Why Is Istio That Shape?

Matt Turner

Software Architecture Gathering, Virtual | November 2022

@mt165 | mt165.co.uk

tetrate

THE ENTERPRISE SERVICE MESH COMPANY

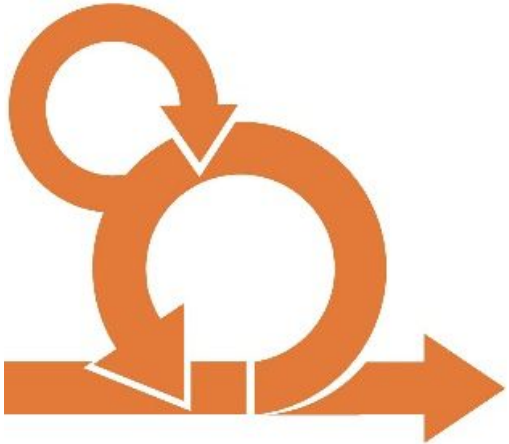
Pop Quiz!

Background

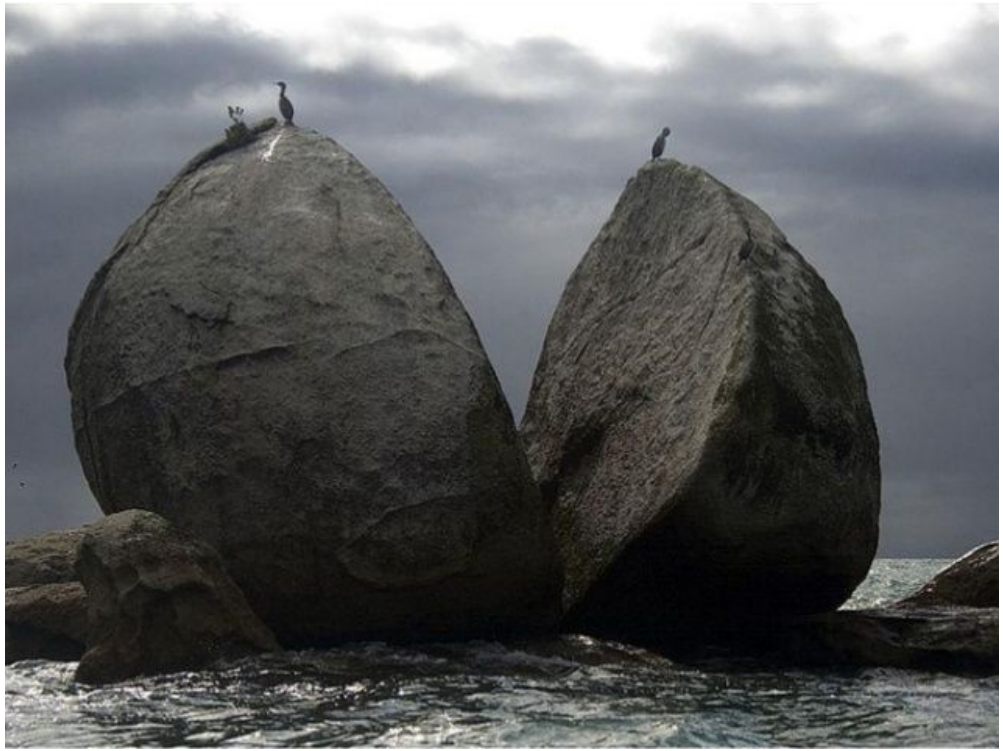
Service Meshes and Istio

Business Value

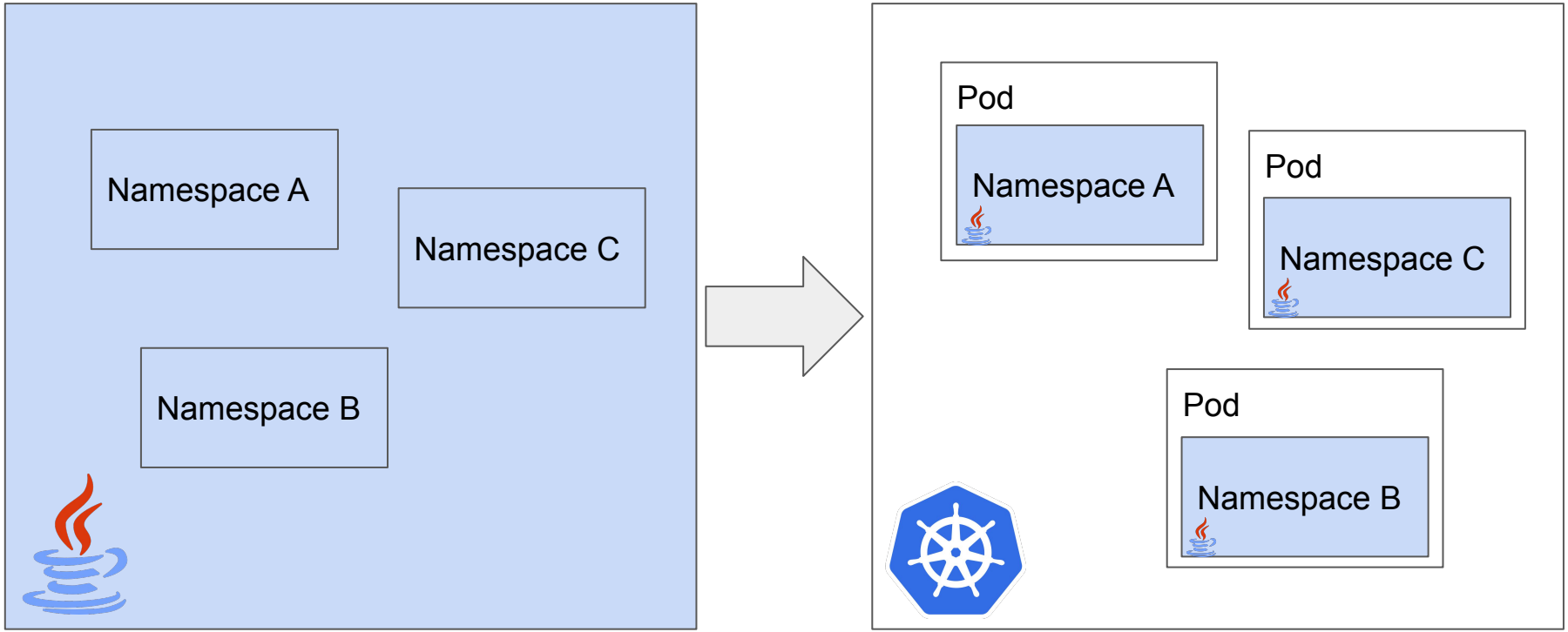
- Top Line
- Bottom Line
- Time to Market & **Speed of Iteration**
- **Risk Reduction**



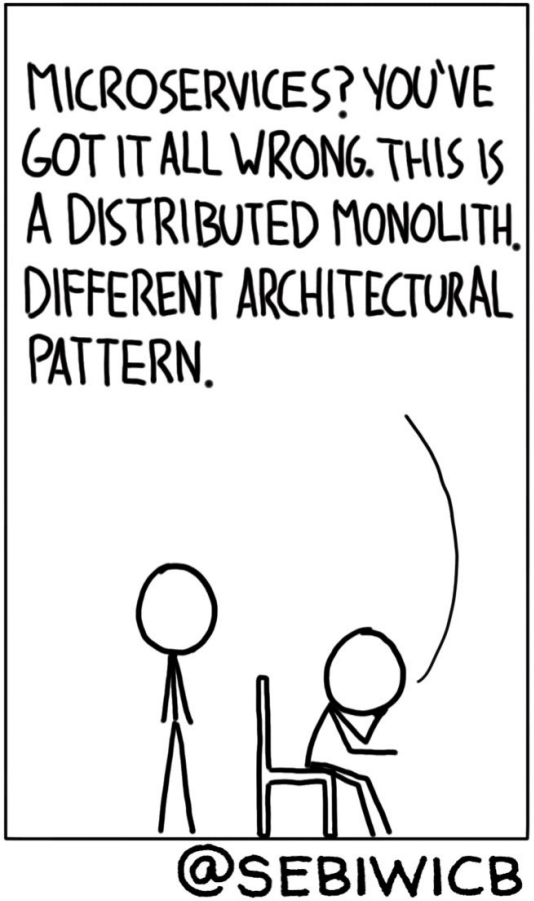
Break the Monolith



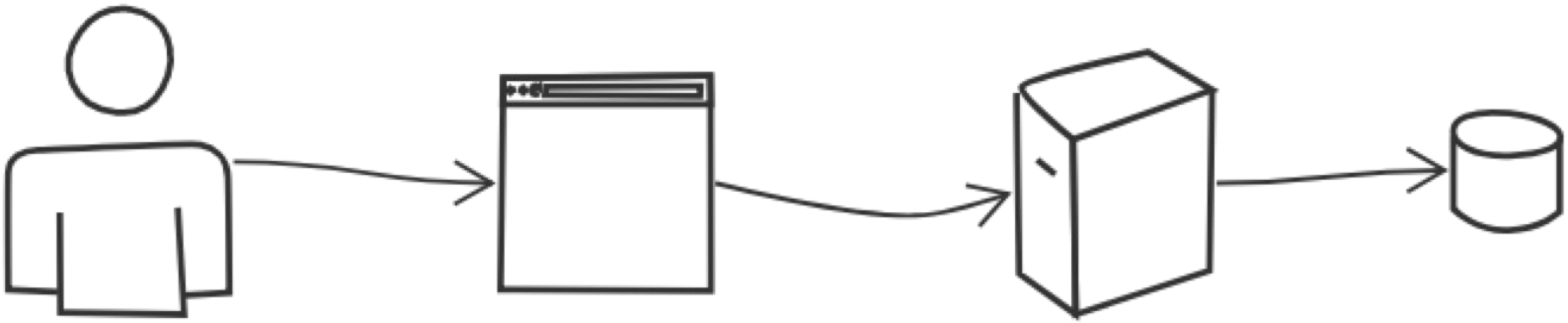
Break the Monolith

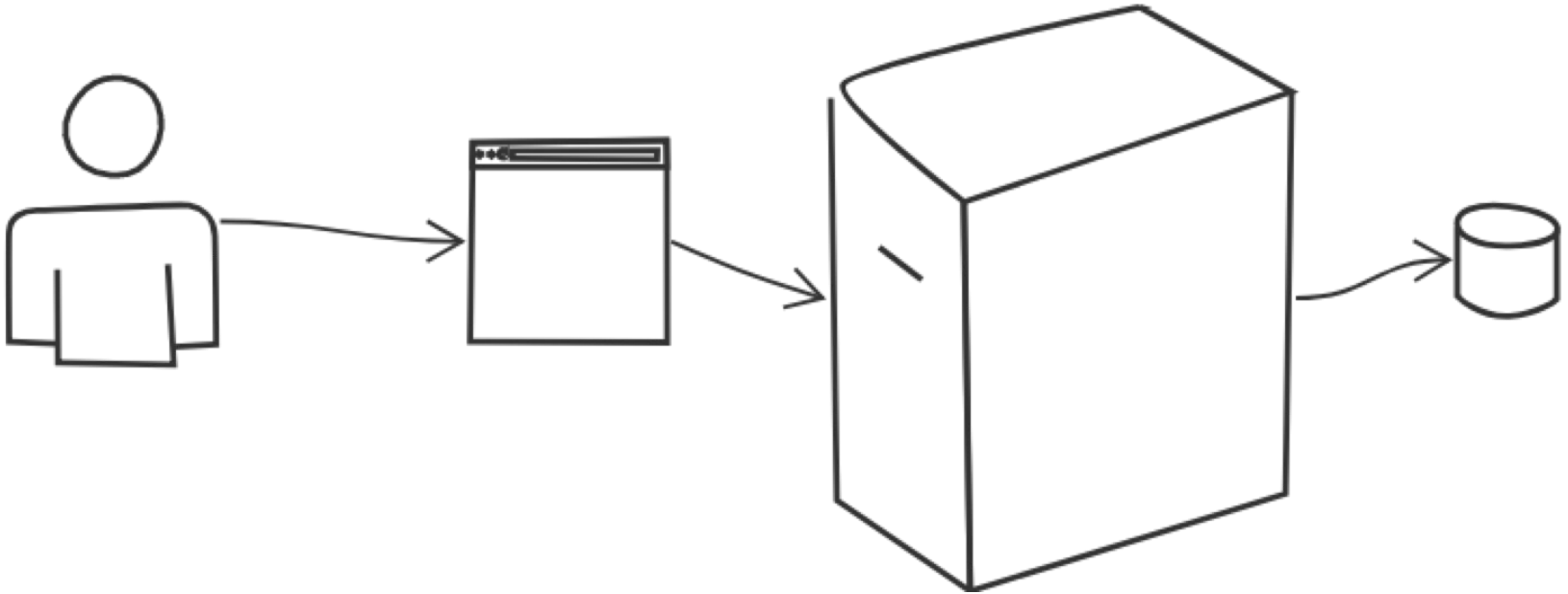


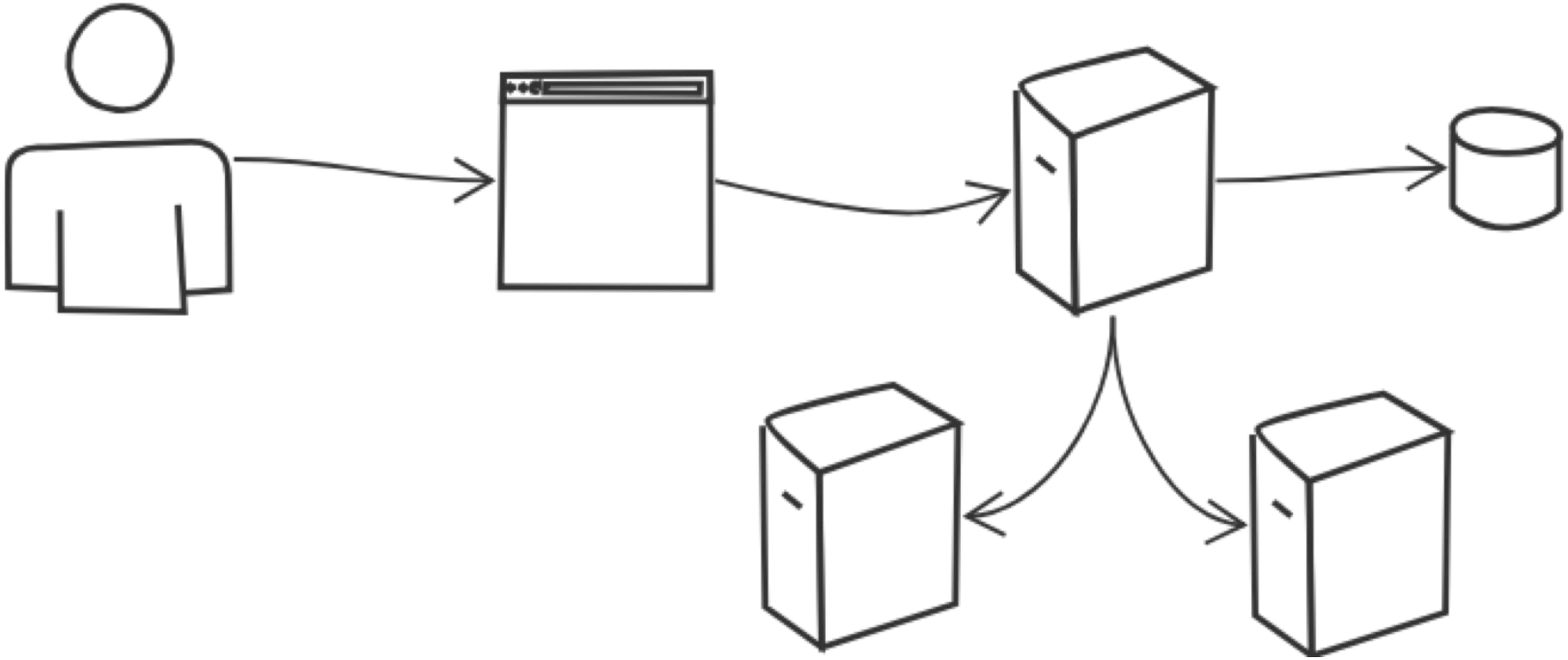
But Don’t Just Distribute It!

