solo.io

The Truth About the Service Mesh Data Plane

Denis Jannot
Director of Field Engineering - EMEA

About me

Denis Jannot

Director of Field Engineering - EMEA @ Solo

@djannot

denis.jannot@solo.io

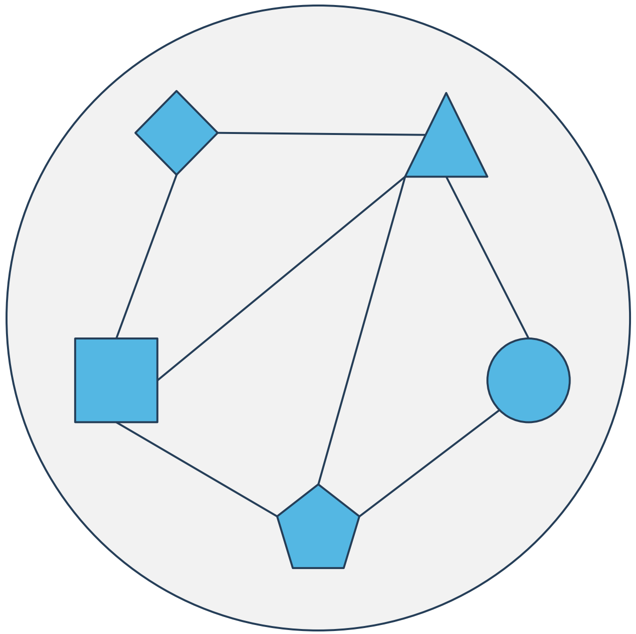
http://www.recorditblog.com

denisjannot

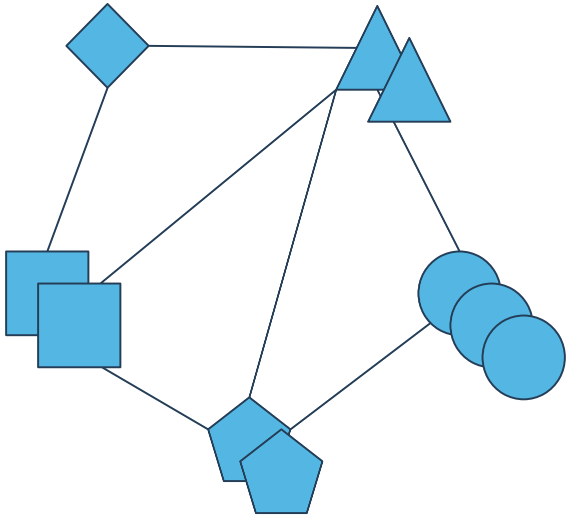
solo.io

From Monolith to Service Mesh

From Monolith to Microservices

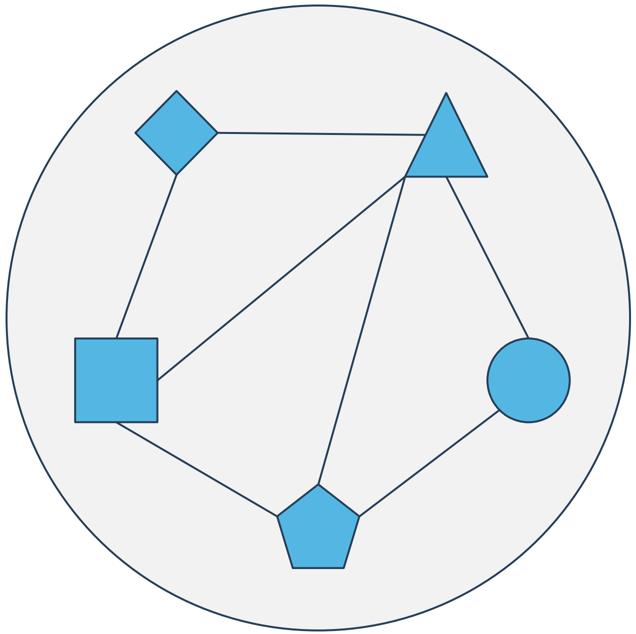


