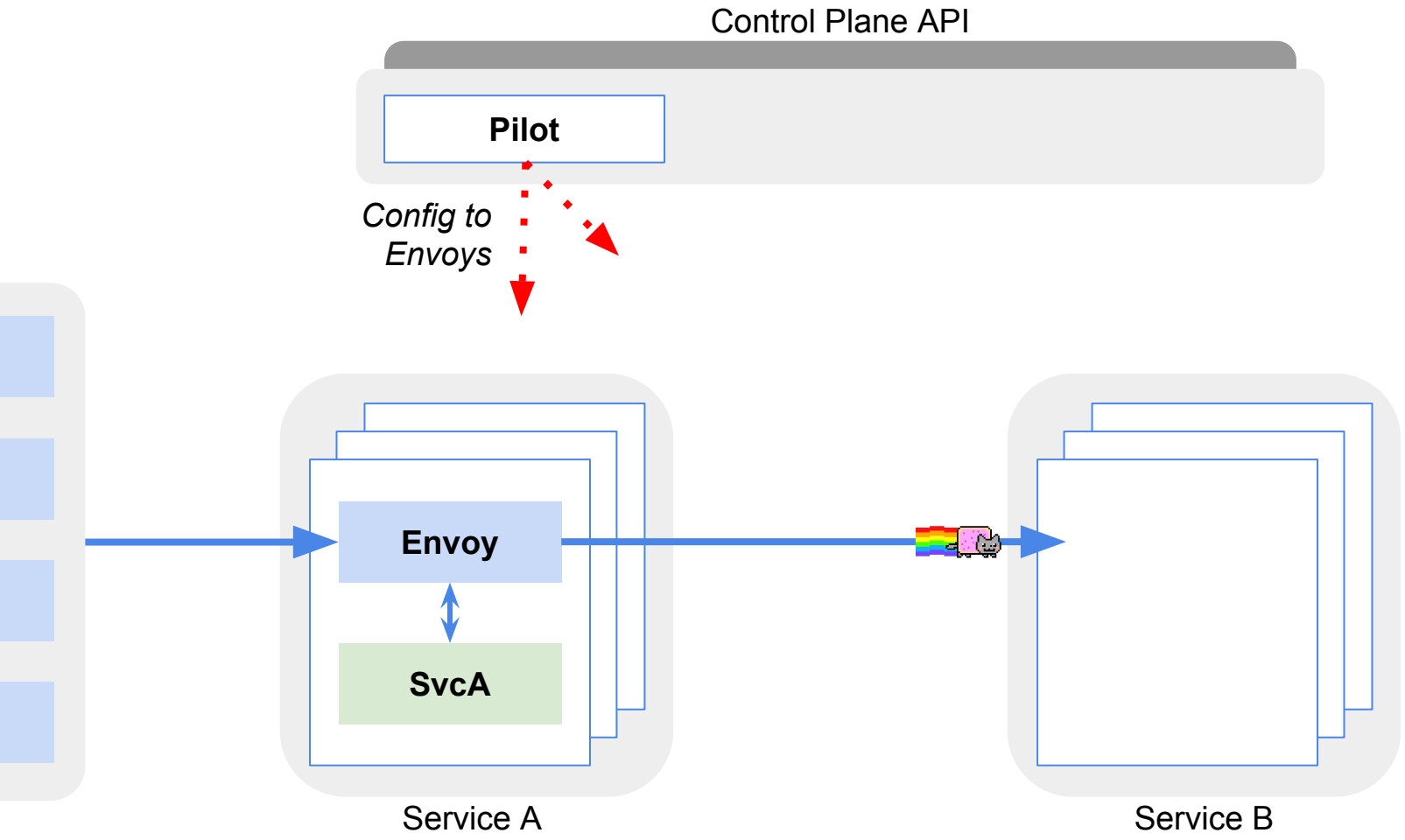
Circuit Breaking

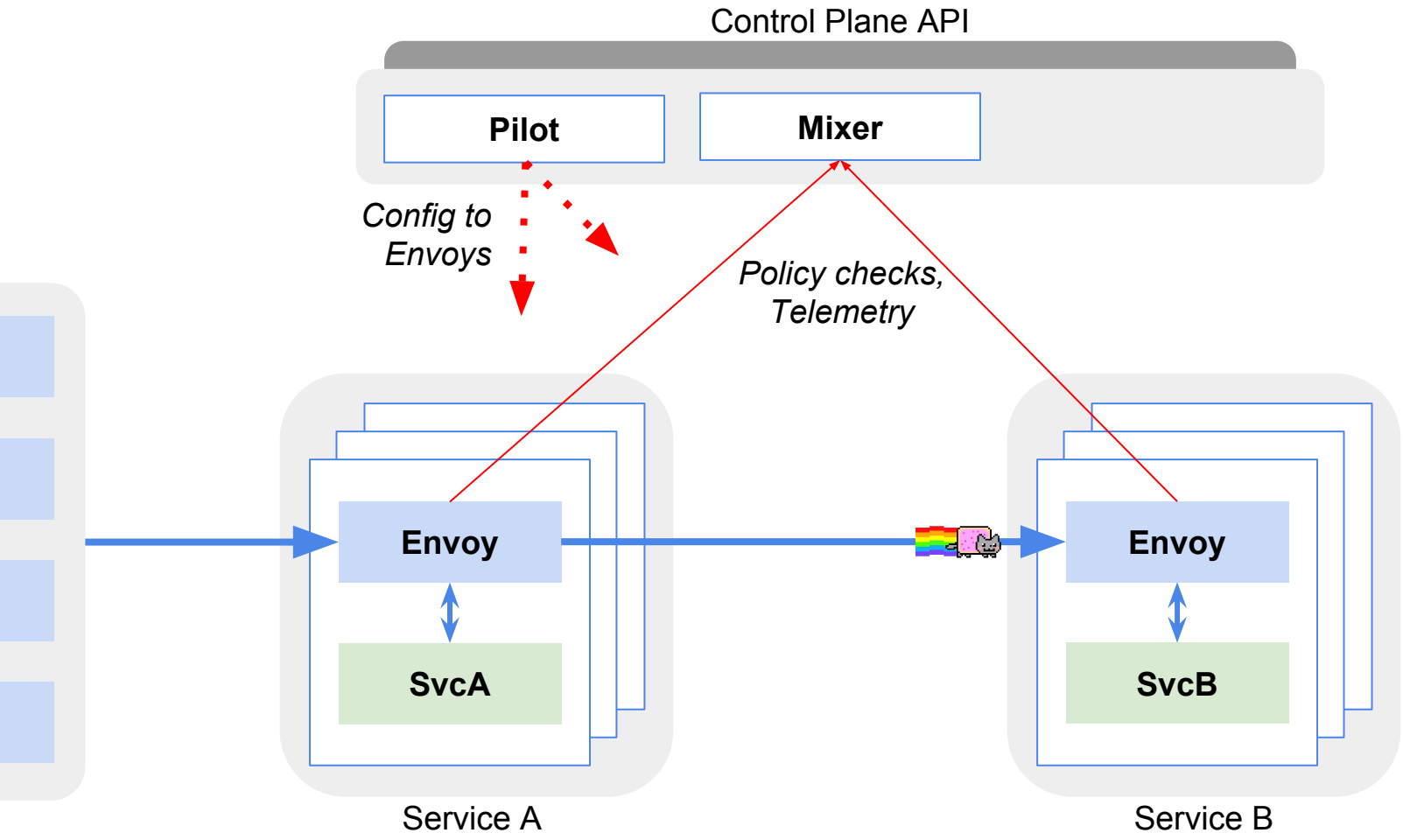
```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: httpbin
spec:
  host: httpbin
  trafficPolicy:
    outlierDetection:
      consecutiveErrors: 1
      interval: 1s
      baseEjectionTime: 3m
      maxEjectionPercent: 100
```

