Building Simplified Service Mesh APIs for Developers

Lin Sun & Ying Zhu

Lin Sun

Director of Open Source, Solo.io

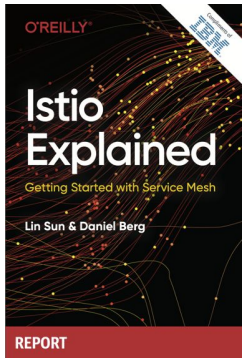
@linsun_unc

lin.sun@solo.io

linkedin.com/pub/lin-sun/1/…

6500+ contributions

TOC & Steering Member



Ambassador

207
IBM Patents

Ying Zhu

Infrastructure Engineer, Airbnb

@ying_95z

ying.zhu@airbnb.com

linkedin.com/in/ying-zhu-763a3879/

AirMesh

Airbnb's next generation service mesh based on Istio

Agenda

- Why? - what problem we are trying to solve
- How? - our approach in solving the problem
  - Airbnb's story
  - Solo's story
  -
  - ...

Why

Airbnb Scale

30+
Clusters

1k+
Services

20k+
Pods

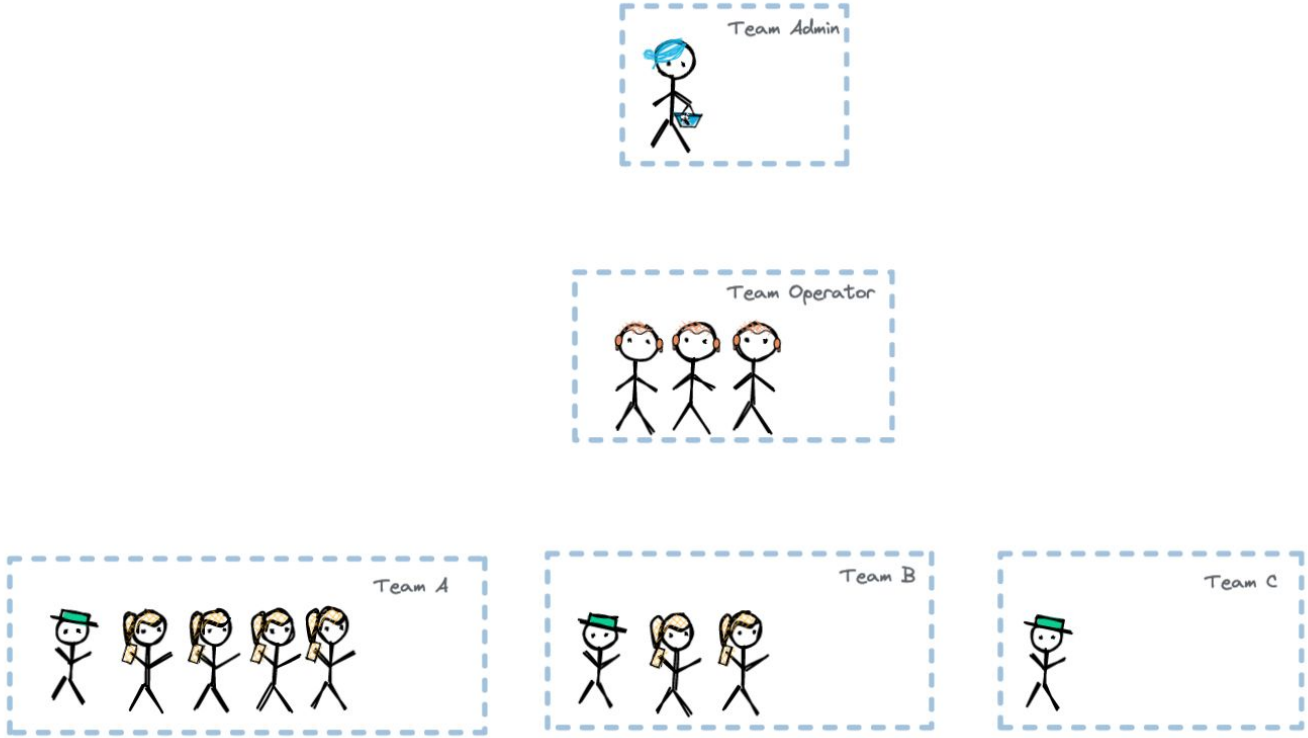
2021 IstioCon “Airbnb on Istio”

Istio API is complex **and** evolving.

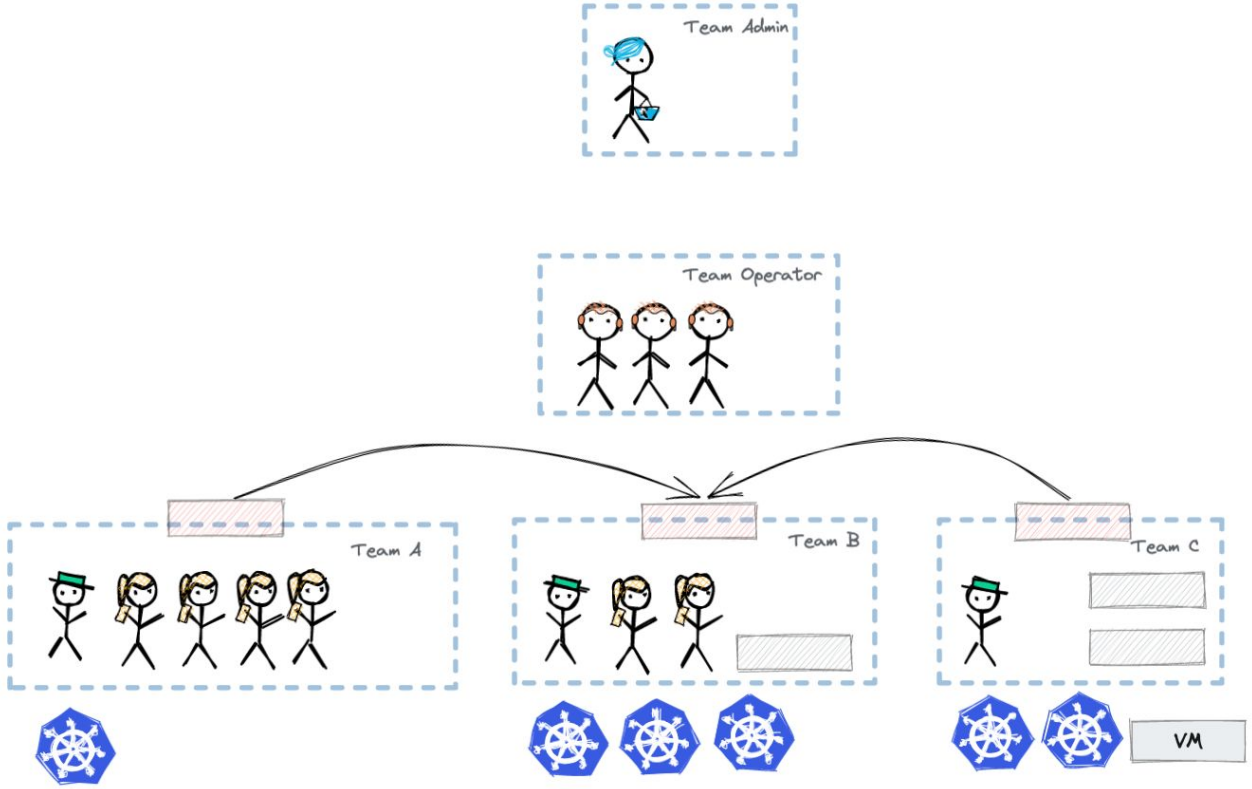
Istio API is complex and evolving.

