Istio 101

Prague Containers Meetup

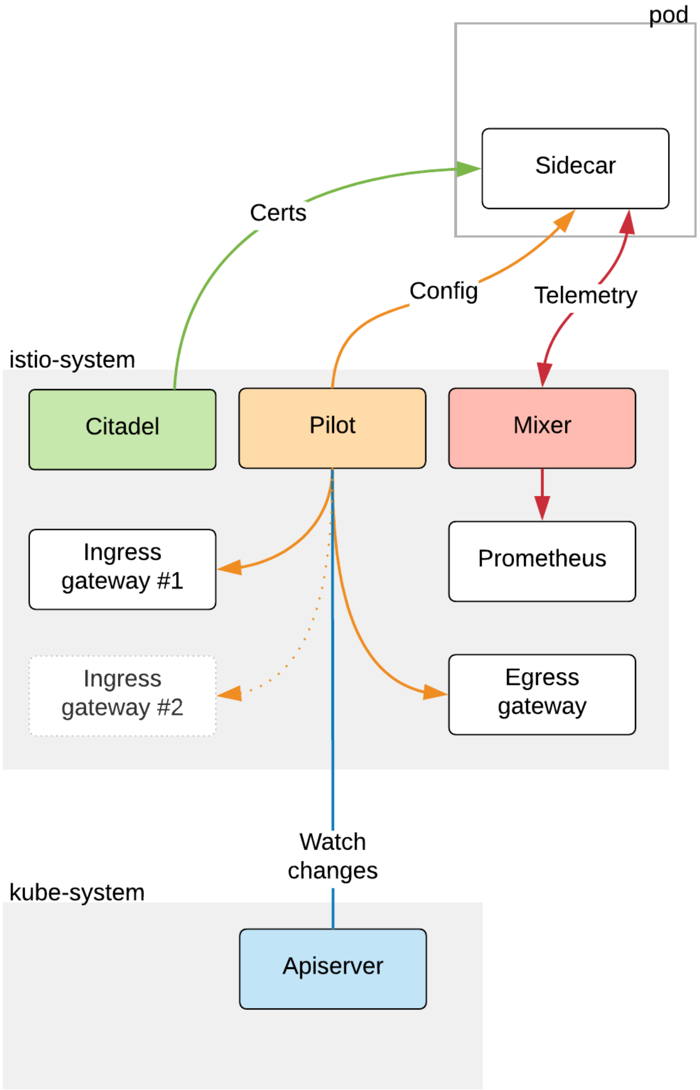
2018-11-07

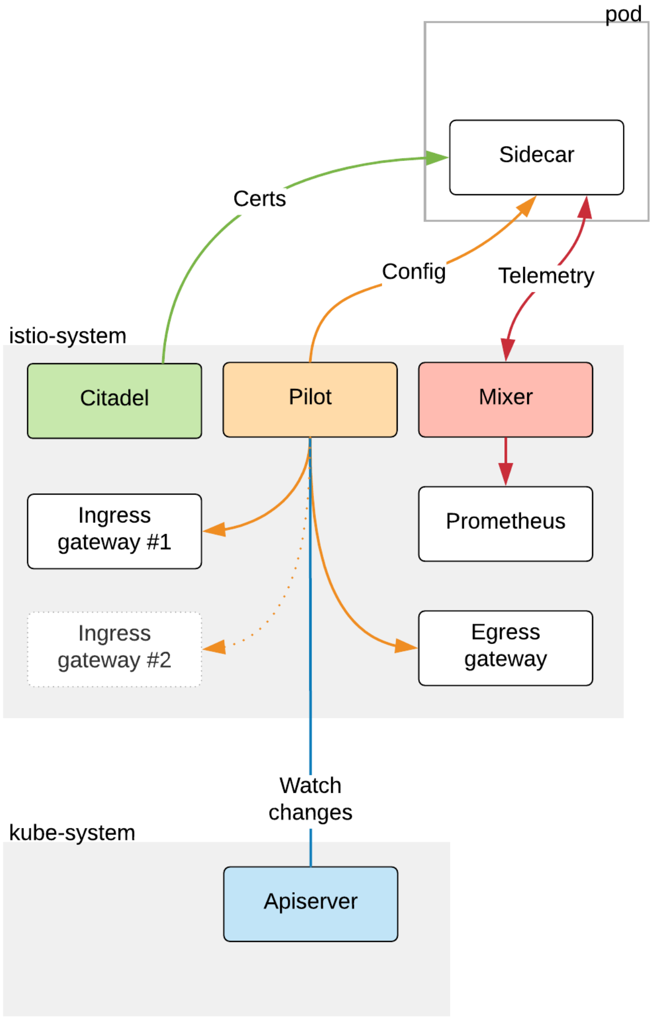
Istio

Services mesh using sidecar pattern to steer traffic in cluster.