Fault Injection

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - route:
    - destination:
        host: ratings
        subset: v1
      fault:
        delay:
          percent: 100
          fixedDelay: 7s
```

Mixer

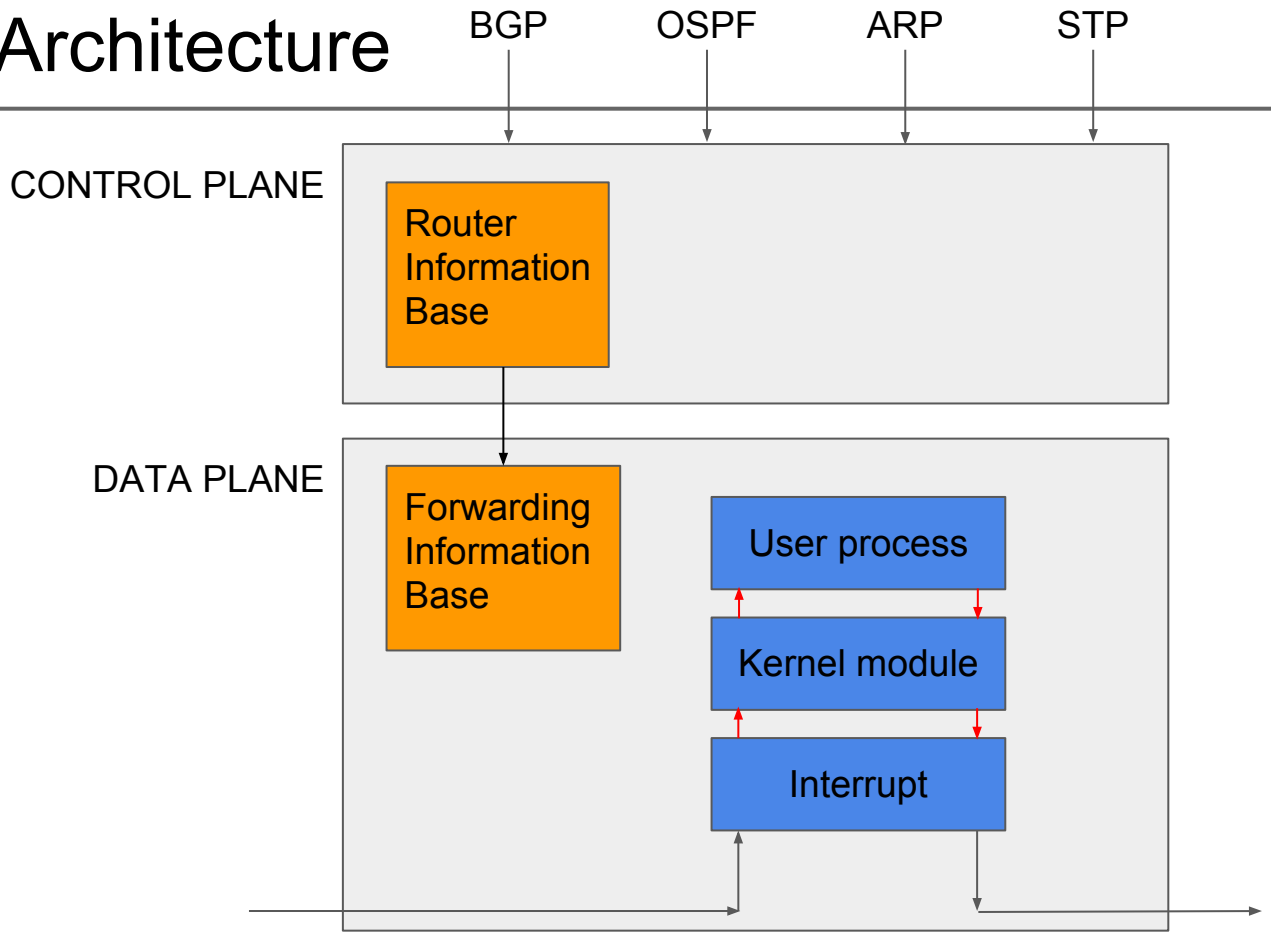

Control Plane API
Pilot
Config to Envoys
Envoy
SvcA
Service A
Service B


Control Plane API
Pilot
Mixer
Config to Envoys
Policy checks, Telemetry
Envoy
SvcA
Envoy
SvcB