At Airbnb, we believe that product engineers should focus on improving the product for our users, instead of keeping up with the underlying infra changes.

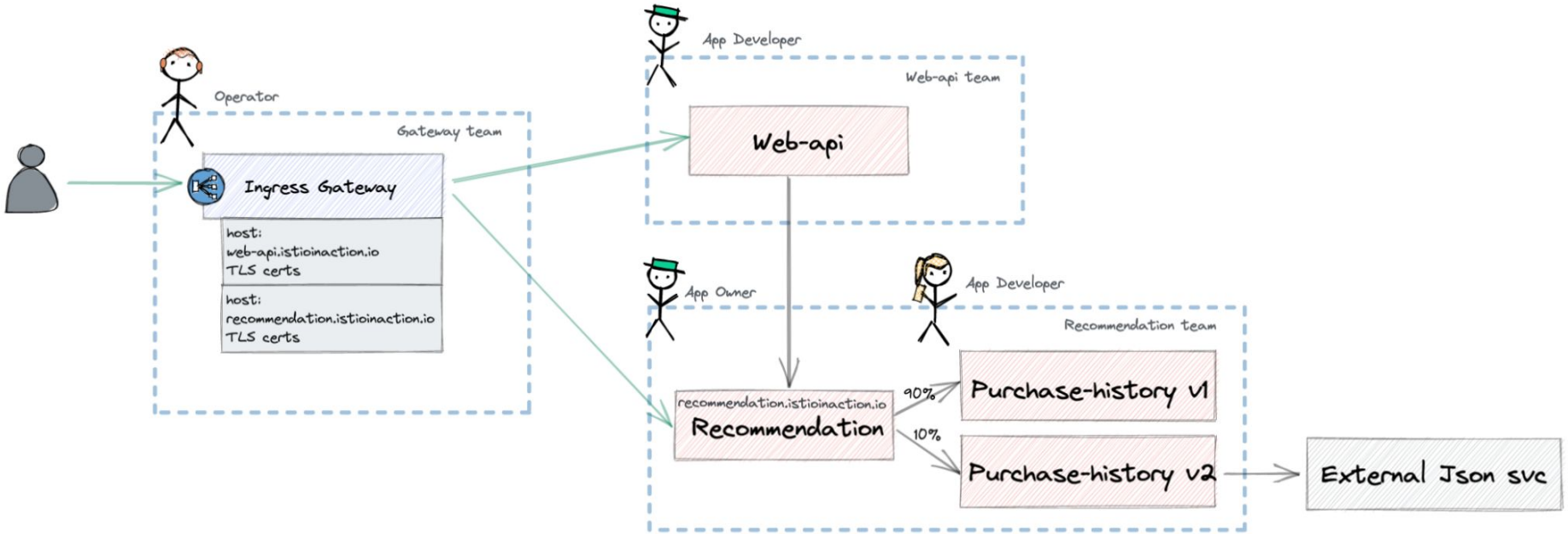
Teams



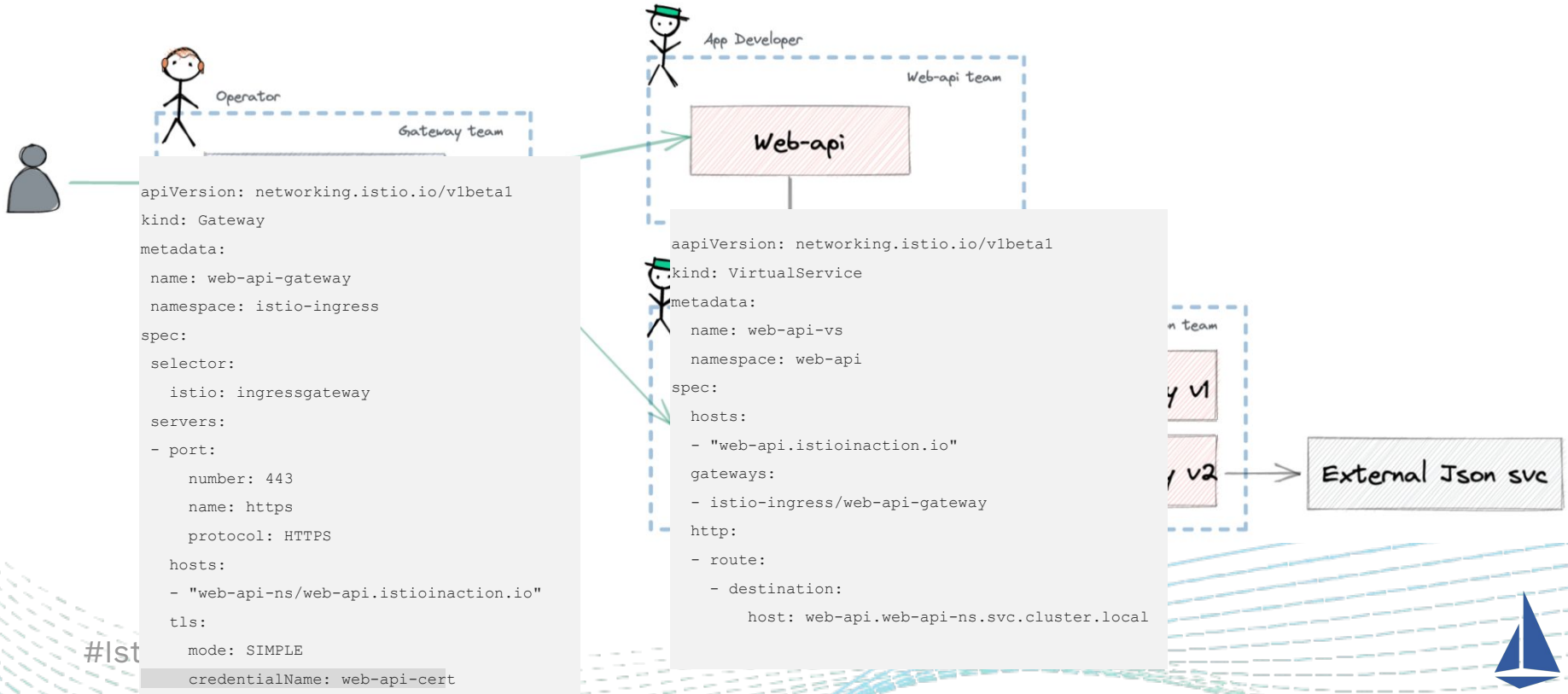
Services



Examples


Operator
Gateway team
Ingress Gateway
host:
web-api.istioinaction.io
TLS certs
host:
recommendation.istioinaction.io
TLS certs
App Developer
Web-api team
Web-api
App Owner
App Developer
Recommendation team
recommendation.istioinaction.io
Recommendation
90%
10%
Purchase-history v1
Purchase-history v2
External Json svc

