Istio at Splunk

Bernard Van De Walle
April 2022

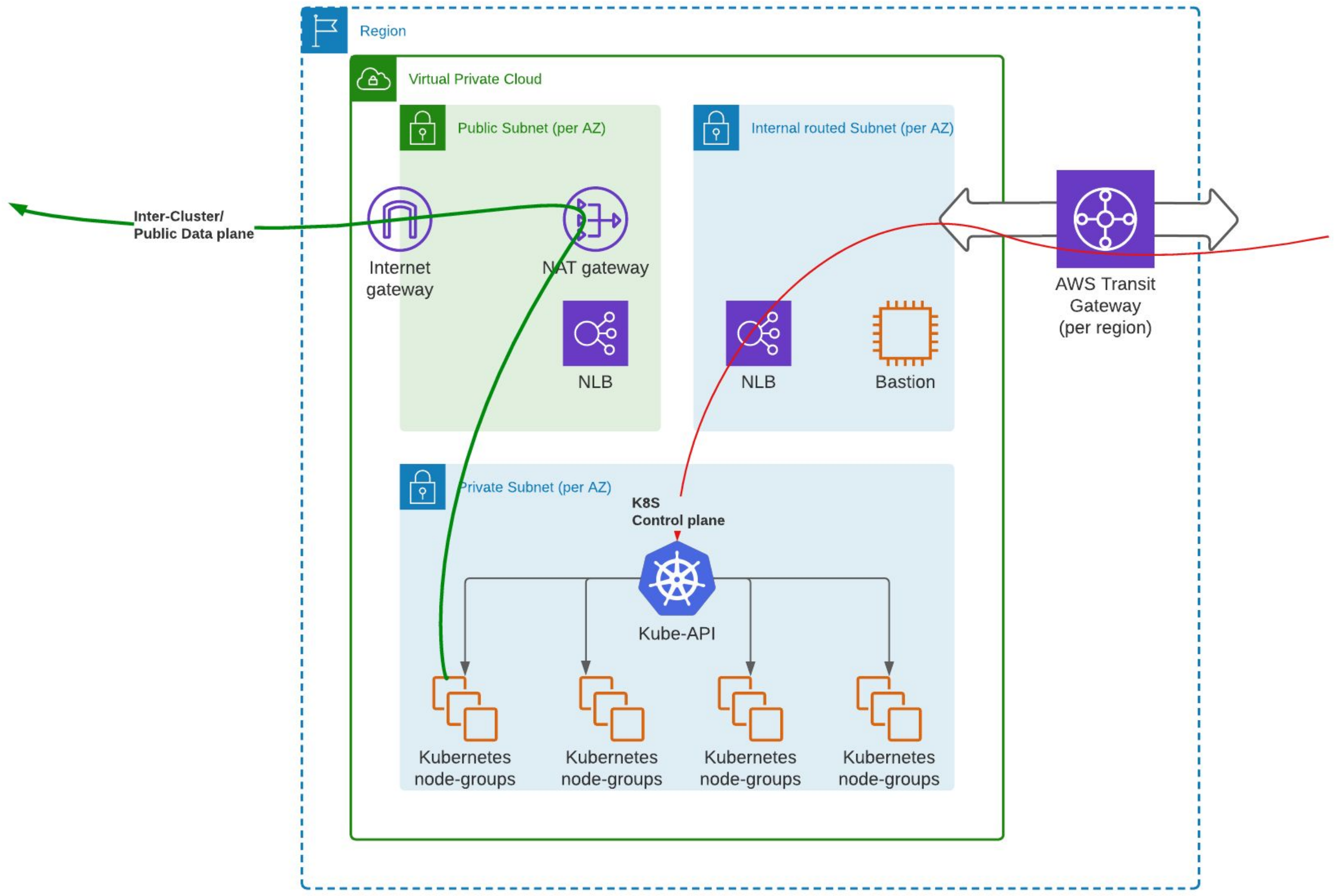
splunk> turn data into doing®

Istio at Splunk

- **Splunk Cloud platform**: Cloud Native splunk
- **Traffic Engineering team**
  - Connectivity (VPC, Multi-Cloud,...)
  - DNS (Millions of records)
  - **Application Traffic Management** (ATM)

Splunk Infrastructure

- ~ 35 K8S clusters
  - Distributed across all regions
  - AWS and GCP
- One cluster per VPC
- Cluster nodes deployed on private IP space
- Internal connectivity through internally routable subnets
- All workload connectivity through NLBs/Gateways