Service A
Service B

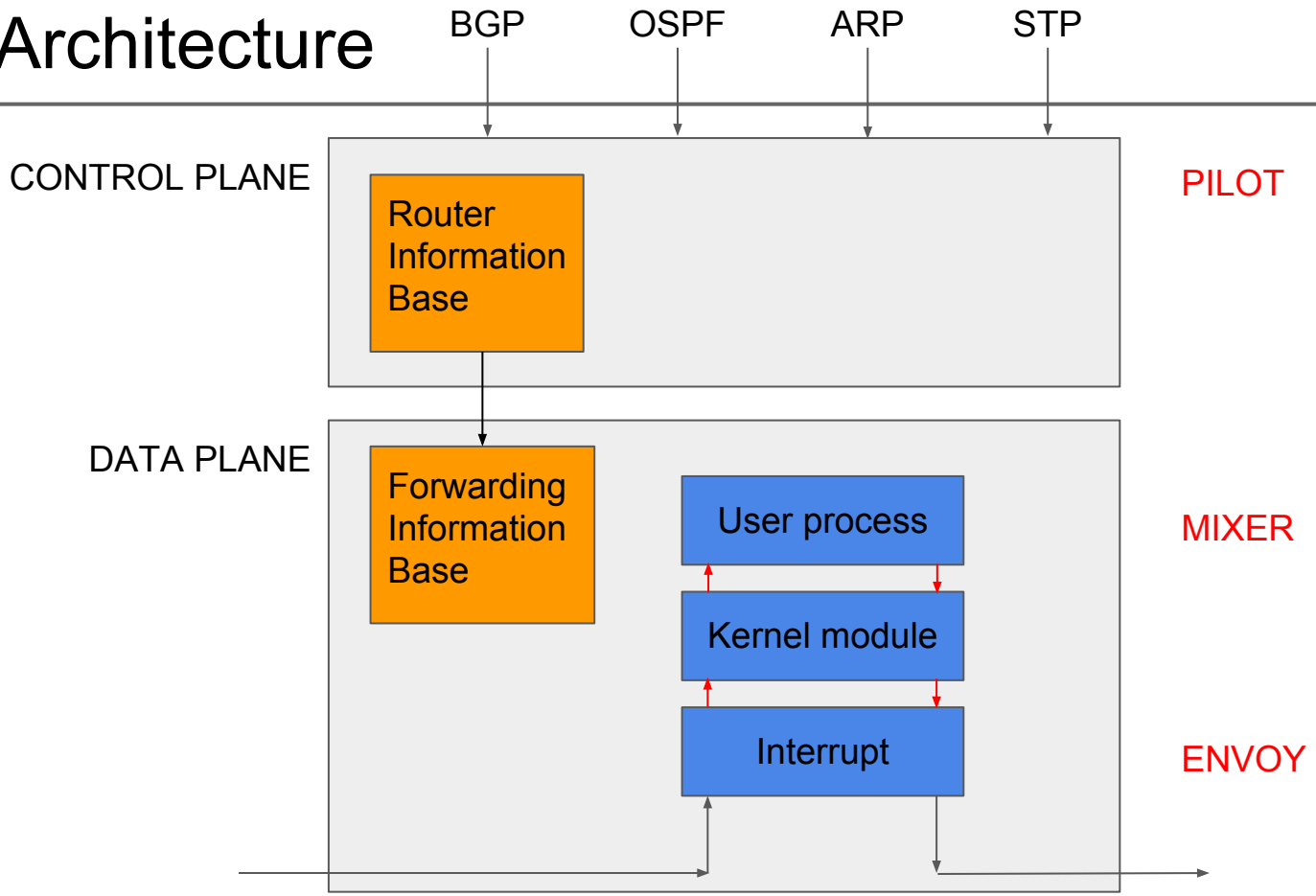
IP 5-tuple

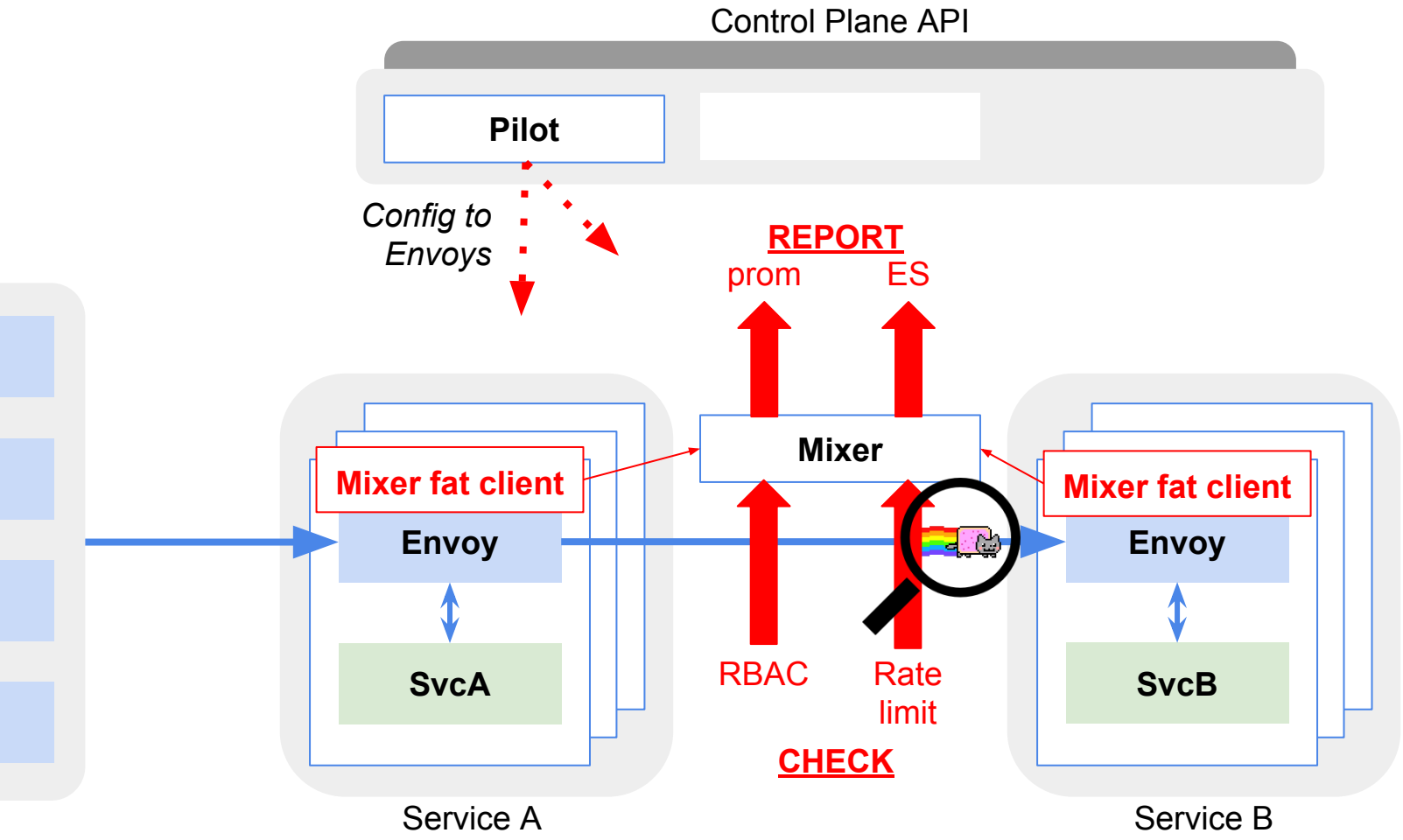
(src_addr, src_port, dst_addr, dst_port, proto)

IP Router Architecture



IP Router Architecture




Control Plane API
Pilot
Config to Envoys
REPORT
prom
ES
Mixer
Mixer fat client
Envoy
SvcA
Mixer fat client
Envoy
SvcB
RBAC
Rate limit
CHECK
Service A
Service B

Demo: Tracing

Demo: Metrics

Demo: Service Graph

Logs