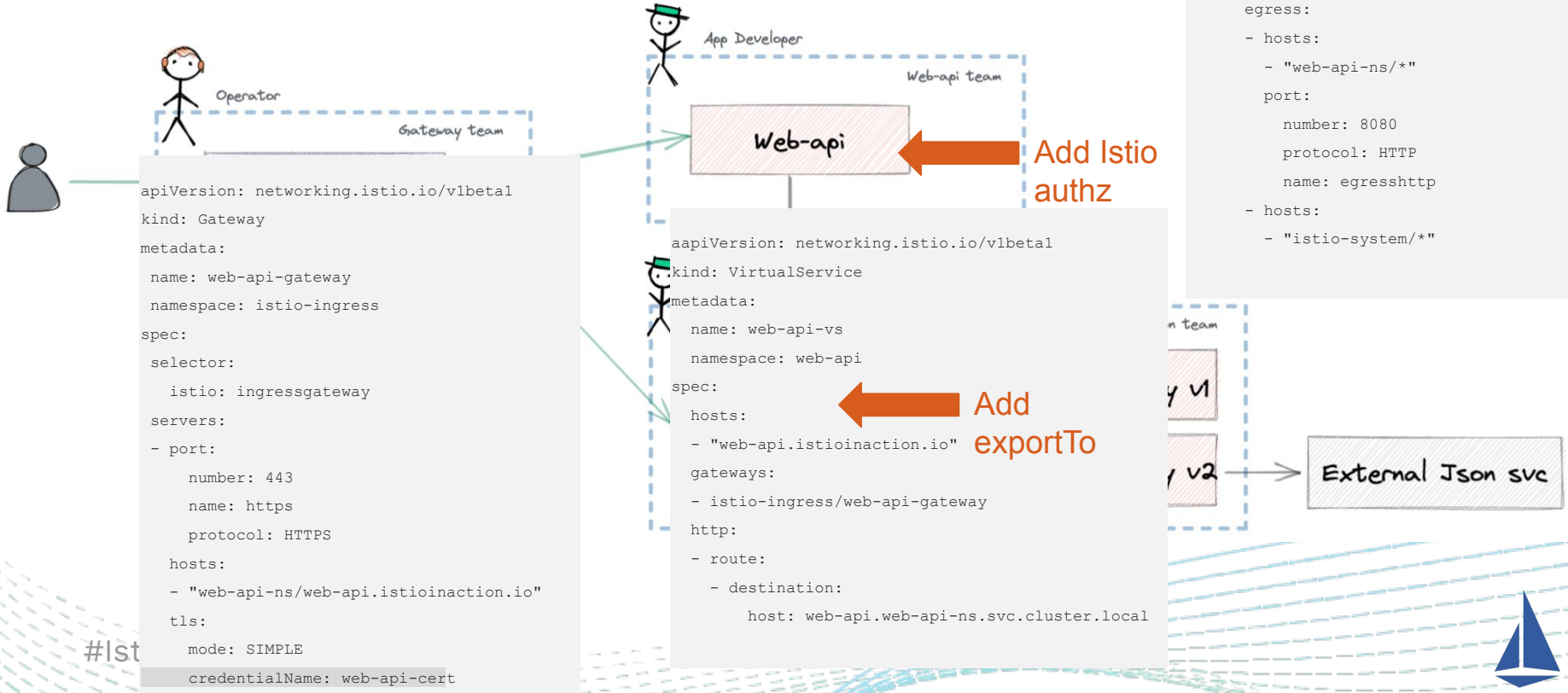
With Istio




```
apiVersion: networking.istio.io/v1beta1
kind: Gateway
metadata:
 name: web-api-gateway
 namespace: istio-ingress
spec:
 selector:
   istio: ingressgateway
 servers:
 - port:
     number: 443
     name: https
     protocol: HTTPS
   hosts:
   - "web-api-ns/web-api.istioinaction.io"
   tls:
     mode: SIMPLE
     credentialName: web-api-cert
```

```
aapiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: web-api-vs
  namespace: web-api
spec:
  hosts:
  - "web-api.istioinaction.io"
  gateways:
  - istio-ingress/web-api-gateway
  http:
  - route:
    - destination:
        host: web-api.web-api-ns.svc.cluster.local
```

What is missing?




```
apiVersion: networking.istio.io/v1beta1
kind: Gateway
metadata:
 name: web-api-gateway
 namespace: istio-ingress
spec:
 selector:
   istio: ingressgateway
 servers:
 - port:
     number: 443
     name: https
     protocol: HTTPS
   hosts:
   - "web-api-ns/web-api.istioinaction.io"
   tls:
     mode: SIMPLE
     credentialName: web-api-cert
```

```
aapiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: web-api-vs
  namespace: web-api
spec:
  hosts:
  - "web-api.istioinaction.io"
  gateways:
  - istio-ingress/web-api-gateway
  http:
  - route:
    - destination:
        host: web-api.web-api-ns.svc.cluster.local
```

```
apiVersion:
networking.istio.io/v1beta1
kind: Sidecar
metadata:
 name: default
spec:
 egress:
 - hosts:
   - "web-api-ns/*"
   port:
     number: 8080
     protocol: HTTP
     name: egresshttp
 - hosts:
   - "istio-system/*"
```

Istio Quiz Time!

HTTP retry and timeout, which resource?

TCP connect timeout and keepalive, which resource?

Outlier detection?

RateLimit?

How to make this simple for our users?

Salesforce

Helm starter Istio

An Istio optimized Helm starter

- Mesh-service
  - Everything you need to run a service in the mesh
- Ingress-service
  - Add's ingress gateway configuration to mesh-service
- Mesh-egress
  - Configures TLS egress and policy
- Auth-policy
  - Configures mTLS authorization policy

https://github.com/salesforce/helm-starter-istio

helm-starter-istio

An Istio starter template for Helm.

Stop fiddling with Istio and Kubernetes YAML and start building. This starter sets up everything you need to get a container running in Istio correctly the first time.

