



solo.io

Envoy, Service Mesh, and Beyond

Oct, 2019

Agenda

- | | | | |
|----|-------------------|----|--|
| 01 | Who is Solo.io | 04 | Service Mesh Interface |
| 02 | How Solo.io helps | 05 | Service Mesh at Scale |
| 03 | Enterprise Envoy | 06 | Solo.io Vision
(Kubecon and beyond) |



The Transformation Journey with Solo.io

Solo.io connects the world's applications with APIs and service mesh across any infrastructure. Simplify the journey to a cloud-native future with the flexibility and control to digitally transform at the pace of your business.



Founded

2017



Based

Cambridge, MA



Funding

15M

THE TRANSFORMATION JOURNEY



MODERNIZATION



SCALING



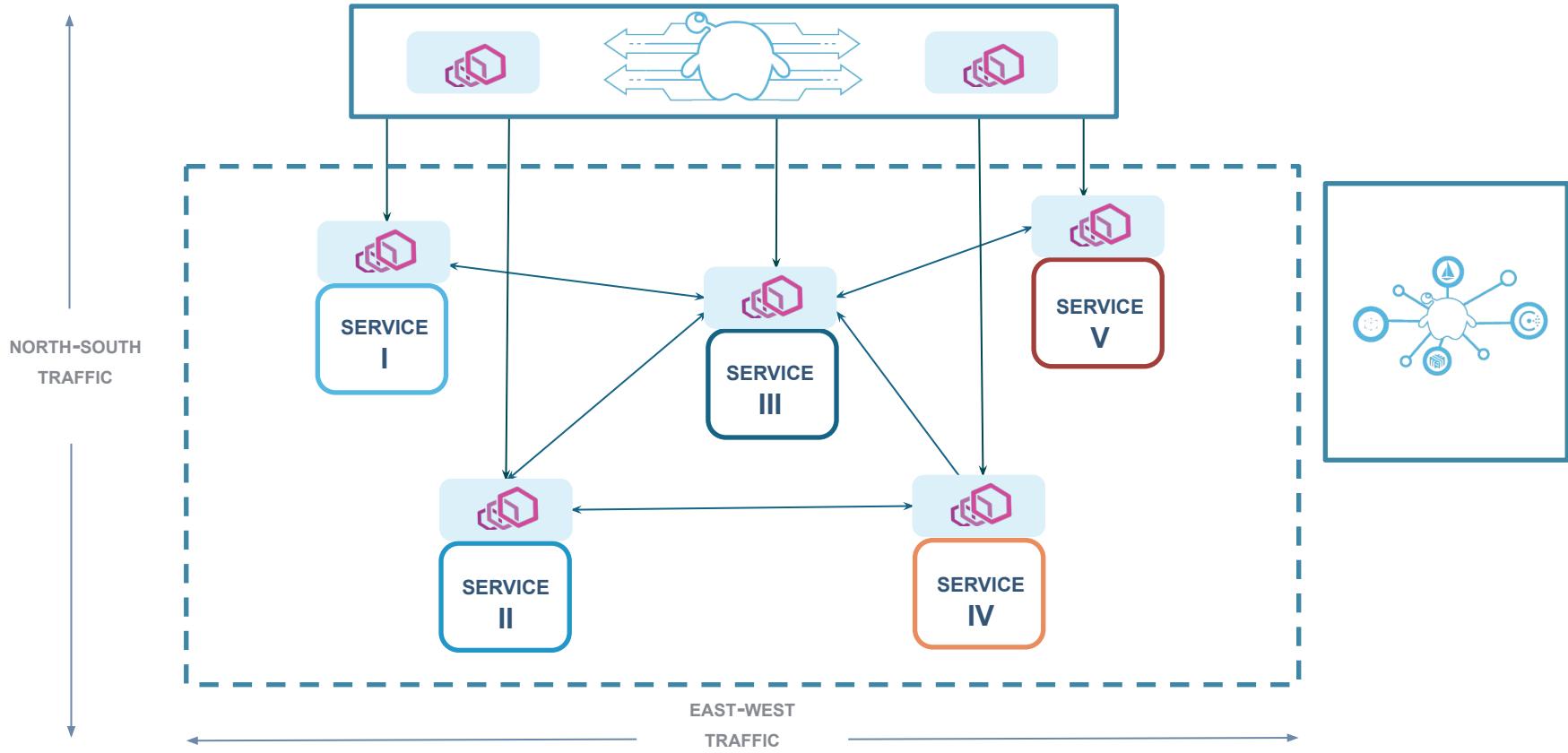
APPLICATION HEALTH

INNOVATION

Why Solo.io

- Better control over network routing, visibility, and security at multiple levels
- Crawl, walk, run strategy with minimal disruption and migration at own pace
- Hedge against volatility in the ecosystem with an eye on the critical end game
- Solve real challenges of adopting service mesh at scale
- Solo.io a key innovator in service mesh ecosystem

Goal: Programmable Application Network



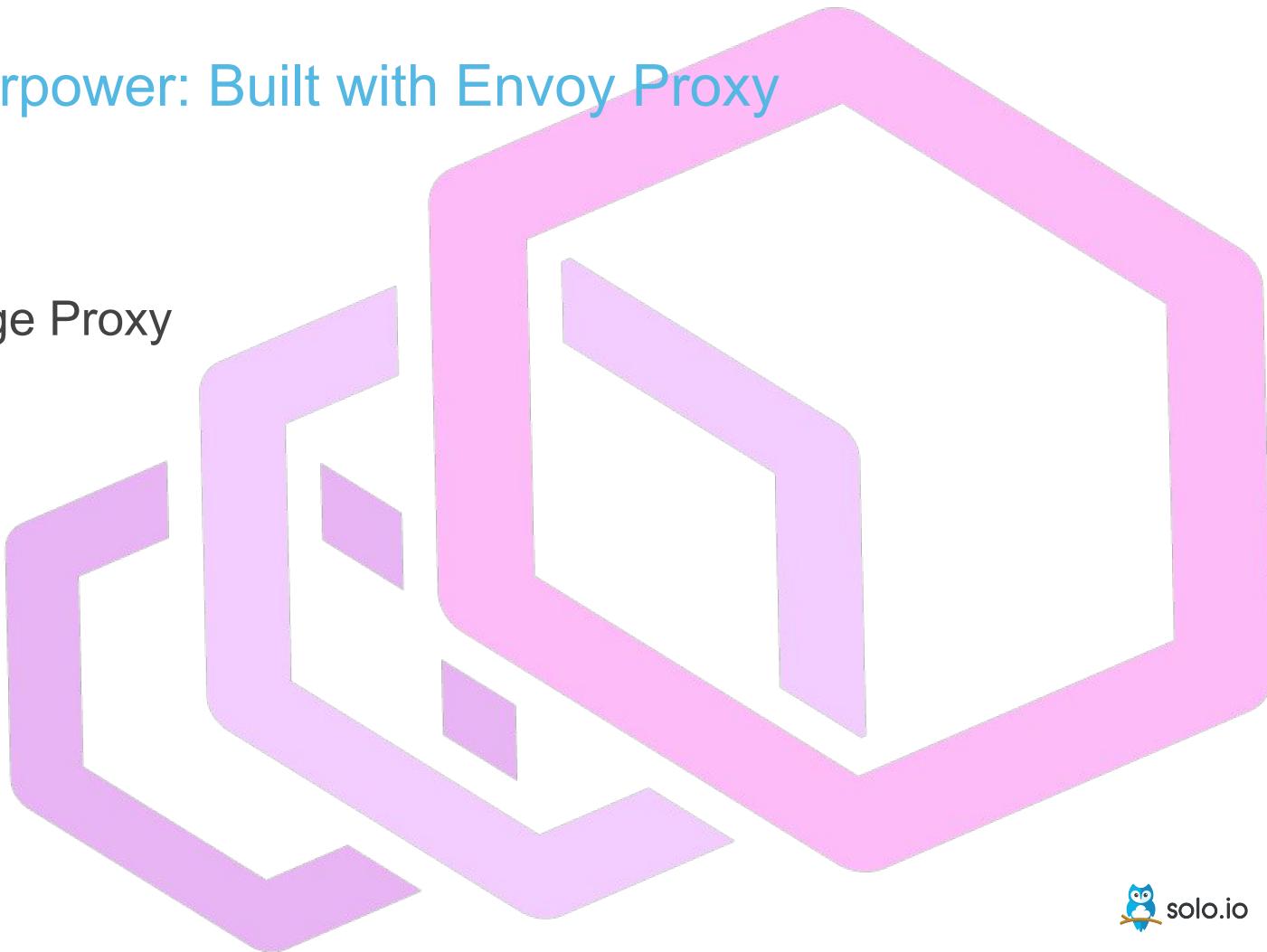
Solo.io's Superpower: Built with Envoy Proxy