```
apiVersion: "config.istio.io/v1alpha2"
kind: logentry
metadata:
  name: newlog
  namespace: istio-system
spec:
  severity: '"info"'
  timestamp: request.time
  variables:
    source: source.labels["app"] | source.workload.name | "unknown"
    user: source.user | "unknown"
    destination: destination.labels["app"] | destination.workload.name | "unknown"
    responseCode: response.code | 0
    responseSize: response.size | 0
    latency: response.duration | "0ms"
  monitored_resource_type: '"UNSPECIFIED"'
```

Logs

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: rule
metadata:
  name: newlogtofluentd
  namespace: istio-system
spec:
  match: "true" # match for all requests
  actions:
   - handler: handler.fluentd
     instances:
     - newlog.logentry
```

Logs

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: fluentd
metadata:
  name: handler
  namespace: istio-system
spec:
  address: "fluentd-es.logging:24224"
```

ACLs / Authorization

```yaml
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRole
metadata:
  name: details-reviews-viewer
  namespace: default
spec:
  rules:
  - services: ["details.default.svc.cluster.local", "reviews.default.svc.cluster.local"]
    methods: ["GET"]
```

ACLs / Authorization

```yaml
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRoleBinding
metadata:
  name: bind-details-reviews
  namespace: default
spec:
  subjects:
  - user: "cluster.local/ns/default/sa/bookinfo-productpage"
  roleRef:
    kind: ServiceRole
    name: "details-reviews-viewer"
```

Rate Limiting

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: memquota
metadata:
  name: handler
  namespace: istio-system
spec:
  quotas:
  - name: requestcount.quota.istio-system
    maxAmount: 500
    validDuration: 1s
```

Rate Limiting

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: quota
metadata:
  name: requestcount
  namespace: istio-system
spec:
  dimensions:
    source: request.headers["x-forwarded-for"] | "unknown"
    destination: destination.labels["app"] | destination.workload.name | "unknown"
    destinationVersion: destination.labels["version"] | "unknown"
```

Rate Limiting