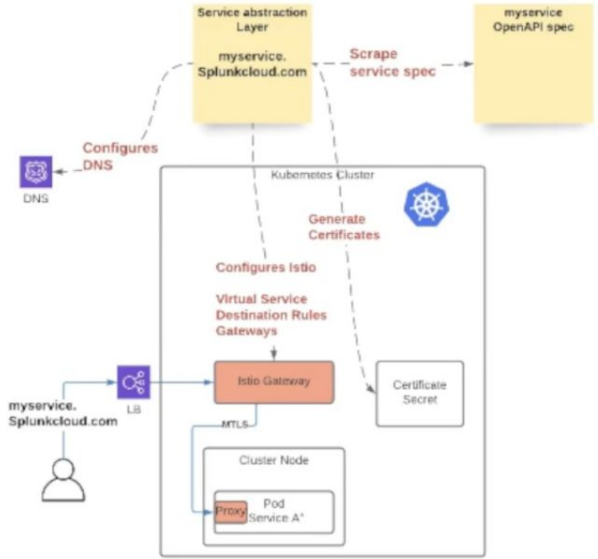
Features

- Fastest way to get a new service into the Istio mesh
- Simplified Istio ingress configuration
- Simplified Istio port configuration
- ConfigMap driven by `values.yaml`, to facilitate easy Helm value overriding
- Creates the following Kubernetes and Istio objects
  - Service
  - Deployment
  - ConfigMap (optional)
  - VirtualService
  - DestinationRule
  - PodDisruptionBudget
  - HorizontalPodAutoscaler (optional)
  - ServiceAccount (optional)

Splunk

Service abstraction layer

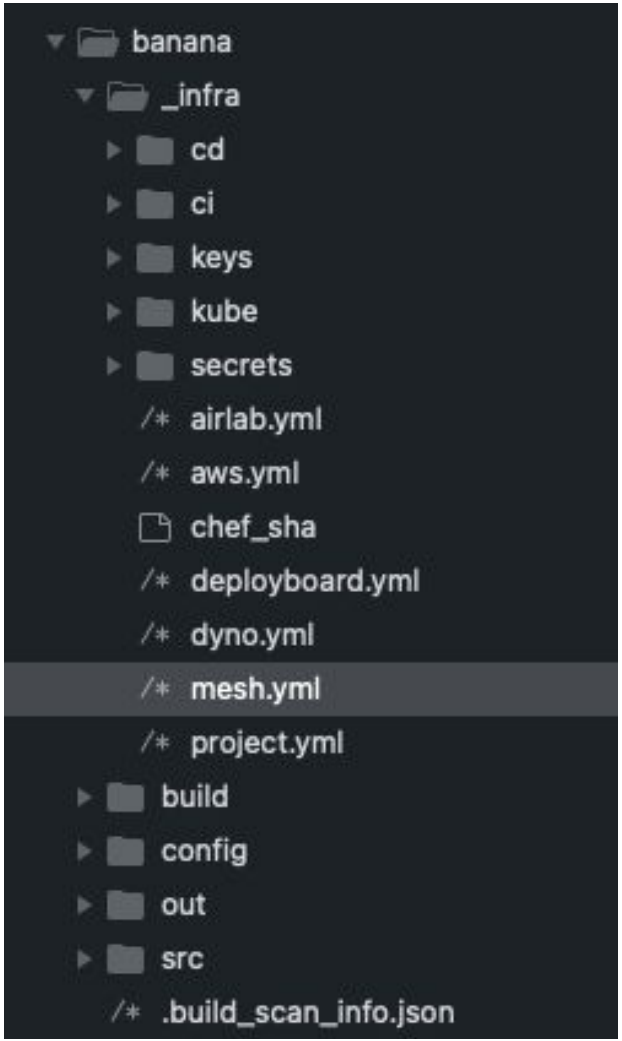
- "Golden path" abstraction layer for 80% of the use cases
- A single abstraction layer for:
  - VirtualServices, DestinationRules, Gateways and ServiceEntry CRD
  - Certificate management
  - DNS management
- single OpenAPI spec per service
- Abstraction Layer controller scrapes those openAPI specs



Airbnb

mesh.yml

defined together with each service



mesh.yml

follows AirMesh API

```yaml
apiVersion: v1beta1

banana-canary:
  extends: banana-production

banana-canary-baseline:
  extends: banana-production

banana-production:
  type: service
  mesh: ea1.us
  proxyConfig:
    cpuRequest: 800m
    memoryRequest: 256Mi
    memoryLimit: 256Mi
  dependentServices:
  - host: sitar-service-production.sitar-service-production
  ports:
    app:
      protocol: http
      number: 5070
      authz:
        all:
          allowList:
          - dealer-production.dealer-production
      serverOptions:
        mirror:
          host: postverta-staging.postverta-staging.svc.ea1.us.a
          port: 11862
          percent: 1
```

mesh.yml

convert during CI

Istio CRs

- DestinationRule
- VirtualService
- AuthorizationPolicy
- Sidecar
- …

mesh.yml

convert during CI

Istio CRs

- DestinationRule
- VirtualService
- AuthorizationPolicy
- Sidecar
- …

deploy into k8s cluster

Istio API is feature based

AirMesh API is workload based

AirMesh object types:

- App
- Service
- VMApp
- VMService
- External
- …


dependency
authz
connection pool settings
outlier detection
timeout and retry
banana-production:
type: service
mesh: eai.us
dependentServices:
- host: "foo-production.foo-production"
ports:
banana-production:
number: 5070
protocol: http
authz:
- allows:
- "bar-production.bar-production"
- "*.baz-production"
serverOptions:
connectionPool:
maxPendingRequests: 1024
clientOptions:
outlierDetection:
consecutive5xxErrors: 2
consecutiveGatewayErrors: 1
ejectionInterval: 2s
minEjectionTime: 10ms
requestOptions:
timeout: 5s
retry:
attempts: 5

External Service

```
banana-db:
  type: external
  mesh: eal.us
  namespace: mysql-production
  endpoints:
  - name: banana-01
    address: aurora-banana-01.cluster-fake.us-east
    locality: us-east-1/us-east-1a
  - name: airmaster-02
    address: aurora-banana-02.cluster-fake.us-east
    locality: us-east-1/us-east-1b
  labels:
    tier: production
  healthCheckedBy: mysql-hc
  ports:
    banana:
      number: 3306
      protocol: tcp
      clientOptions:
        tls:
          mode: DISABLE
        outlierDetection:
          consecutiveConnectionErrors: 2
          ejectionInterval: 2s
          minEjectionTime: 10ms
```

To reduce verbosity, extension and override feature is provided in AirMesh API.