Intercept incoming and outgoing traffic from pods and give you tools to manage this traffic

Using Kubernetes CRD for configuration objects.


pod
Sidecar
Certs
Config
Telemetry
istio-system
Citadel
Pilot
Mixer
Ingress gateway #1
Prometheus
Ingress gateway #2
Egress gateway
Watch changes
kube-system
Apiserver



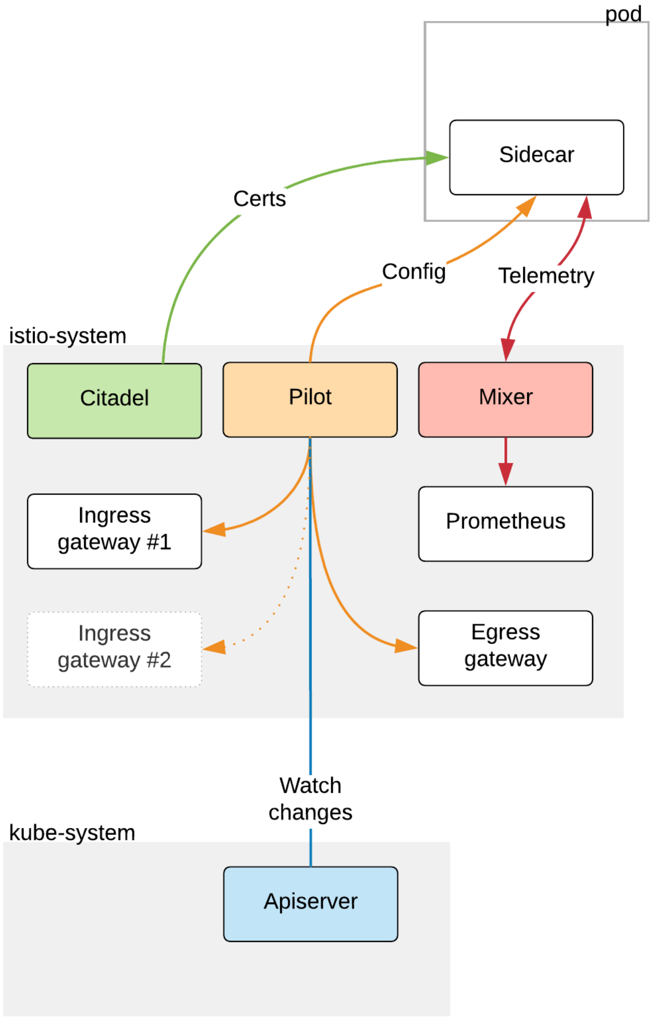
Sidecar

- envoy proxy
- Meditate and inspect network connection
- Get configuration from Pilot
- Running in all containers
- Transparent for application