Cloud Native Edge Proxy

01 | Performance

02 | Extensibility

03 | Community



GLOO FEATURES

Gloo is a fully-featured edge Gateway and Ingress Controller built on Envoy Proxy.

Gloo was built to extend the capabilities of the service mesh.

Gloo runs on any platform bare-metal, VM, Kubernetes, and Cloud.

Connect	Secure	Control
<ul style="list-style-type: none">Auto Service DiscoveryHTTP RoutingTCP RoutinggRPC WebCORSKubernetes ServicesConsul ServicesRequest transformationsService Mesh Integration	<ul style="list-style-type: none">TLSOpen Policy AgentAWS CognitoIntegrations<ul style="list-style-type: none">• Vault Secrets• Management• Let's Encrypt <p>ENTERPRISE</p> <ul style="list-style-type: none">JWT AuthenticationOAuth / OIDCLDAPWeb Application Firewall (WAF)	<ul style="list-style-type: none">TimeoutsRetriesTraffic ShadowingPrometheus / GrafanaAccess LoggingUsage StatisticsTracing <p>ENTERPRISE</p> <ul style="list-style-type: none">Rate LimitingAdmin Dashboard

Solo.io Core Use Cases

1. Application modernizing with Gloo as starting point for service mesh
2. SMI Standard for Service Mesh enabling Full featured programmable network
3. Scaling up Service Mesh for multi cluster / multi cloud

Use Case Drill Down

Service Mesh Adoption is a Journey



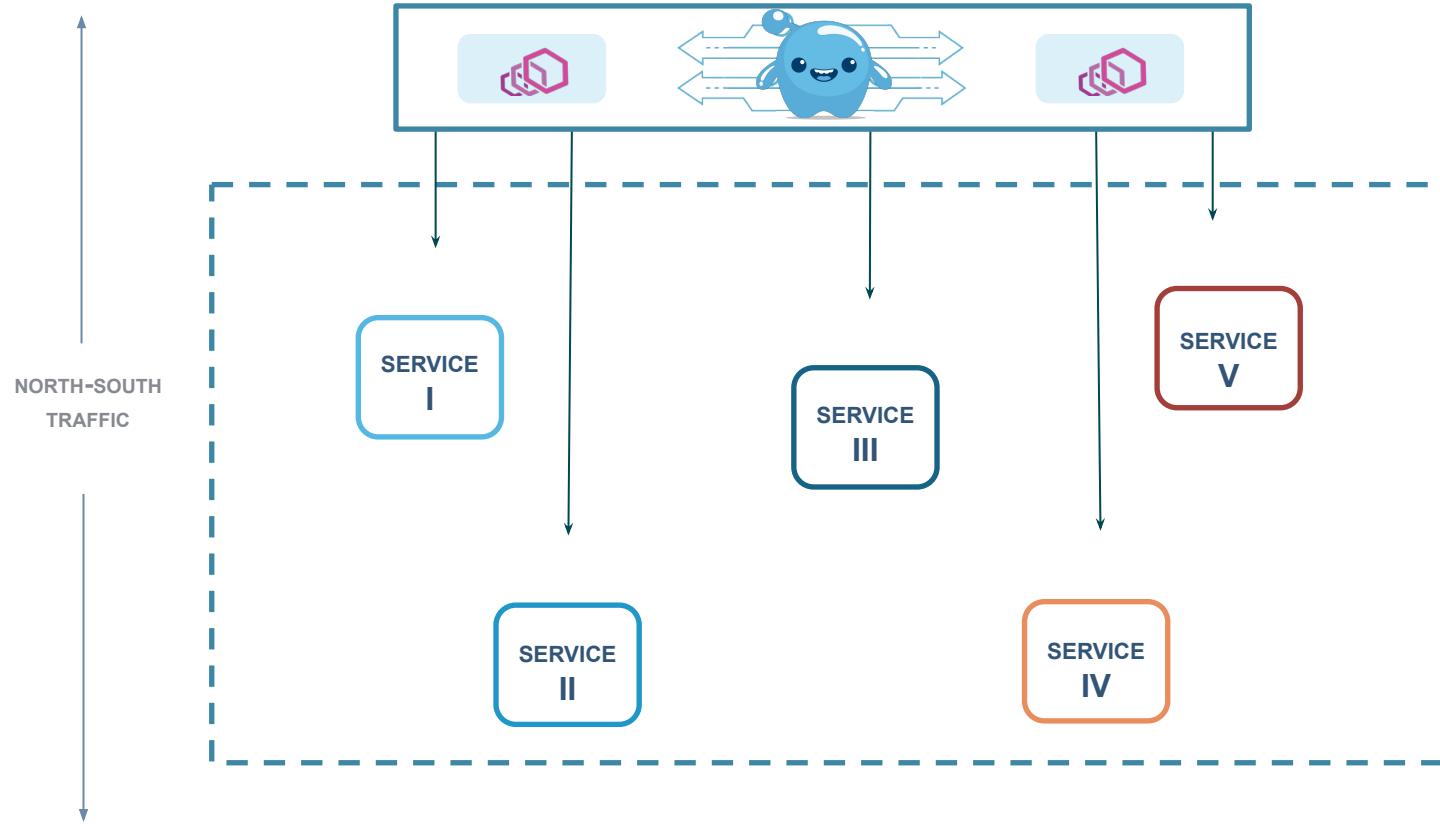
Start with Gloo Edge Gateway



“Gloo Gateway is really Enterprise Envoy”

-- large financial services company

Programmable Application Network: Edge



Getting Started with Gloo Edge Gateway

Secure existing and new applications, and enable applications to evolve at best rate for them.