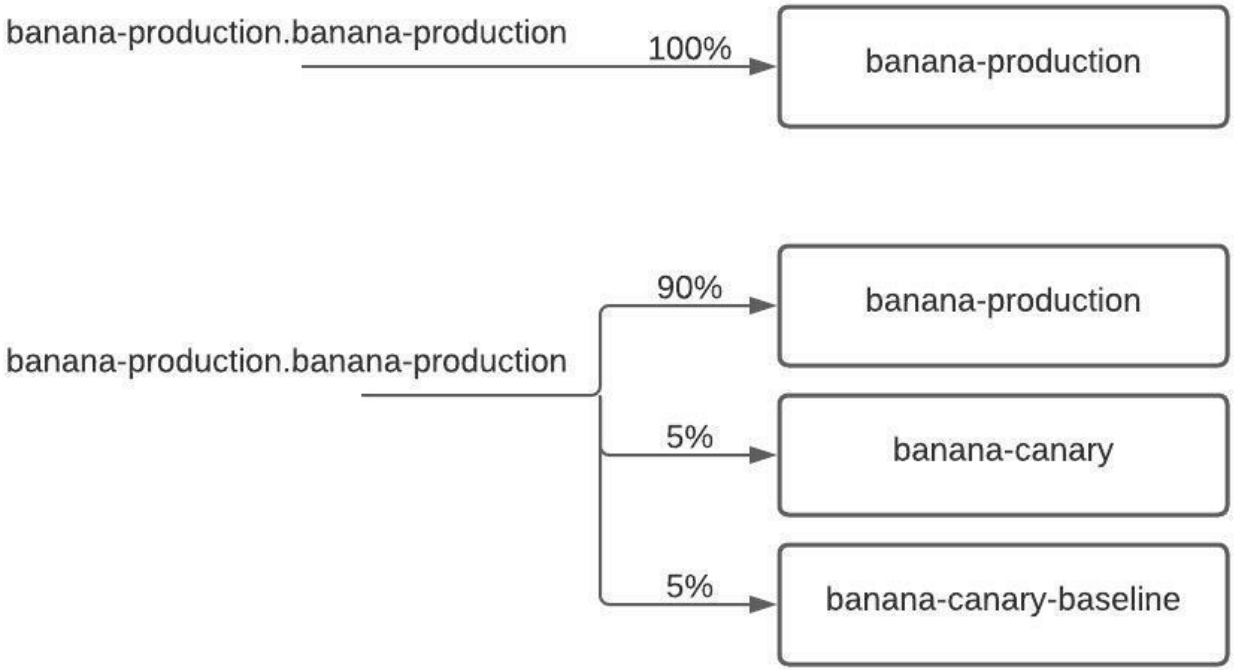
extension

```
apiVersion: v1beta1

banana-canary:
  extends: banana-production

banana-canary-baseline:
  extends: banana-production

banana-production:
  type: service
  mesh: ea1.us
  proxyConfig:
    cpuRequest: 800m
    memoryRequest: 256Mi
    memoryLimit: 256Mi
  dependentServices:
  - host: sitar-service-production.sitar-service-production
  ports:
    app:
      protocol: http
      number: 5070
      authz:
        all:
```

```yaml
production:
  type: service
  mesh: ea1.us
  namespace: app-with-extend-production
  name: app-with-extend-production
  dependentServices:
    - host: foo-production.foo-production
    - host: foo-canary.foo-canary
  ports:
    port:
      number: 8080
      protocol: http
      authz:
        - allows:
            - airbnb-admin.airbnb-admin
            - bar-production.bar-production
            - bar-canary.bar-canary

app-with-extend-canary:
  extends: production
  ports:
    port:
      number: 8081
```

override

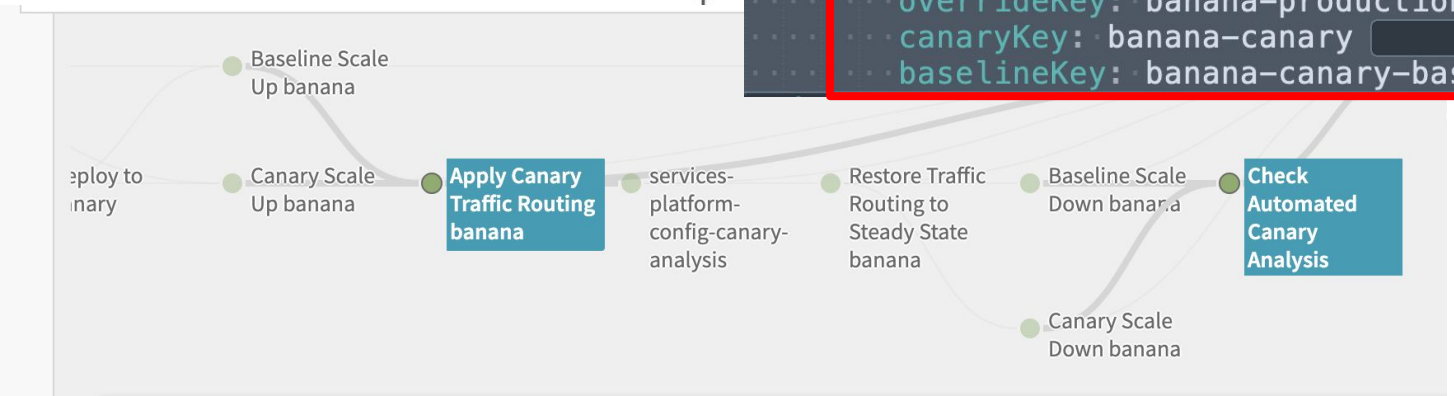
Traffic Routing (ACA)



