
cisco
You make possible

Understanding Istio Service Mesh on Kubernetes

Shannon McFarland – CCIE#5245
Distinguished Engineer
@eyepv6

DEVNET-2022

cisco Live!
Barcelona | January 27-31, 2020

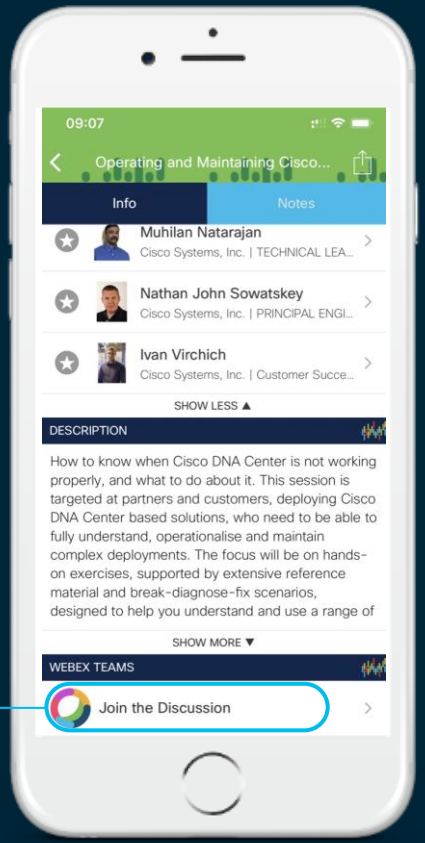
Cisco Webex Teams

Questions?

Use Cisco Webex Teams to chat with the speaker after the session

How

1. Find this session in the Cisco Events Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space



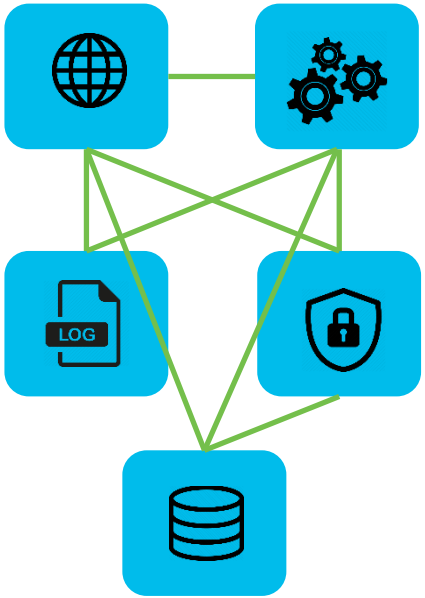
Agenda

- What is a Service Mesh and why do I want one?
- Istio Overview
- Demo
- How to get Istio
- How to contribute to Istio

What is a Service Mesh?

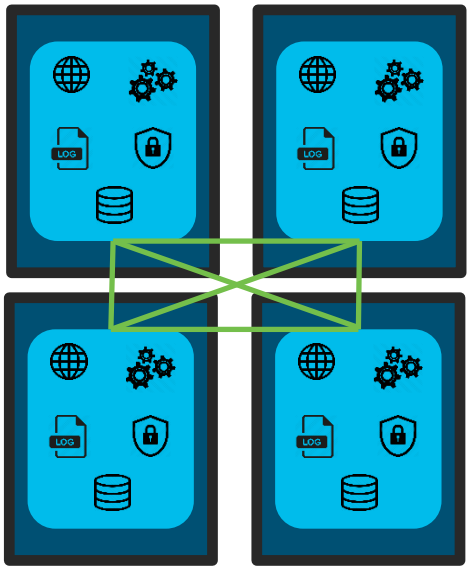
What's a Microservice?