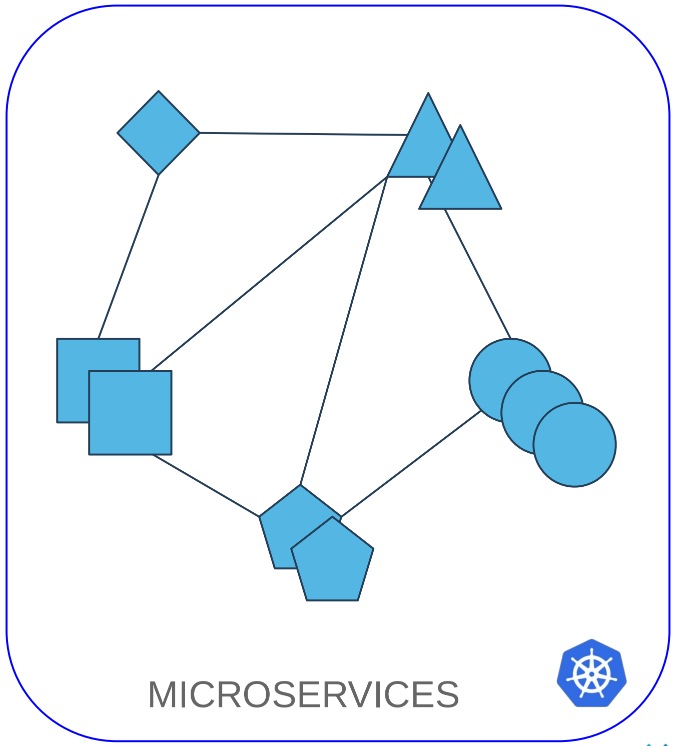
MONOLITH



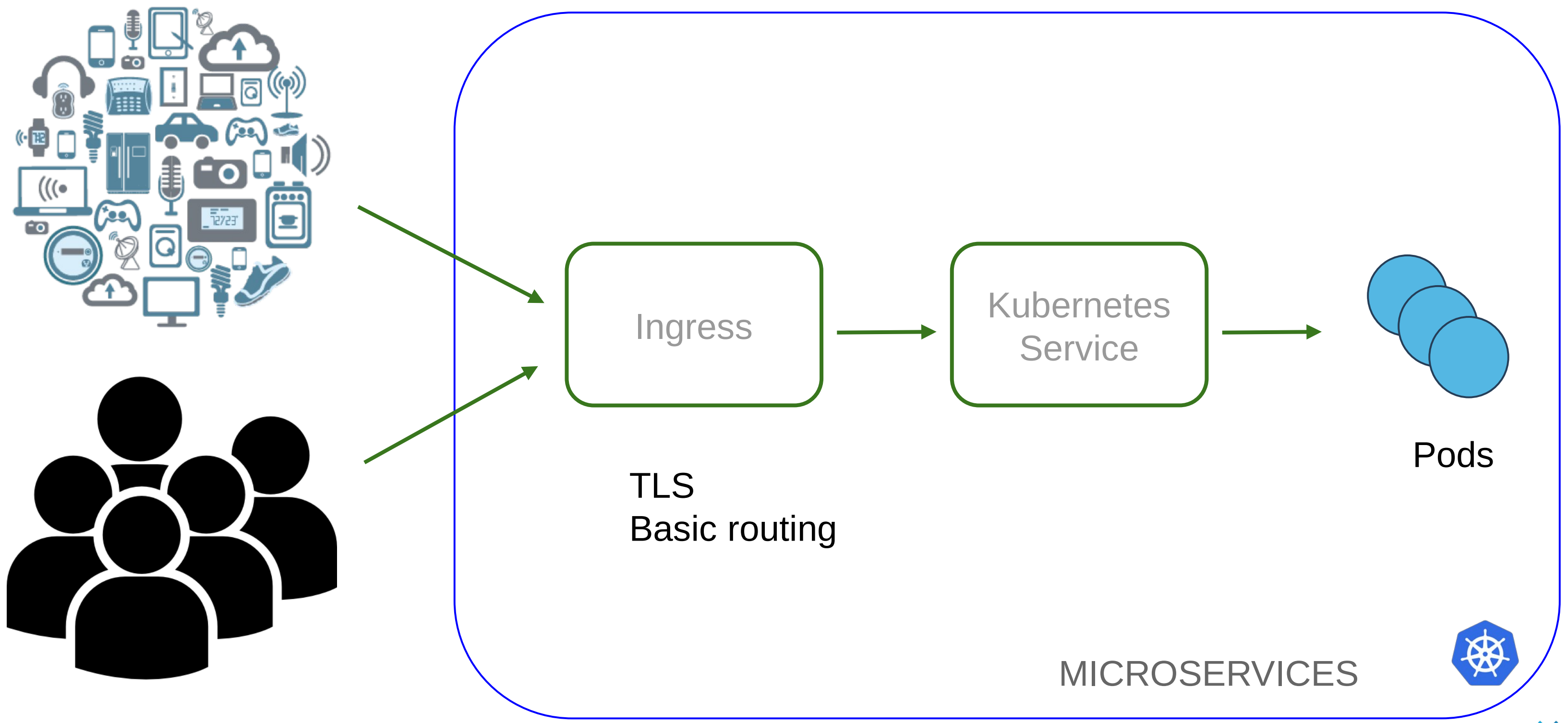
MICROSERVICES

Kubernetes became the most popular platform

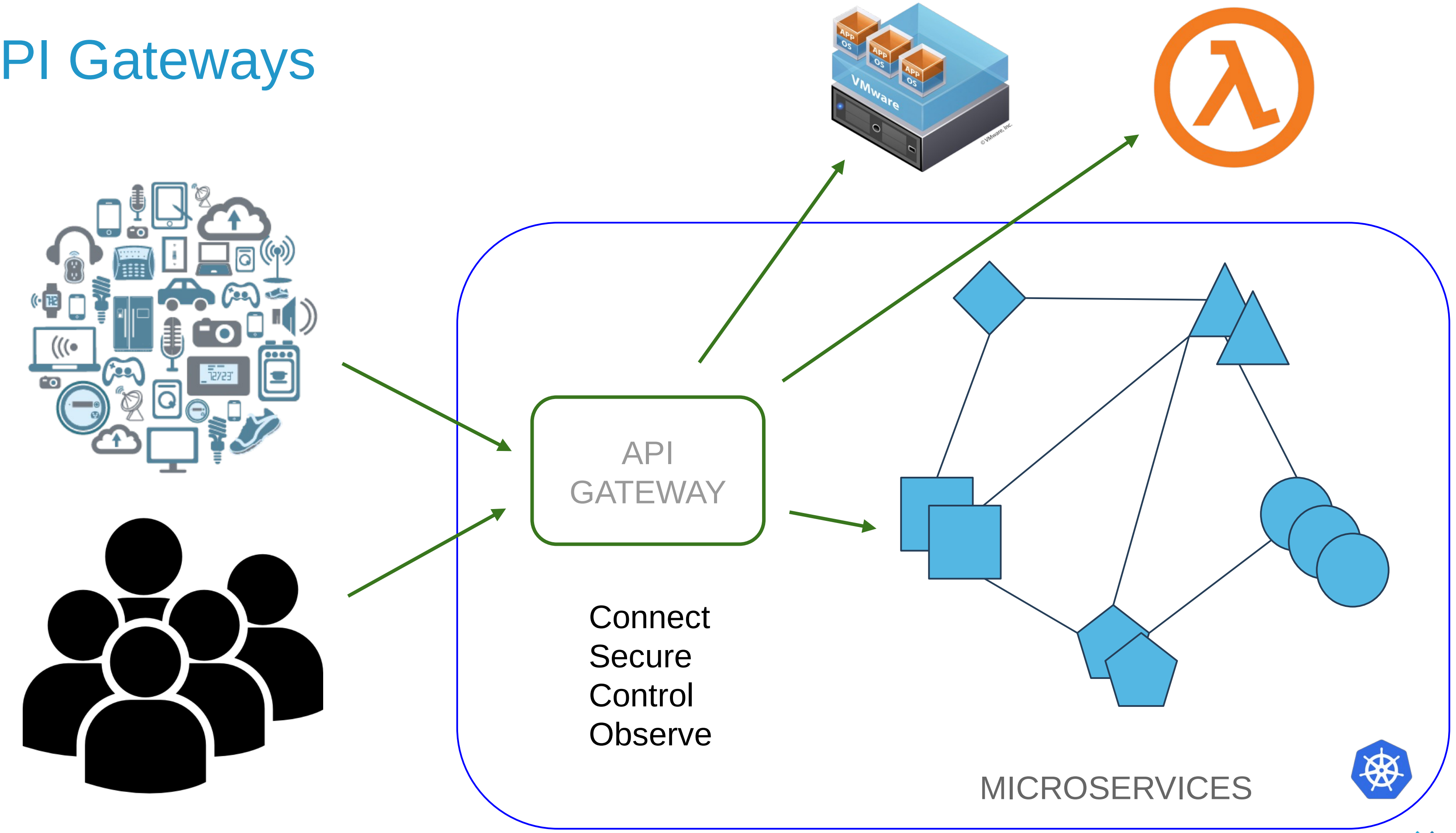




How do you expose your apps ? The Ingress way



API Gateways

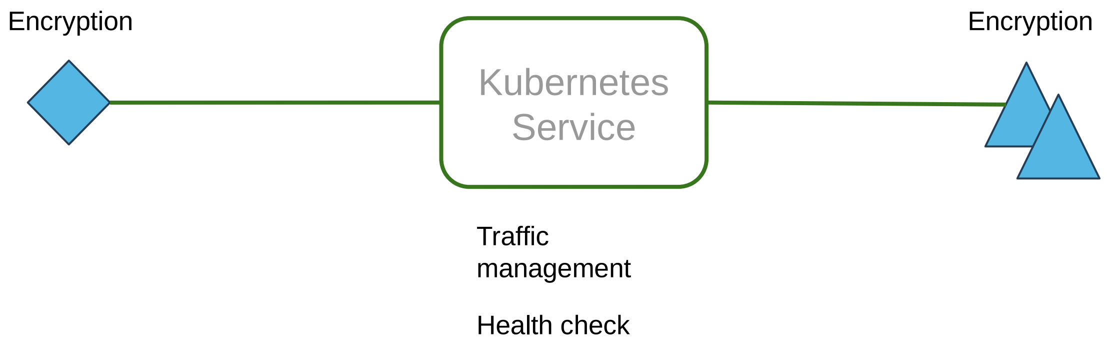


Service to Service communication requirements

- Identity Management
- Encryption
- Certificate Management
- Traffic Management
- Health check
- Access Control
- Telemetry
- ...