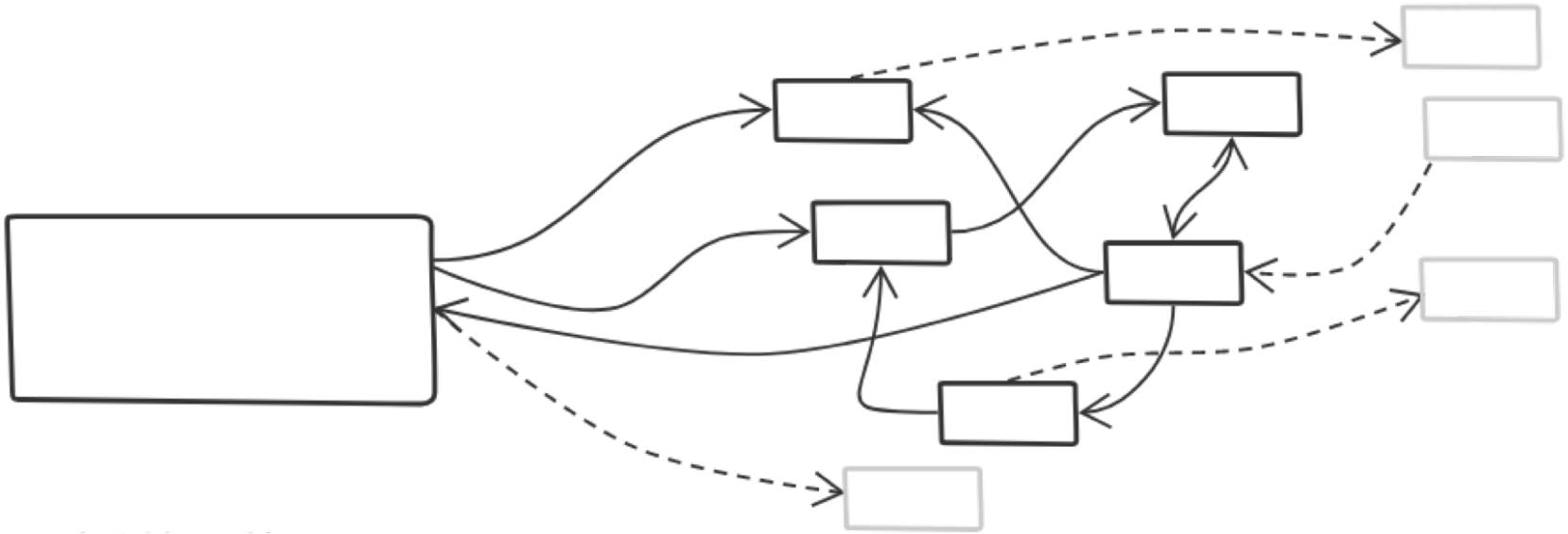


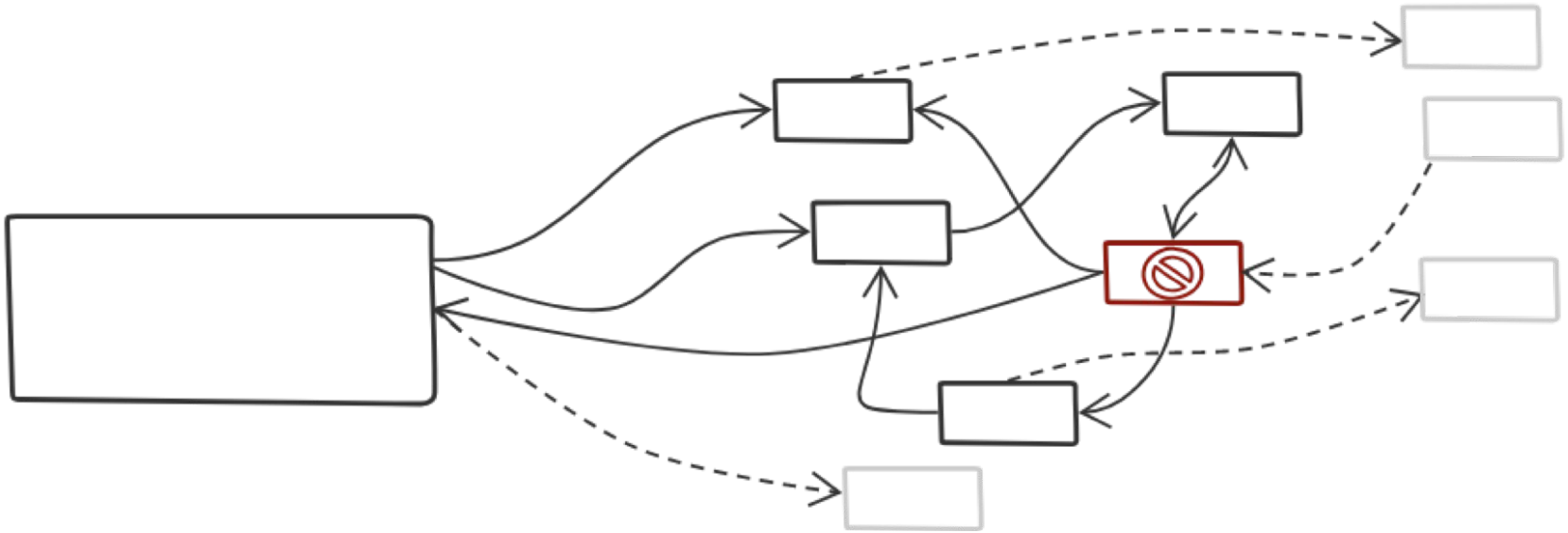
Technical Implications

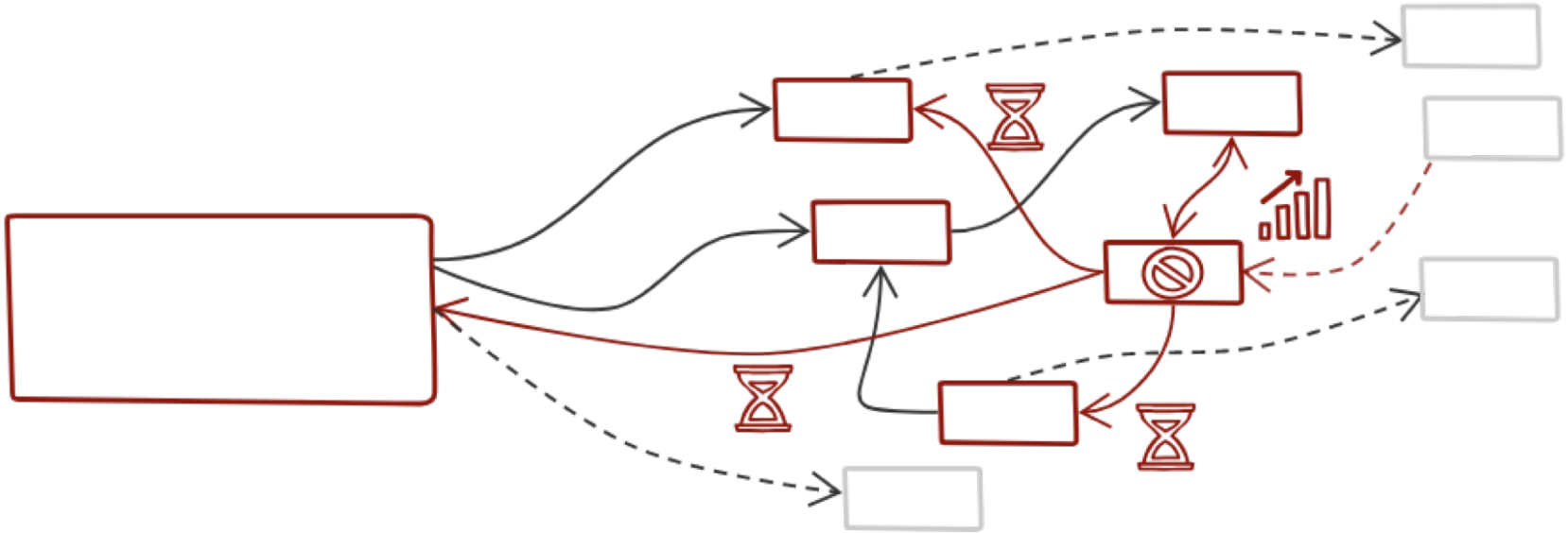


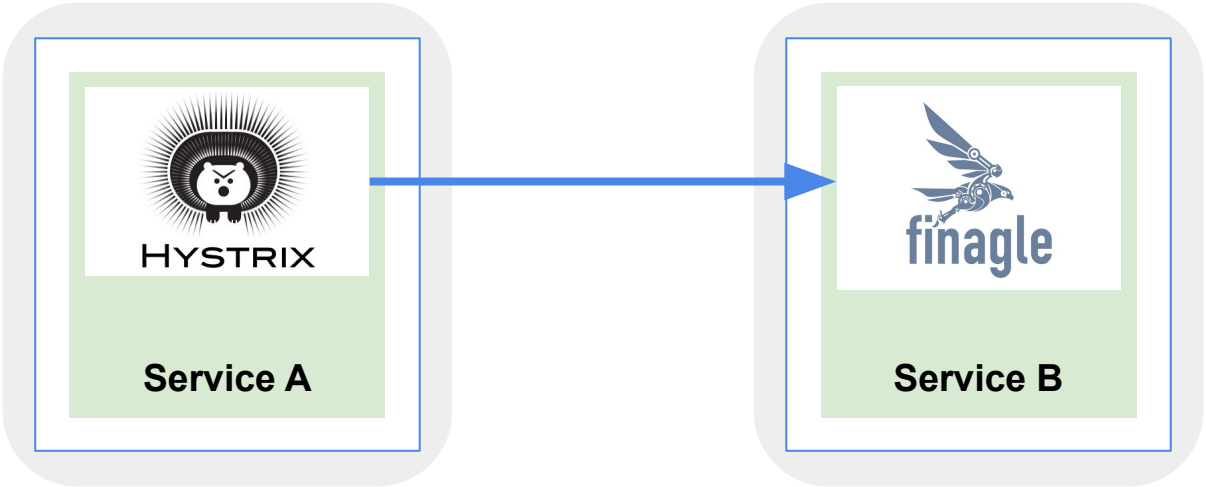






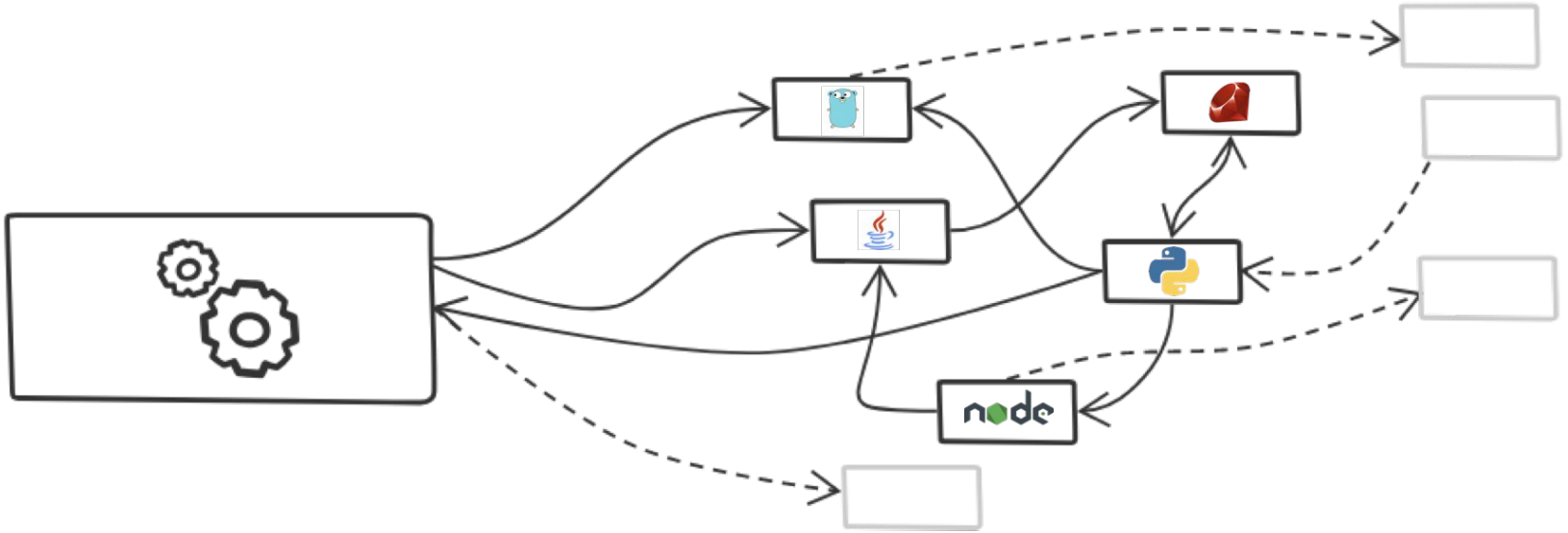




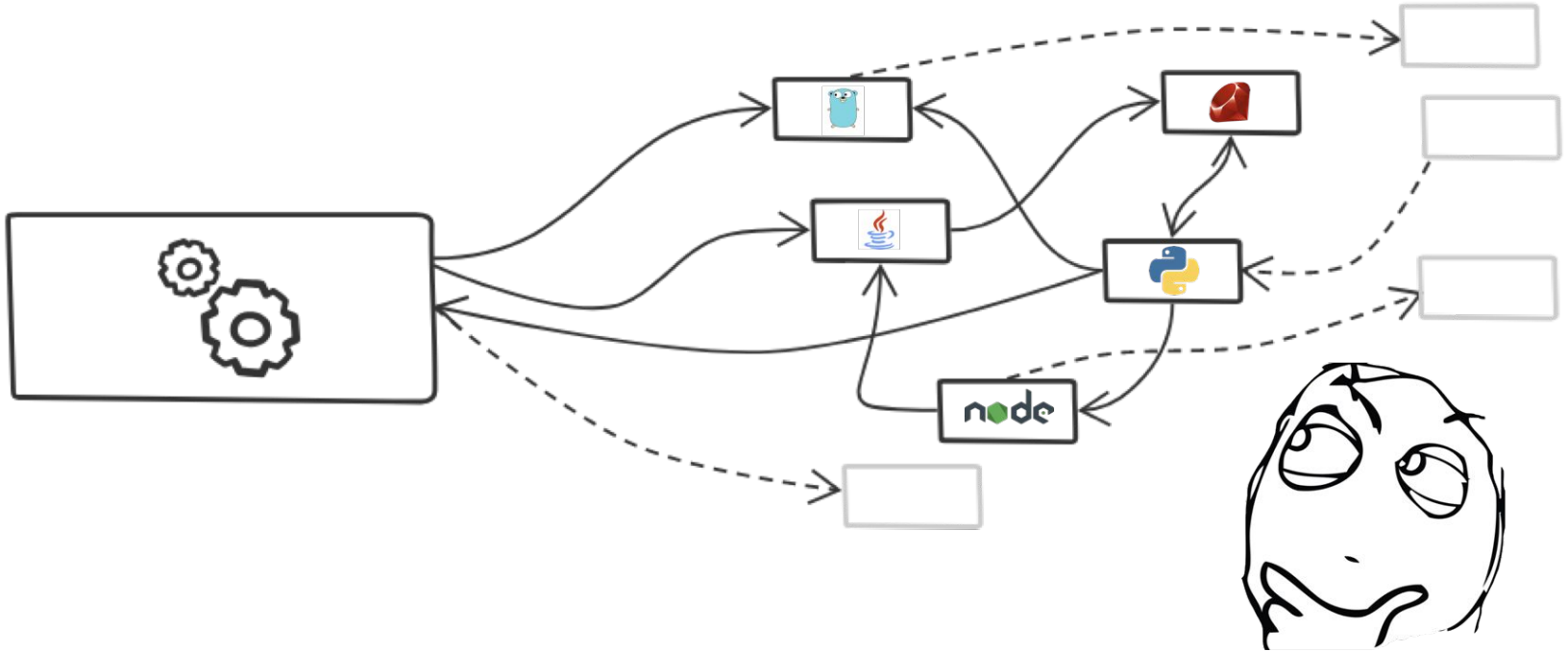

HYSTRIX
Service A
finagle
Service B

Lots of Options: In-process libraries and frameworks

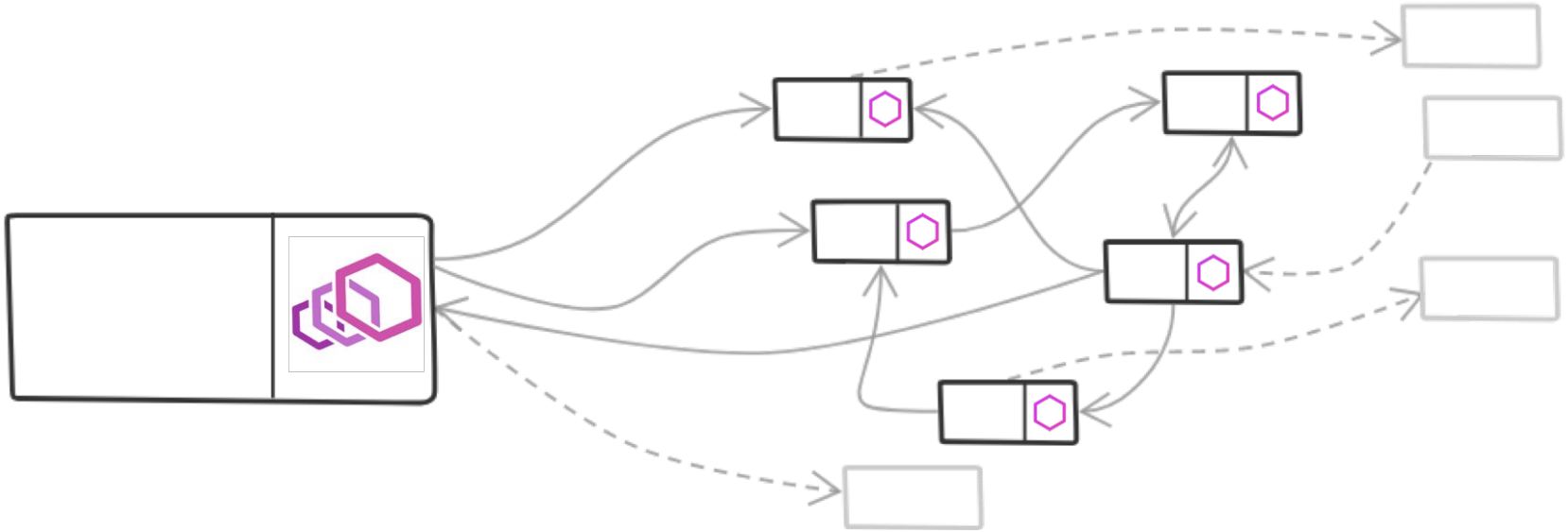