Monolithic Application

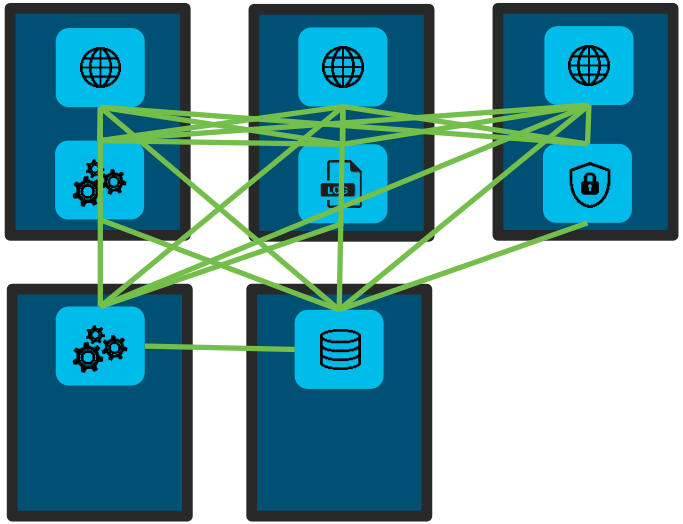


Microservices Application

Scaling a Microservice Application



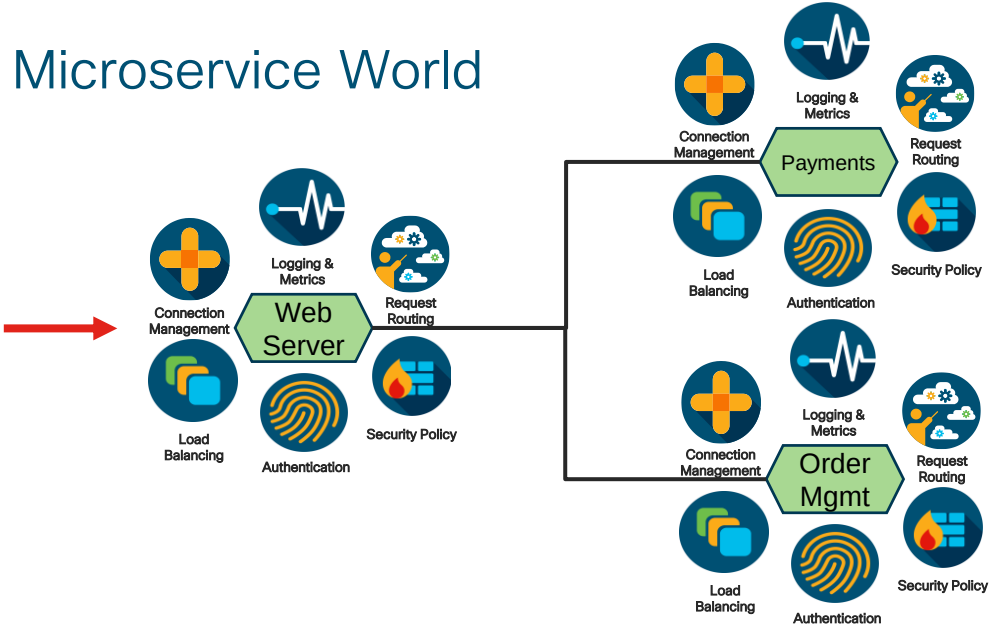
Monolithic Application



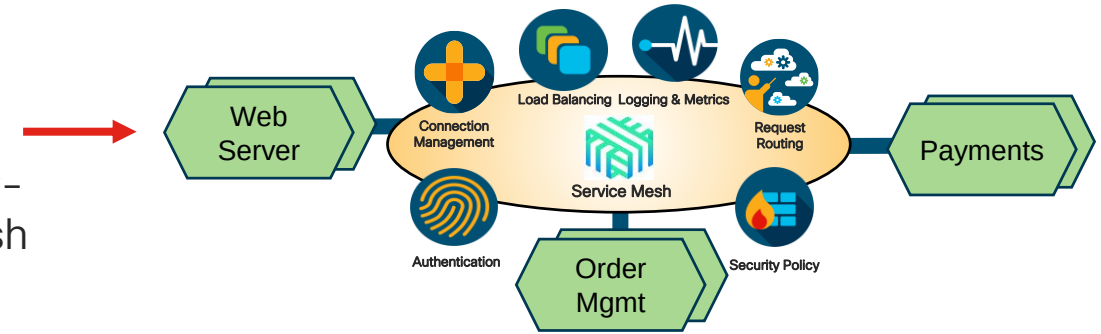
Microservice Application

Service Features in a Microservice World

- Deploy all the service features as independent components



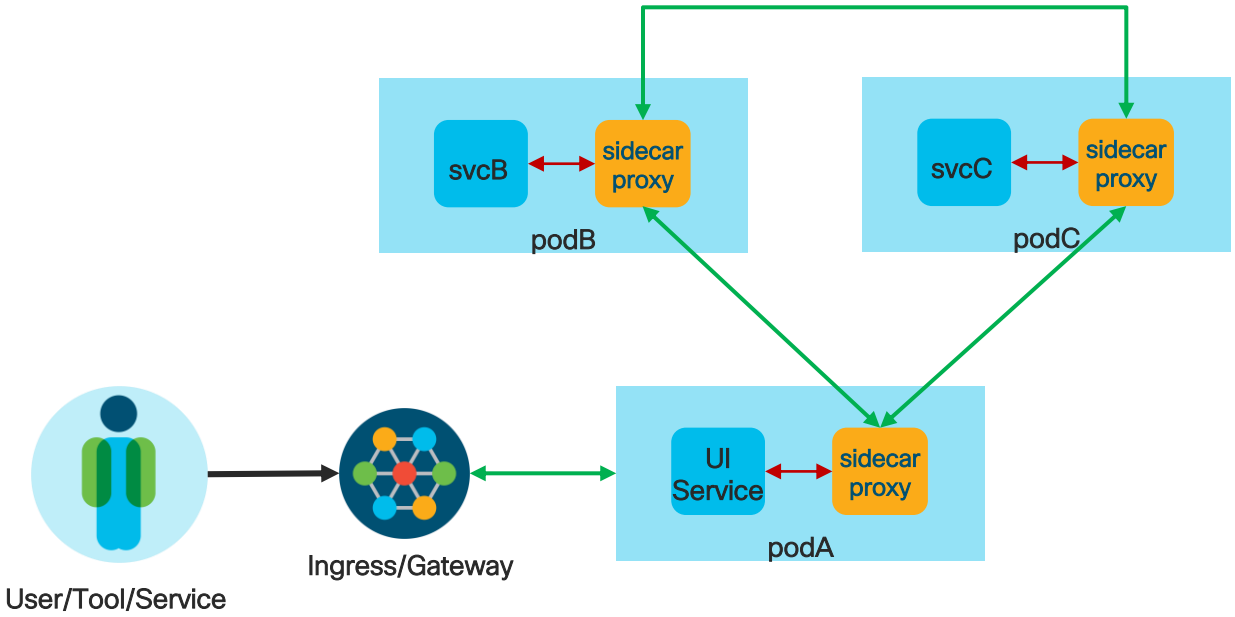
- Offload features for service-to-service communication to a policy-driven secure service mesh



Stuff I Want to Do

- I want to deploy a microservice
- I want to deploy using Kubernetes
- I have a bunch of requirements such as the need to handle:
  - Service failures
  - Retries
  - Circuit breaking
  - Topology changes
  - Monitoring
  - Tracing
  - Encryption between services
  - and more

What is a Service Mesh?