Our requirements

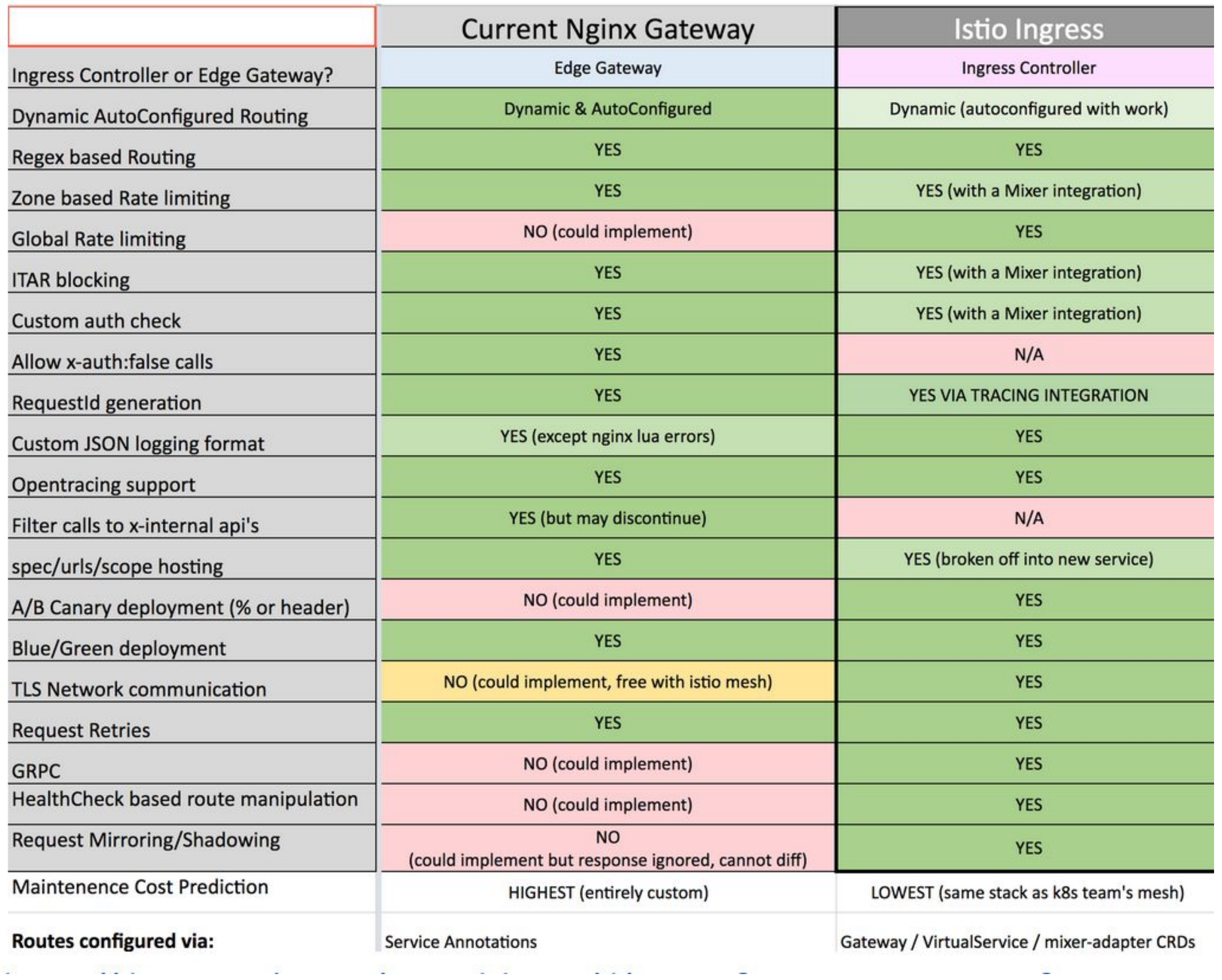
- mTLS and in-transit encryption
- Observability
- Managed Ingress
- Pluggable AuthN/AuthZ

First Iteration

- Nginx for Ingress
  - Custom controller
  - Managed through OpenResty and Lua scripts
  - AuthN/AuthZ enforcement
- mTLS managed by workloads
  - Certificates/Keys loaded at startup
- Observability through workloads libraries

Standardize on Istio

- Ingress through Istio gateways
- mTLS enforced through Istio
- Pluggable observability
- Mixer and out-of-process adapters

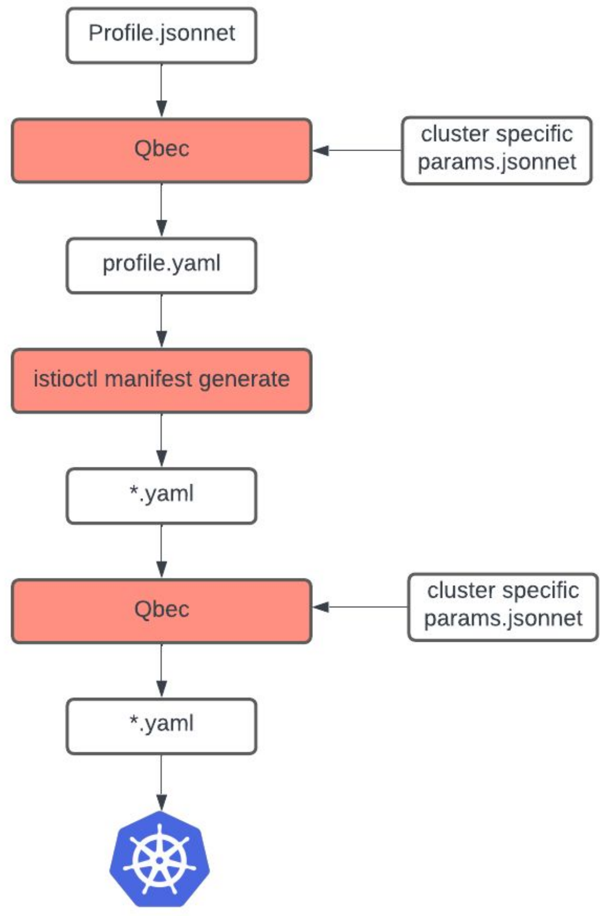


	Current Nginx Gateway	Istio Ingress
Ingress Controller or Edge Gateway?	Edge Gateway	Ingress Controller
Dynamic AutoConfigured Routing	Dynamic & AutoConfigured	Dynamic (autoconfigured with work)
Regex based Routing	YES	YES
Zone based Rate limiting	YES	YES (with a Mixer integration)
Global Rate limiting	NO (could implement)	YES
ITAR blocking	YES	YES (with a Mixer integration)
Custom auth check	YES	YES (with a Mixer integration)
Allow x-auth:false calls	YES	N/A
RequestId generation	YES	YES VIA TRACING INTEGRATION
Custom JSON logging format	YES (except nginx lua errors)	YES
Opentracing support	YES	YES
Filter calls to x-internal api's	YES (but may discontinue)	N/A
spec/urls/scope hosting	YES	YES (broken off into new service)
A/B Canary deployment (% or header)	NO (could implement)	YES
Blue/Green deployment	YES	YES
TLS Network communication	NO (could implement, free with istio mesh)	YES
Request Retries	YES	YES
GRPC	NO (could implement)	YES
HealthCheck based route manipulation	NO (could implement)	YES
Request Mirroring/Shadowing	NO (could implement but response ignored, cannot diff)	YES
Maintenence Cost Prediction	HIGHEST (entirely custom)	LOWEST (same stack as k8s team's mesh)
Routes configured via:	Service Annotations	Gateway / VirtualService / mixer-adapter CRDs