solo.io

Service to Service communications



Third Party

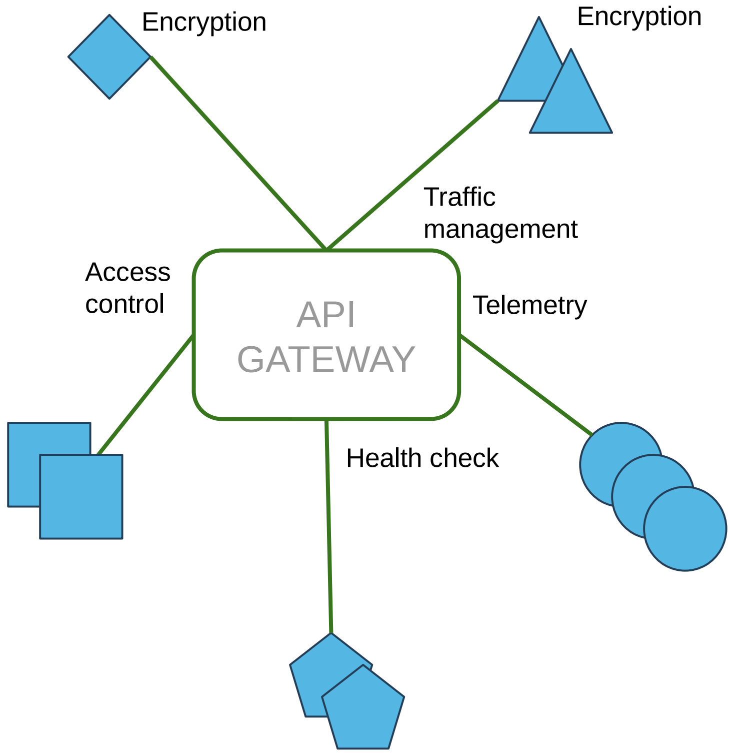
Access control

Identity Management

Certificate management

Telemetry

Service to Service communications

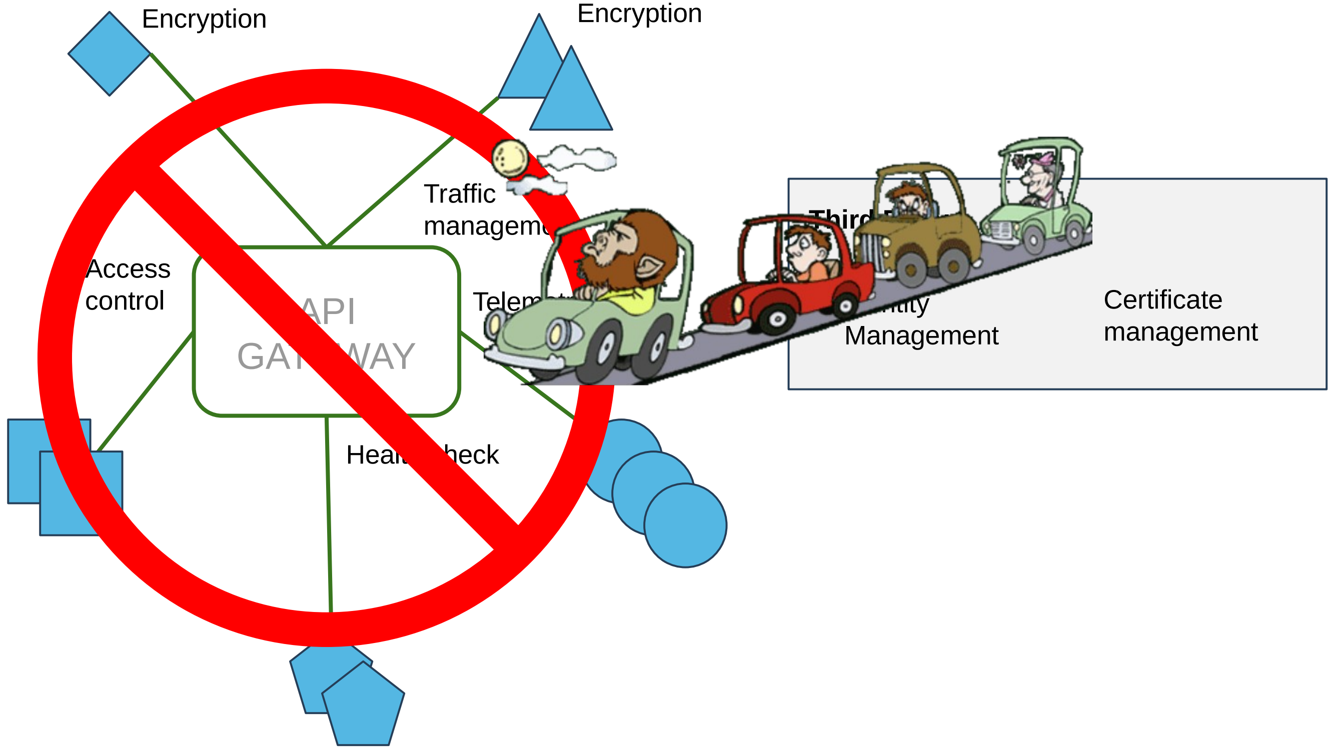


Third Party

Identity Management

Certificate management

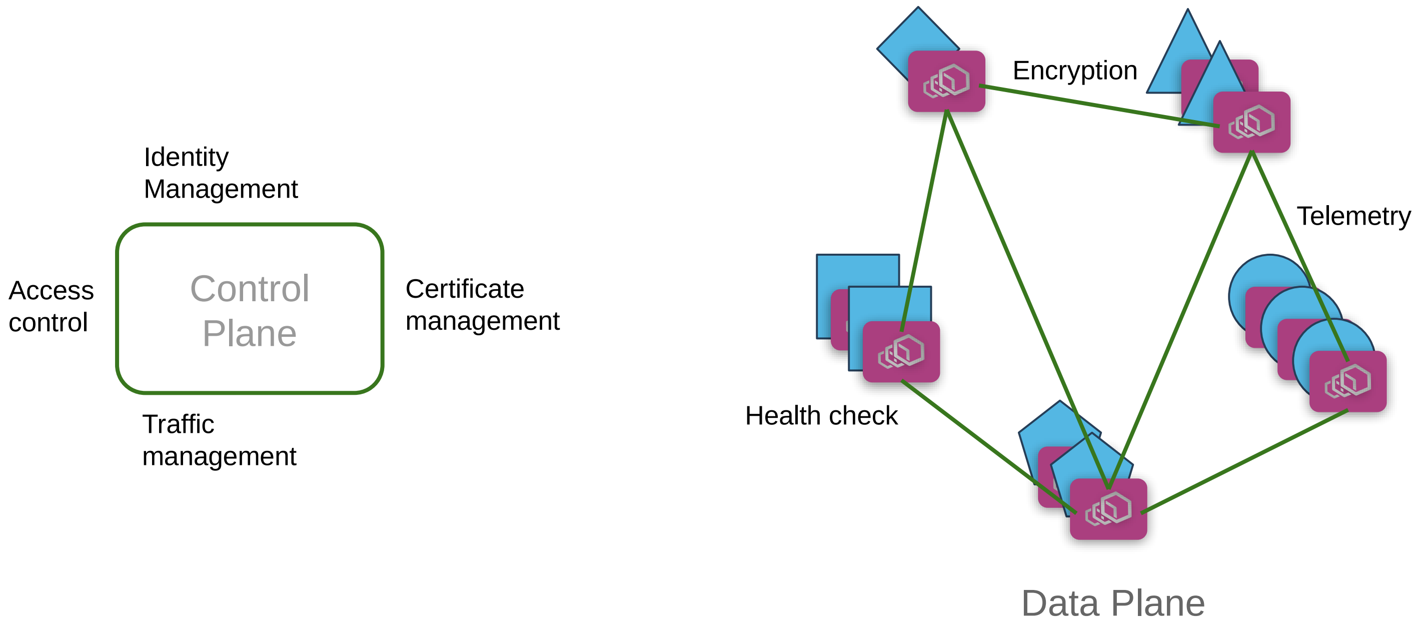
Service to Service communications



MICROSERVICES

solo.io

Service to Service communications



SERVICE MESH

solo.io

Why Envoy Proxy for Service Mesh Data Plane

- Neutral Foundation (CNCF)
- Large, diverse, vibrant community
- Built ground up for dynamic services environment
- Dynamic configuration, driven by API
- Highly extensible
- L7 filters (HTTP/1, HTTP/2, gRPC, redis, mysql, Kafka, etc)
- Deep signals telemetry out of the box
- Versatile deployment options

airbnb	amazon web services™	Booking.com	cookpad
DigitalOcean	ebay	f5	Google
GO-JEK	GRUBHUB	IBM	Medium
Microsoft	NETFLIX	Pinterest	salesforce
	stripe	Square	Tencent 腾讯
twilio	Uber	verizon✓	vmware®
vsco	YAHOO! JAPAN	yelp	