```yaml
apiVersion: config.istio.io/v1alpha2
kind: rule
metadata:
  name: quota
  namespace: istio-system
spec:
  actions:
  - handler: handler.memquota
    instances:
    - requestcount.quota
```

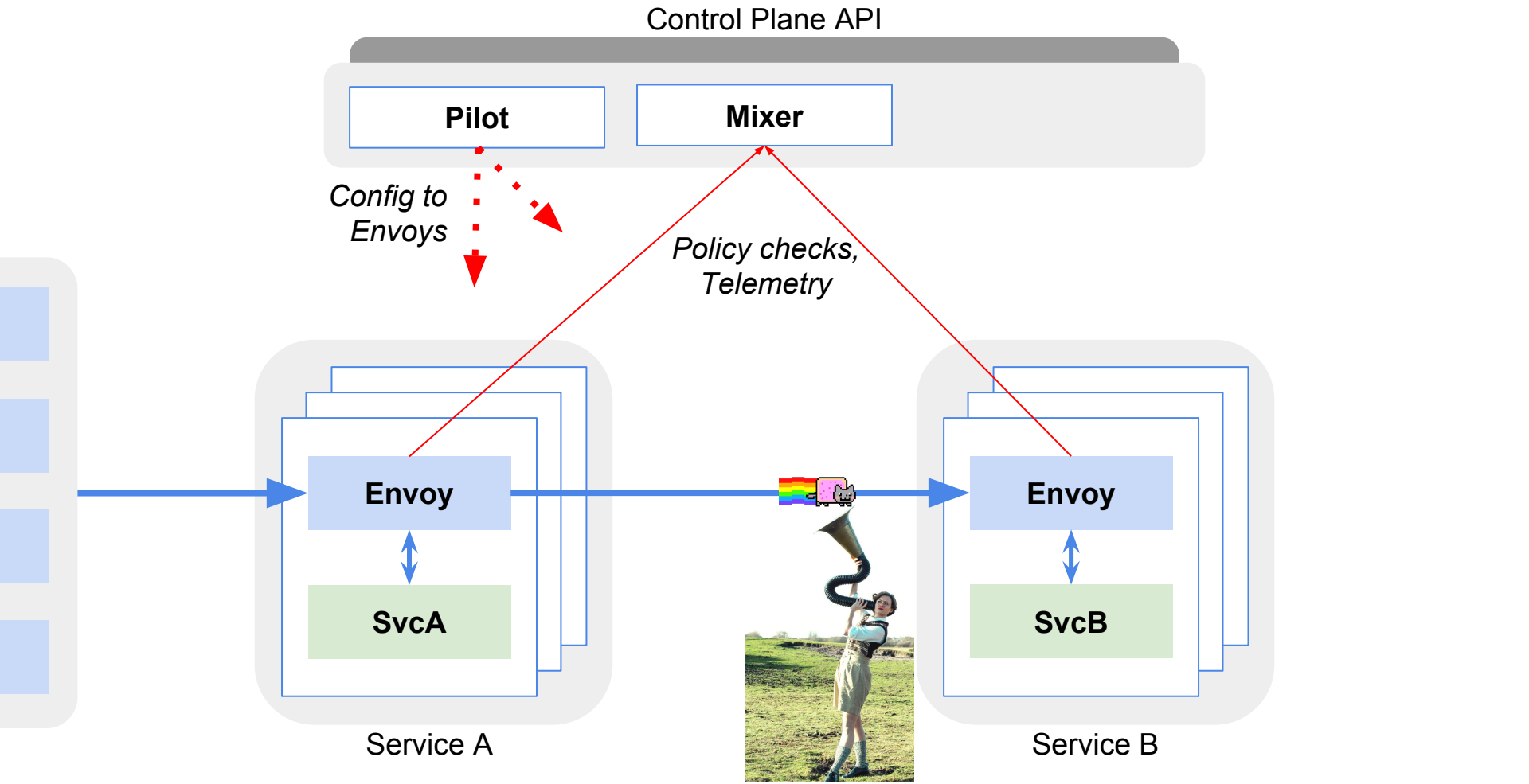
Rate Limiting

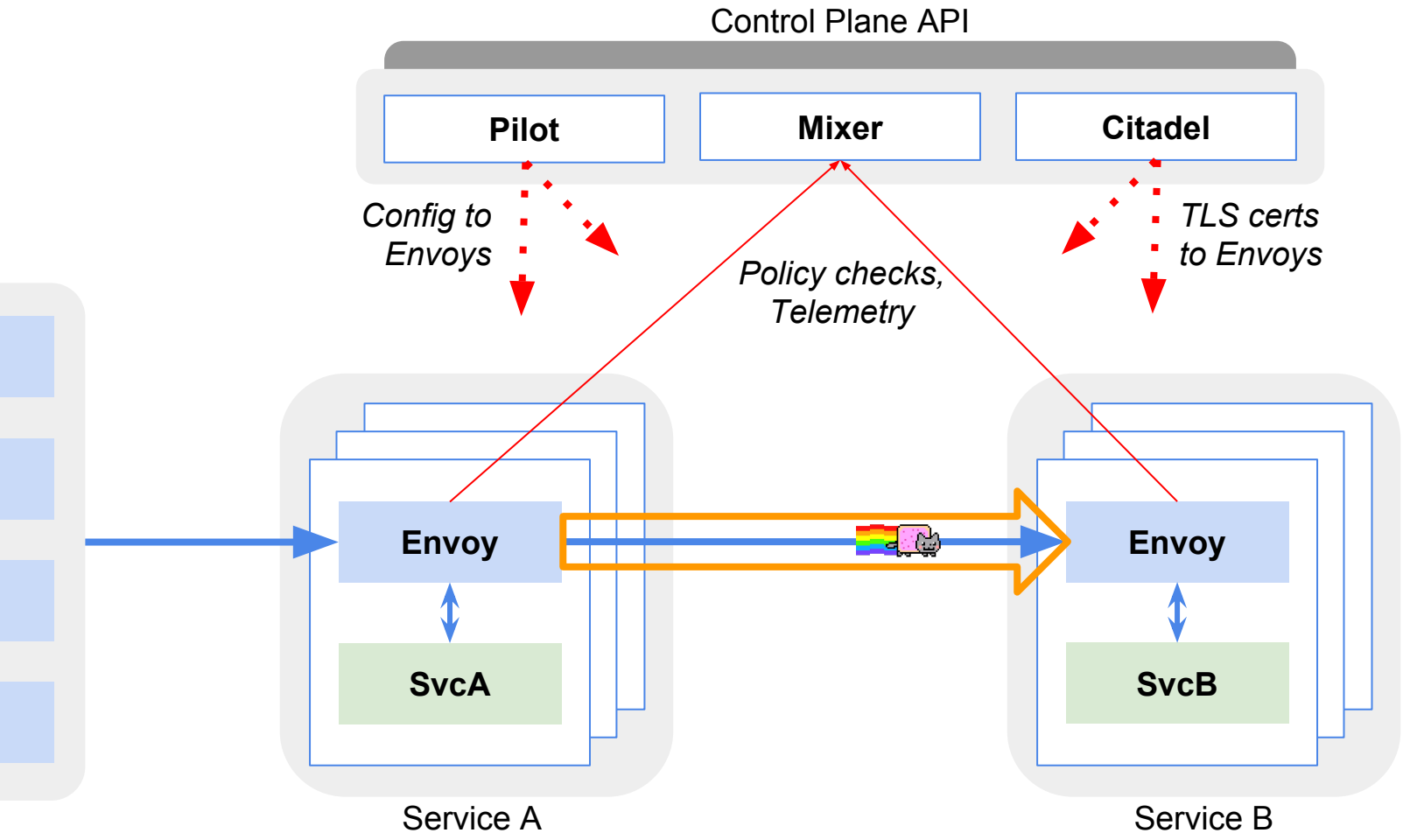
```yaml
apiVersion: config.istio.io/v1alpha2
kind: QuotaSpec
metadata:
  name: request-count
  namespace: istio-system
spec:
  rules:
  - quotas:
    - charge: 1
      quota: requestcount
```

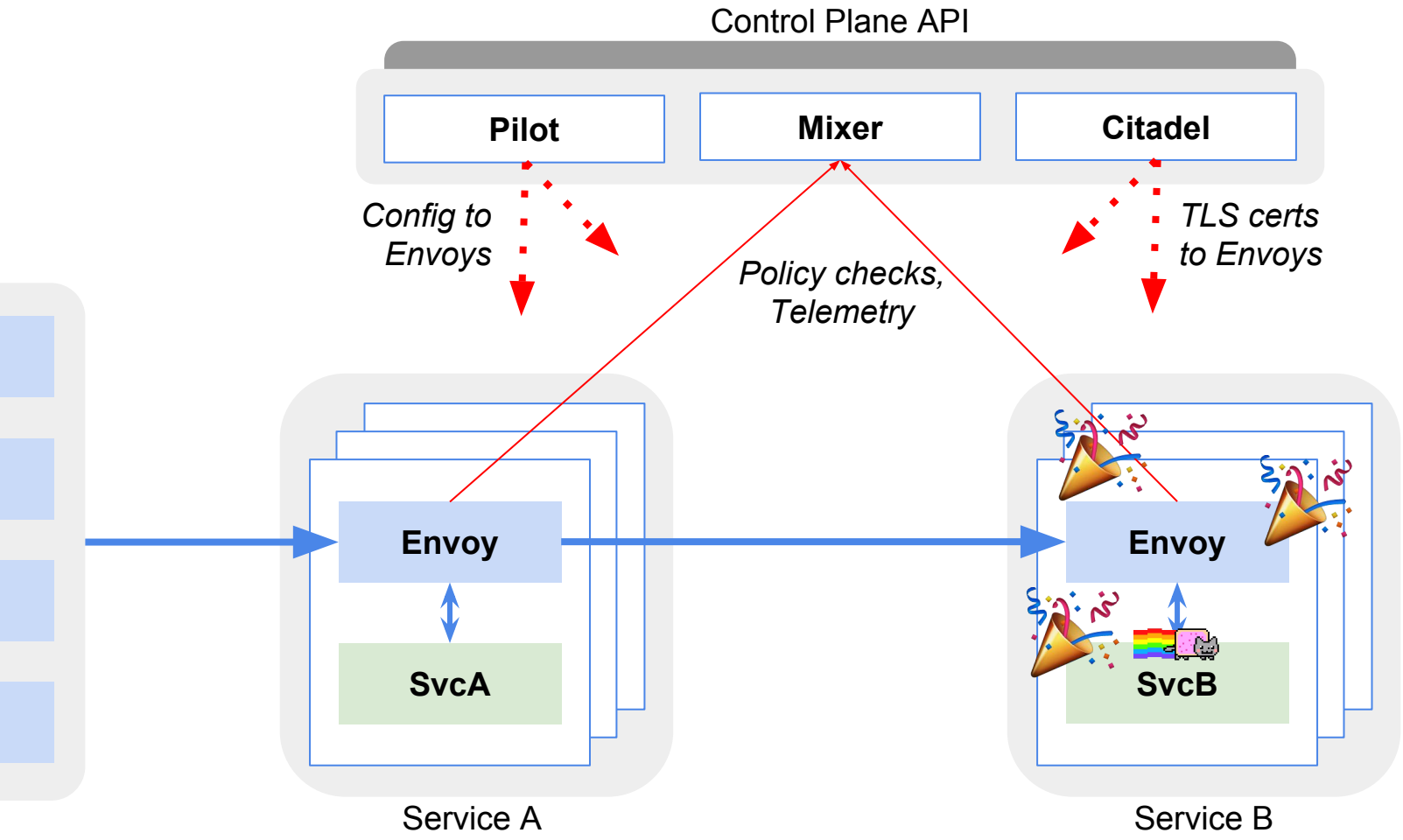
Rate Limiting

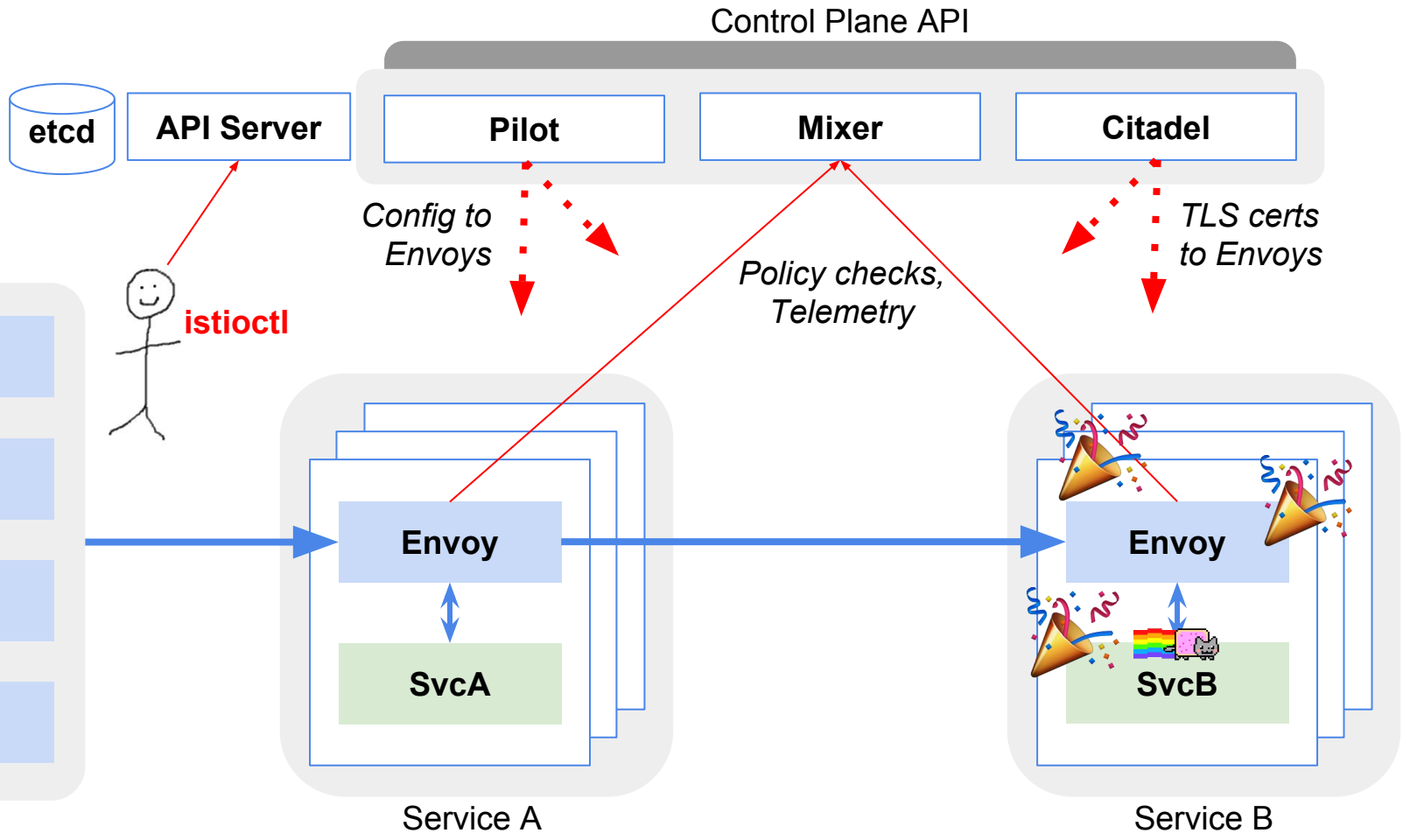
```yaml
apiVersion: config.istio.io/v1alpha2
kind: QuotaSpecBinding
metadata:
  name: request-count
  namespace: istio-system
spec:
  quotaSpecs:
  - name: request-count
    namespace: istio-system
  services:
  - name: productpage