splunk> turn data into doing

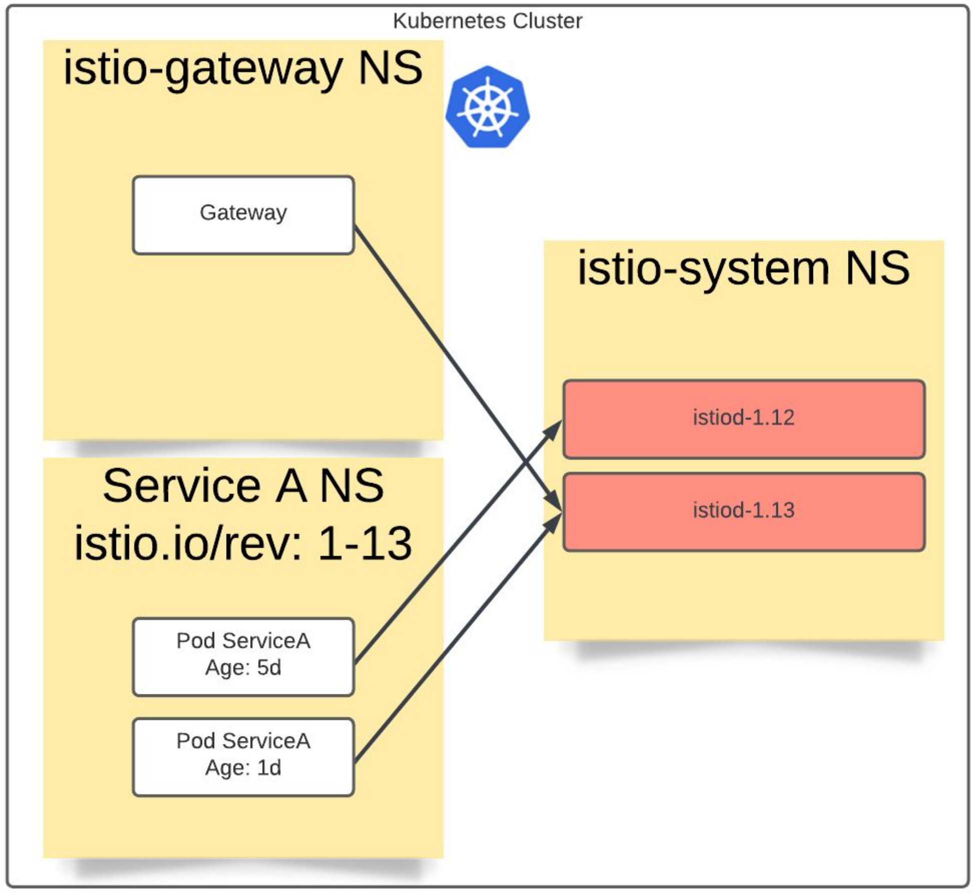
Istio installation

- Upstream istioctl
- Define as many things as possible programmatically on the istioctl profile
- Load and modify the resulting YAML
- Defined in jsonnet/QBEC (https://qbec.io/)


Profile.jsonnet
Qbec
cluster specific params.jsonnet
profile.yaml
istioctl manifest generate
*.yaml
Qbec
cluster specific params.jsonnet
*.yaml