Polyglot environments



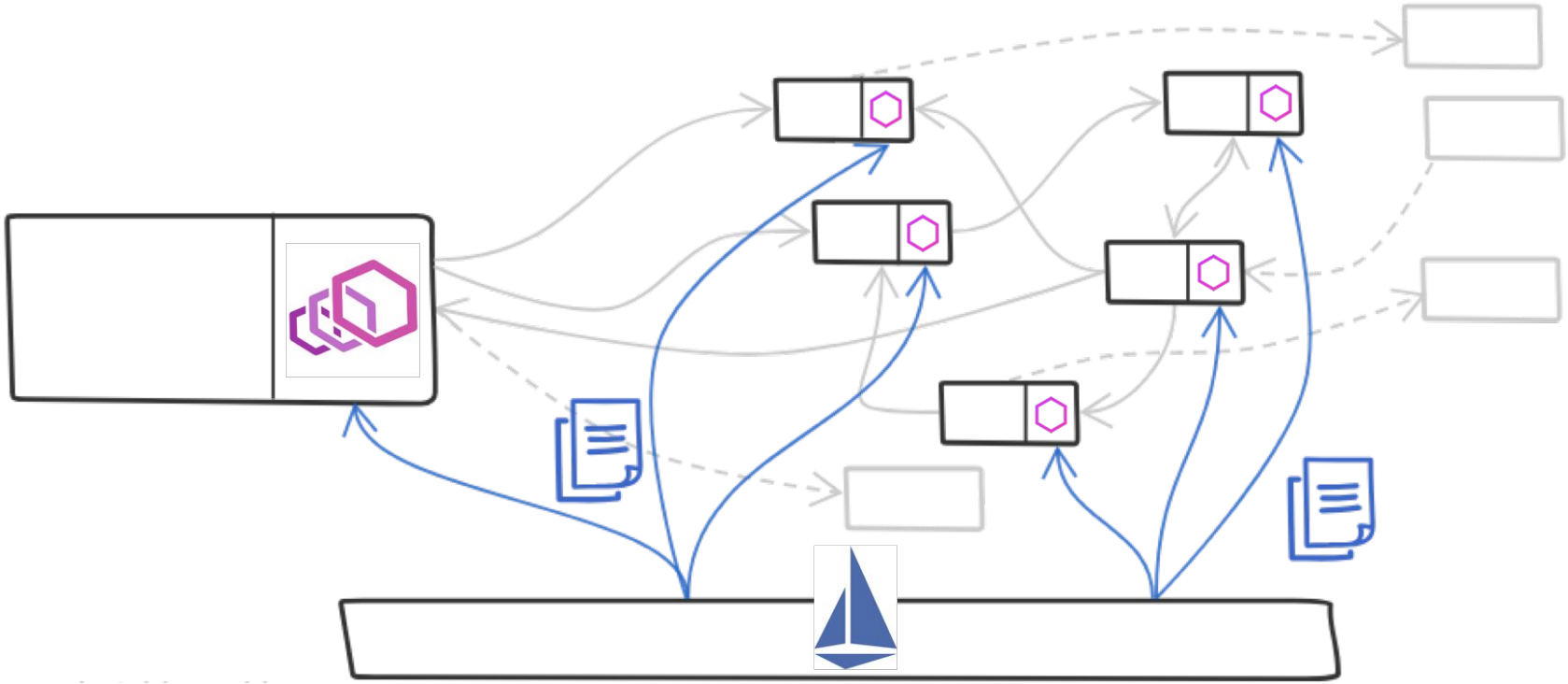
Polyglot environments



Take the Network Smarts out of the Process



Add a Control Plane

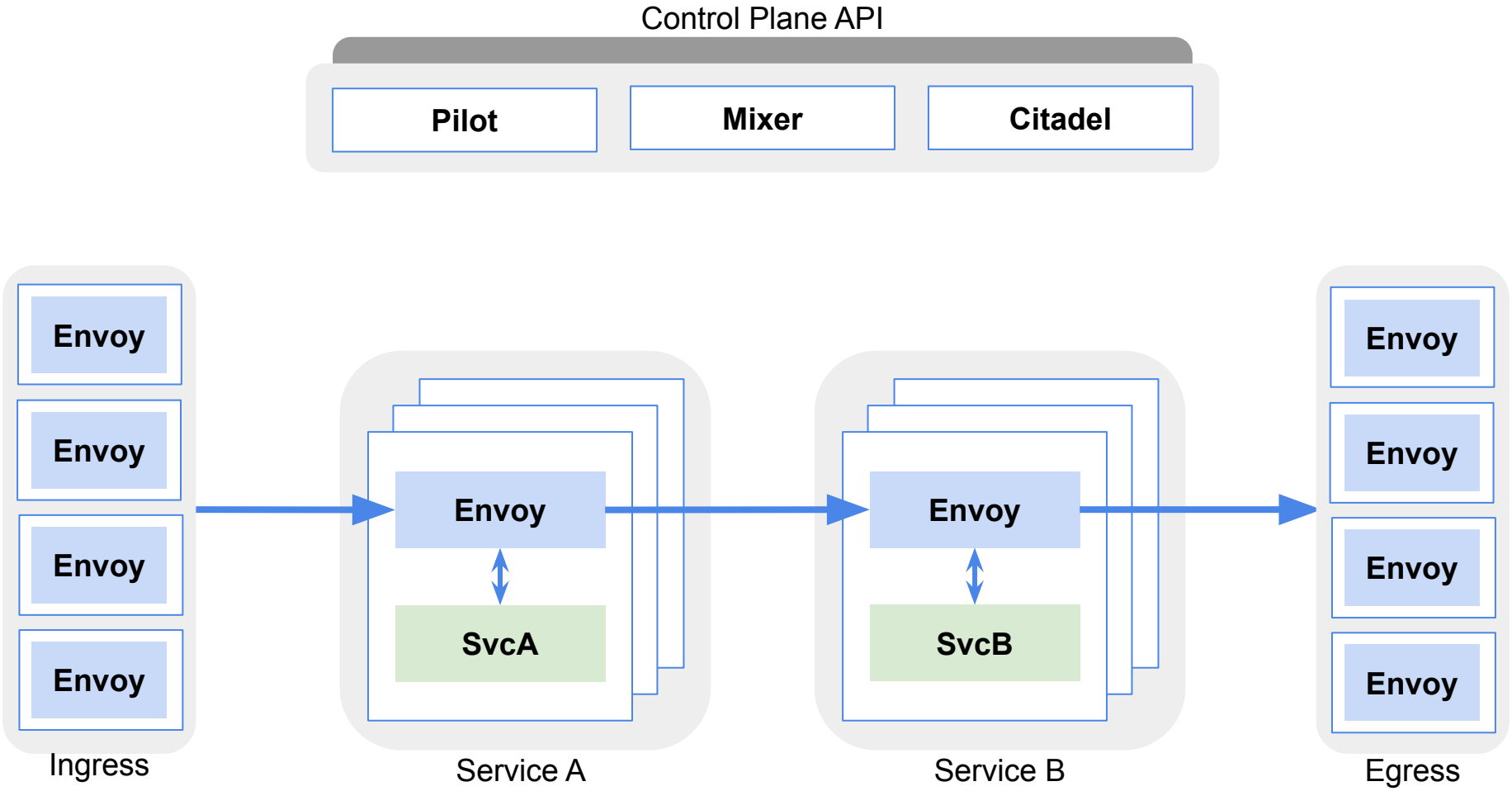


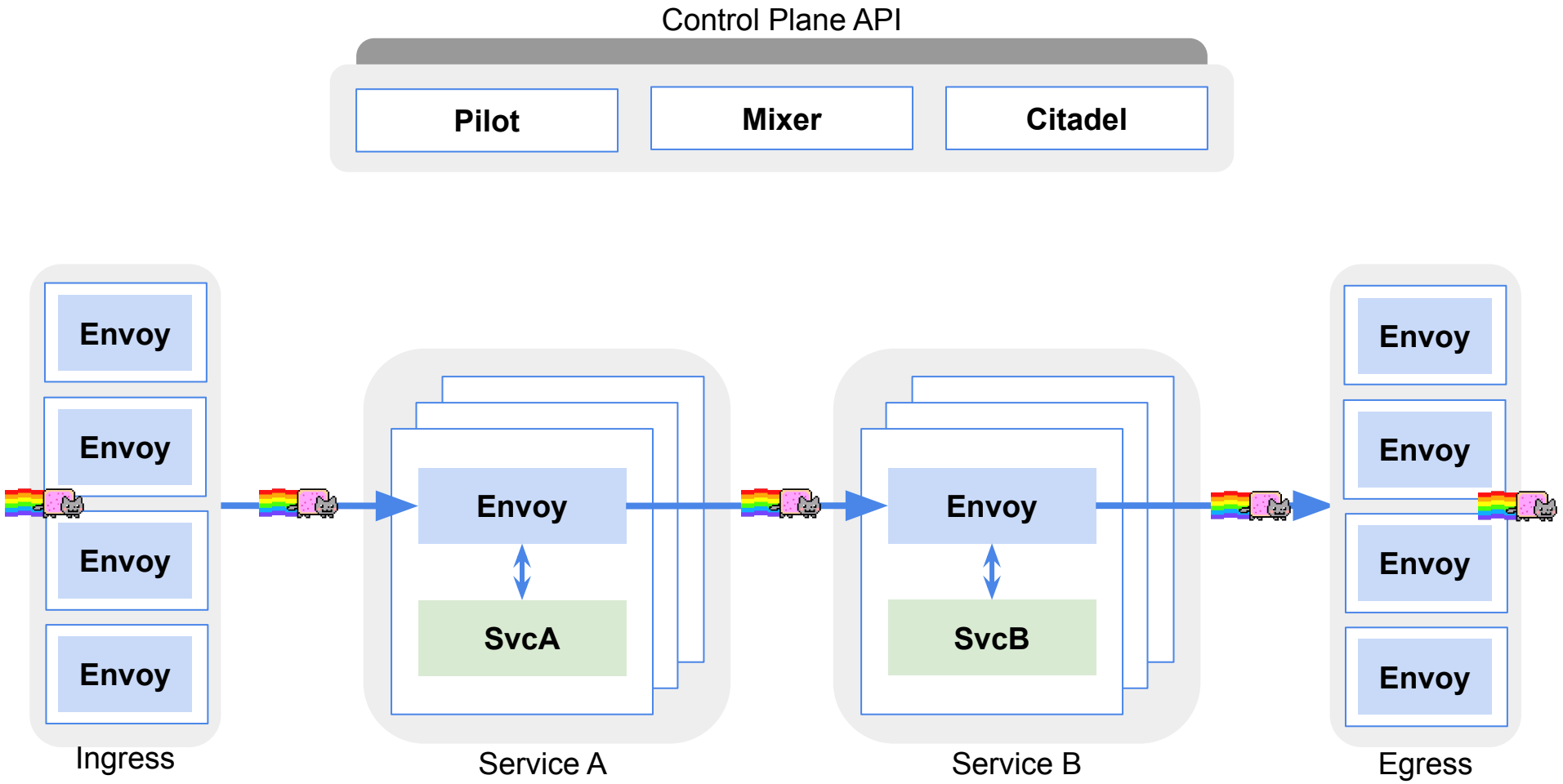
Istio

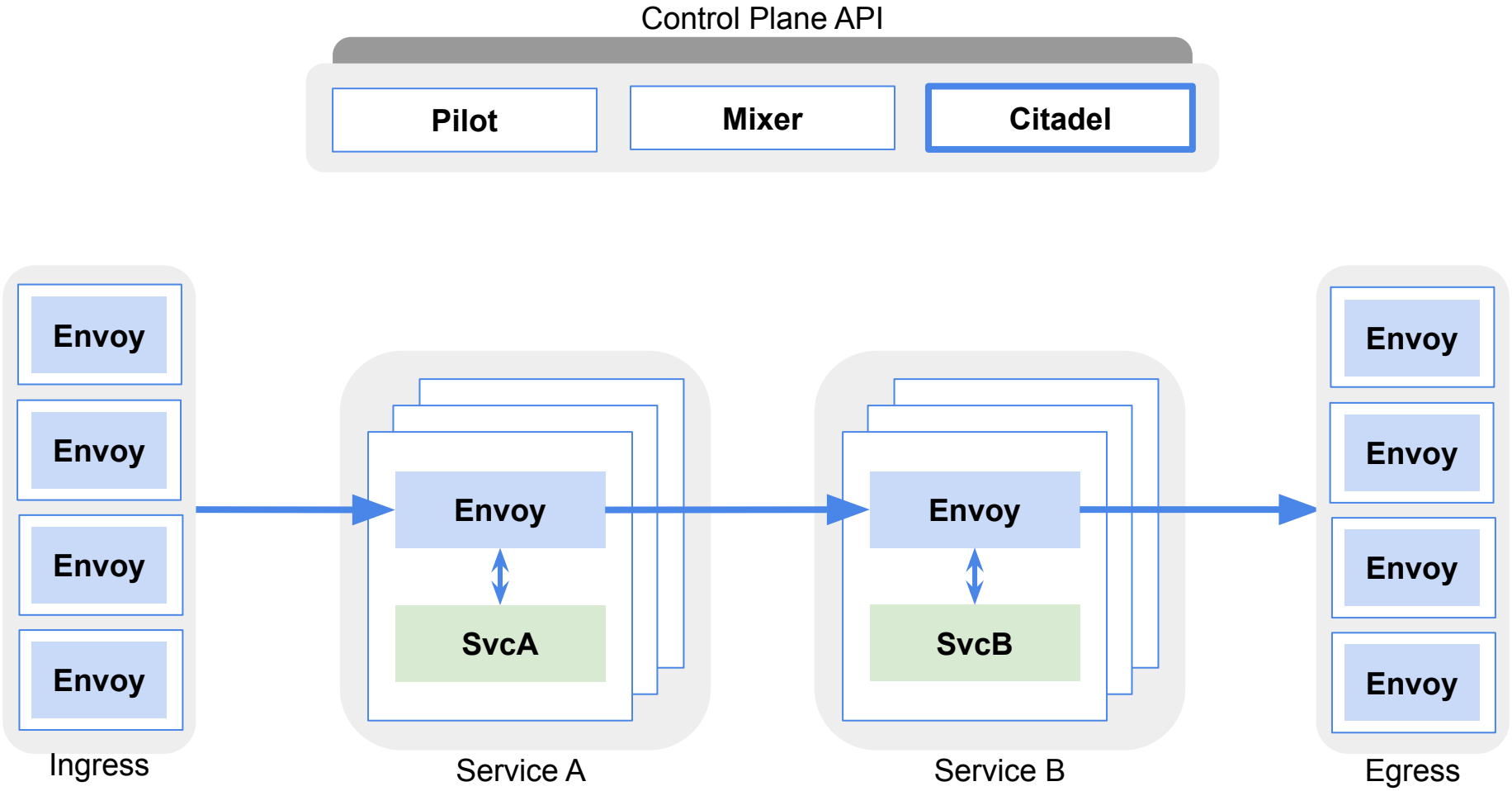
“An open platform to **connect**, **secure**, **control**, and **observe** services.”

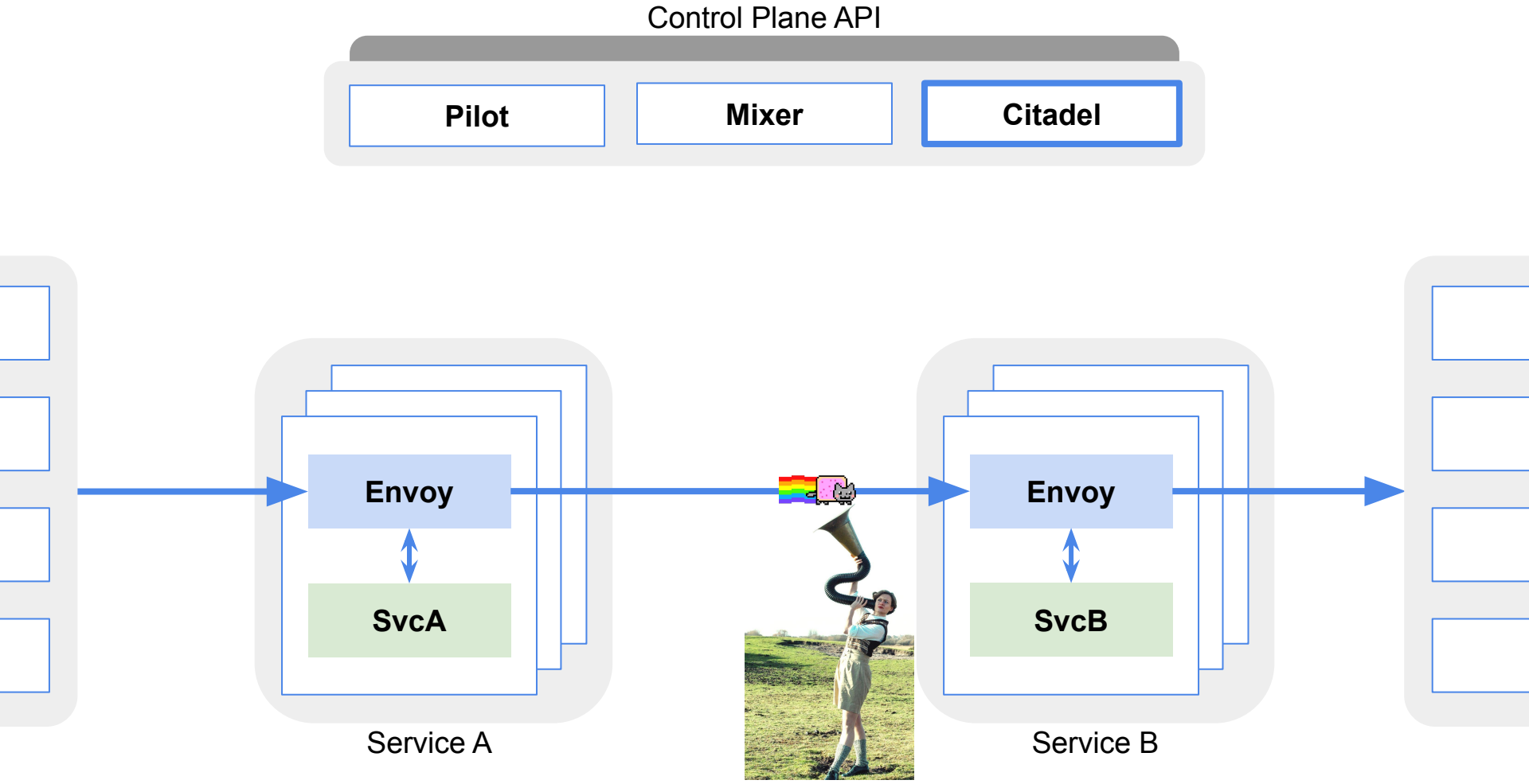
What Shape is Istio?