    namespace: default
```

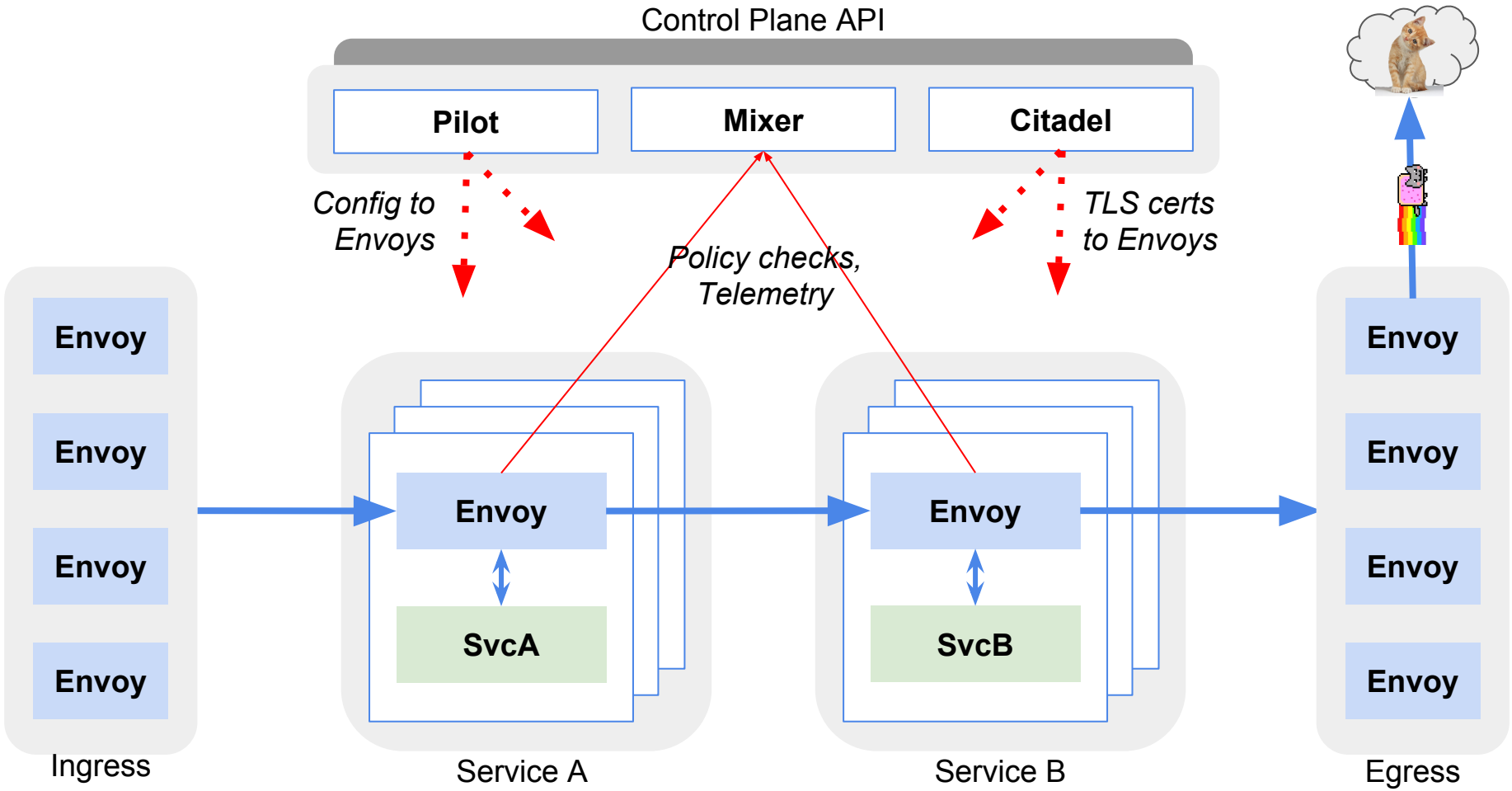
Citadel, mTLS, Egress


Control Plane API
Pilot
Mixer
Config to Envoys
Policy checks, Telemetry
Envoy
SvcA
Envoy
SvcB
Service A
Service B


Control Plane API
Pilot
Mixer
Citadel
Config to Envoys
Policy checks, Telemetry
TLS certs to Envoys
Envoy
SvcA
Envoy
SvcB
Service A
Service B


Control Plane API
Pilot
Mixer
Citadel
Config to Envoys
Policy checks, Telemetry
TLS certs to Envoys
Envoy
SvcA
Envoy
SvcB
Service A
Service B


Control Plane API
etcd
API Server
Pilot
Mixer
Citadel
Config to Envoys
Policy checks, Telemetry
TLS certs to Envoys
istioctl
Envoy
SvcA
Envoy
SvcB
Service A
Service B


Control Plane API
Pilot
Mixer
Citadel
Config to Envoys
Policy checks, Telemetry
TLS certs to Envoys
Envoy
Envoy
Envoy
Envoy
Ingress