- Kubernetes mutating webhook
- Injected using

```
istioctl kube-inject
```

- Generates telemetry

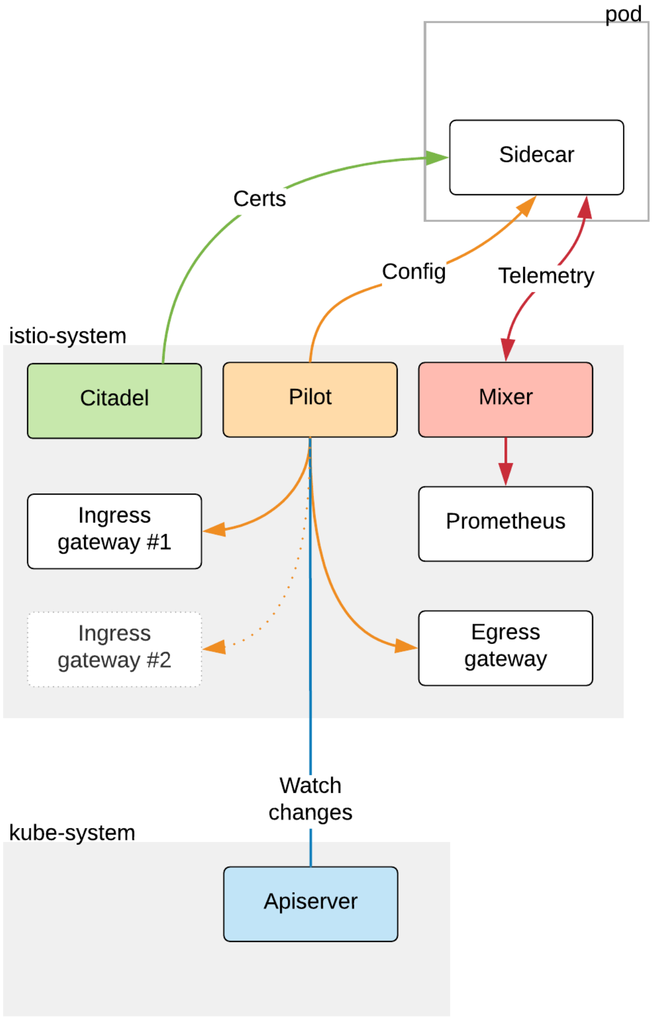


Pilot

- Convert Istio configuration to envoy configuration
- PUSH config to sidecars (end gateways)

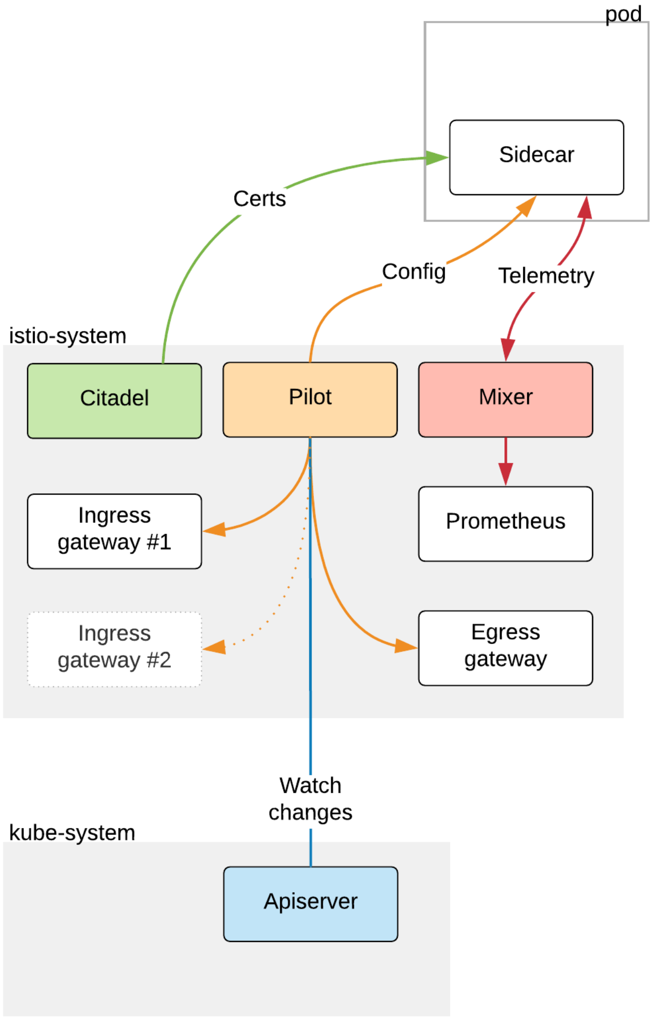
- Service discovery
- Traffic management
- Request handling (timeouts, circuit breaking)