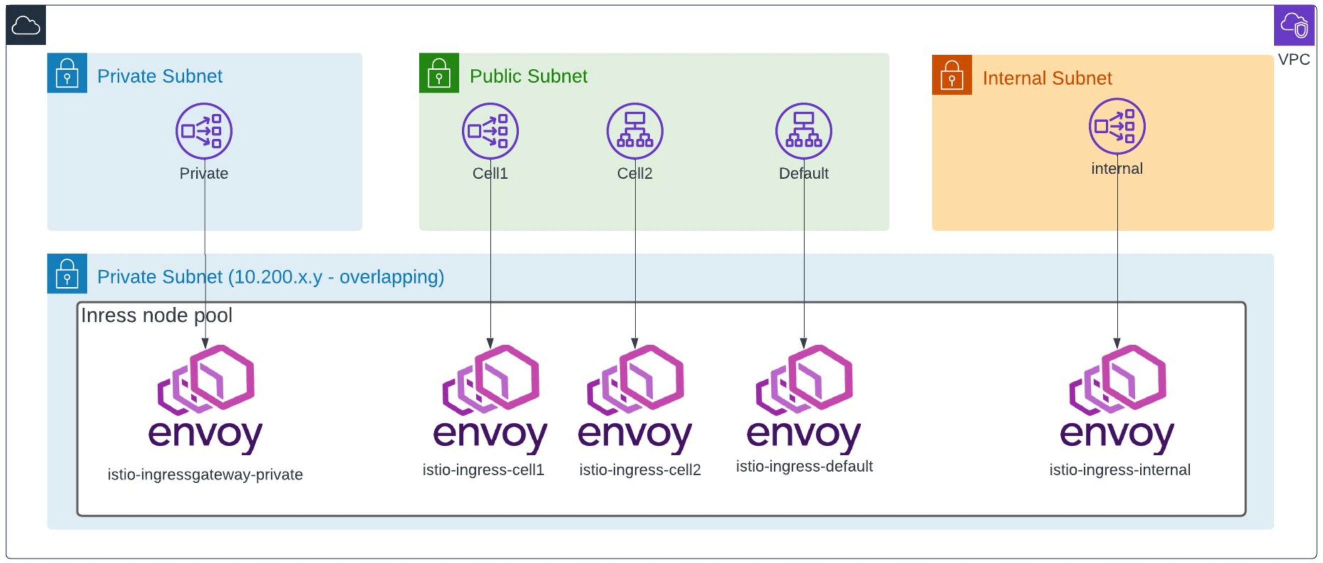
Istio upgrades

- Major versions
  - Deploy dual control-plane
  - Use *istio.io/rev* annotations
- Minor versions
  - Upgrade in-place
- Gateways get rolling-deployed directly
- Pods/Nodes get force-redeployed after 7 days



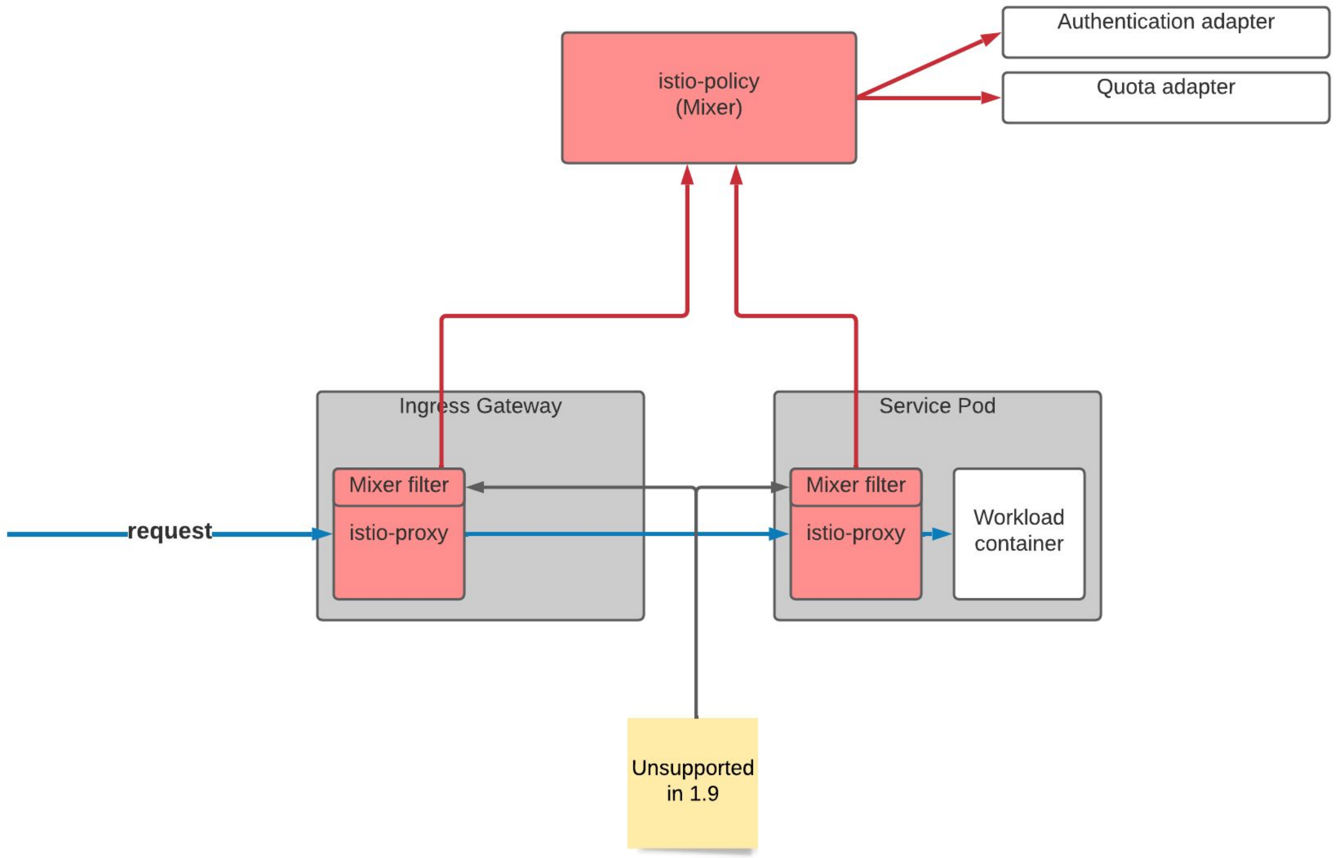
Gateway management

- NLB/Gateway pair per
  - Workload isolation
  - Internal/External/Private
- NLB defined through Terraform module
- Istio gateways defined through istioctl
- Gateways co-scheduled on Ingress nodes