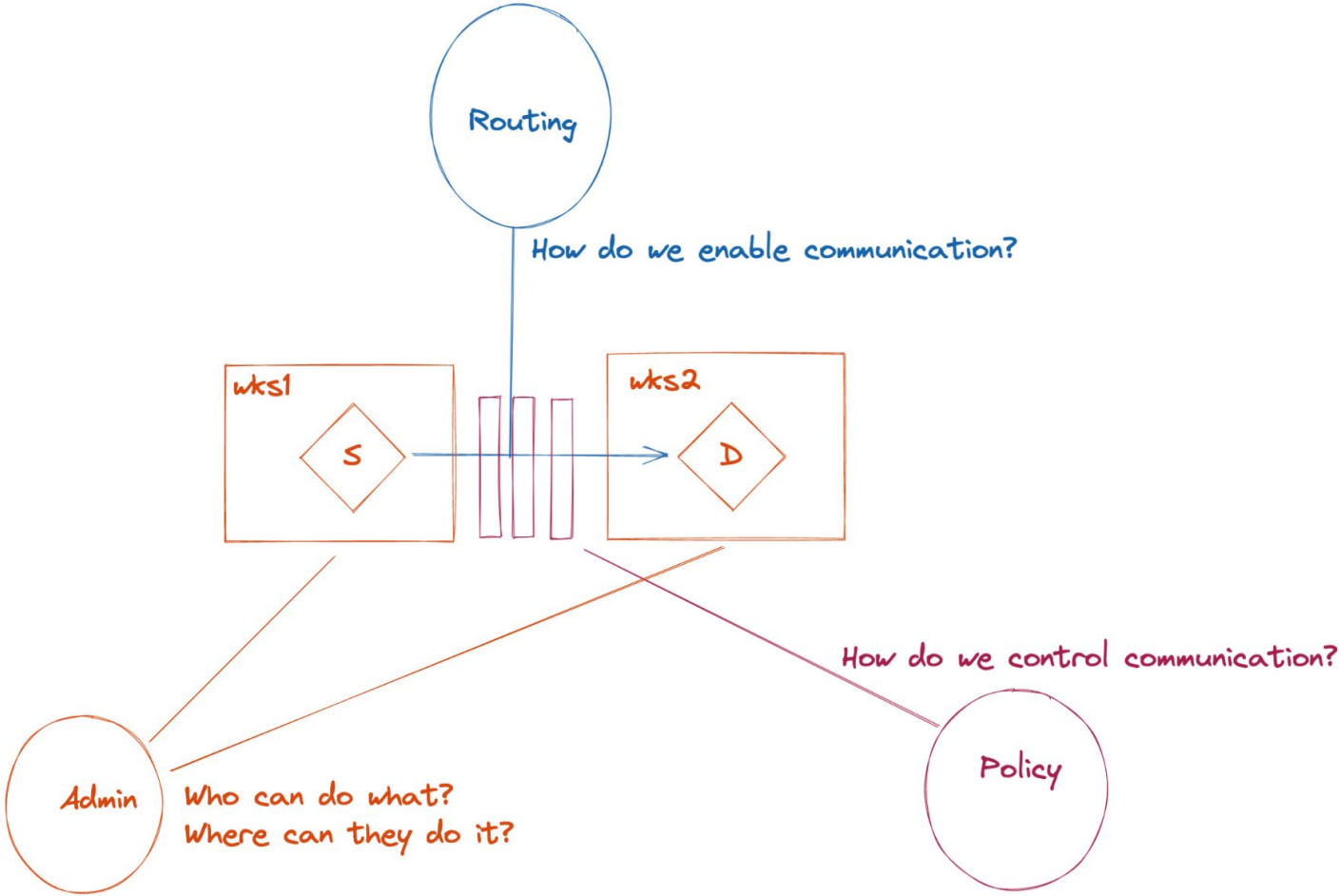
Traffic Routing

banana-production.banana-production

```yaml
- name: aca
  type: automatedCanaryAnalysis
  title: aca
  parameters:
    canaryConfigs:
    - name: services-platform-config
      scoreThreshold: 75
      durationMinutes: 30
      delayMinutes: 5
      intervalMinutes: 10
    scaling:
      replicas: 2
      meshContext: mesh-mgmt-a-ca1-us
      percentTrafficPerEnv: 5
      meshKeys:
        overrideKey: banana-production
        canaryKey: banana-canary
        baselineKey: banana-canary-basel
```



Solo.io

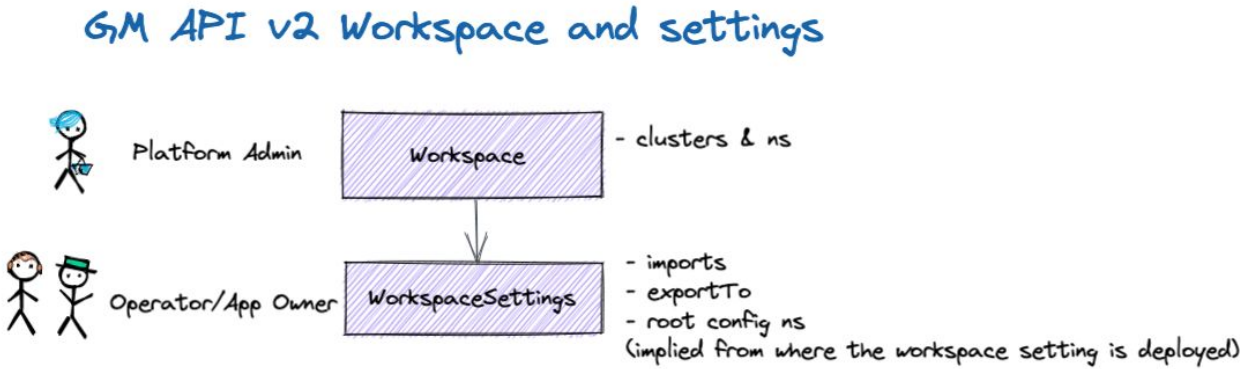

Routing
How do we enable communication?
wks1
S
wks2
D
How do we control communication?
Admin
Who can do what?
Where can they do it?
Policy

Listening to our users

- Multi-tenancy and isolation among teams
- Application centric approach vs cluster centric approach
- Simple policy reuse
- Delegate as much as possible

Workspace

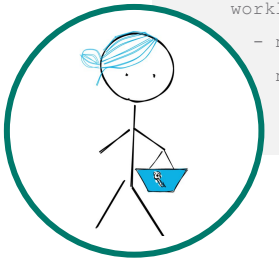
- A ***workspace*** is a logical boundary for a team
- Provides the foundation for multi-tenancy features
- Makes it easy to onboard teams
- Same UX for single cluster and multi-cluster



Pam onboards new teams

Team per cluster

```
apiVersion: admin.gloo.solo.io/v2
kind: Workspace
metadata:
  name: ratings
  namespace: gloo-mesh
  labels:
    team: ratings
    gloo.solo.io/exportToGateway:
tier1
spec:
  workloadClusters:
    - name: cluster1
      namespaces:
        - name: *
```



Team per namespaces

```
apiVersion: admin.gloo.solo.io/v2
kind: Workspace
metadata:
  name: recommendation
  namespace: gloo-mesh
  labels:
    team: recommendation
    gloo.solo.io/exportToGateway:
tier1
spec:
  workloadClusters:
    - name: cluster2
      namespaces:
        - name: recommendation
```

Dynamic team

```
apiVersion: admin.gloo.solo.io/v2
kind: Workspace
metadata:
  name: web-api
  namespace: gloo-mesh
  labels:
    team: web-api
    gloo.solo.io/exportToGateway:
tier1
spec:
  workloadClusters:
    - selector:
        region: us-east
      namespaces:
        - name: web*
```

Pam onboards the gateway team

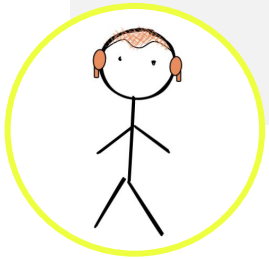
Workspace

```
apiVersion: admin.gloo.solo.io/v2
kind: Workspace
metadata:
  name: n-s-gateway
  namespace: gloo-mesh
  labels:
    team: n-s-gateway
spec:
  workloadClusters:
    - name: cluster-gateway
      namespaces:
        - name: istio-n-s-gateway
```