Challenges

- Ingress/Egress Security
- Starting with Legacy
- Have existing cross-cutting code
- Cognitive Overload
- How to skill up team

Gloo Benefits

- Strong security capabilities
- Meets you where your app is now
- Help with infrastructure transition
- Migrate common concerns at your speed
- Use as much as team needs/is ready for
- Gain relevant experience on your journey to Service Mesh

GLOO FEATURES

Gloo is a fully-featured edge Gateway and Ingress Controller built on Envoy Proxy.

Gloo was built to extend the capabilities of the service mesh.

Gloo runs on any platform bare-metal, VM, Kubernetes, and Cloud.

Connect	Secure	Control
<ul style="list-style-type: none">Auto Service DiscoveryHTTP RoutingTCP RoutinggRPC WebCORSKubernetes ServicesConsul ServicesRequest transformationsService Mesh Integration	<ul style="list-style-type: none">TLSOpen Policy AgentAWS CognitoIntegrations<ul style="list-style-type: none">• Vault Secrets• Management• Let's Encrypt <p>ENTERPRISE</p> <ul style="list-style-type: none">JWT AuthenticationOAuth / OIDCLDAPWeb Application Firewall (WAF)	<ul style="list-style-type: none">TimeoutsRetriesTraffic ShadowingPrometheus / GrafanaAccess LoggingUsage StatisticsTracing <p>ENTERPRISE</p> <ul style="list-style-type: none">Rate LimitingAdmin Dashboard

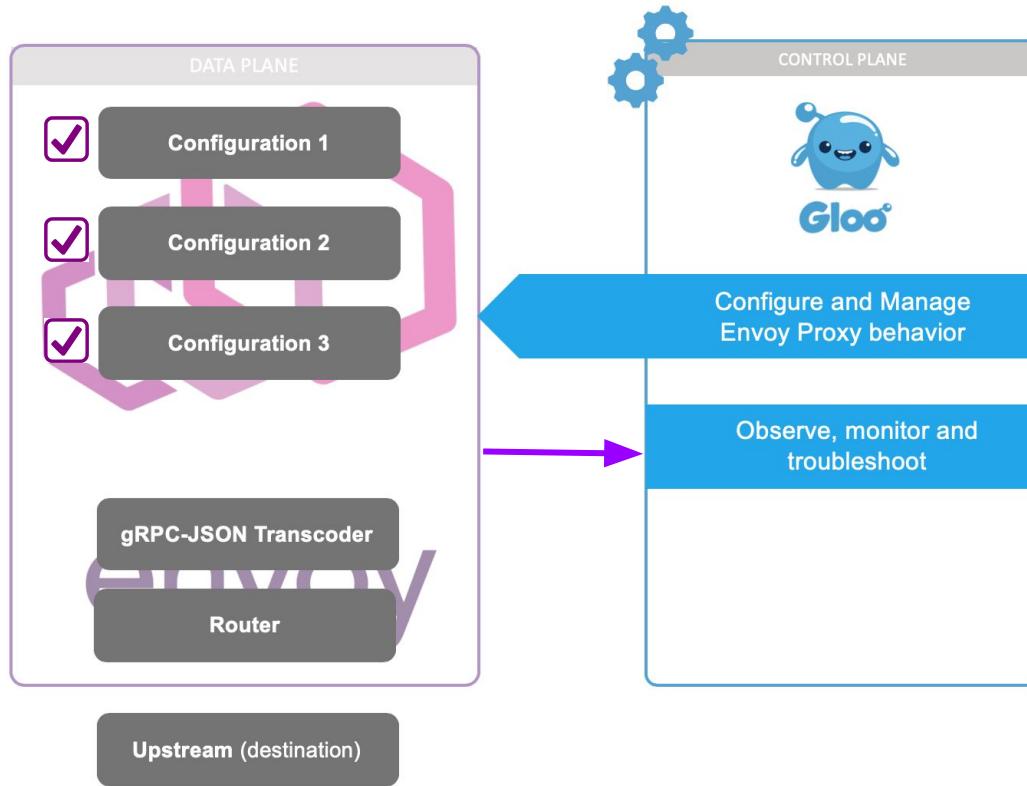


GLOO ARCHITECTURE

Gloo is a flexible and extensible control plane traffic handled by Envoy Proxy for microservices environments. Gloo configures the behavior of the Envoy Proxy data plane to ensure secure application connectivity and policy-based traffic management.



How Gloo Works: Envoy Proxy Control Plane



Gloo configures and manages Envoy

Incoming traffic flows through Envoy Proxy

Traffic and users are checked, validated, verified before passing through

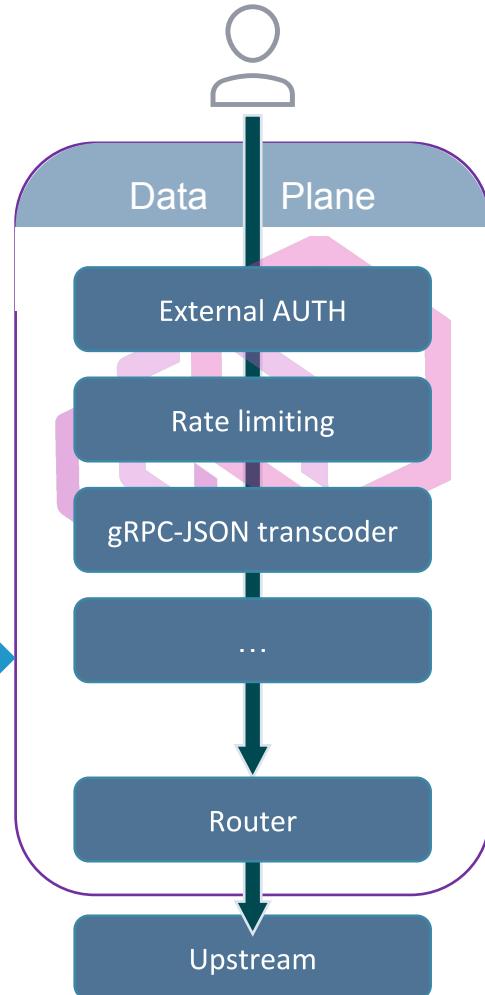
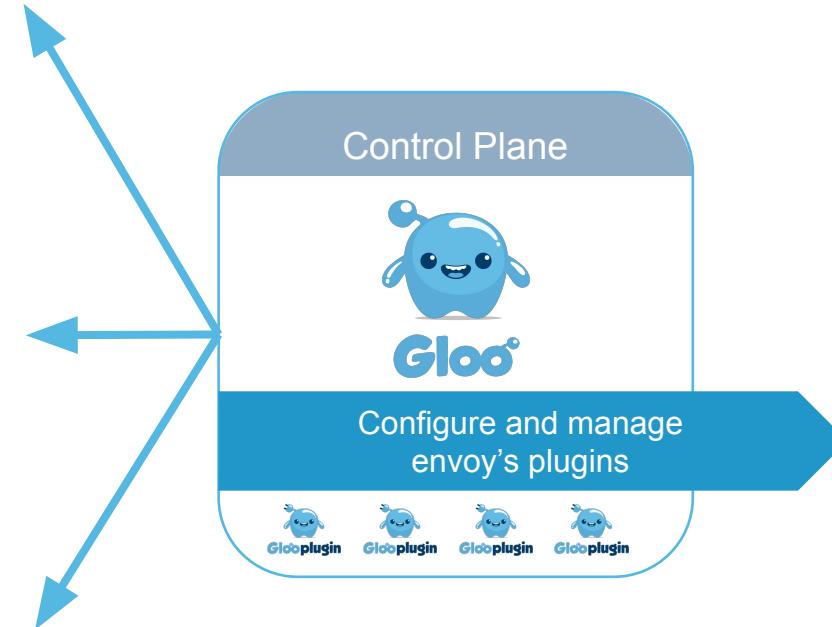
ENVIRONMENT