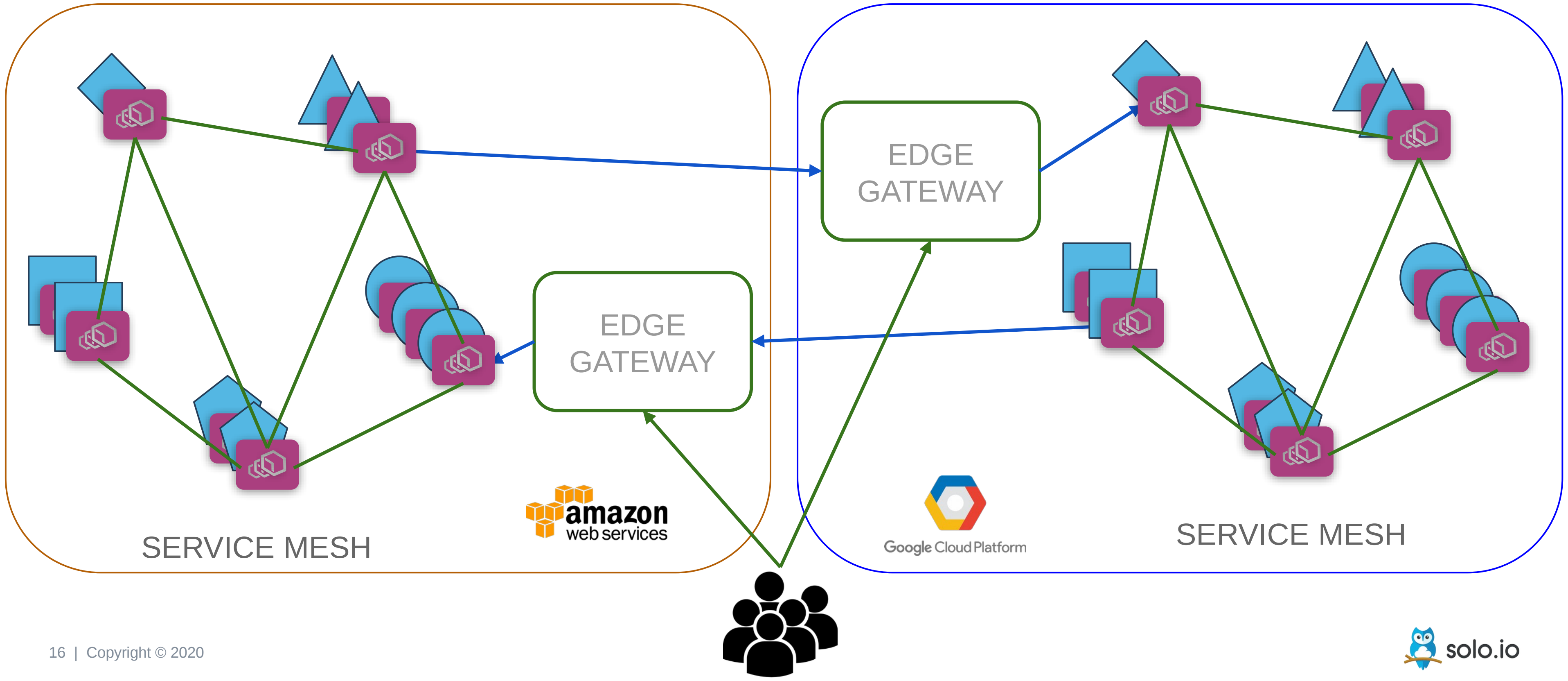
solo.io

Common challenges with Service Mesh

Adoption challenges

- Which one to choose ?
- Who's going to support it ?
- Fitting with existing services (sidecar lifecycle, race conditions, etc)
- Non container environments / hybrid env ?
- No good way to manage multiple clusters

Multicluster Service Mesh



solo.io

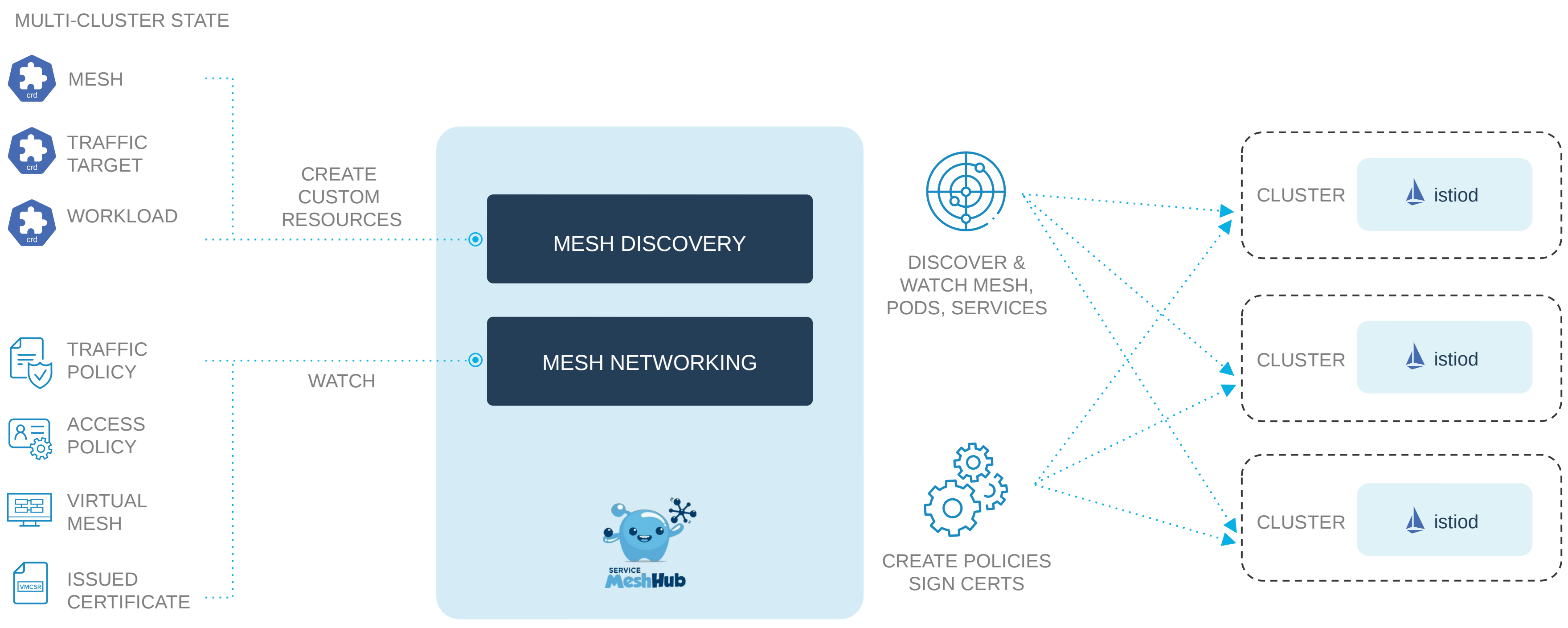
Multicluster Service Mesh challenges

- You need Federated Trust and Identity
- You need to allow communications between clusters
- You need to manage access control globally
- You need to define a Disaster Recovery strategy
- You need to secure the Edge as well
- All of the above is highly complex