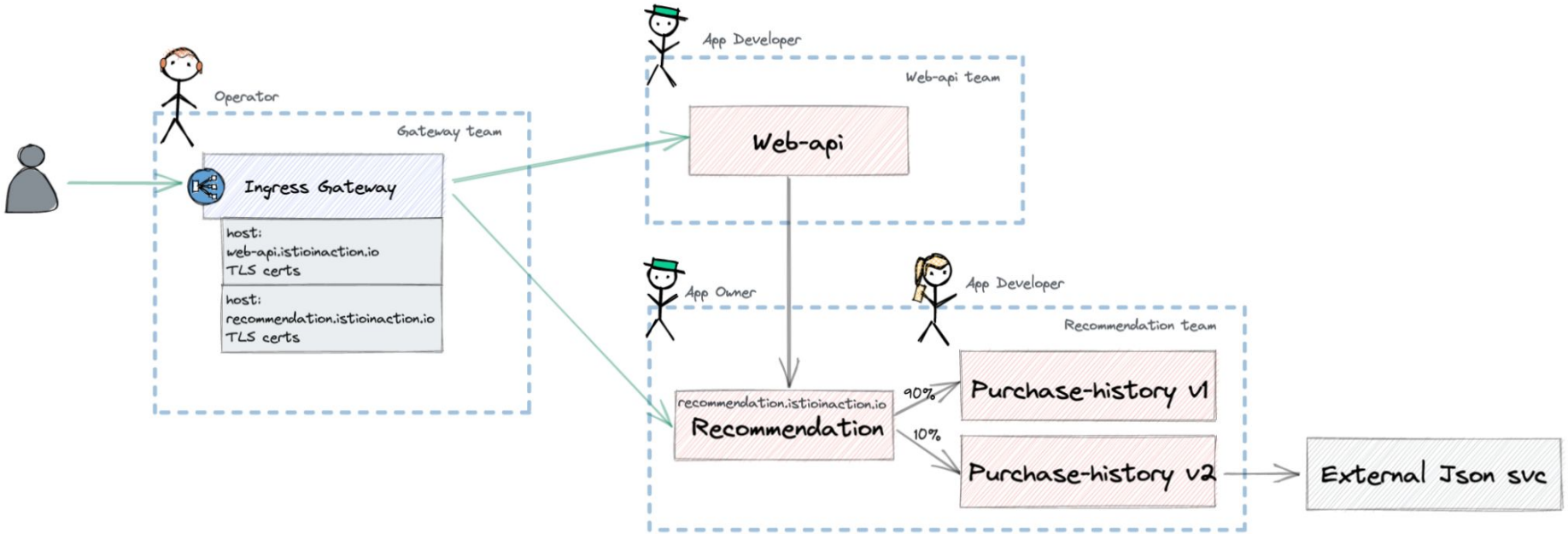
Oliver defines team settings

WorkspaceSettings

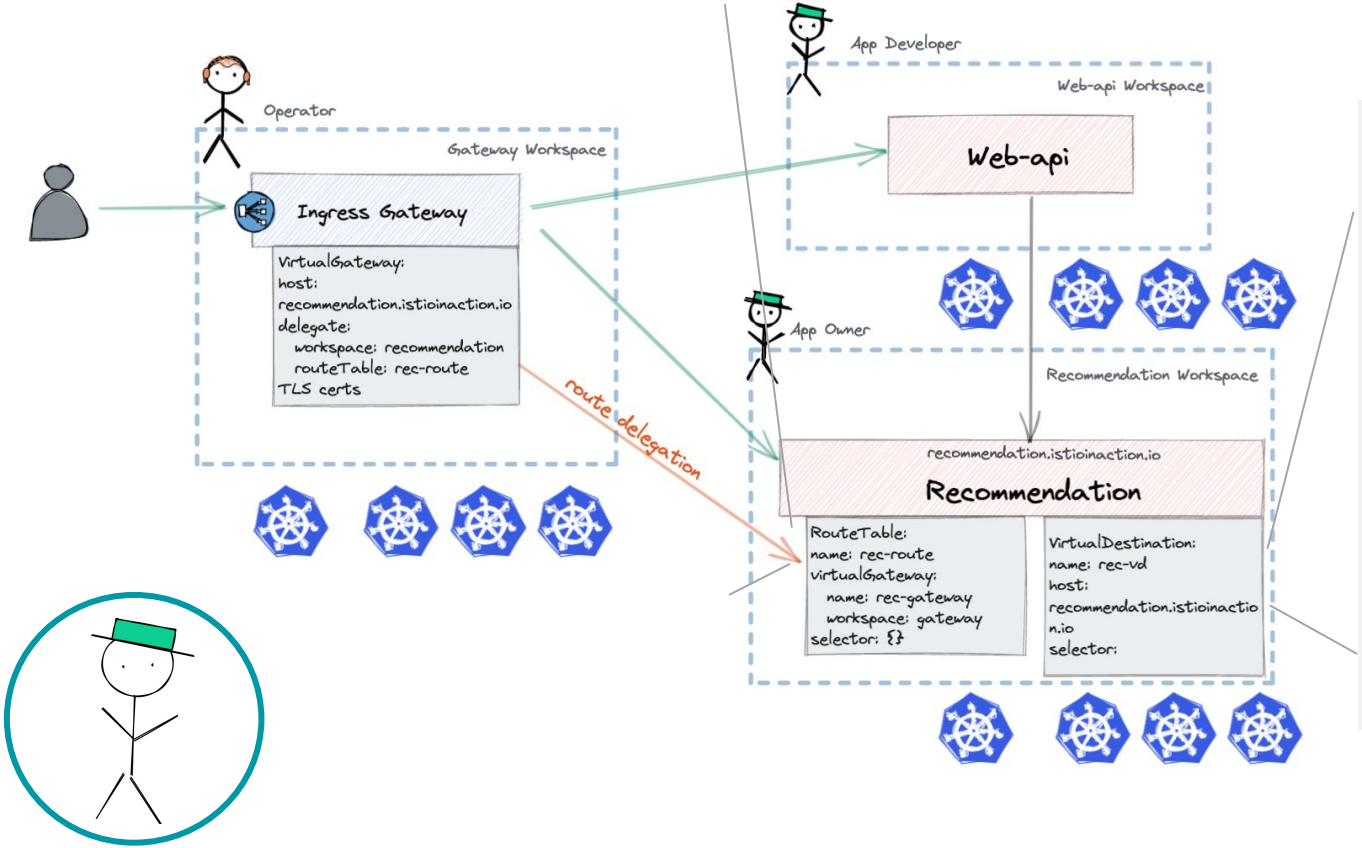
```
apiVersion: admin.gloo.solo.io/v2
kind: WorkspaceSettings
metadata:
  name: n-s-gateway
  namespace: istio-n-s-gateway
spec:
  imports:
    - selector:
        gloo.solo.io/exportToGateway: tier1
```