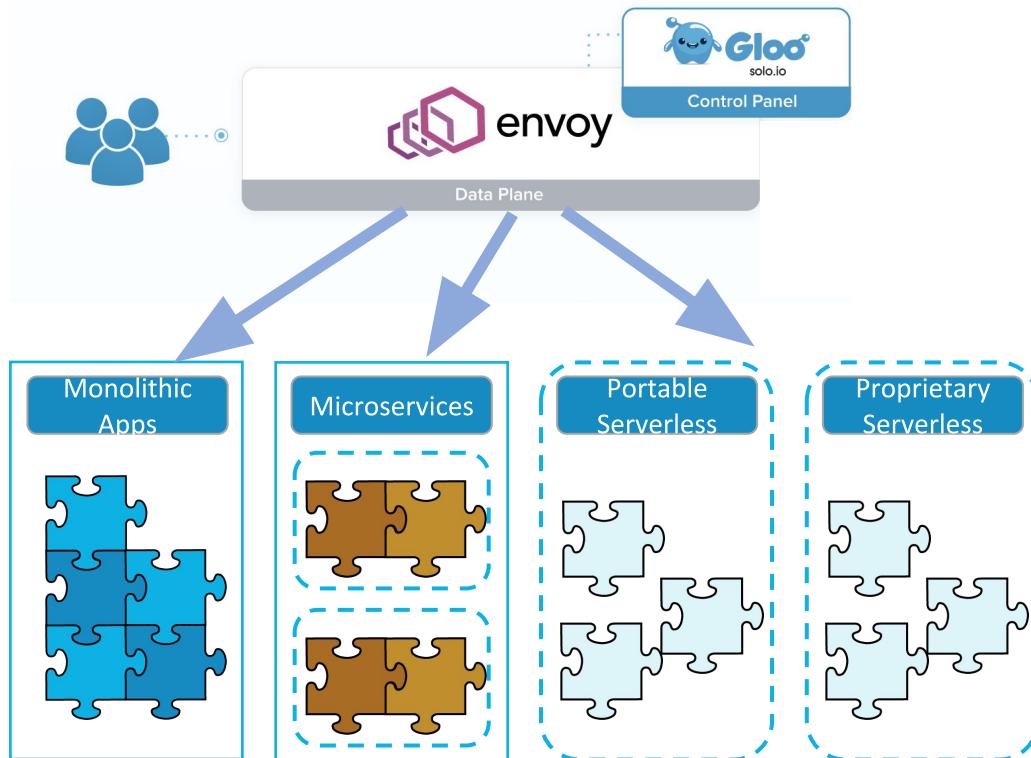
SECRET



CONFIGURATION



Hybrid Routing to Any Programming Paradigm



 Search this site...[Documentation](#)

Installing Knative

[Getting Started with App Deployment](#)

[Gardener](#)

[Gloo on a Cluster](#)

[IBM Cloud Kubernetes Service](#)

[Minikube](#)

[Pivotal Container Service](#)

[Azure Kubernetes Service](#)

[Customizing your install](#)

[Docker - Mac OS](#)

[Google Kubernetes Engine](#)

[IBM Cloud Private](#)

[Installing Istio for Knative](#)

[On existing cluster](#)

Installing Knative

Knative depends on an Ingress/Gateway which is capable of routing requests to Knative Services.

Currently, two options exist which provide this functionality: [Istio](#), the Envoy-based Service Mesh, and [Gloo](#), the Envoy-based API Gateway.

Installing Knative with Gloo

[Install with Gloo](#): Gloo functions as a lightweight gateway for Knative. Choose this option if you don't require service mesh in your cluster and want a lighter-weight alternative to Istio. Note that the *Knative Eventing component* is not supported by Gloo at this time.

Installing Knative with Istio

Istio is a popular service mesh that includes a Knative-compatible ingress. Choose this option if you wish to use Istio service mesh features. You will also need to choose this installation option if you wish to use the Knative Eventing component, which currently depends on Istio.



Service Mesh and Edge Gateway Overlap

Capability	Service Mesh	Edge/Gloo
Traffic Control	✓	✓
Traffic Routing	✓	✓
TLS/mTLS	✓	✓
Network Observability	✓	✓
Policy Enforcement	✓	✓

Edge Capabilities Not Provided by Mesh Gateway

Capability	Service Mesh	Edge/Gloo
OAuth/OIDC	✗	✓
Web Application Firewall	✗	✓
Message transformation	✗	✓
Request/response caching	✗	✓
Spike arrest/rate limit	✗	✓
JWT claim routing	✗	✓
HMAC validation	✗	✓

Gloo Gateway Complements Service Mesh

- Extend with powerful edge capabilities (as seen previously)
- Integrate with mTLS (SDS)
- Integrate with telemetry collection
- Integrate with distributed tracing
- Provide full North-South & East-West networking abstraction

Service Mesh Interface



SERVICE MESH IS AN “EXCITING” SPACE...