Service Mesh Hub

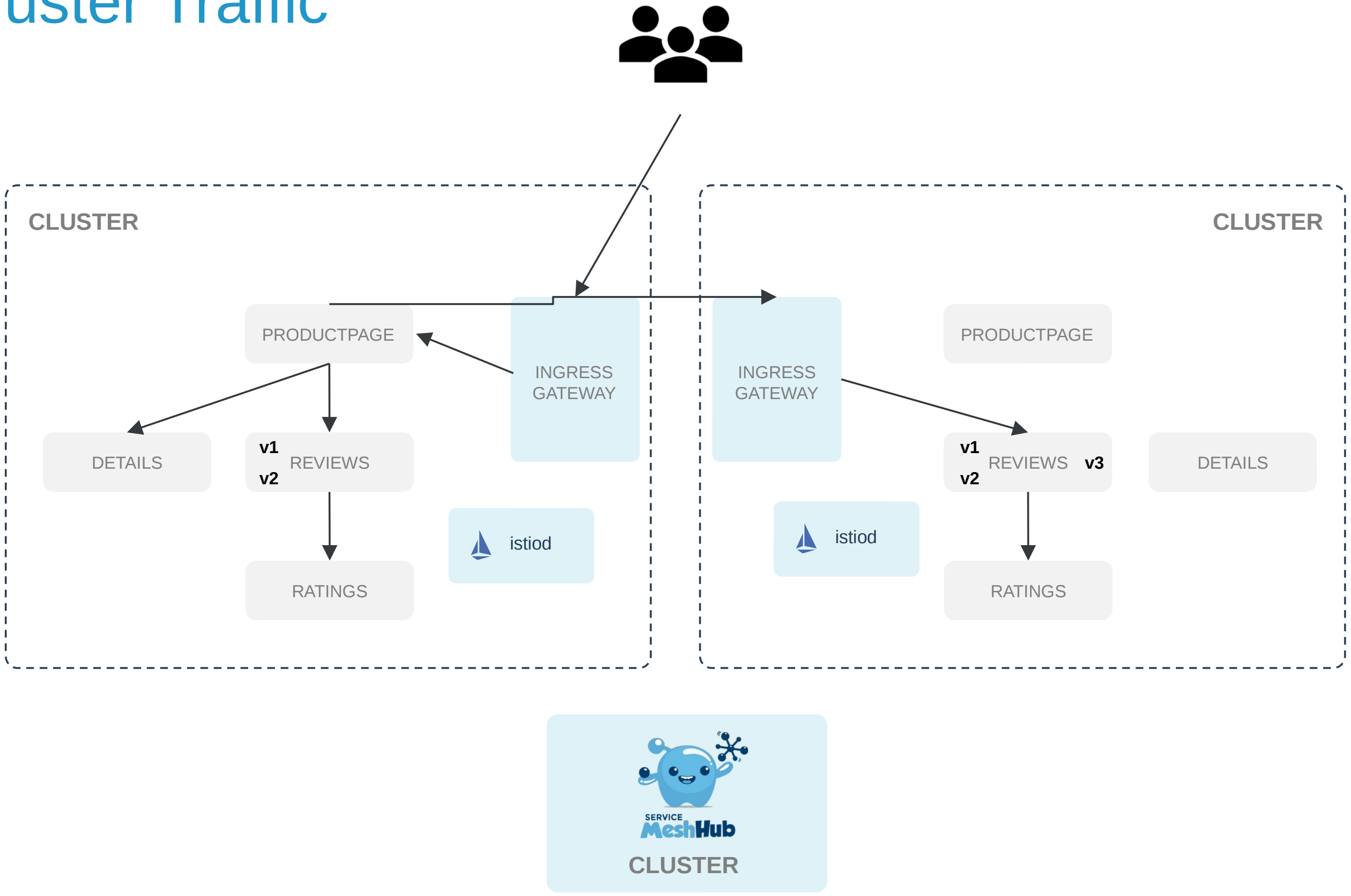
Manage your service mesh deployments across multiple clusters and multiple meshes

Service Mesh Hub is simplifying everything



solo.io

Multi-cluster Traffic



solo.io

Traffic Policy

CLUSTER 1

```yaml
apiVersion: networking.smh.solo.io/v1alpha2
kind: TrafficPolicy
metadata:
  namespace: service-mesh-hub
  name: simple
spec:
  destinationSelector:
  - kubeServiceRefs:
      services:
        - clusterName: kind2
          name: reviews
          namespace: default
  trafficShift:
    destinations:
      - kubeService:
          clusterName: kind3
          name: reviews
          namespace: default
          subset:
            version: v3
        weight: 75
      - kubeService:
          clusterName: kind2
          name: reviews
          namespace: default
          subset:
            version: v1
        weight: 15
      - kubeService:
          clusterName: kind2
          name: reviews
          namespace: default
          subset:
            version: v2
        weight: 10
```

solo.io

Demo

Service Mesh Hub

CLUSTER 2

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
   labels:
    cluster.multicluster.solo.io: kind2
    owner.networking.smh.solo.io:
service-mesh-hub
  name: reviews
  namespace: default
spec:
  hosts:
  - reviews.default.svc.cluster.local
  http:
  - route:
    - destination:
        host:
reviews.default.svc.kind3.global
        subset: version-v3
      weight: 75
    - destination:
        host:
reviews.default.svc.cluster.local
        subset: version-v1
      weight: 15
    - destination:
        host:
reviews.default.svc.cluster.local
        subset: version-v2
      weight: 10
```

CLUSTER 2

```
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  labels:
    cluster.multicluster.solo.io: kind2
    owner.networking.smh.solo.io:
service-mesh-hub
  name: reviews.default.svc.kind3.global
  namespace: istio-system
spec:
  host: reviews.default.svc.kind3.global
  subsets:
  - labels:
      cluster: kind3
    name: version-v3
  - labels:
      cluster: kind3
    name: version-v1
  - labels:
      cluster: kind3
    name: version-v2
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```