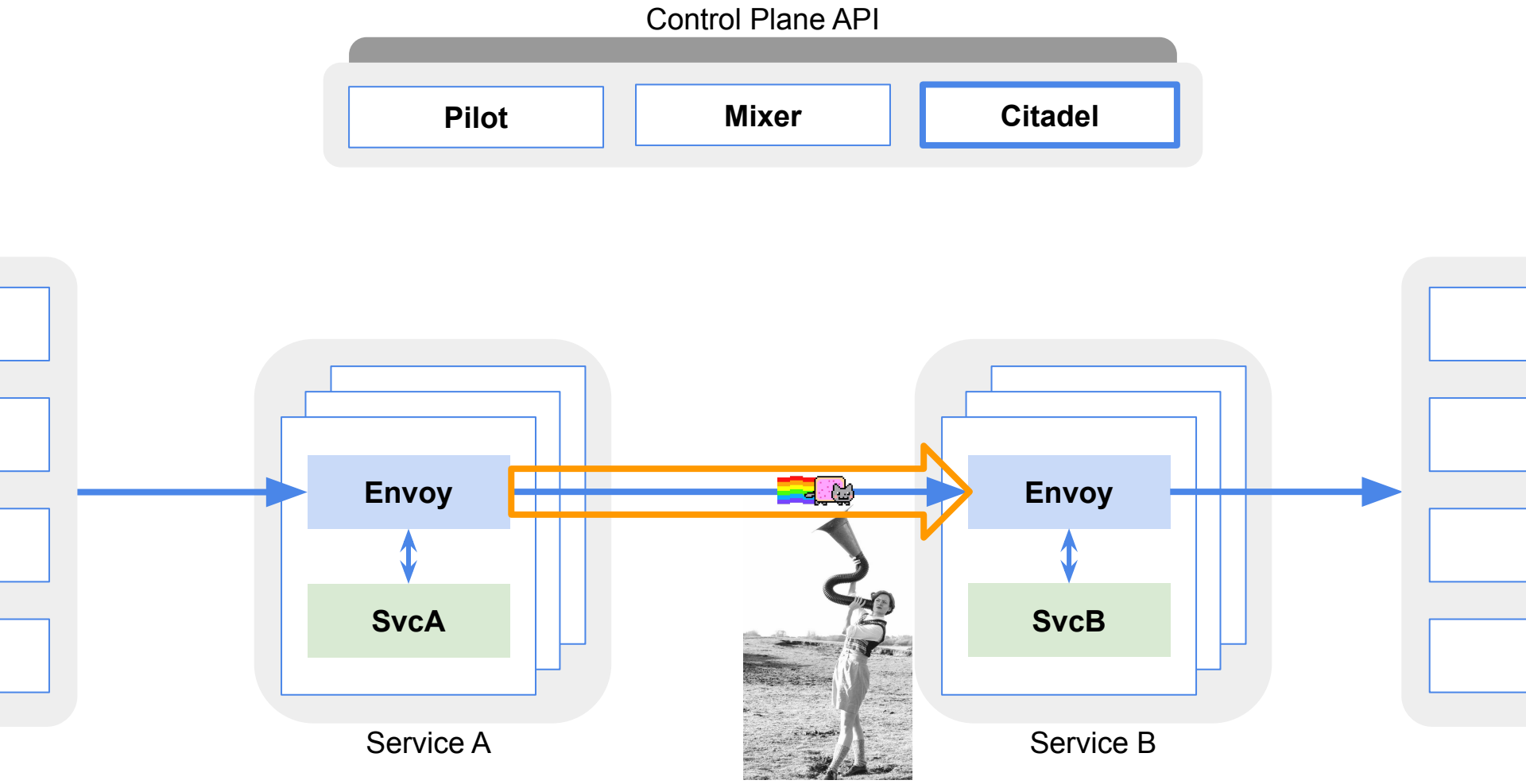
Istio 0.1 - 1.4

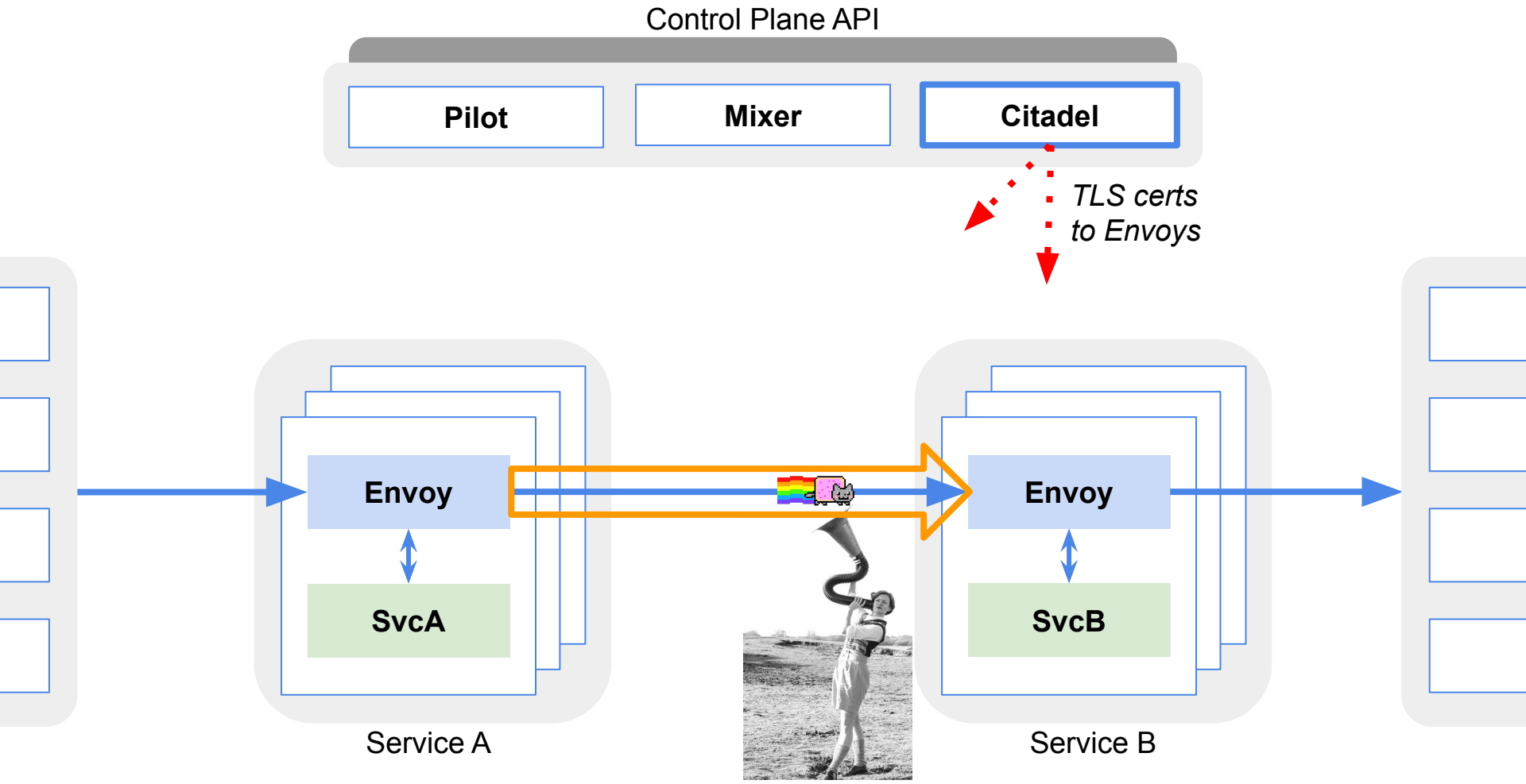

Control Plane API
Pilot
Mixer
Citadel
Envoy
Envoy
Envoy
Envoy
Ingress
Envoy
SvcA
Service A
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress

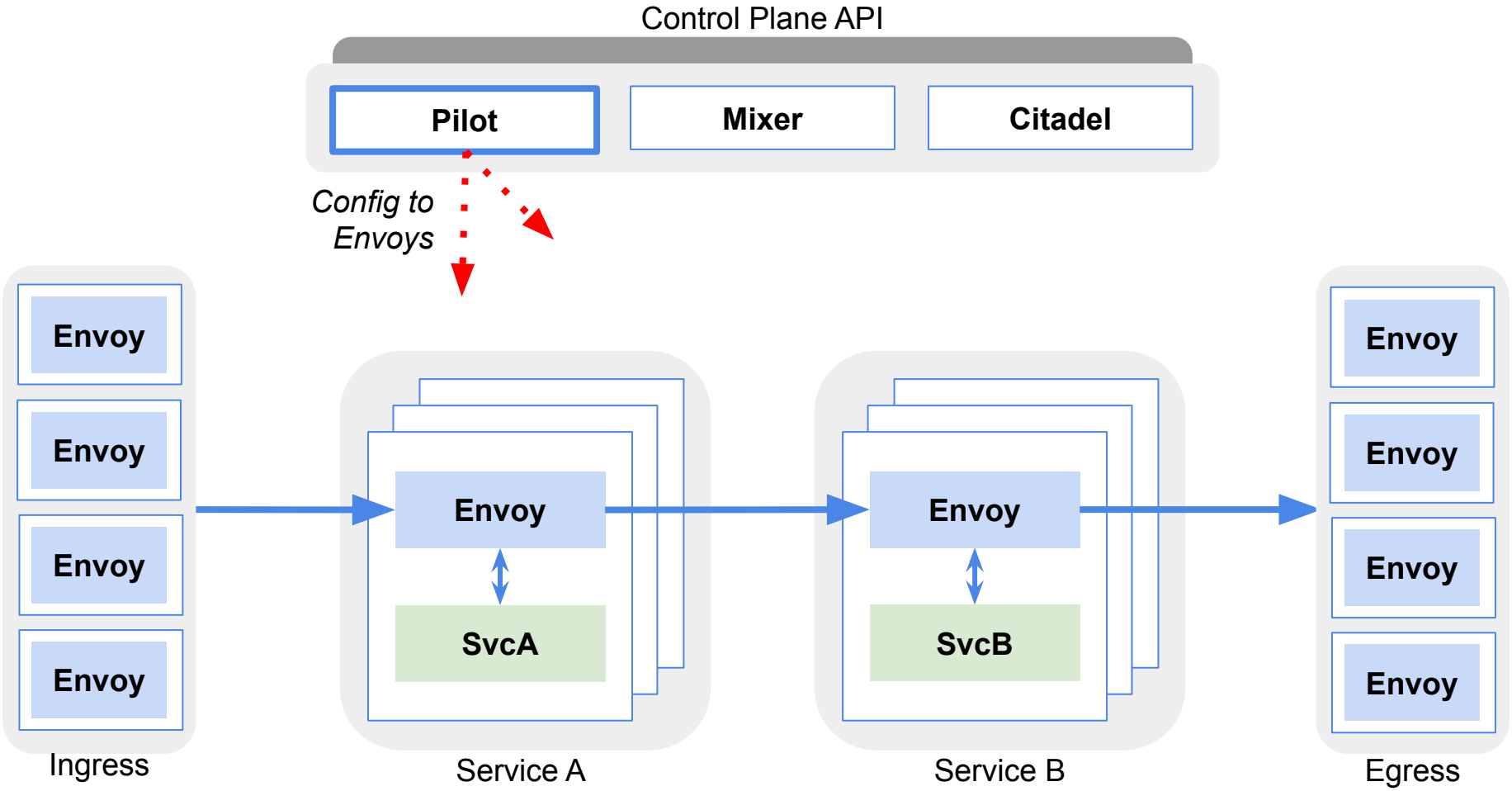

Control Plane API
Pilot
Mixer
Citadel
Envoy
Envoy
Envoy
Envoy
Ingress
Envoy
SvcA
Service A
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress

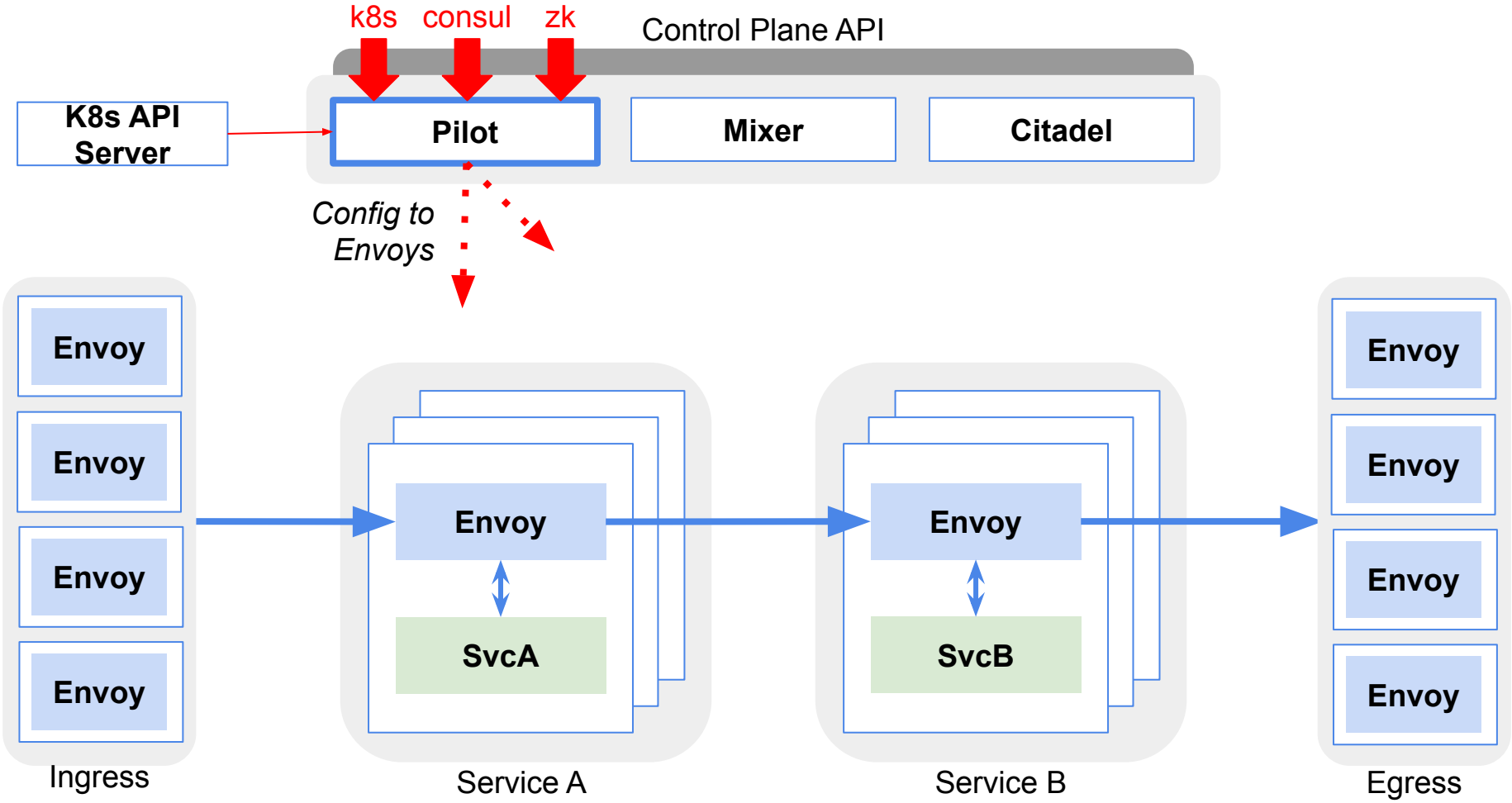

Control Plane API
Pilot
Mixer
Citadel
Envoy
Envoy
Envoy
Envoy
Ingress
Envoy
SvcA
Service A
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress