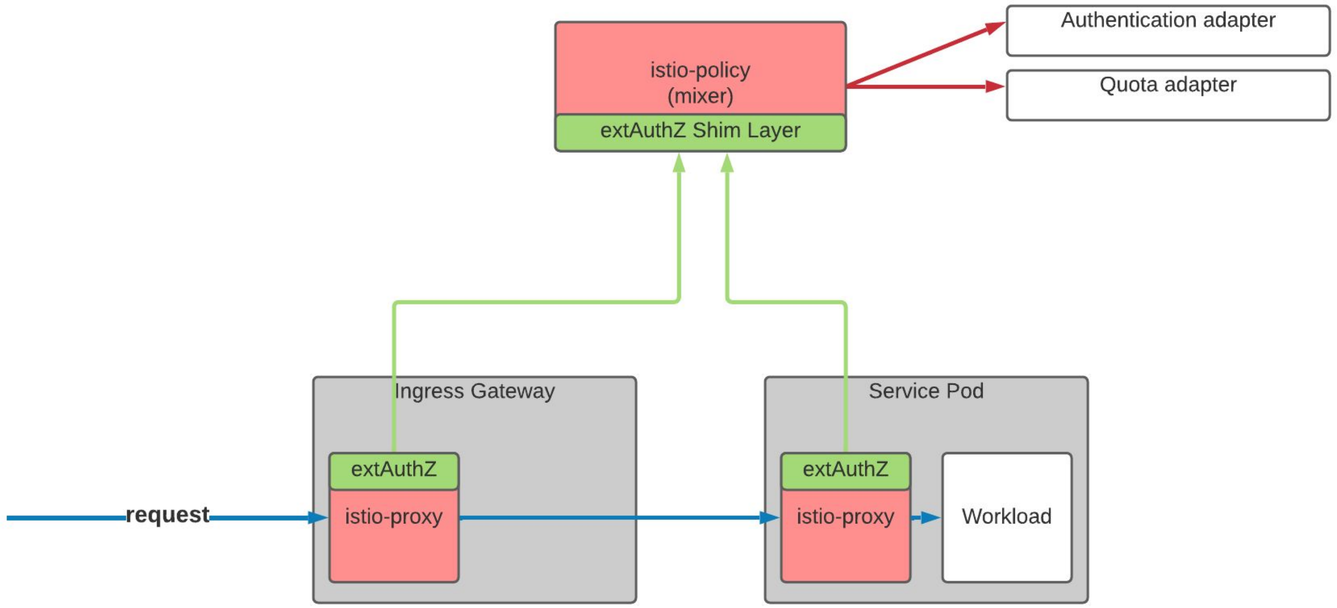
Authentication & Routing

- Heavy usage of Mixer (istio-policy)
- Out of process adapters:
  - Authentication
  - Quota enforcement
  - ITAR requirements
  - Logging



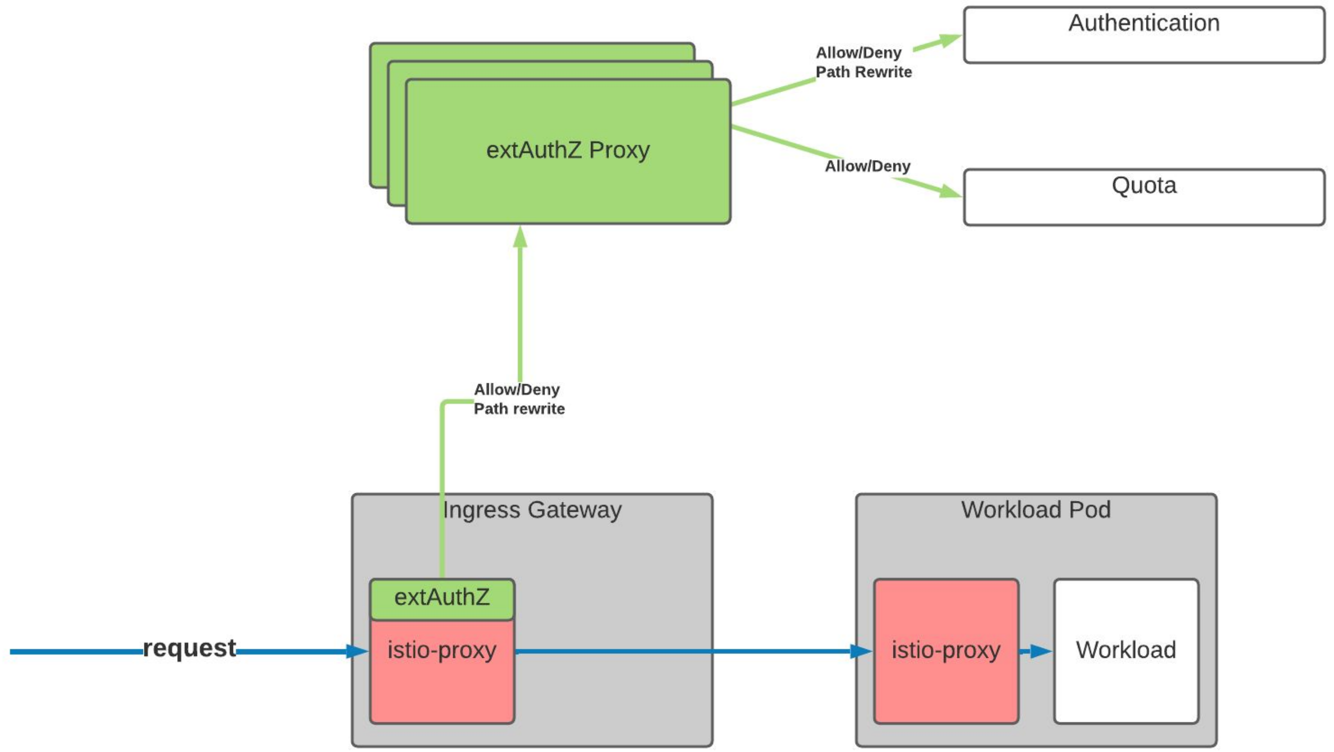
Authentication & Routing

- Mixer removed with Istio 1.8
- Use Envoy ExtAuthZ Shim Layer for Mixer



Authentication & Routing

- Build our own ExtAuthZ Server
  - Use it as a proxy to different components
- Build a WASM filter for logging