CLUSTER 2

```
apiVersion: networking.istio.io/v1beta1
kind: ServiceEntry
metadata:
  labels:
    cluster.multicluster.solo.io: kind2
    owner.networking.smh.solo.io:
service-mesh-hub
  name: reviews.default.svc.kind3.global
  namespace: istio-system
spec:
  addresses:
  - 253.124.25.94
  endpoints:
  - address: 172.18.0.230
    labels:
      cluster: kind3
    ports:
      http: 15443
  hosts:
  - reviews.default.svc.kind3.global
  location: MESH_INTERNAL
  ports:
  - name: http
    number: 9080
    protocol: TCP
  resolution: DNS
```

solo.io

CLUSTER 3

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  labels:
    cluster.multicluster.solo.io: kind3
    owner.networking.smh.solo.io: service-mesh-
hub
  name: virtual-mesh.service-mesh-hub
  namespace: istio-system
spec:
  configPatches:
  - applyTo: NETWORK_FILTER
    match:
      context: GATEWAY
      listener:
        filterChain:
          filter:
            name:
envoy.filters.network.sni_cluster
        portNumber: 15443
    patch:
      operation: INSERT_AFTER
      value:
        name:
envoy.filters.network.tcp_cluster_rewrite
        typed_config:
          '@type':
type.googleapis.com/istio.envoy.config.filter.n
etwork.tcp_cluster_rewrite.v2alpha1.TcpClusterR
ewrite
          cluster_pattern: \.kind3.global$
          cluster_replacement: .cluster.local
  workloadSelector:
    labels:
      istio: ingressgateway
```

CLUSTER 3

```
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  labels:
    cluster.multicluster.solo.io: kind3
    owner.networking.smh.solo.io:
service-mesh-hub
  name: reviews
  namespace: default
spec:
  host:
reviews.default.svc.cluster.local
  subsets:
  - labels:
      version: v3
    name: version-v3
  - labels:
      version: v1
    name: version-v1
  - labels:
      version: v2
    name: version-v2
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```

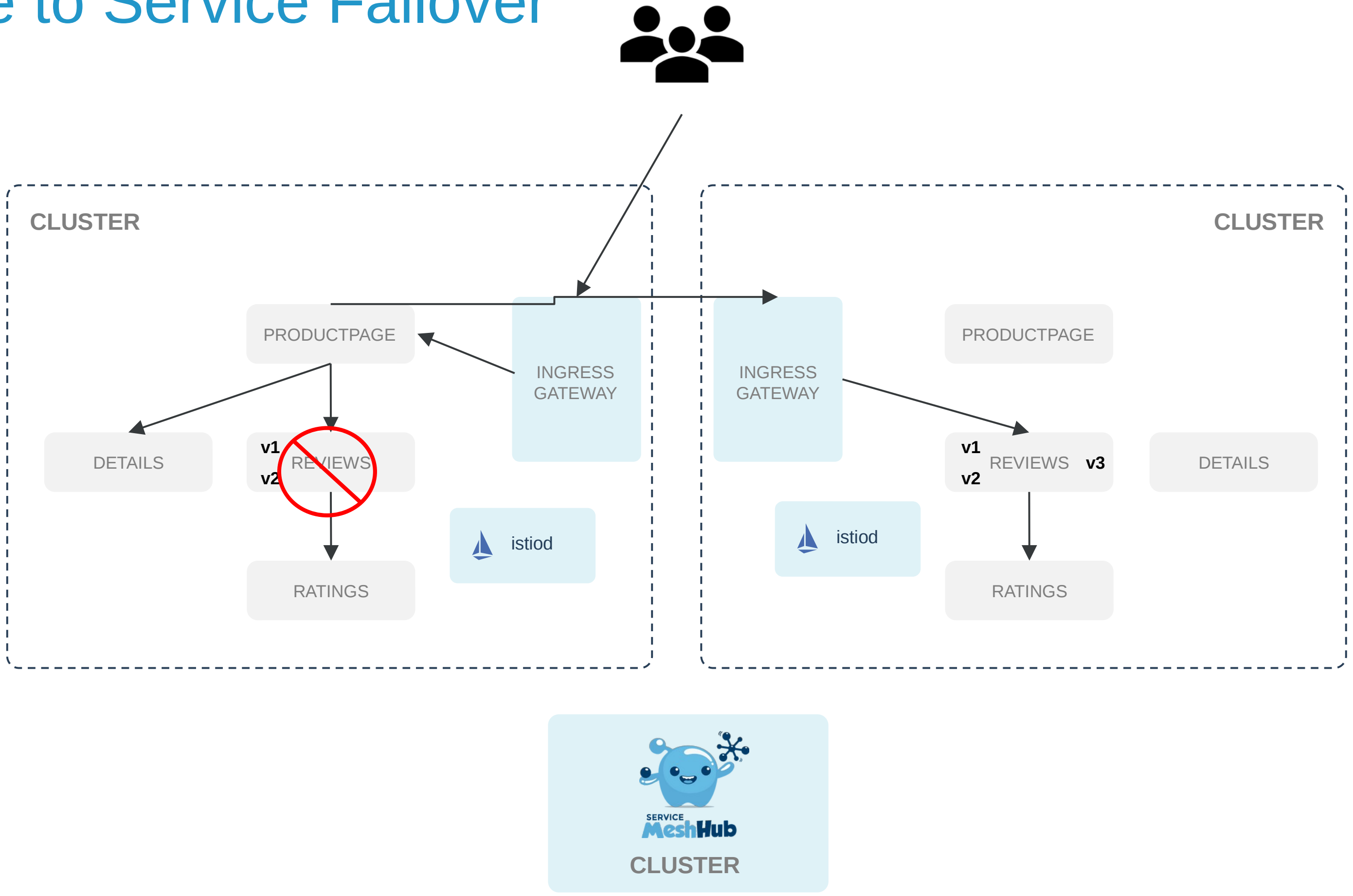
CLUSTER 3

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: reviews
    service: reviews
  name: reviews
  namespace: default
spec:
  clusterIP: 10.97.193.52
  ports:
  - name: http
    port: 9080
    protocol: TCP
    targetPort: 9080
  selector:
    app: reviews
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```