Control Plane API
Pilot
Mixer
Citadel
Envoy
SvcA
Service A
Envoy
SvcB
Service B


Control Plane API
Pilot
Mixer
Citadel
Envoy
SvcA
Service A
Envoy
SvcB
Service B

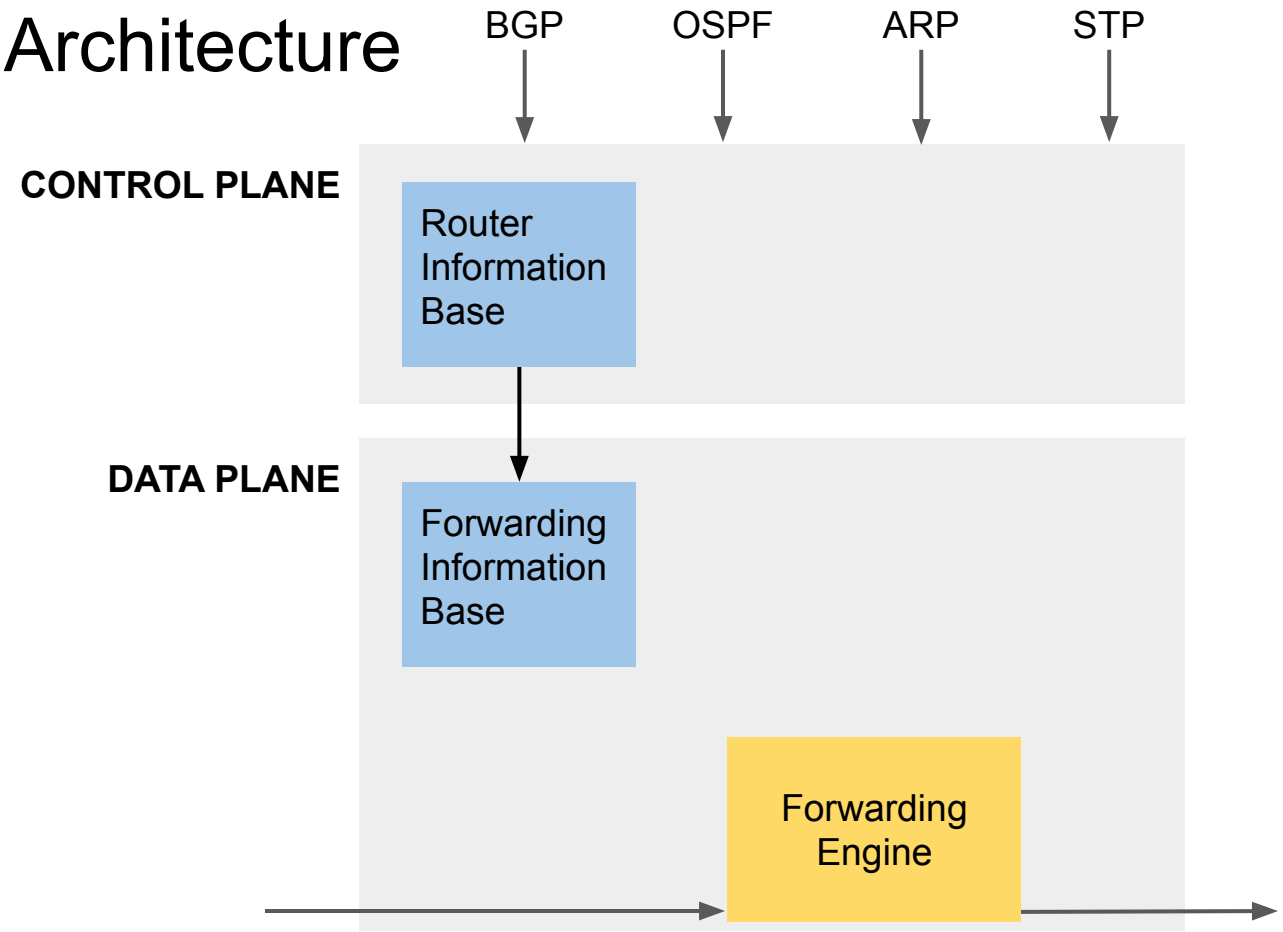

Control Plane API
Pilot
Mixer
Citadel
TLS certs
to Envoys
Envoy
SvcA
Service A
Envoy
SvcB
Service B


Control Plane API
Pilot
Mixer
Citadel
Config to Envoys
Envoy
Envoy
Envoy
Envoy
Ingress
Envoy
SvcA
Service A
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress

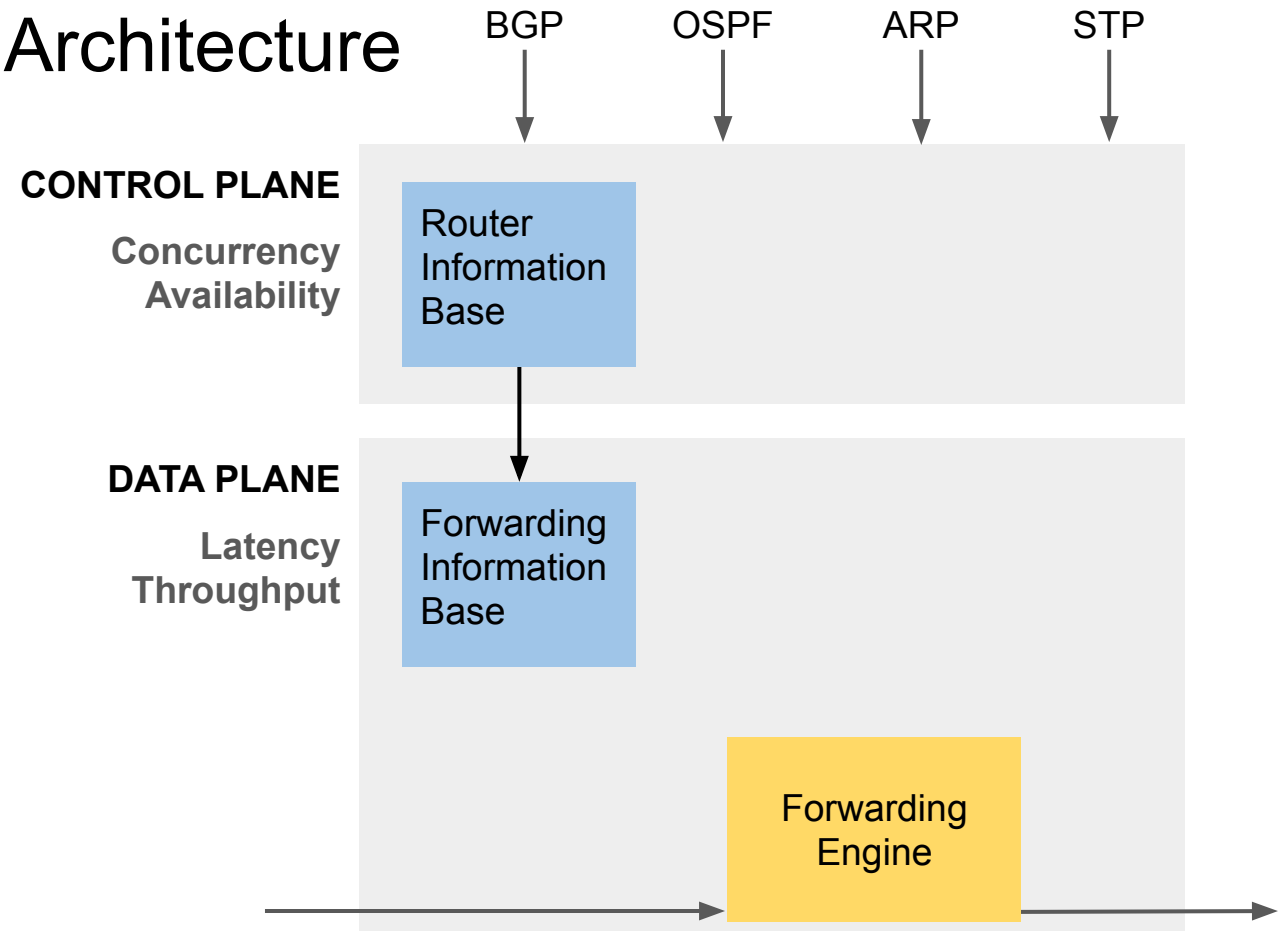

k8s
consul
zk
Control Plane API
K8s API Server
Pilot
Mixer
Citadel
Config to Envoys
Envoy
Envoy
Envoy
Envoy
Ingress
Envoy
SvcA
Service A
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress

Have We Seen This Before?