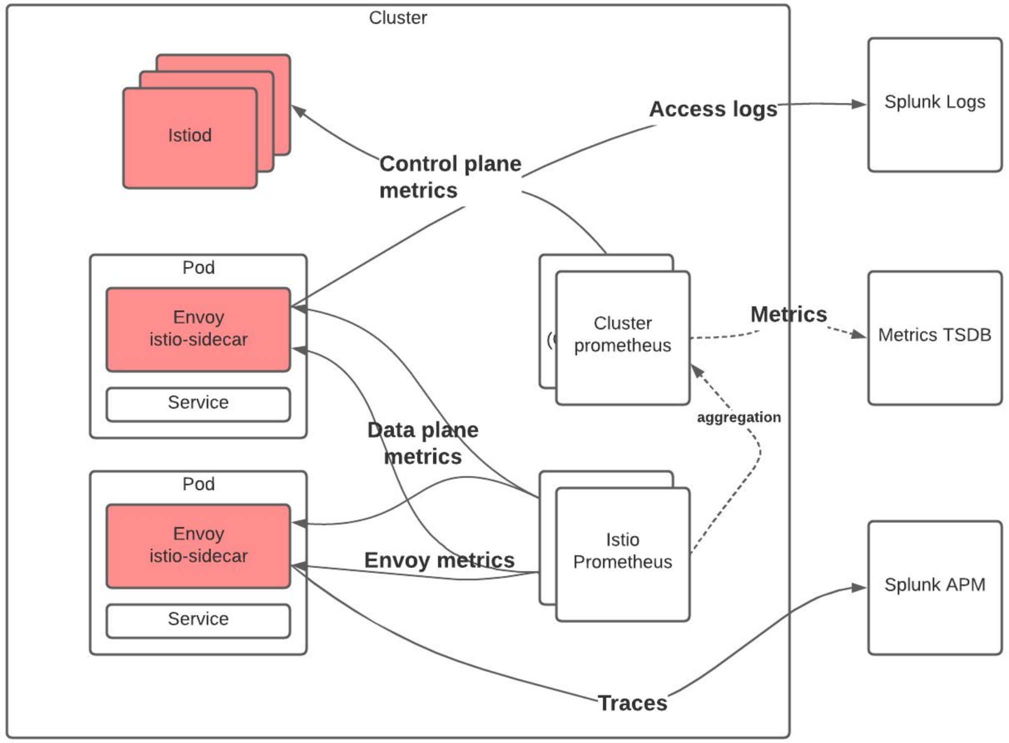


Friday, April 29 (English)

9:00-9:35 **Lessons Learned: Developing WASM filter for logging use-case**
by Amey Bhide & Takeshi Yoneda

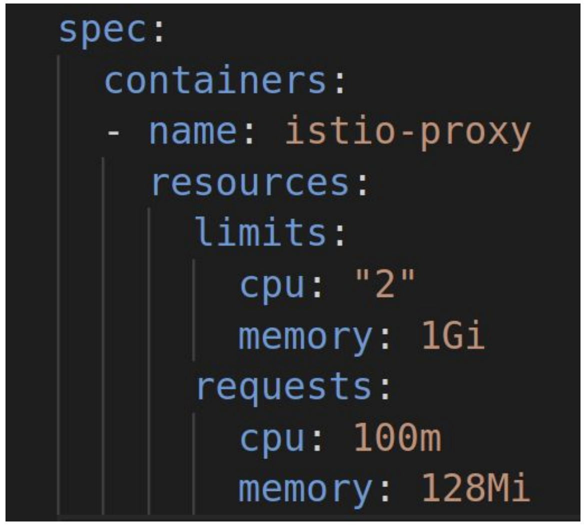
Istio observability

- **Access Logs** sent to … Splunk
- **Data plane metrics** to dedicated Istio prometheus instances
  - Aggregated and sent to cluster prometheus
- **Control plane metrics** sent to cluster prometheus instances
- **Traces** sent to Splunk APM

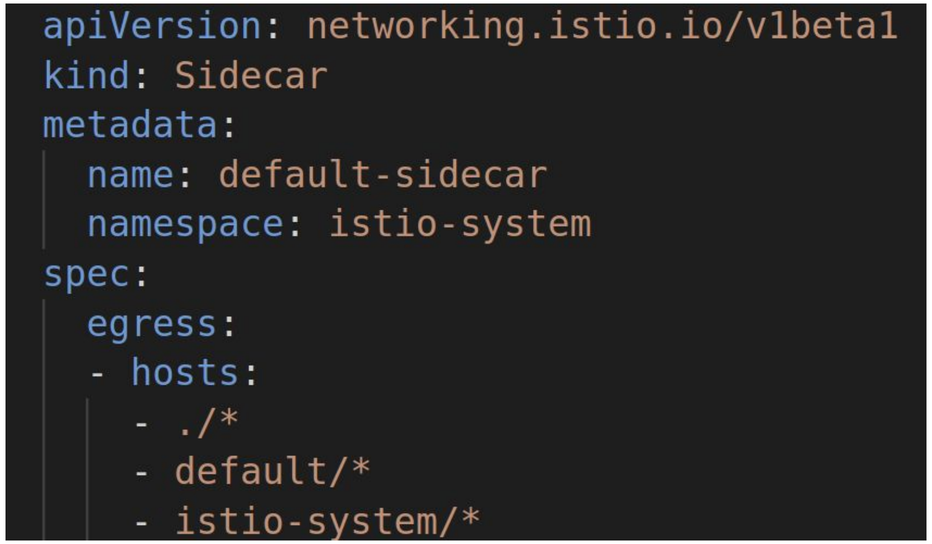


Resource management

- Gateway right-sizing
  - CPU HPA set to ~60%
  - baseline set by historical usage
- Sidecar right-sizing
  - Default sizing
  - Sidecar CRDs for the win