CLUSTER 3

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: reviews
    istio.io/rev: default
    pod-template-hash: d978546db
    security.istio.io/tlsMode: istio
    service.istio.io/canonical-name: reviews
    service.istio.io/canonical-revision: v3
    version: v3
  name: reviews-v3-d978546db-dj59b
  namespace: default
spec:
...
```

Service to Service Failover



solo.io

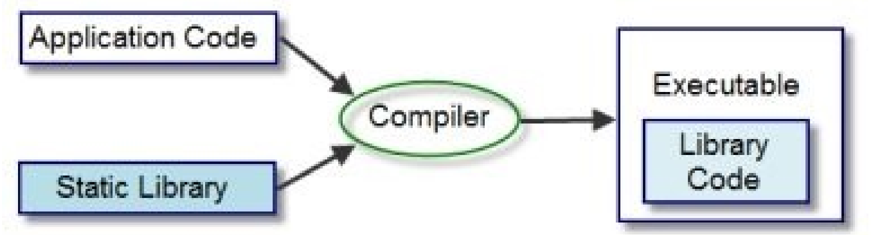
What’s next ?

Web Assembly

Customize Envoy Proxy with WebAssembly

solo.io

Extending Envoy Proxy - Adding Custom Filters

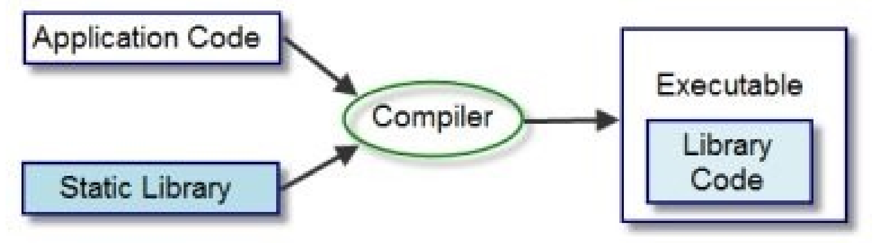


THE OLD WAY:

- Write filter in C++
- Statically link it to Envoy (requires compiling Envoy)
- Ship and deploy new version of Envoy

solo.io

Extending Envoy Proxy - Adding Custom Filters



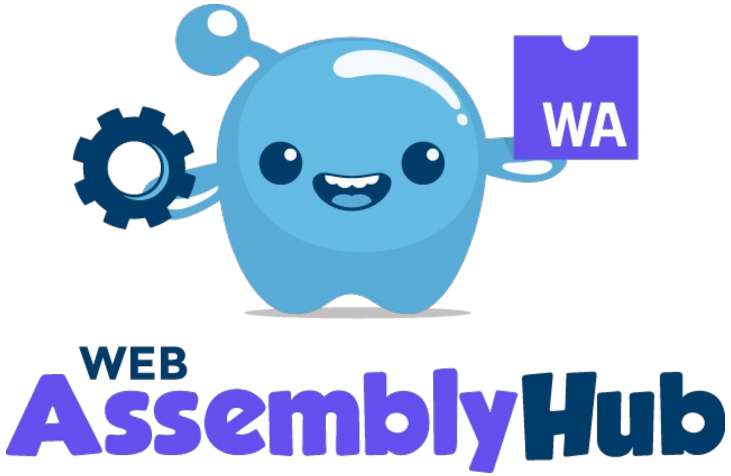
THE OLD WAY:

- Write filter in C++
- Statically link into Envoy (requires compiling Envoy)
- Ship and deploy new version of Envoy

Introducing WebAssembly Hub and `wasme`

Build, Deploy, and Publish

- Write filter in *any language*
- Compile to .wasm module
- Dynamically load in Envoy Proxy during runtime
- Publish and share filters
- With or without Service Mesh



https://webassemblyhub.io

BUILD
WA
PUBLISH
DISCOVER
DEPLOY
AssemblyHub
Explore Modules

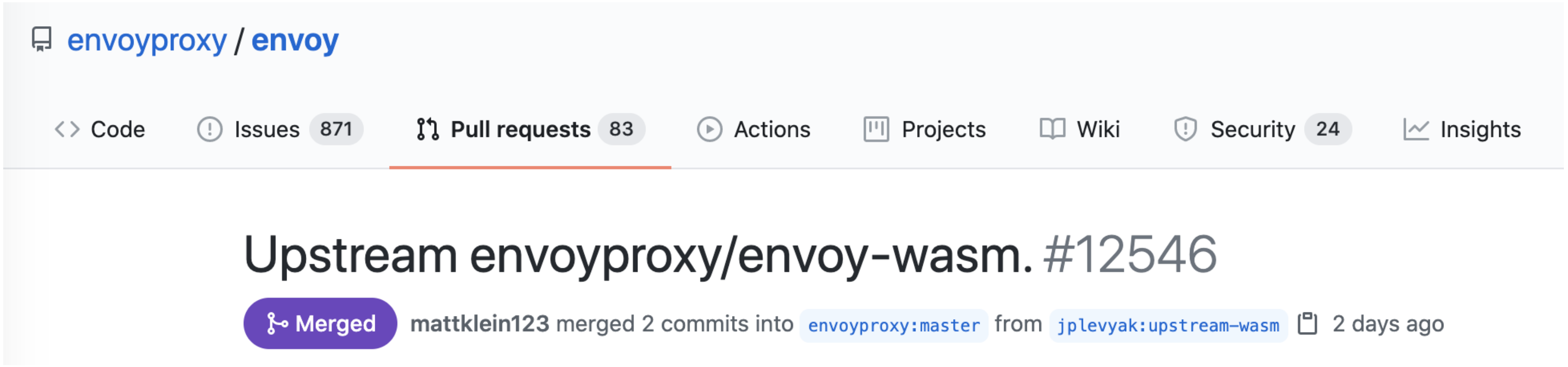
solo.io

Demo

Web Assembly

Current state of WASM in Envoy

- https://www.solo.io/blog/the-state-of-webassembly-in-envoy-proxy/
- https://github.com/envoyproxy/envoy/pull/12546
- https://www.youtube.com/watch?v=8fty-sqFyoY


envoyproxy / envoy

Upstream envoyproxy/envoy-wasm. #12546

Merged mattklein123 merged 2 commits into envoyproxy:master from jplevyak:upstream-wasm 2 days ago

solo.io

Thank you !

solo.io