LINKERD

NGINX



HashiCorp

Consul



Istio



Kuma



Kong



AWS App Mesh



WITH MANY CHOICES COMES GREAT RESPONSIBILITY

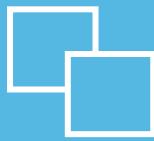
Need to choose a service mesh provider



Hard to tell which one is best for you without trying



Complicated on-boarding
(complex settings)



Different API for different meshes

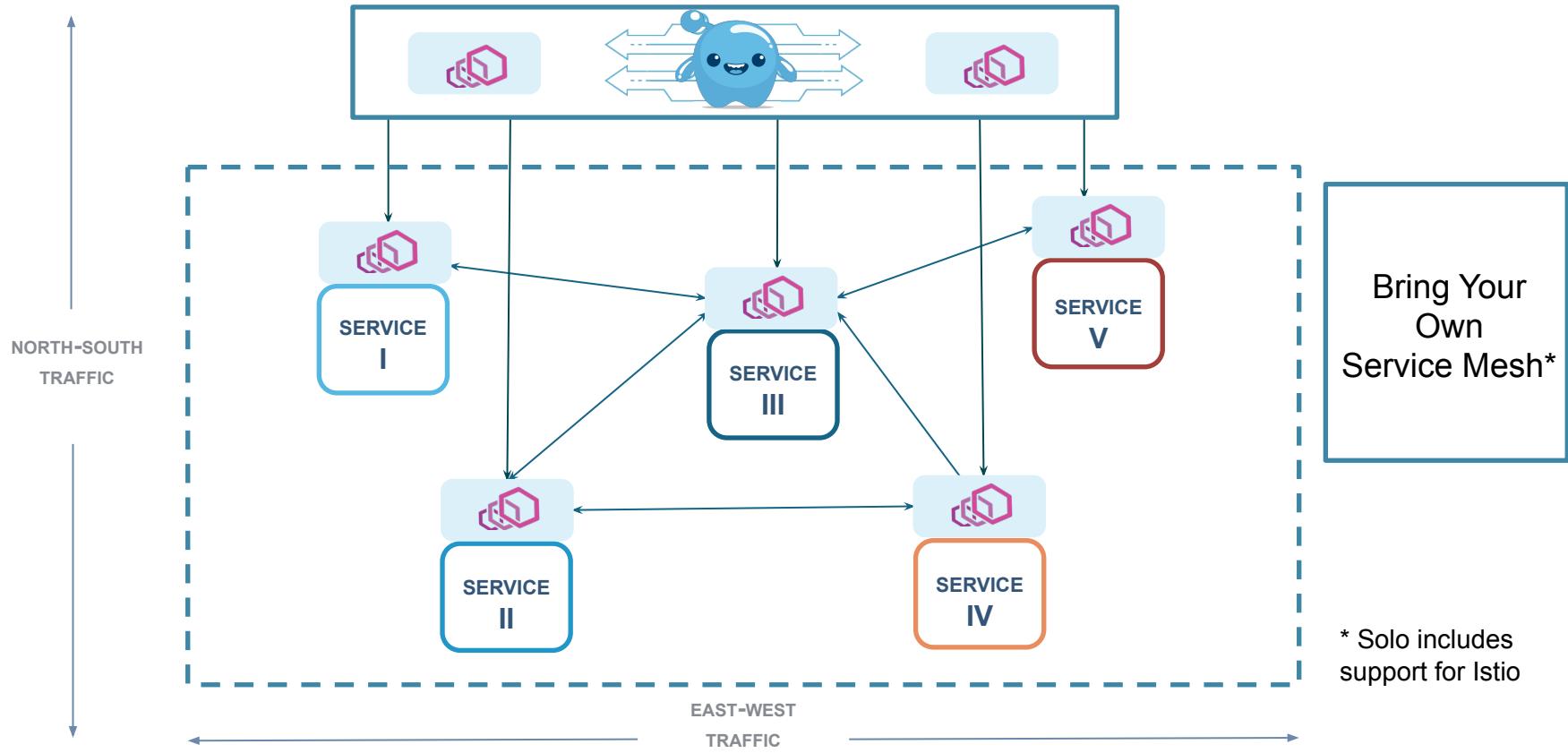


Risk of being locked-in to provider

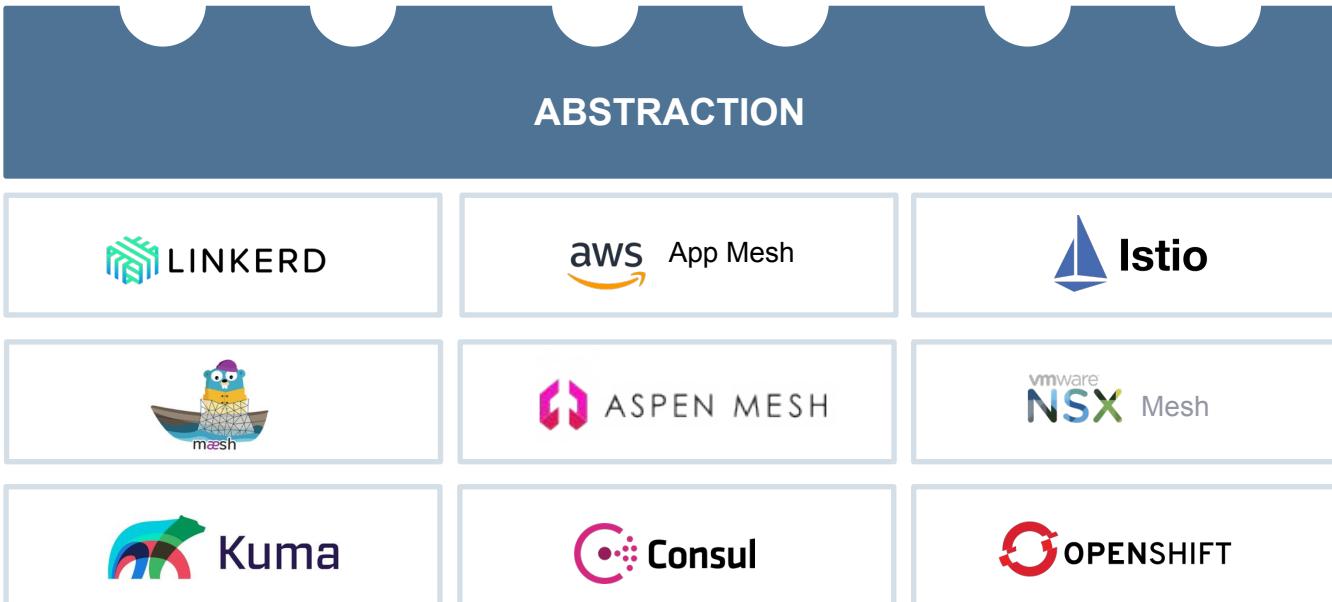


AWS App Mesh

Programmable Application Network: East-West



MAKING IT SIMPLE: ABSTRACTING THE MESH API



- Dead-simple API: simpler workflow, migration, prevents vendor-locking
- Single pane for all meshes