- *"The term **service mesh** is often used to describe the network of microservices that make up such applications and the interactions between them" – istio.io*
- Infrastructure layer for service-to-service communication
- Uses mesh of sidecar proxies:
  - Can inspect API transactions at Layer 7 or layer 3/4
  - Intelligent routing rules can be applied between endpoints

Istio Overview

Istio – Operator and Developer Goals

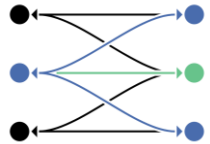
Operator Goals

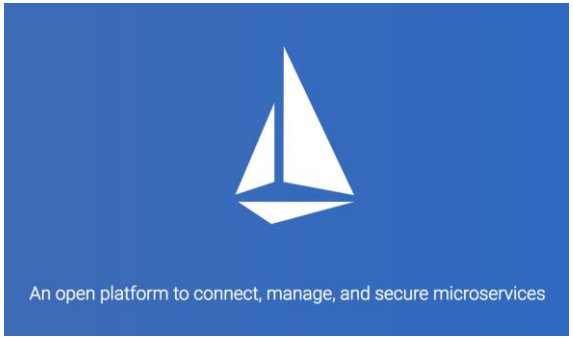
- ✓ Enable operators to move deployments between environments with less friction

Developer Goals

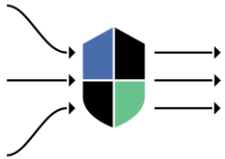
- ✓ Make services easier to implement for developers

Intelligent Routing and Load Balancing

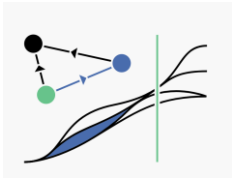




Policy Enforcement and Service Security



In-Depth Telemetry and Reporting



Resilience Across Languages and Platforms

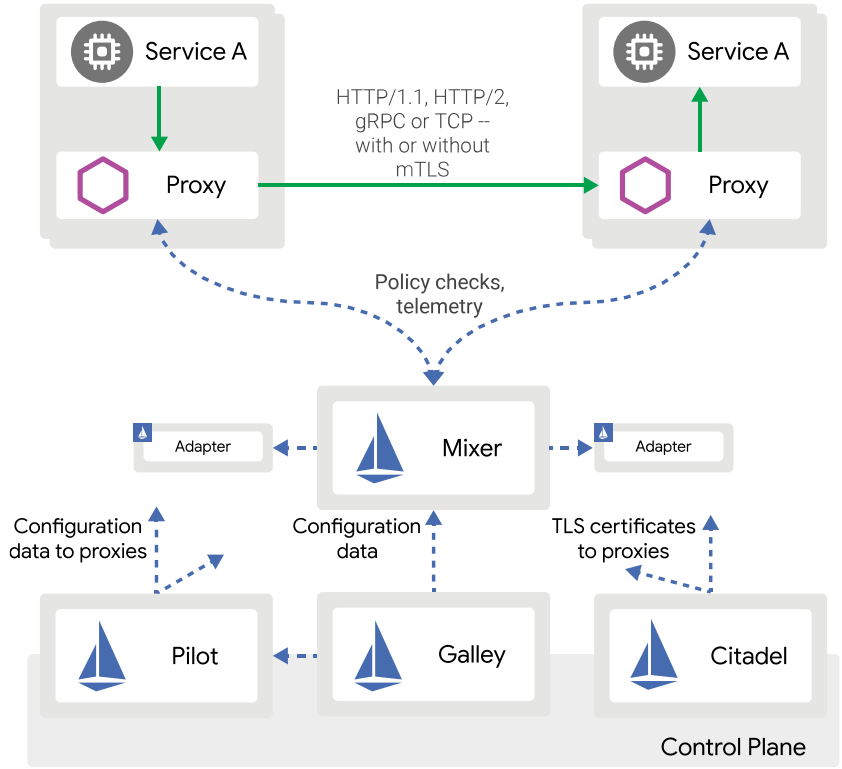
Istio Overview

- Open source project started by Google and IBM with help from the Envoy team at Lyft
  - https://istio.io/
  - https://github.com/istio
  - https://www.envoyproxy.io/
- https://istio.io/docs/concepts/what-is-istio/
  - Automatic load balancing for HTTP, gRPC, WebSocket, and TCP traffic
  - Fine-grained control of traffic behavior with rich routing rules, retries, failovers, and fault injection
  - A pluggable policy layer and configuration API supporting access controls, rate limits and quotas
  - Automatic metrics, logs, and traces for all traffic within a cluster, including cluster ingress and egress
  - Secure service-to-service authentication with strong identity assertions between services in a cluster