Envoy
SvcA
Service A
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress

Egress Routing

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: https-wikipedia-org
spec:
  hosts:
  - wikipedia.org
  ports:
  - number: 443
    name: https
    protocol: HTTPS
  location: MESH_EXTERNAL
  resolution: DNS
  endpoints:
  - address: istio-egressgateway.istio-system.svc.cluster.local
    ports:
      http: 443
```

Egress Routing

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: https-wikipedia-org-egress
spec:
  selector:
    istio: egressgateway
  servers:
  - port:
      number: 443
      name: https-wikipedia-org-egress-443
      protocol: TLS # Mark as TLS as we are passing HTTPS through.
    hosts:
    - wikipedia.org
    tls:
      mode: PASSTHROUGH
```

Egress Routing

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: egress-wikipedia-org
spec:
  hosts:
  - wikipedia.org
  gateways:
  - https-wikipedia-org-egress
  tls:
  - match:
    - ports: 443
      sniHosts:
      - wikipedia.org
    route:
    - destination:
        host: egress-wikipedia-org
```

Egress Routing

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: egress-https-wikipedia-org
spec:
  hosts:
  - egress-wikipedia-org
  ports:
  - number: 443
    name: https
    protocol: HTTPS
  location: MESH_EXTERNAL
  resolution: DNS
  endpoints:
  - address: wikipedia.org
    ports:
      http: 443
```

Recap

We learned:

- How a packet traverses an Istio/Envoy/Kubernetes system
- What control plane calls are made in that process
- A useful mental model for reasoning about, and debugging Istio

Thanks!

@mt165