Examples

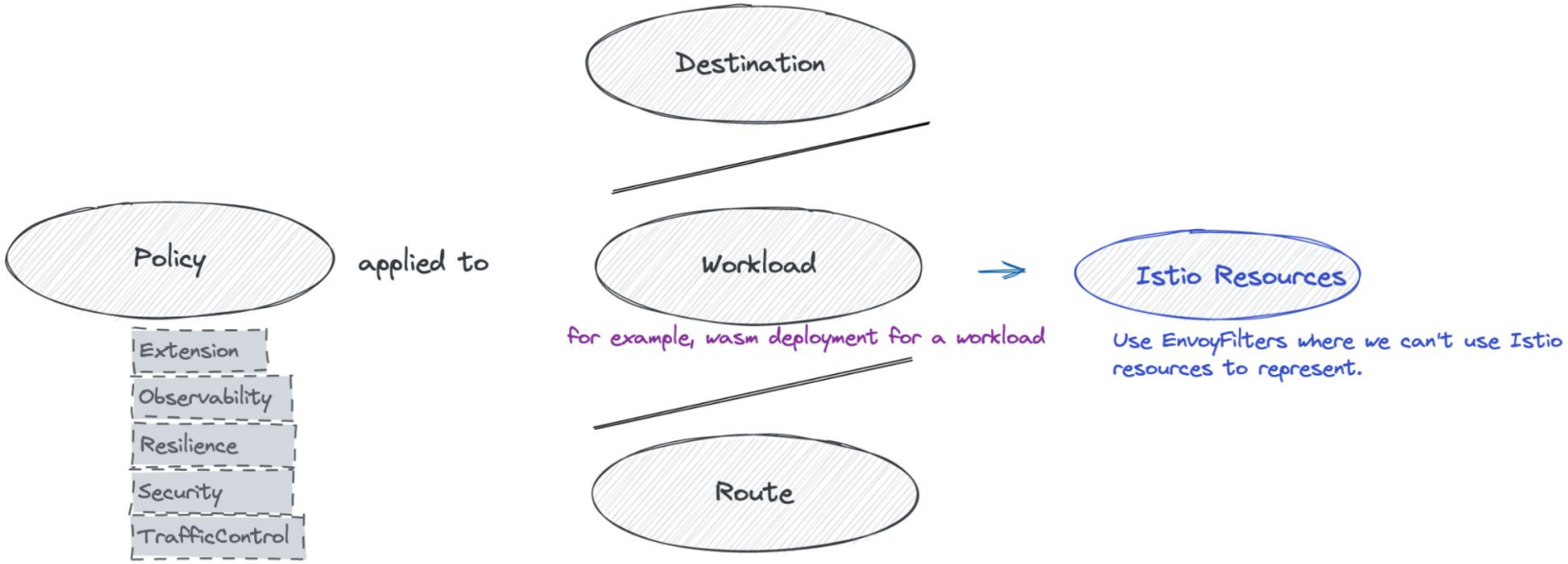

Operator
Gateway team
Ingress Gateway
host:
web-api.istioinaction.io
TLS certs
host:
recommendation.istioinaction.io
TLS certs
App Developer
Web-api team
Web-api
App Owner
App Developer
Recommendation team
recommendation.istioinaction.io
Recommendation
90%
10%
Purchase-history v1
Purchase-history v2
External Json svc

Virtual destination




```yaml
apiVersion: networking.gloo.solo.io/v2
kind: VirtualDestination
metadata:
  name: rec-vd
  namespace: recommendation-root-ns
spec:
  hosts:
    - 'recommendation.istioinaction.io'
  services:
    - labels:
        app: recommendation
      namespace: recommendation-ns
      cluster: cluster1 # optional
  ports:
    - number: 9080
```

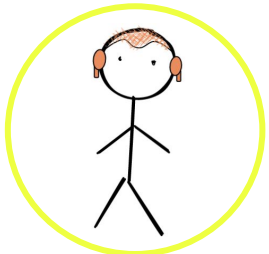
Policies

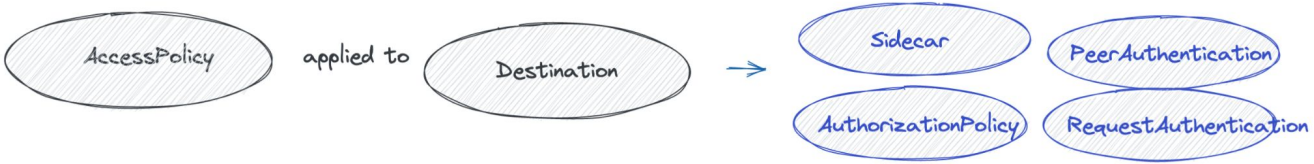

Destination
Policy
applied to
Workload
Istio Resources
for example, wasm deployment for a workload
Use EnvoyFilters where we can't use Istio resources to represent.
Extension
Observability
Resilience
Security
TrafficControl
Route

Access Policy

```
apiVersion:
security.policy.gloo.solo.io/v2
kind: AccessPolicy
metadata:
 name: access-policy
spec:
 applyToDestinations:
 - selector:
     labels:
       strict: enabled
 config:
   authn:
     tlsMode: STRICT
```

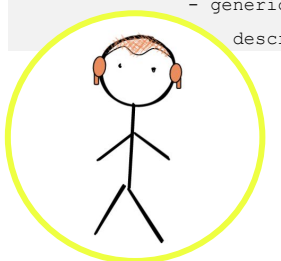
```
apiVersion:
security.policy.gloo.solo.io/v2
kind: AccessPolicy
metadata:
 name: access-policy
spec:
 applyToDestinations:
 - selector:
     labels:
       app: recommendation
 config:
   authn:
     tlsMode: STRICT
   authz:
     allowedClients:
       - serviceAccountSelector:
           labels:
             app: web-api
       - serviceAccountSelector:
           labels:
             istio: ingressgateway
```





RateLimiting Policy

```
apiVersion: trafficcontrol.policy.gloo.solo.io/v2
kind: RateLimitPolicy
metadata:
  name: 100-req-per-min-policy
  namespace: bar-ns
spec:
  # applies rules to route/destination
  applyToRoutes:
    - route:
        labels:
          ratelimit: 100-req-per-min
  config:
    ratelimitServerConfig:
      namespace: gloo-mesh-addons
      name: rl-server-config
    raw:
      ratelimits:
        - actions:
            - genericKey:
                descriptorValue: counter
```



```
apiVersion: networking.gloo.solo.io/v2
kind: RouteTable
metadata:
  name: rec-routes
  namespace: recommendation-root-ns
spec:
  hosts:
   - 'recommendation.istioinaction.io'
  virtualGateways:
    - name: my-gateway
      workspace: n-s-gateway
  selector: {}
  http:
    - name: basic-route
      forwardTo:
        destinations:
          - name: recommendation
            namespace: recommendation-ns
            port:
              number: 9080
```

Add labels

```
apiVersion: networking.gloo.solo.io/v2
kind: RouteTable
metadata:
  name: rec-routes
  namespace: recommendation-root-ns
spec:
  hosts:
   - 'recommendation.istioinaction.io'
  virtualGateways:
    - name: my-gateway
      workspace: n-s-gateway
  workloadSelectors:
    - {}
  http:
    - name: basic-route
      labels:
        ratelimit: 100-req-per-min
      forwardTo:
        destinations:
          - name: recommendation
            namespace: recommendation-ns
            port:
              number: 9080
```

Conclusion

- A **simplified, opinionated, user-friendly** API that suits your company's specific needs greatly help with Istio adoption.
- Service owners **don't want to learn** another set of CRDs.

Thank you!

Director of Open Source, Solo.io

@linsun_unc

lin.sun@solo.io

linkedin.com/pub/lin-sun/1/…

Infrastructure Engineer, Airbnb

@ying_95z

ying.zhu@airbnb.com

linkedin.com/in/ying-zhu-763a3879/