Istio Architecture

https://istio.io/docs/concepts/what-is-istio/

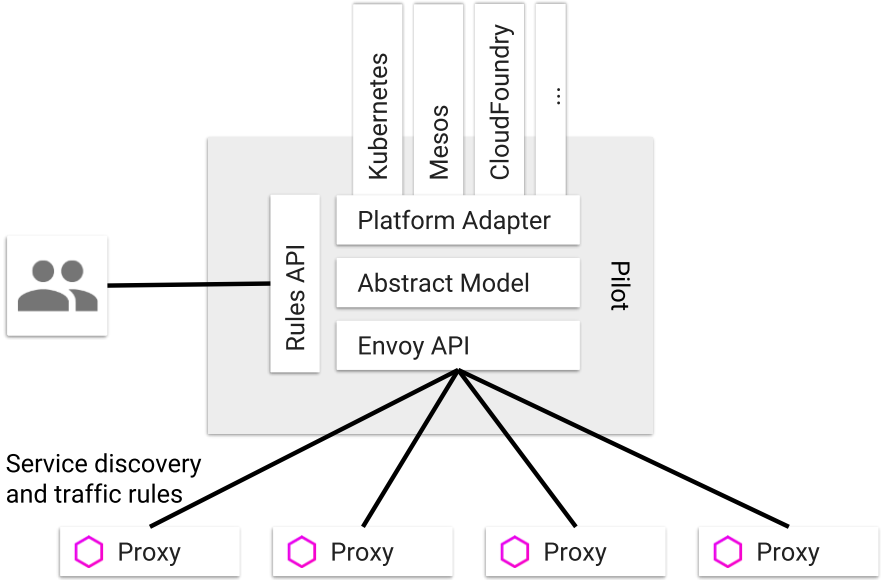
- Pilot
  - Handles service discovery and config data
  - Provides the Envoy proxies with the mesh topology and route rules
- Galley
  - Validates user authored Istio API configuration on behalf of other control plane components
  - Top-level config ingestion, processing and distribution
- Citadel
  - Provides certificates to the Envoy proxies for authentication and authorization
- Mixer
  - Handles Telemetry for back-end systems
  - Provides more sophisticated policy checks
  - Easily pluggable
- Envoy
  - A proxy attached to every microservice
  - The connection point for a microservice to attach to the mesh



Istio Control Plane

Pilot

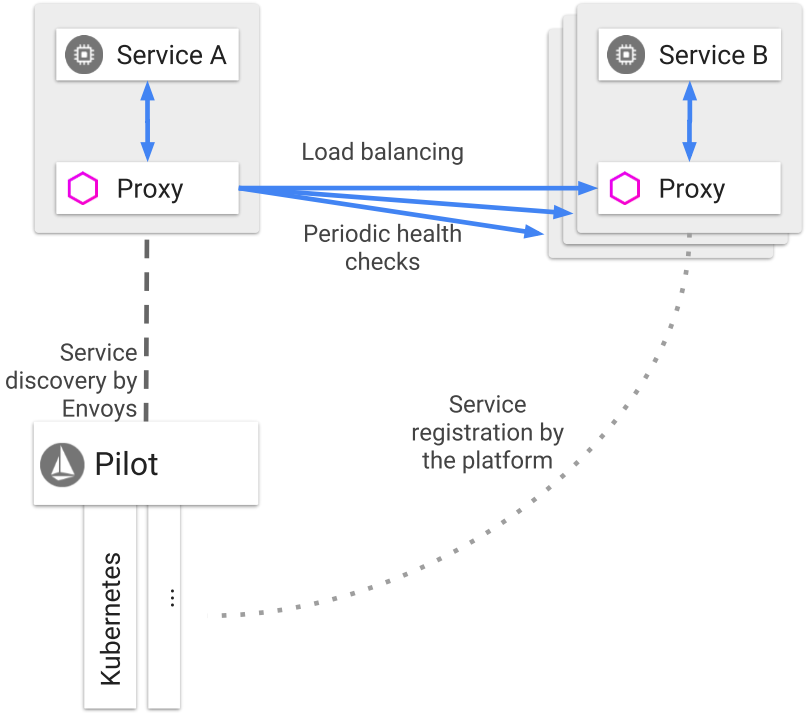
- Responsible for proxy configuration, exposes APIs for service discovery and traffic management (routing rules)
- Sends configuration data to the proxy at runtime
- Maintains a canonical representation of services in the mesh (abstract model)
- Platform-specific adapters (e.g. Kubernetes) watch for changes in pod info, ingress, etc. which is translated into proxy-specific (e.g. Envoy) configuration form