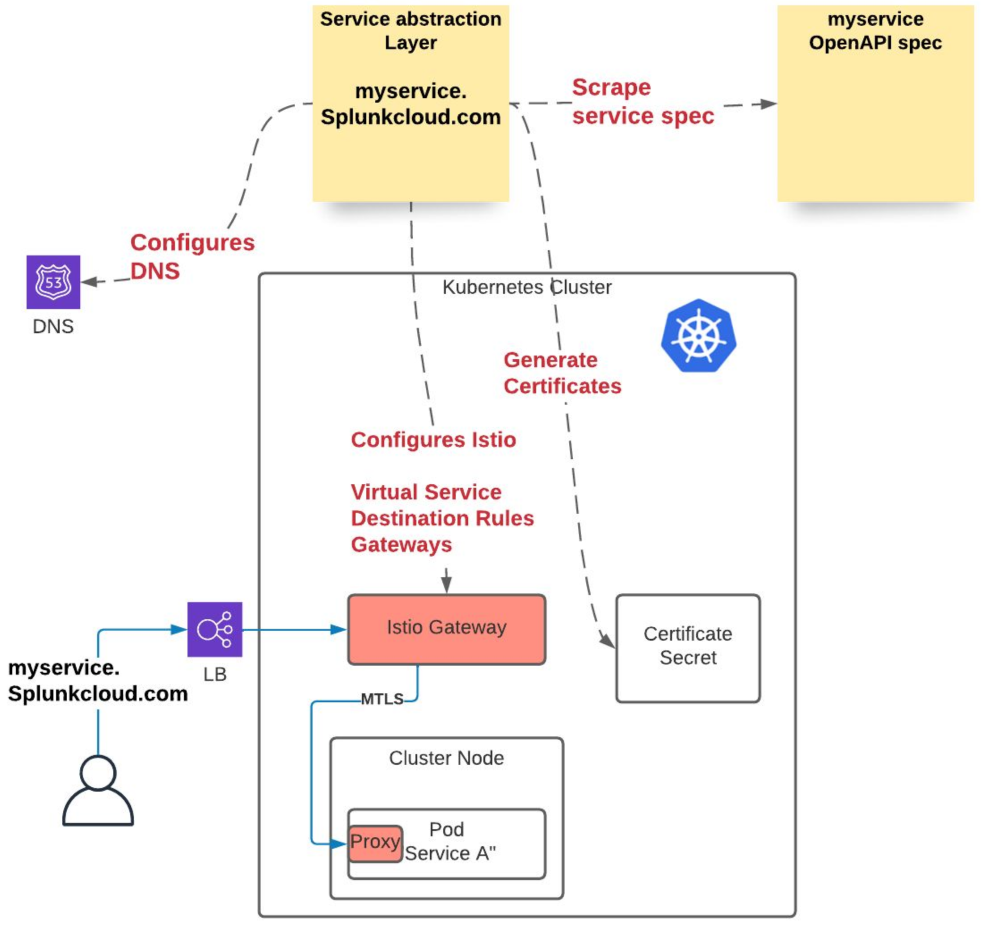
```
spec:
  containers:
  - name: istio-proxy
    resources:
      limits:
        cpu: "2"
        memory: 1Gi
      requests:
        cpu: 100m
        memory: 128Mi
```



```
apiVersion: networking.istio.io/v1beta1
kind: Sidecar
metadata:
  name: default-sidecar
  namespace: istio-system
spec:
  egress:
  - hosts:
    - ./*
    - default/*
    - istio-system/*
```

Service abstraction layer

- "Golden path" abstraction layer for 80% of the use cases
- A single abstraction layer for:
  - VirtualServices, DestinationRules, Gateways and ServiceEntry CRD
  - Certificate management
  - DNS management
- single OpenAPI spec per service
- Abstraction Layer controller scrapes those openAPI specs



Best practices

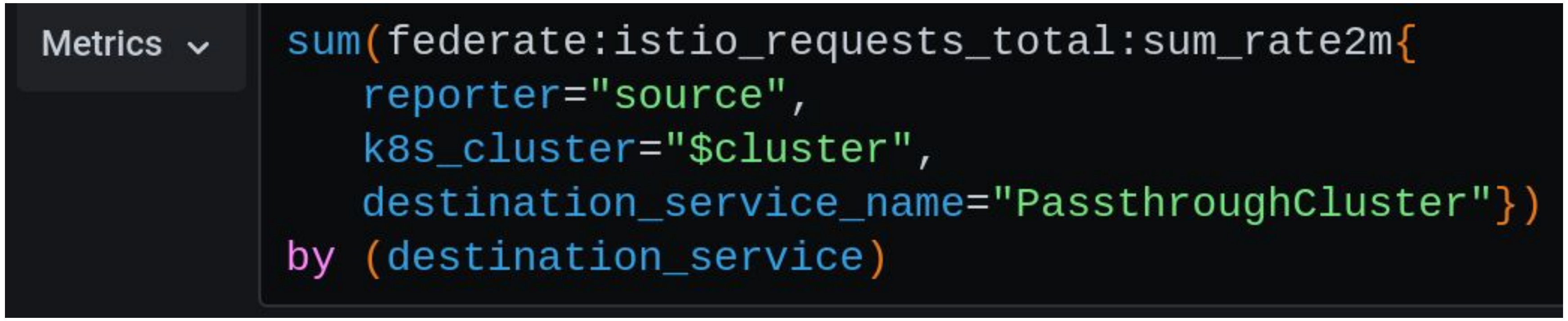
- Naming service ports correctly!
- Deploy Sidecar CRDs per namespace
- Scope internal service with