Introducing SuperGloo: The Service Mesh Orchestration Platform



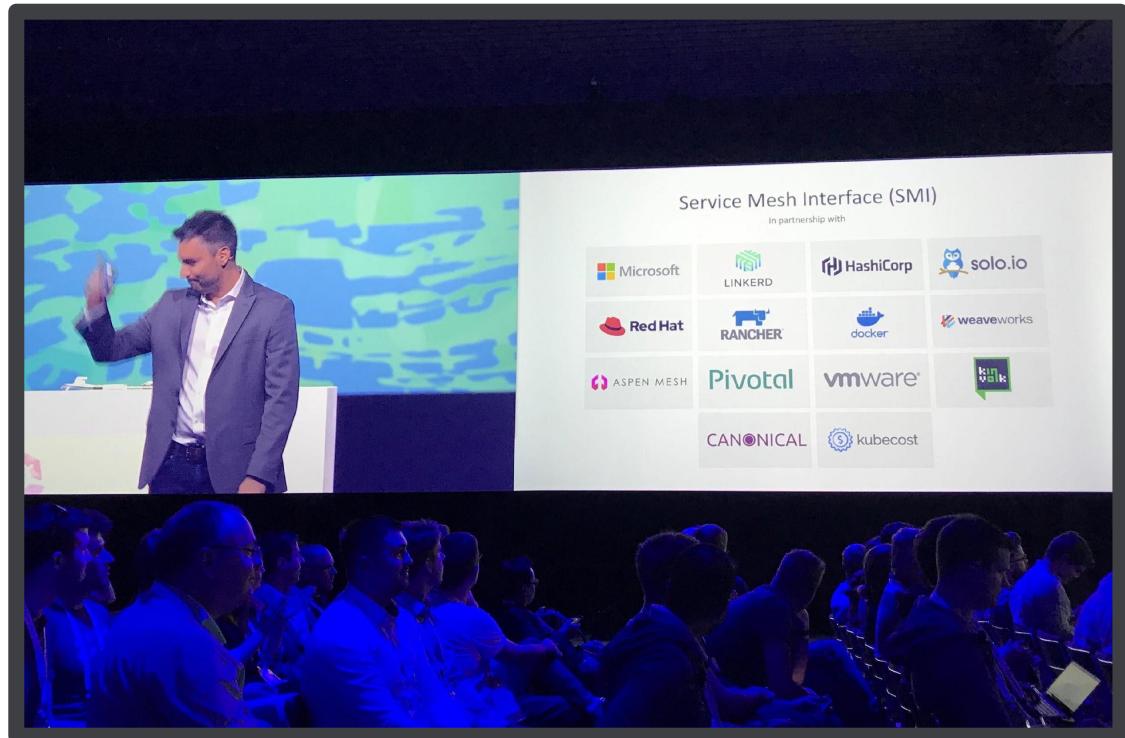
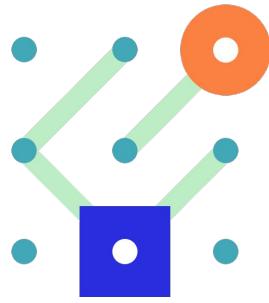
Idit Levine [Follow](#)

Nov 21, 2018 · 3 min read



Today we are thrilled to announce the release of SuperGloo, an open-source project to manage and orchestrate service meshes at scale. SuperGloo is an opinionated abstraction layer that will simplify the installation, management, and operation of your service mesh, whether you use (or plan to use) a single mesh or multiple mesh technologies, on-site, in the cloud, or on any topology that best fits you.

SERVICE MESH INTERFACE [SMI]



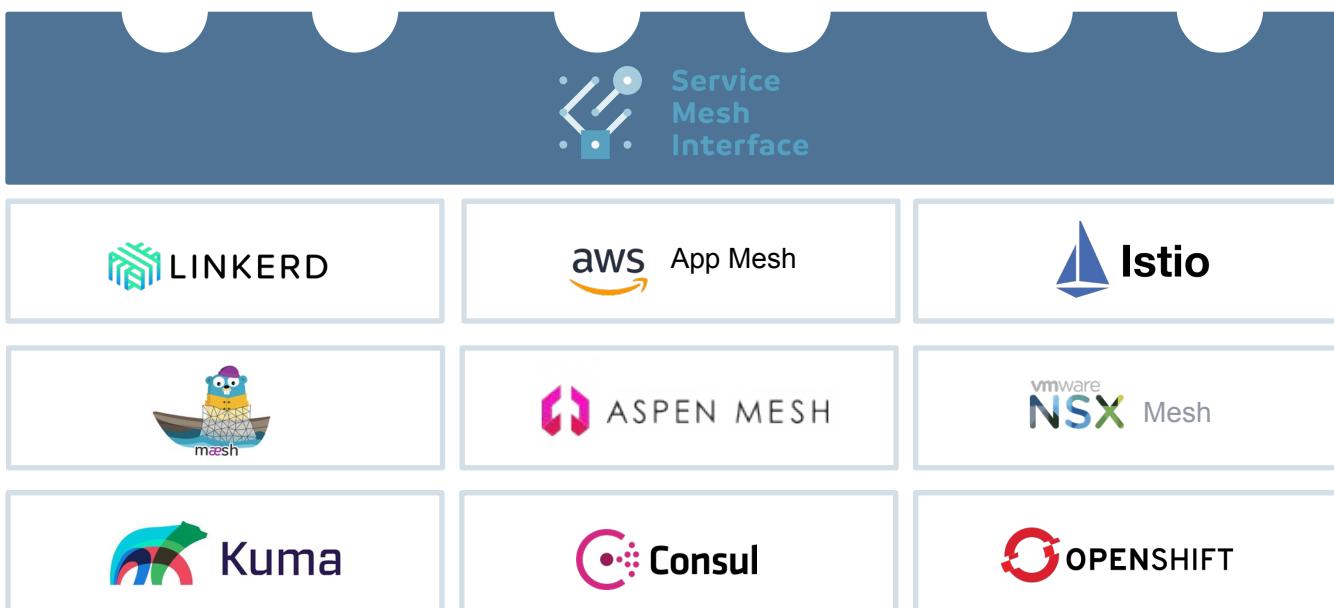
SERVICE MESH INTERFACE

```
kind: TrafficSplit
apiVersion: split.smi-spec.io/v1alpha1
metadata:
  name: my-trafficsplit
spec:
  # The root service that clients use
  # to connect to the destination
  # application.
  service: my-website
  # Services inside the namespace with
  # their own selectors, endpoints and
  # configuration.
  backends:
    - service: my-website-v1
      weight: 50
    - service: my-website-v2
      weight: 50
```

```
kind: HTTPRouteGroup
apiVersion: specs.smi-spec.io/v1alpha1
metadata:
  name: the-routes
matches:
  - name: metrics
    pathRegex: "/metrics"
    methods:
      - GET
  - name: health
    pathRegex: "/ping"
    methods: ["*"]
```

```
kind: TrafficTarget
apiVersion: access.smi-spec.io/v1alpha1
metadata:
  name: path-specific
  namespace: default
destination:
  kind: ServiceAccount
  name: service-a
  namespace: default
  port: 8080
specs:
  - kind: HTTPRouteGroup
    name: the-routes
    matches:
      - metrics
sources:
  - kind: ServiceAccount
    name: prometheus
    namespace: default
```

SERVICE MESH INTERFACE [SMI]

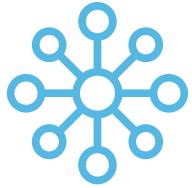


- Dead-simple API: simpler workflow, migration, prevents vendor-locking
- Single pane for all meshes

Scaling Service Mesh



A SERVICE MESH RARELY COMES ALONE



More than one cluster
mean more than one
service mesh instance



Perhaps different teams
in of your organization
prefer different meshes?!