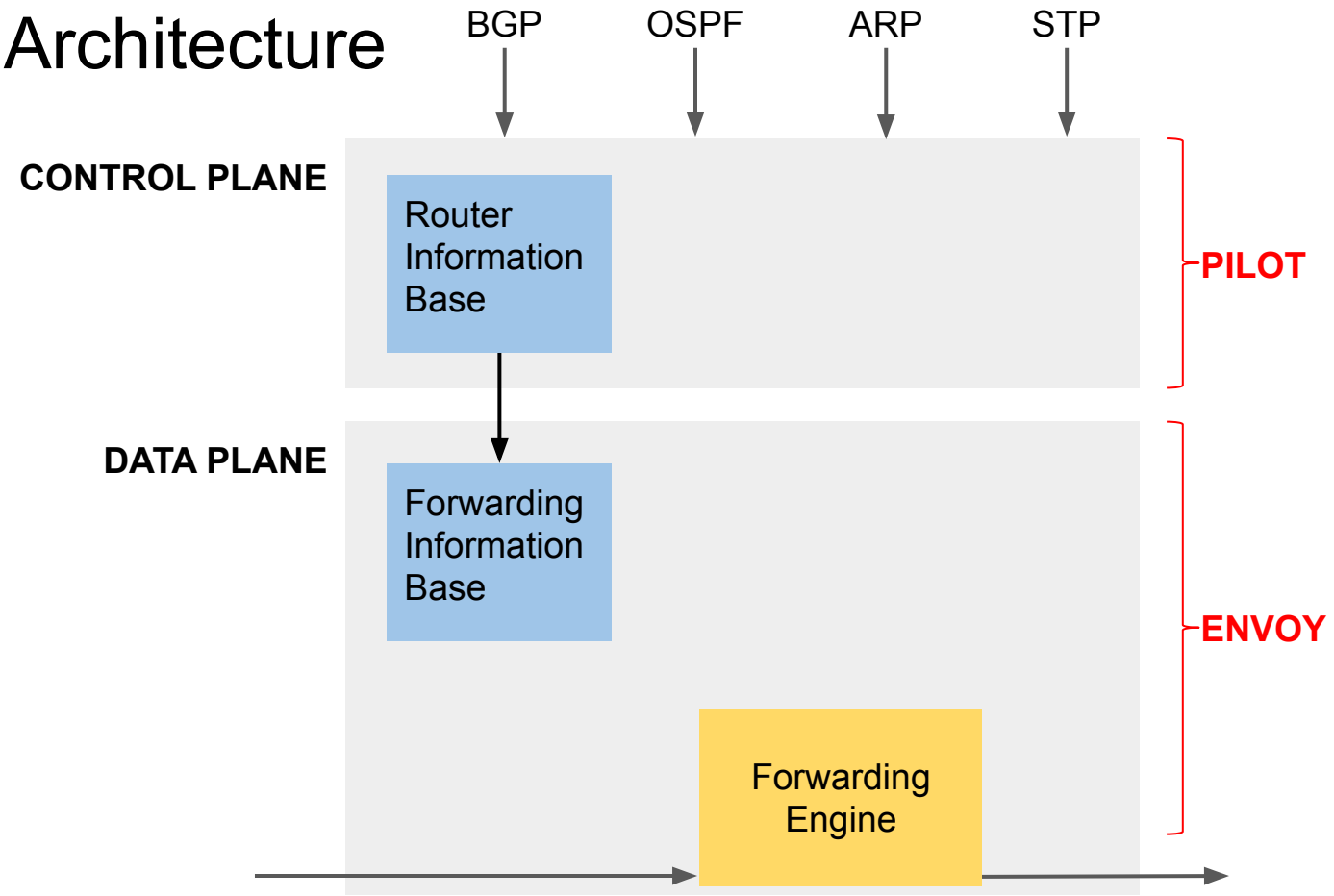
IP Router Architecture

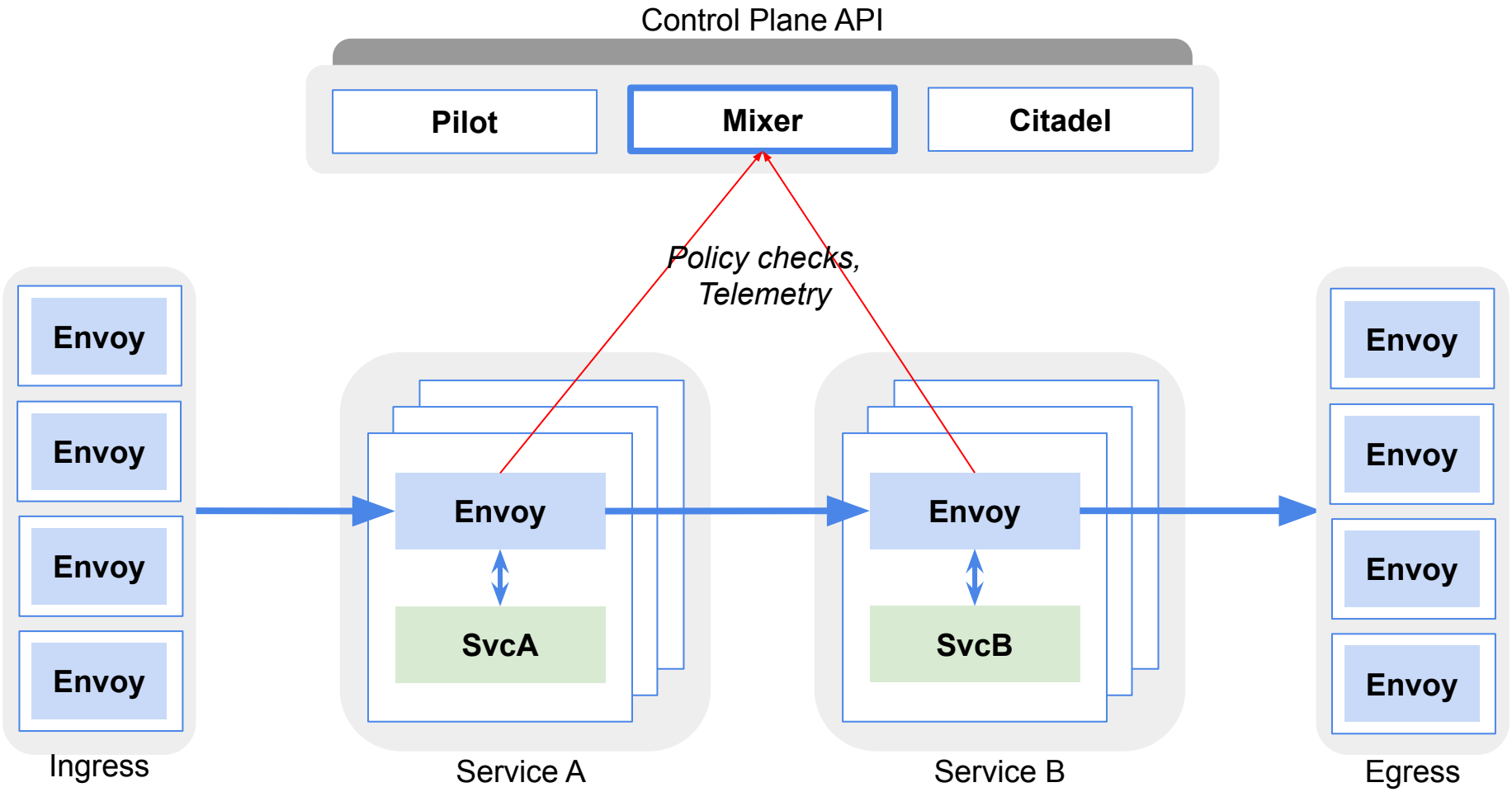


IP Router Architecture

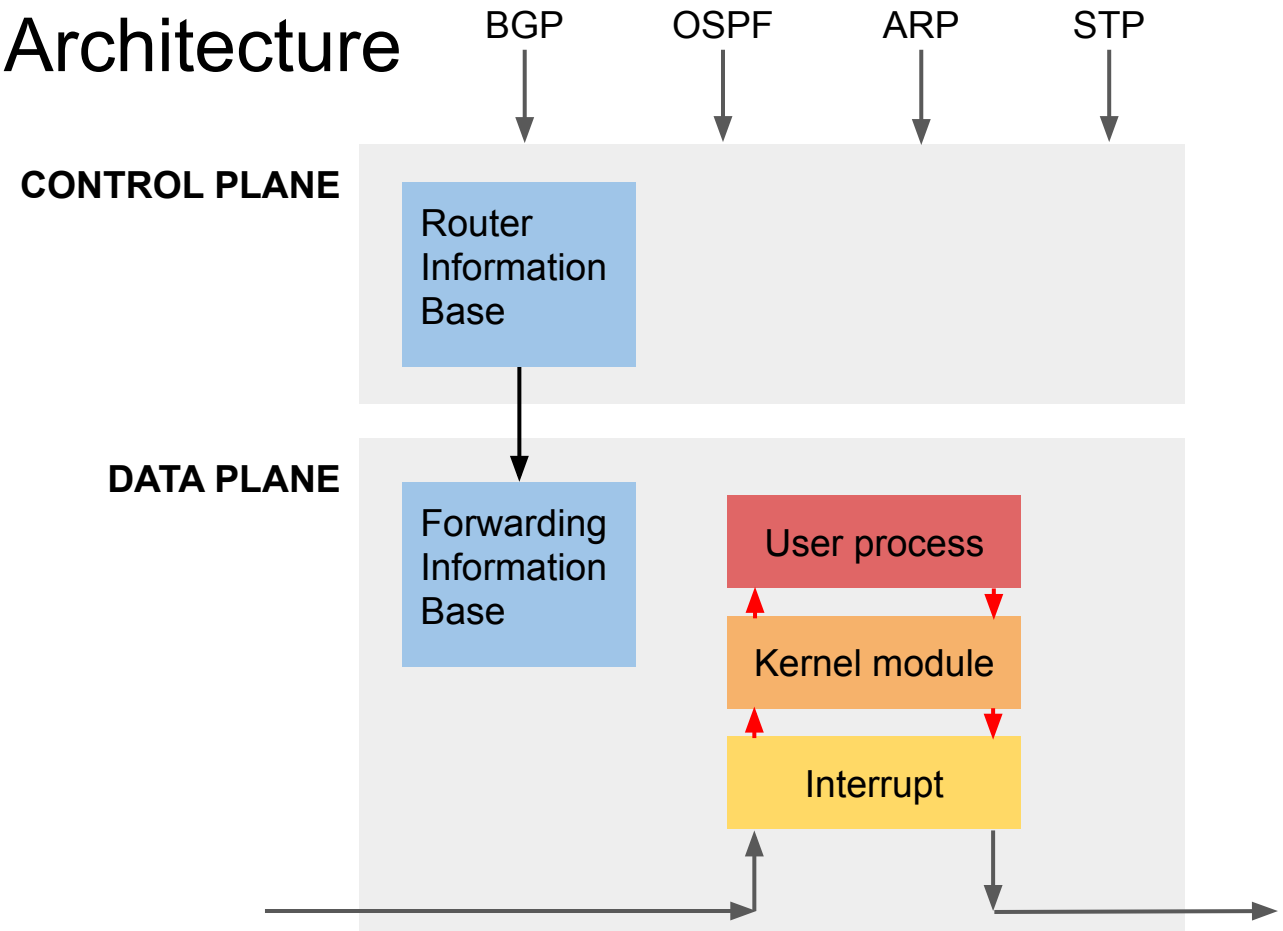


IP Router Architecture

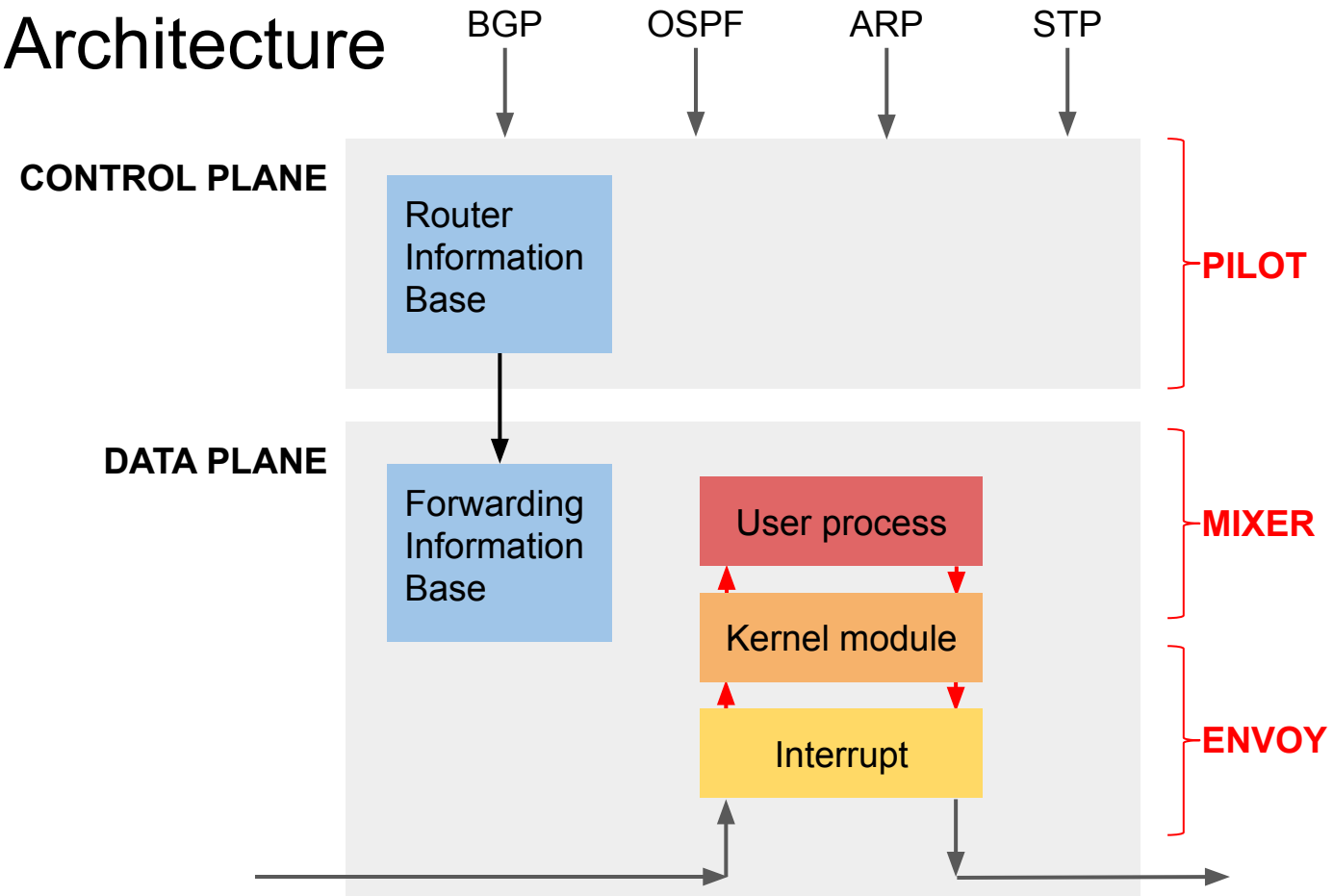


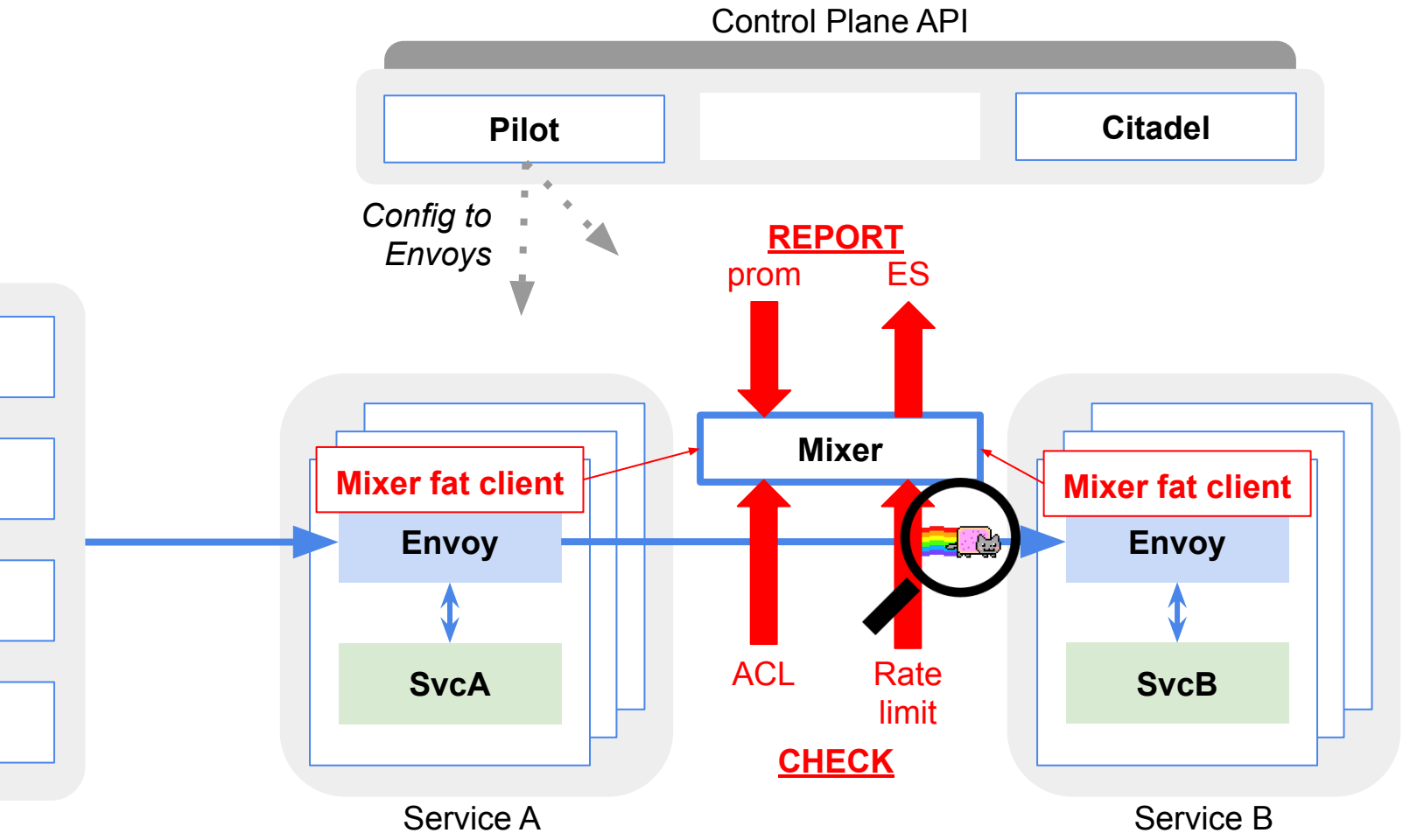

Control Plane API
Pilot
Mixer
Citadel
Policy checks,
Telemetry
Envoy
Envoy
Envoy
Envoy
Ingress
Envoy
SvcA
Service A
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress

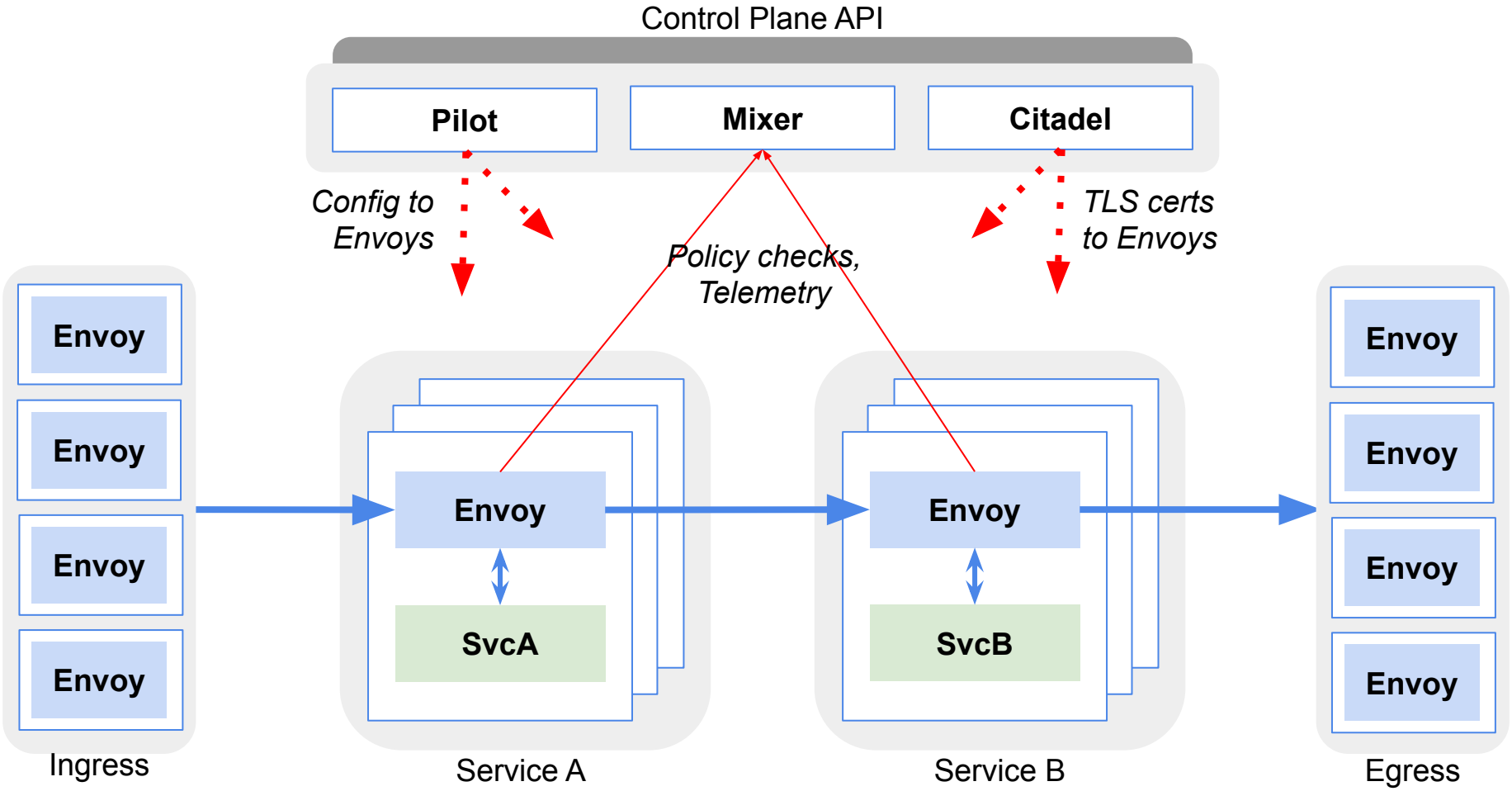
IP Router Architecture



IP Router Architecture




Control Plane API
Pilot
Citadel
Config to Envoys
REPORT
prom
ES
Mixer
Mixer fat client
Envoy
SvcA
Mixer fat client
Envoy
SvcB
ACL
Rate limit
CHECK
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

Why *Isn't* Istio This Shape?

Mixer

Pros

- Recognisable mental mode
- Good availability
- Can horizontally scale
  - Load
  - High availability

Cons

- Performance was never *great*
- Very few things need coördination between proxies

µServices!

Pros

- Decoupled deployments
- Isolated security contexts
- Isolated failure domains
- Individual scaling
- Multiple languages
- Clean mapping to individual teams/domains

Cons

- Complex to debug
- Fiddly to deploy
- Inefficient
  - Message-passing
  - Missed shared cache opportunities

µServices?

Decoupled deployments

- No-one ever did

μServices?

Isolated security contexts

- Citadel is obviously a security component, but…
- Pilot can re-route all your requests
- Mixer can steal all your traffic
- …a breach in any one is as bad as the others

μServices?

Isolated failure domains

- Mixer is the only acutely critical service

μServices?

Individual scaling

- Resource usage was dominated by one component of service: Pilot’s XDS serving

μServices?

Multiple languages

- It’s all Go

µServices?

Clean mapping to individual teams/domains

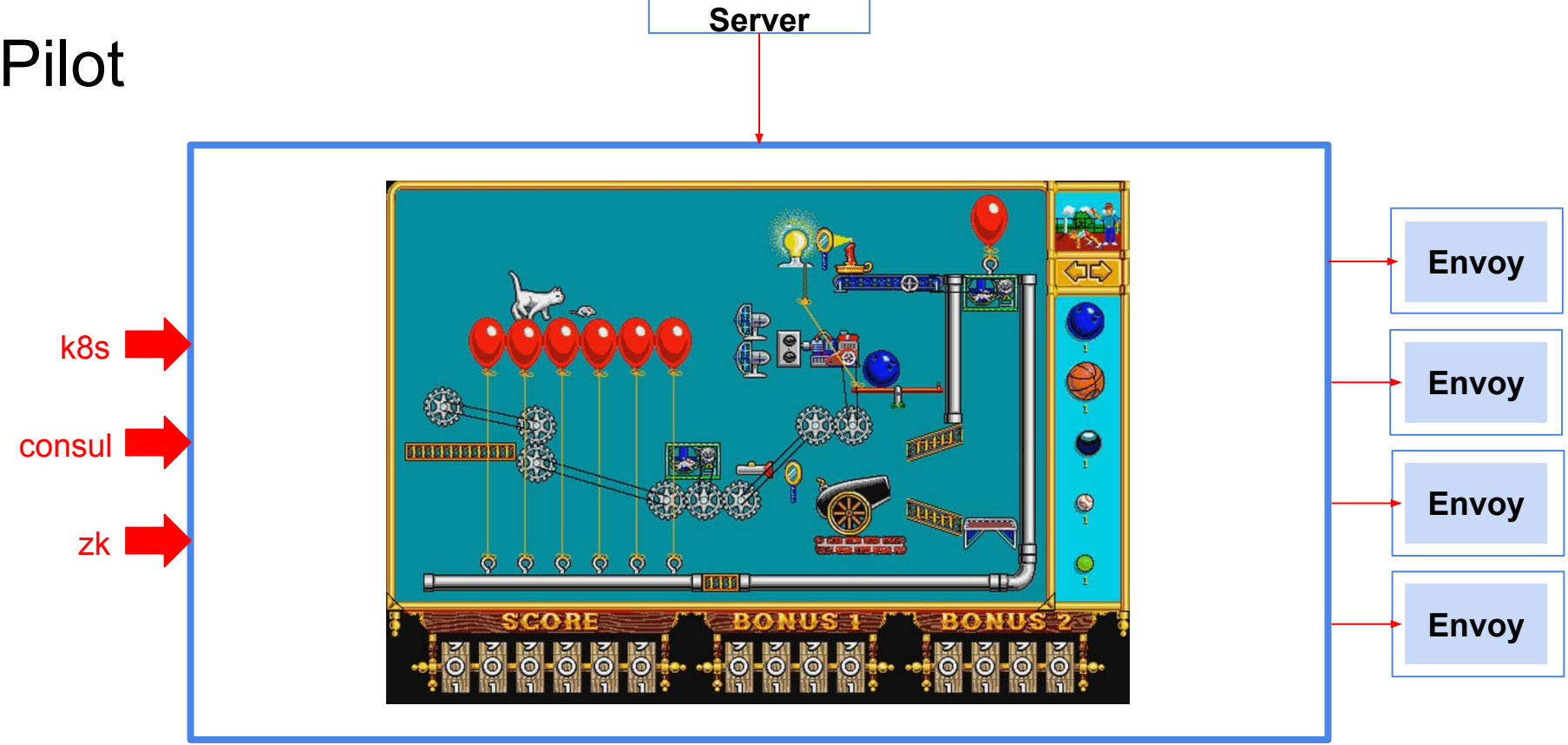
- We’re good at writing monoliths
- µServices were never about this anyway