  `networking.istio.io/exportTo: .`
- Exclude some ports from the mesh with

  `traffic.sidecar.istio.io/excludeInboundPorts:`
- Avoid Headless services as much as possible
- For some workloads, load certificates directly

In-transit encryption

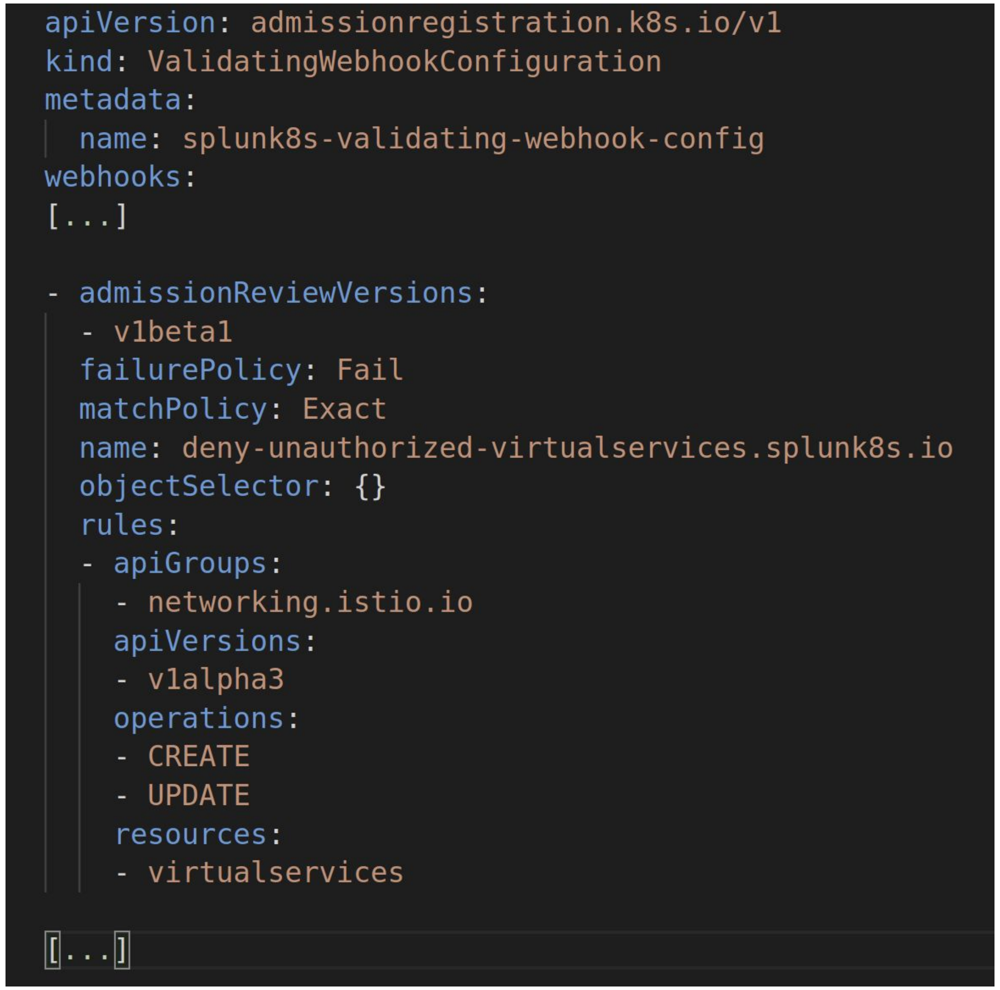
- Run the mesh in permissive mode
- Monitor the passthrough cluster and alert teams not using mTLS
- Consider moving to strict soon



```
sum(federate:istio_requests_total:sum_rate2m{
   reporter="source",
   k8s_cluster="$cluster",
   destination_service_name="PassthroughCluster"})
by (destination_service)
```

Validating Webhook

- Widely used for K8S and Istio
- Validates K8S objects:
  - Service type LB
  - Gateways and VS CRD



```yaml
apiVersion: admissionregistration.k8s.io/v1
kind: ValidatingWebhookConfiguration
metadata:
  name: splunk8s-validating-webhook-config
webhooks:
[...]

- admissionReviewVersions:
  - v1beta1
  failurePolicy: Fail
  matchPolicy: Exact
  name: deny-unauthorized-virtualservices.splunk8s.io
  objectSelector: {}
  rules:
  - apiGroups:
    - networking.istio.io
    apiVersions:
    - v1alpha3
    operations:
    - CREATE
    - UPDATE
    resources:
    - virtualservices

[...]
```

Help our end-user

- 90% of help requests due to 503s from the gateway
  - Runbooks based on Access logs flags
- Debugging workflow
  - Validate Kubernetes configuration
  - Validate Istio configuration
  - Dump Envoy configuration (istioctl)

What’s next

- Multi Cluster service Mesh
    - Global control plane
    - Workload redundancy across clusters
- Advanced traffic engineering:
    - Zone aware routing
    - Blue/Green deployments

Istio

What we learned

- Istio only make sense for specific requirements, at a specific scale
- Incremental stability improvement over the last versions
- Understanding Envoy is **CRITICAL**
- Your users **don’t want to learn** another set of CRDs. Consider an abstraction layer
- Don’t underestimate how much it takes to “Keep the lights on” (Upgrading, Helping users, etc)

Thank You!

(We are hiring)

splunk> turn data into doing®