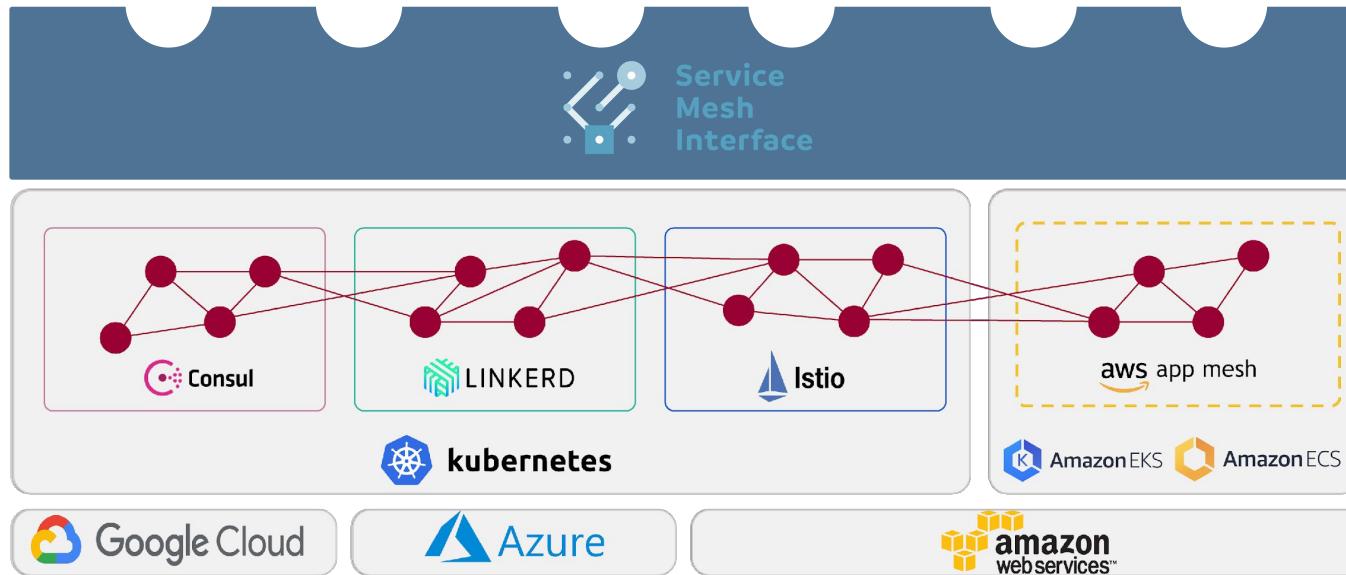


Different public cloud
providers offer
different meshes.

Challenges of managing multi-cluster

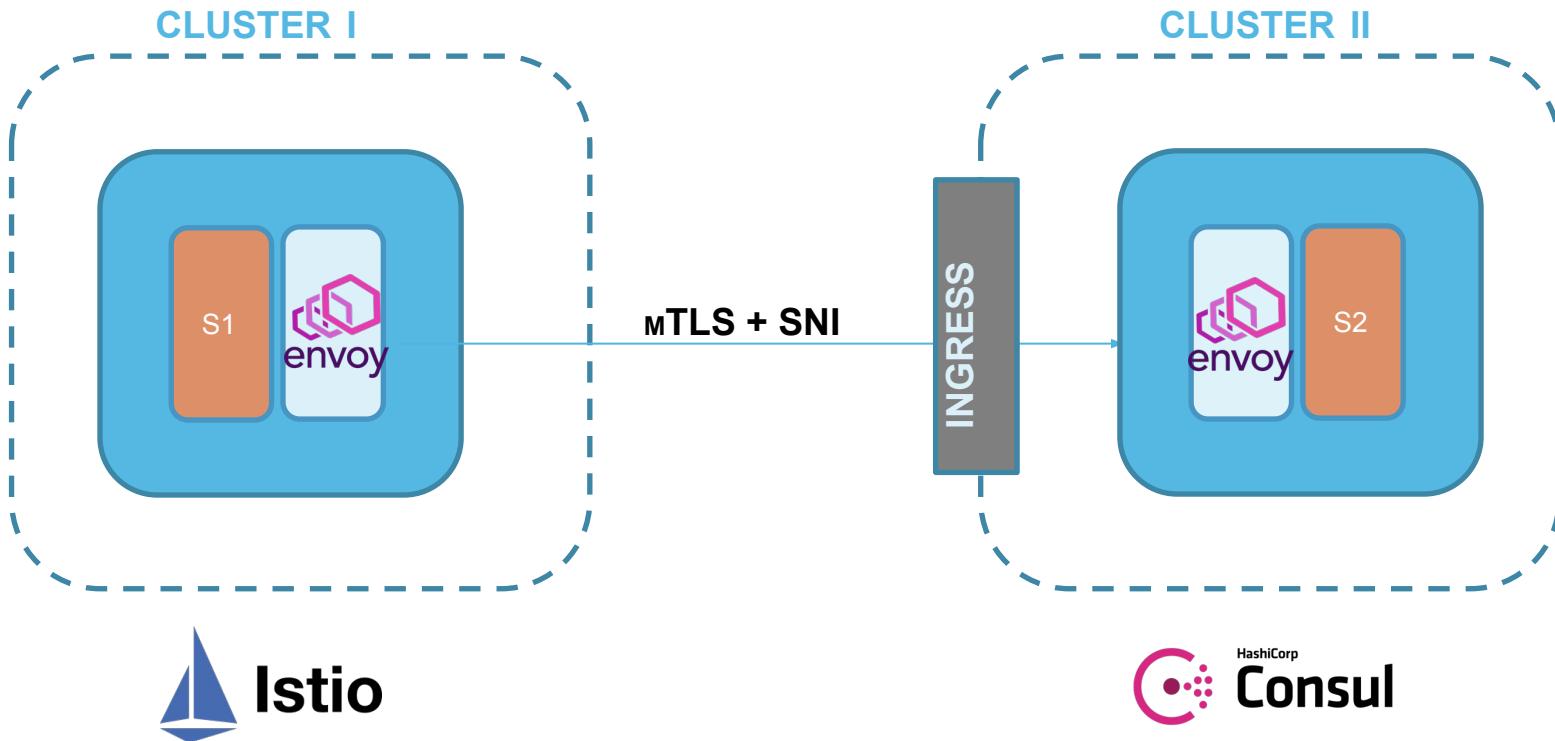
- Unified observability (telemetry, tracing, etc)
- Unified identity and security domain
- Traffic routing across cluster, load balancing
- Configuration of policies
- Discovery, grouping, virtualizing a mesh
- Troubleshooting