- Watch changes in **Kubernetes API** and react on these changes



Mixer

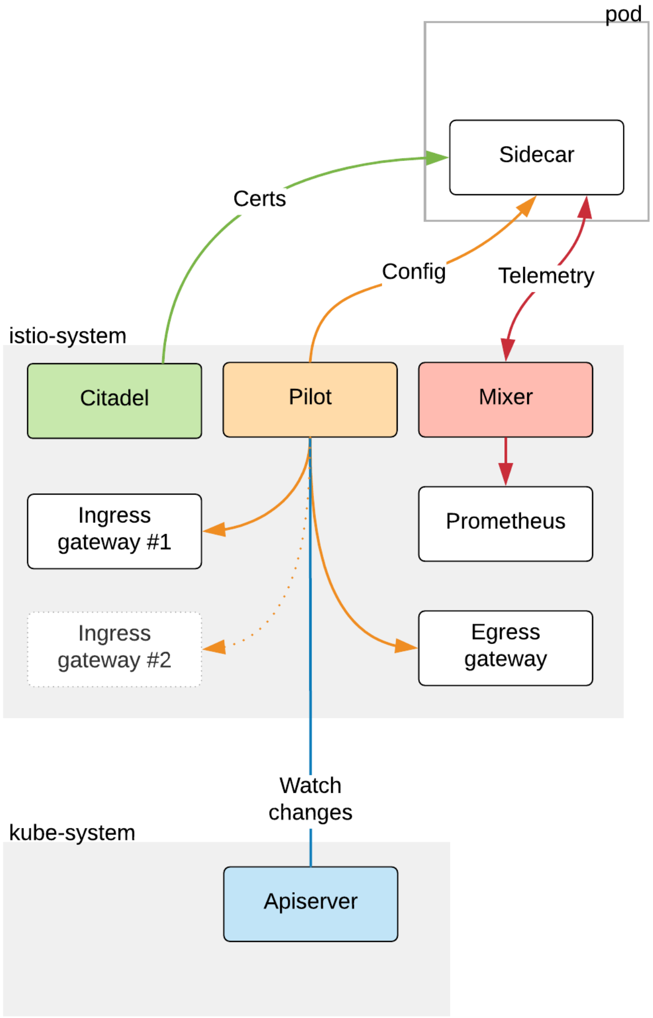
- Receive telemetry from sidecars
- Traffic policies
- Access control
- Quotas



Citadel

- Identity
- Credential management
- RBAC

- Generates certificates for mTLS

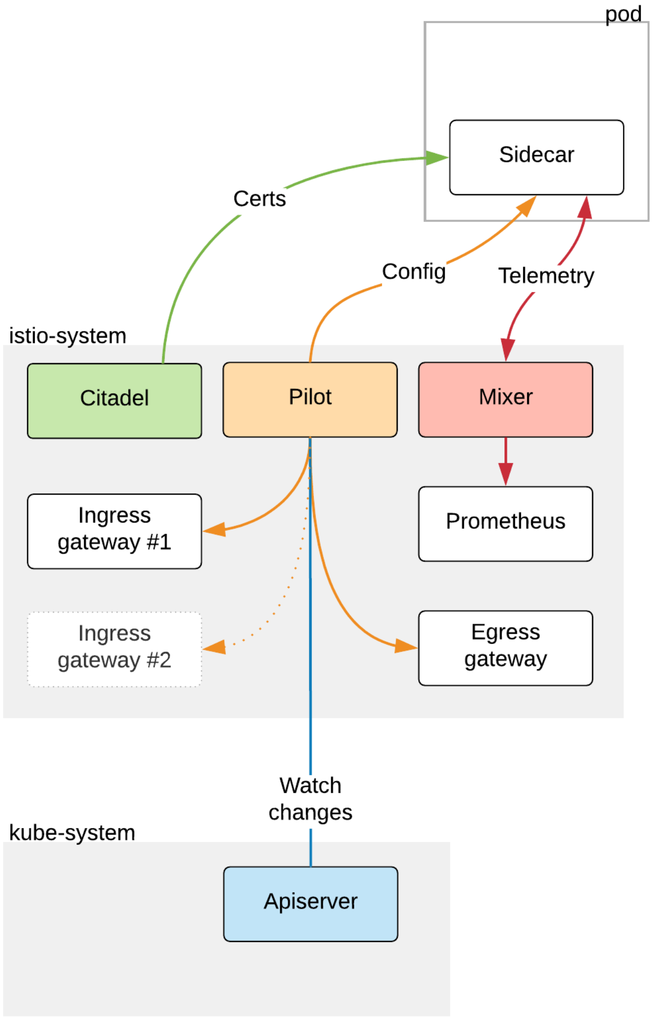


Ingress gateway

- Envoy
- Proxy from outside into the cluster
- Configured via Gateway and VirtualService

- Listen on `:80`
- Requests to `demo.prgcong.cz/istio101`
- Send to `web.demo.svc.cluster.local`

- Problem with `*` host
- Start new instances manually



Egress gateway

- Envoy
- Proxy for outgoing traffic
- Configured via ServiceEntry and VirtualService

Recap

- Reducing complexity for clouds with millions of services
- Adding complexity for smaller clouds

- Complicated as a hell

- Sidecar

- Pilot