Istio 1.5 - Date

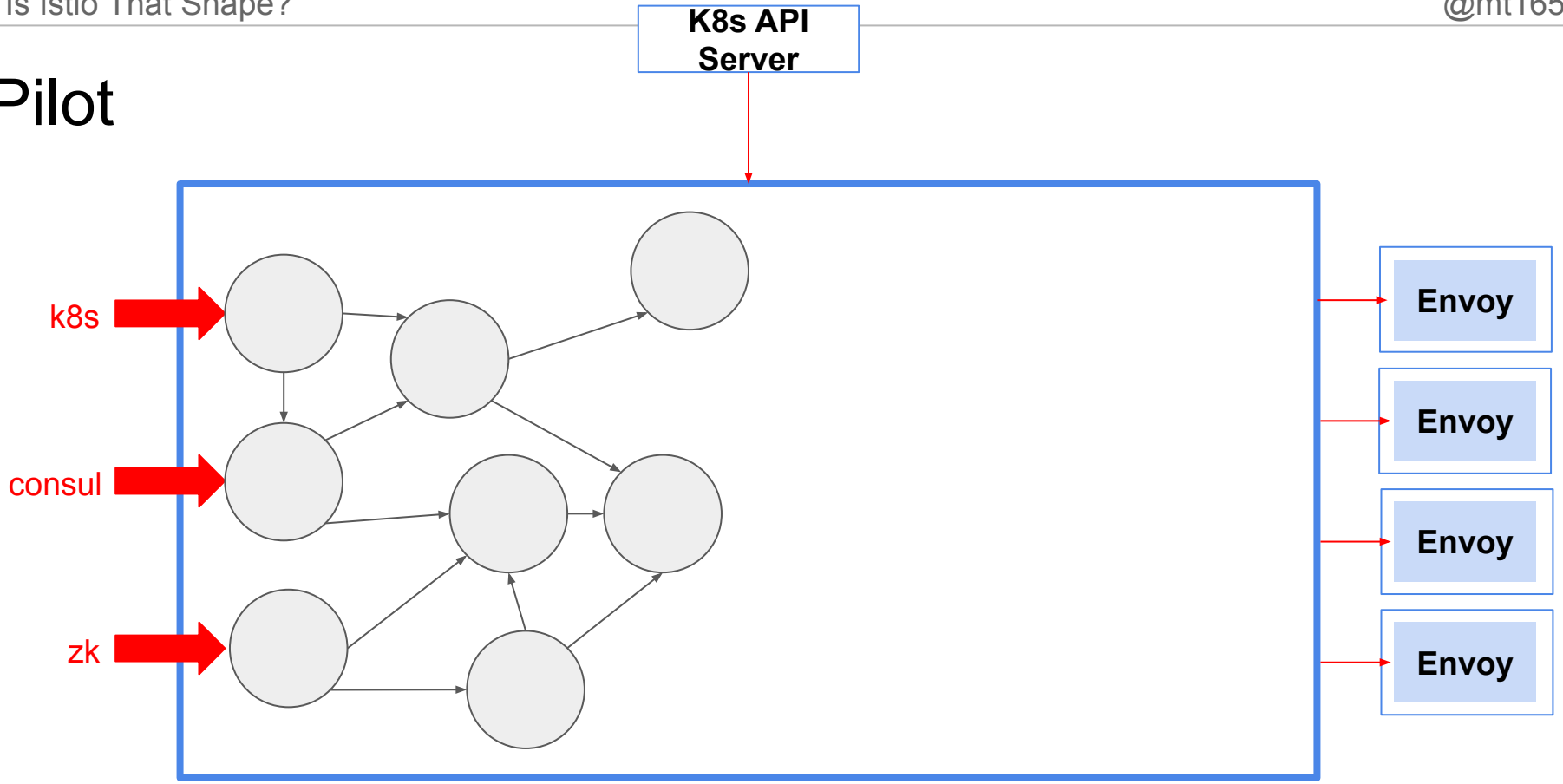
Pilot

- That thing I said about a compiler…
- It’s a nice mental model
- It wasn’t true

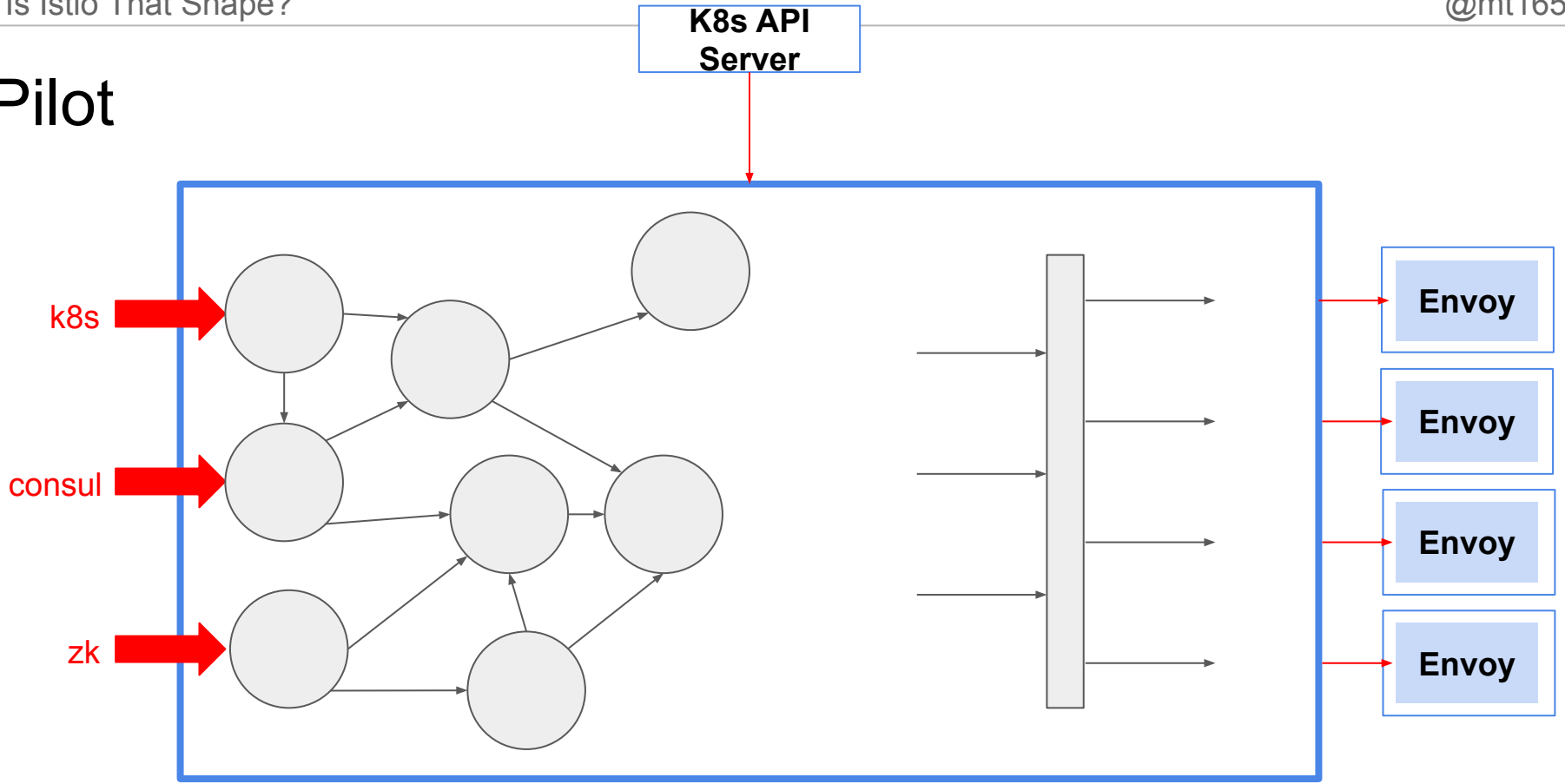
Pilot

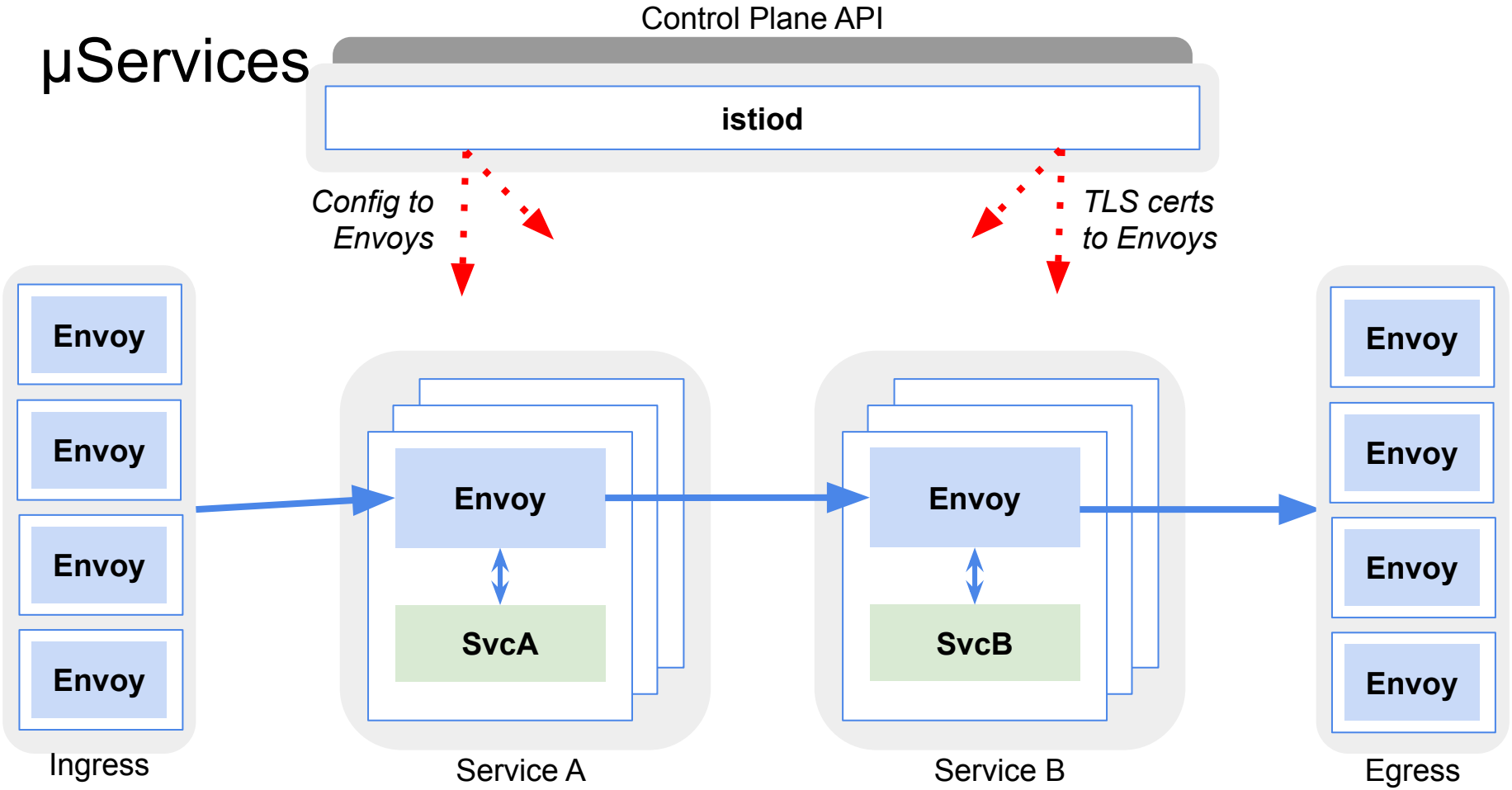

K8s API Server
k8s
consul
zk
SCORE
BONUS 1
BONUS 2
Envoy
Envoy
Envoy
Envoy

Pilot



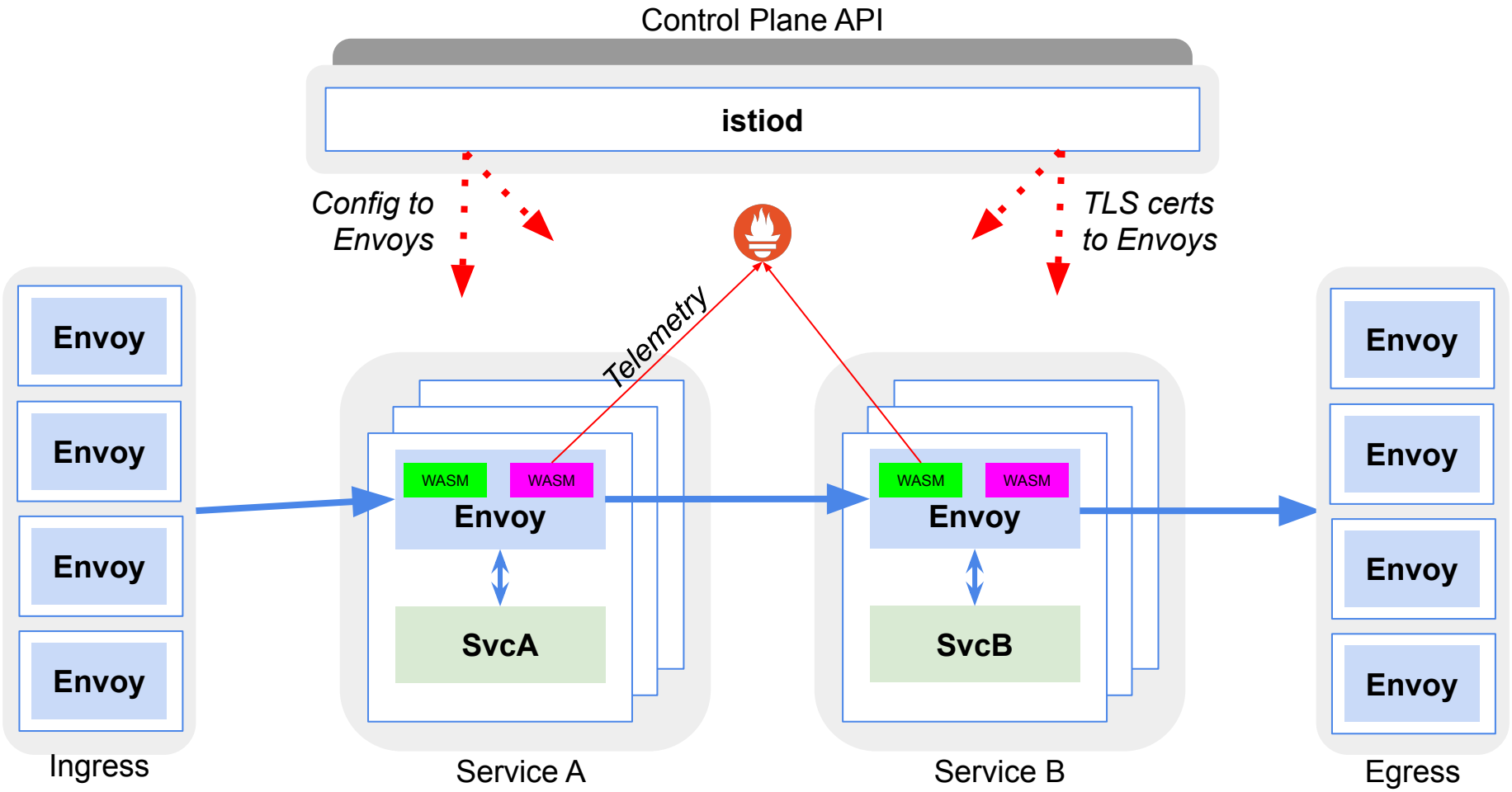
Pilot




μServices
Control Plane API
istiod
Config to Envoys
TLS certs to Envoys
Envoy
Envoy
Envoy
Envoy
Ingress
Envoy
SvcA
Service A
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress

Mixer

- Policy: implemented with (new) Envoy-native features

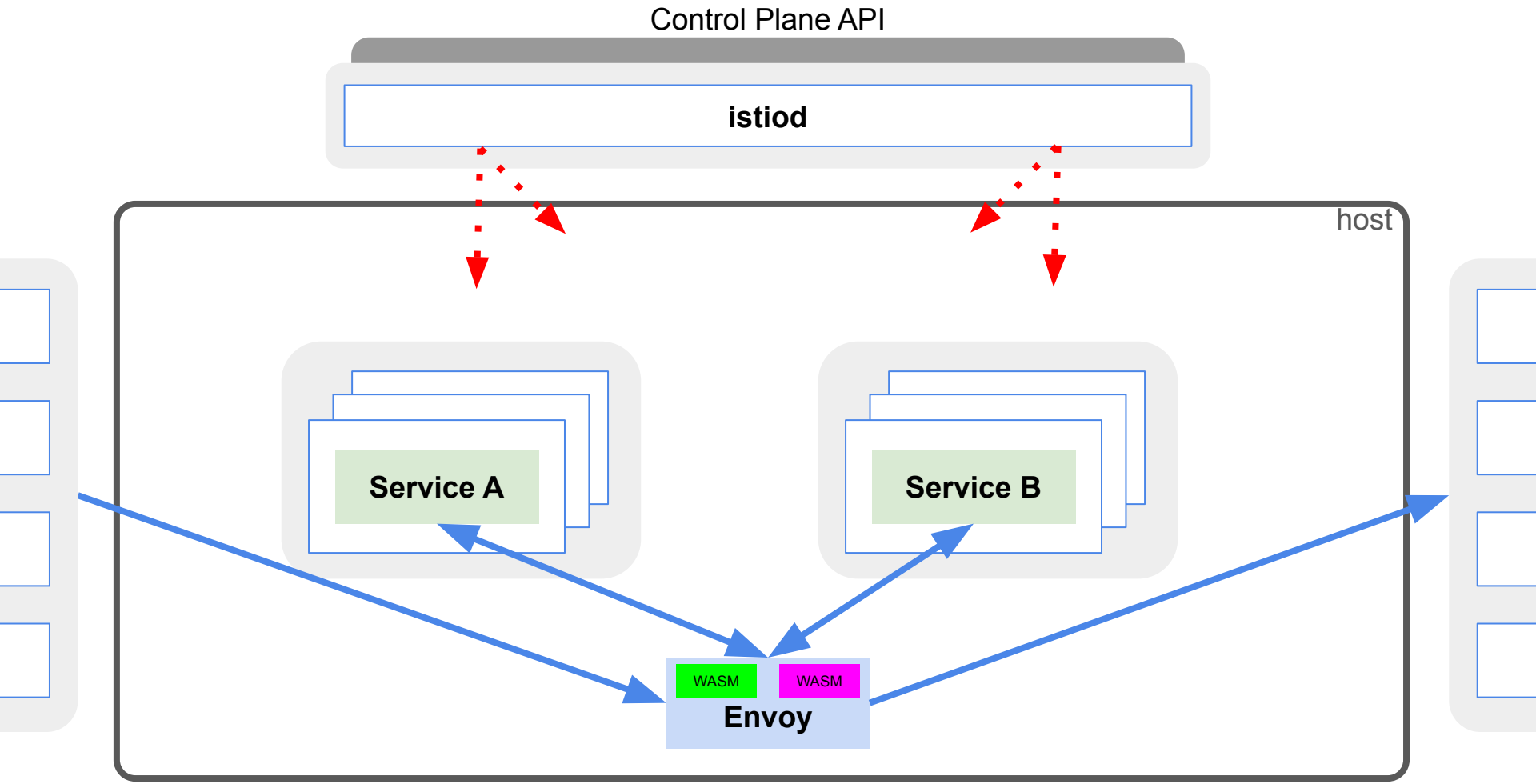

Control Plane API
istiod
Config to Envoys
TLS certs to Envoys
Telemetry
Envoy
Envoy
Envoy
Envoy
Ingress
WASM
WASM
Envoy
SvcA
Service A
WASM
WASM
Envoy
SvcB
Service B
Envoy
Envoy
Envoy
Envoy
Egress

#nomesh

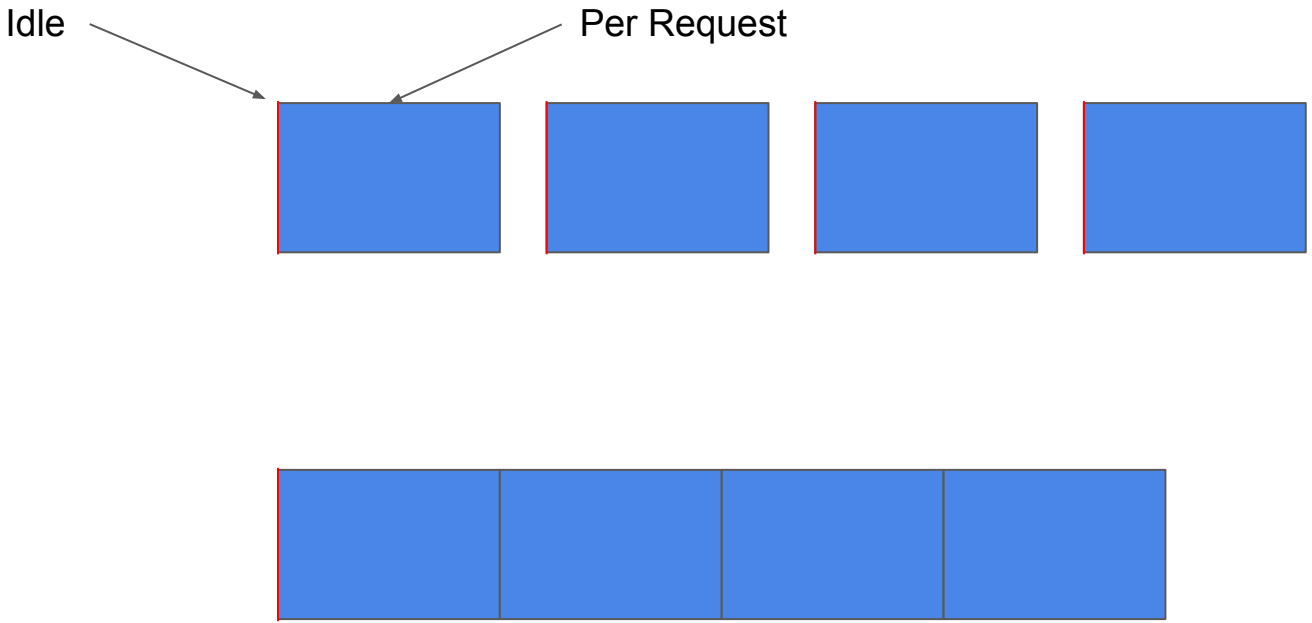
Dataplane Topology - Host-Based Proxy

#sidecarless

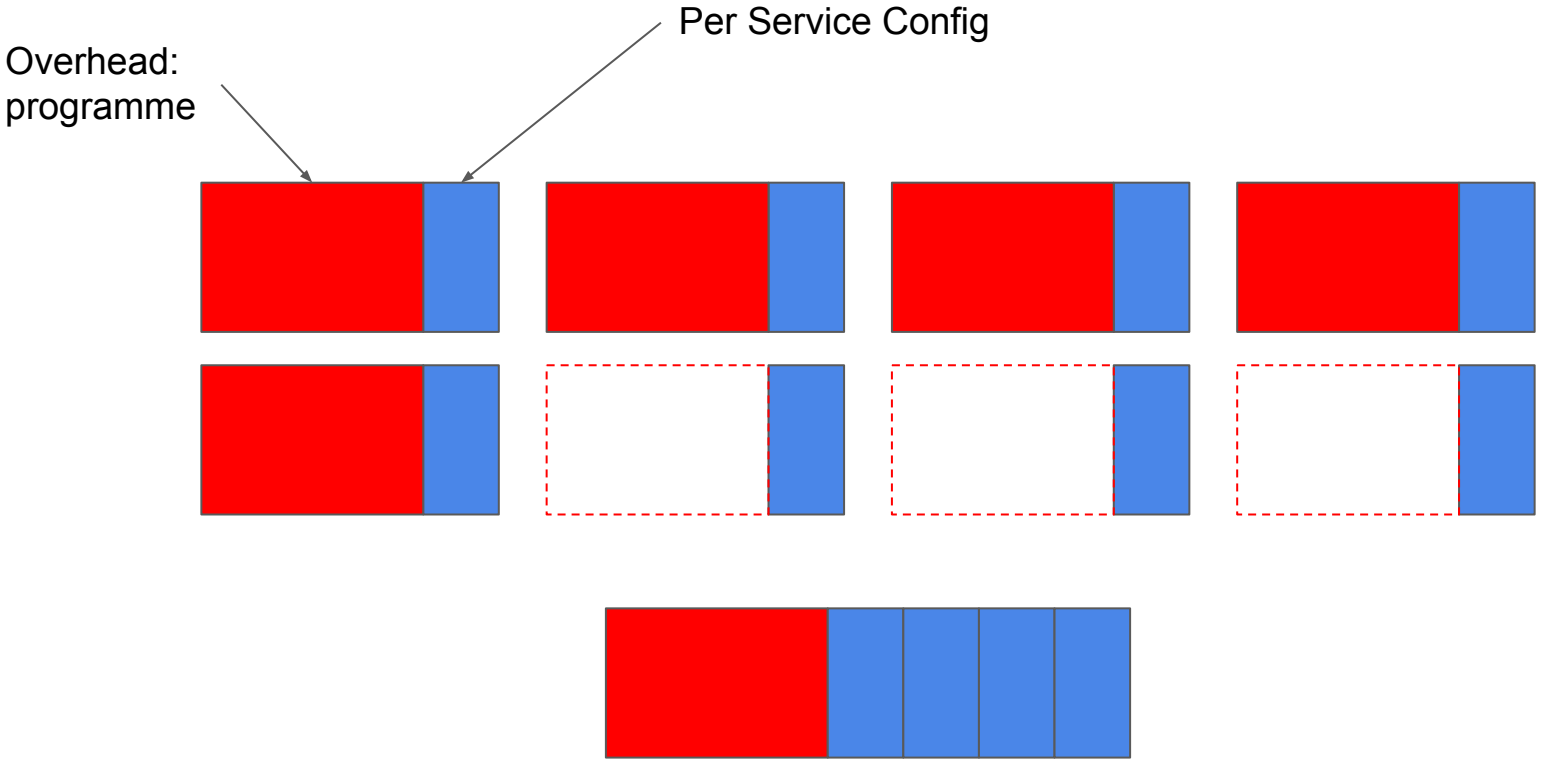
#eBPF


Control Plane API
istiod
host
Service A
Service B
WASM
WASM
Envoy

CPU Usage

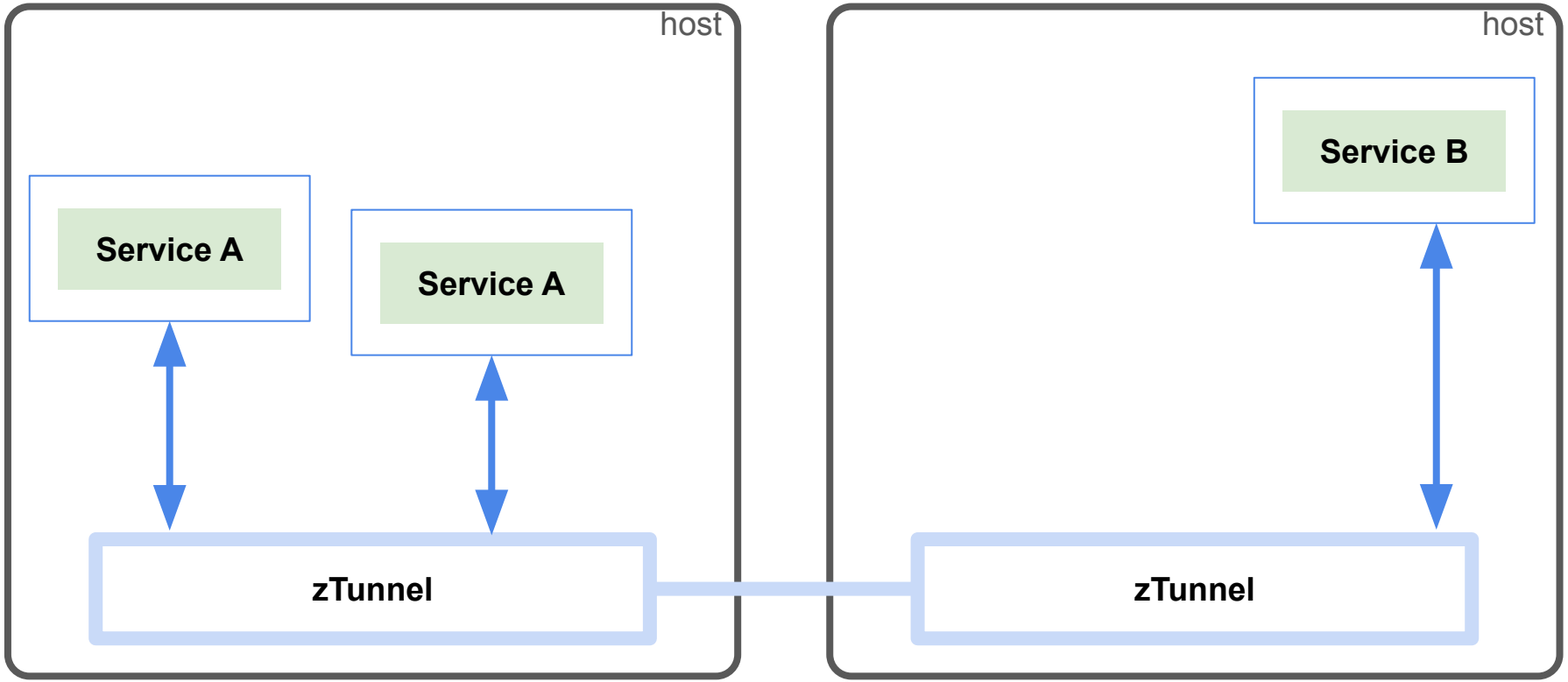


Memory Usage

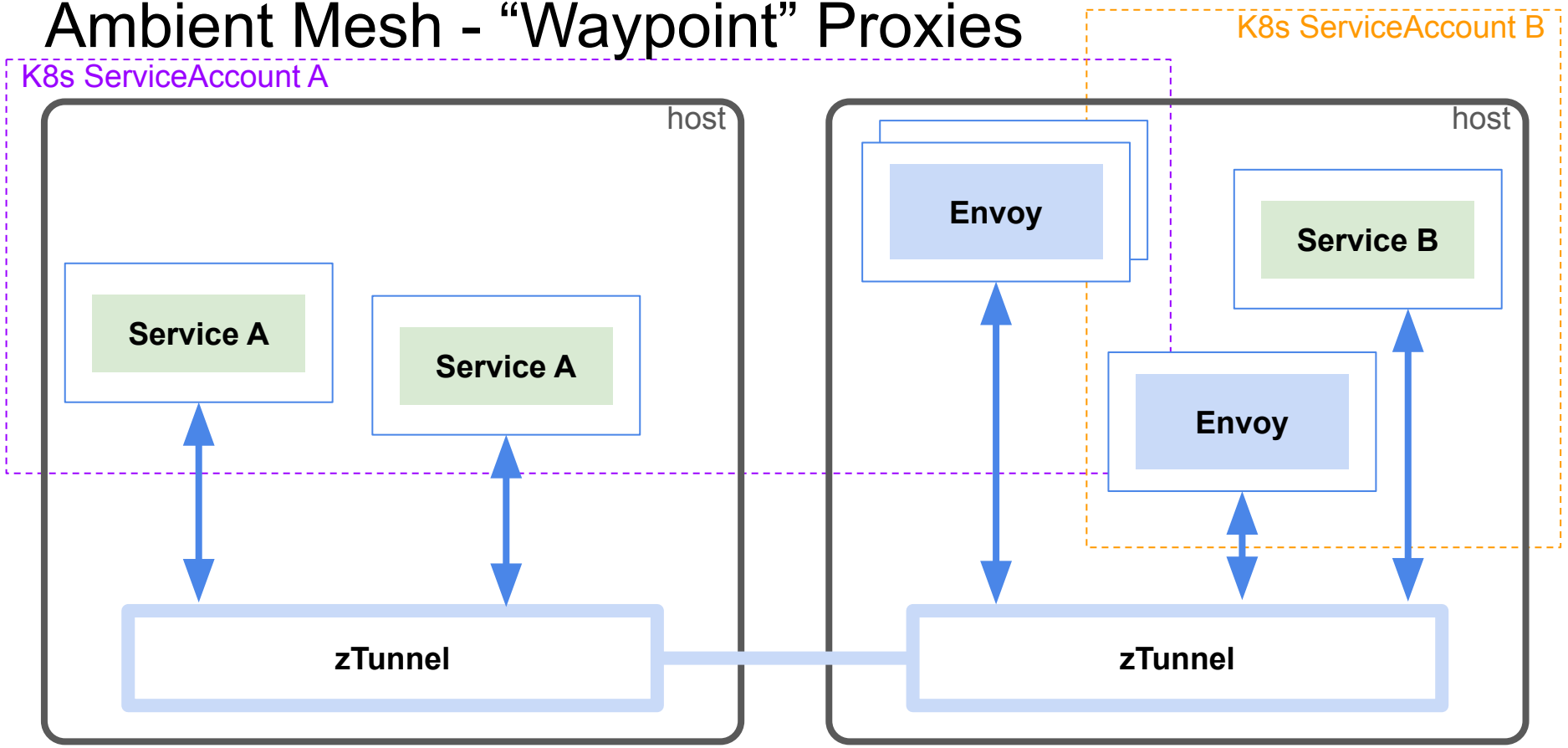


Dataplane Topology - “Ambient Mesh”

Ambient Mesh - zTunnel



Ambient Mesh - “Waypoint” Proxies



Thanks!

Questions	@mt165
Slides Videos Demo code	mt165.co.uk