MAKING IT FLAT: MESHING TOGETHER MULTIPLE MESHES

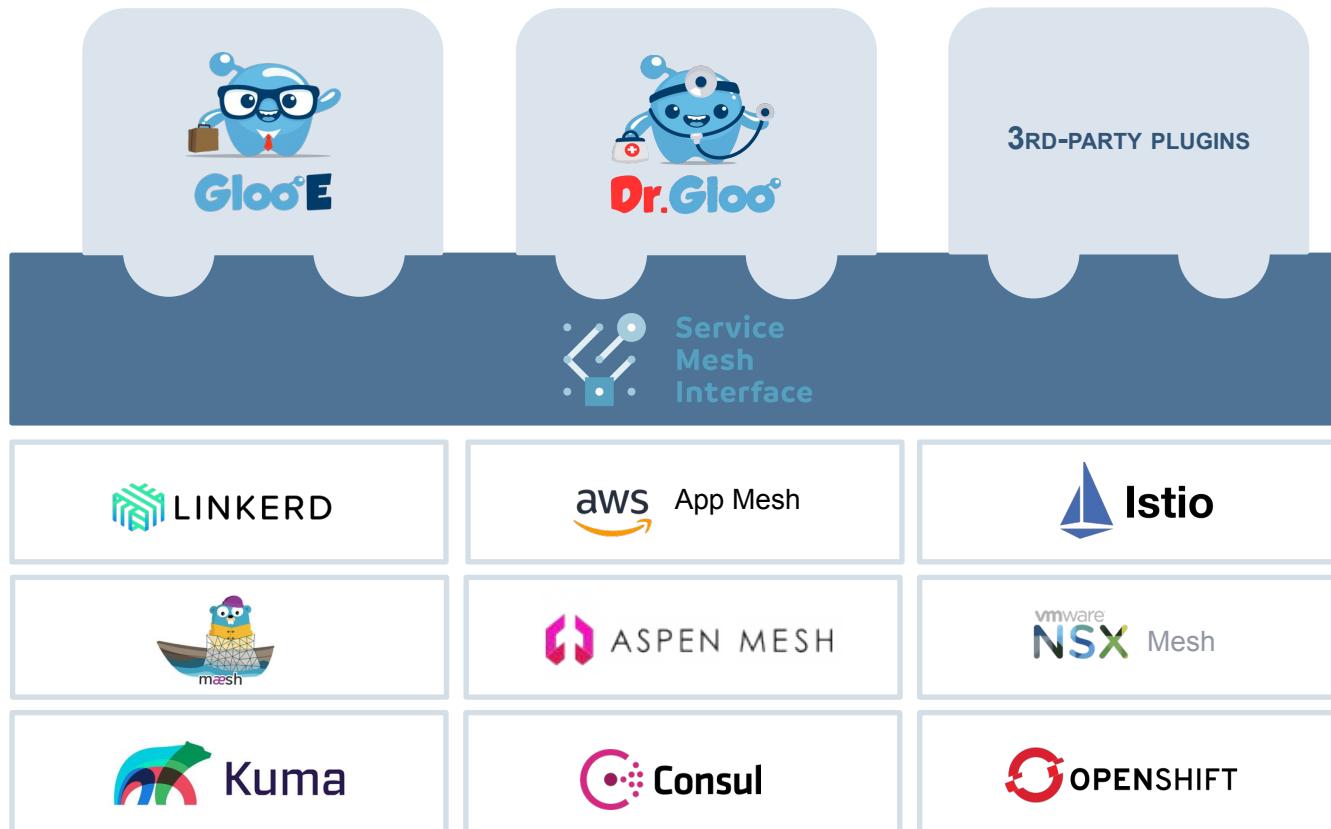


- Flat network across different meshes
- Single pane for all meshes

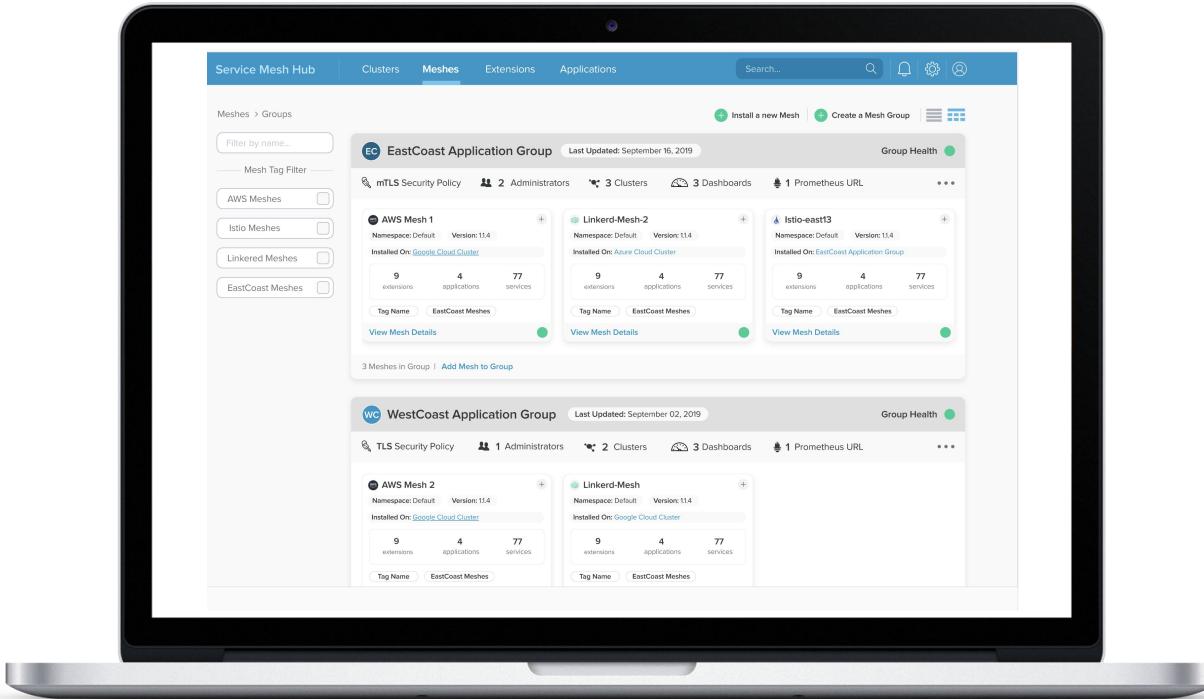
MULTI CLUSTERS



MAKING IT EXTENDABLE: CONTROL PLANE FOR ALL SERVICE MESHES



Let's take a look at Service Mesh Hub!



NEXT ANNOUNCEMENTS FROM SOLO.IO



Summary

- Solo.io products bring the full power of Envoy to you in the best ways your teams can consume
- Best way to manage and scale is to enable the full network abstraction (N-S & E-W)
- Supporting emerging standards (SMI) helps both mitigate disruptions and allow you to realize value sooner
- Solo.io allows you to start piloting now and accelerates your success



solo.io

GLOO OPEN SOURCE
gloo.solo.io

GLOO ENTERPRISE TRIAL
solo.io/gloo-trial

JOIN THE COMMUNITY
slack.solo.io

See you soon

www.solo.io/events



solo.io