https://istio.io/docs/concepts/traffic-management/#pilot-and-envoy

Discovery and Load Balancing

- Service Registration and Discovery: Leverages platform-specific registration/discovery (e.g. K8s = environment variables or **DNS**)
- Pilot consumes info from the service registry and then provides that info in a platform agnostic way – Envoy updates load balancing info

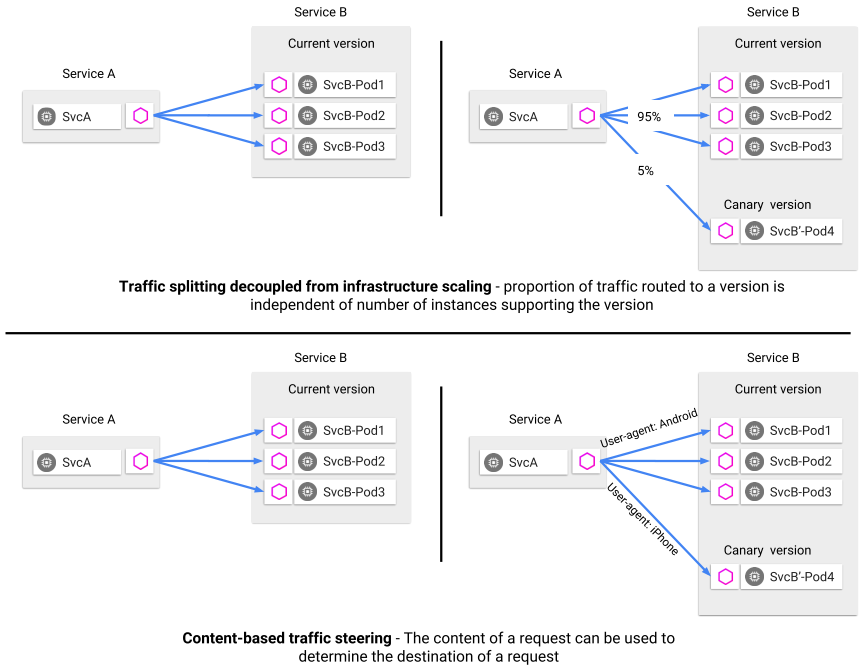


https://istio.io/docs/concepts/traffic-management/#discovery-and-load-balancing

Sophisticated Traffic Routing

- Istio provides a rich set options to route API requests.
- Matching criteria extends to layer 7 across numerous API formats (HTTP1, HTTP2, gRPC, etc.)
- Extends beyond just matching the request path to include headers and cookies.
- Multiple destination choices based on Kubernetes labels (e.g version labels)
- Various algorithms on how the requests are sent
  - Load balanced with a couple different hash options
  - Rate limited or quota enforced
  - Delay insertion
  - Circuit breakers
  - Time-outs and retries configurable per route

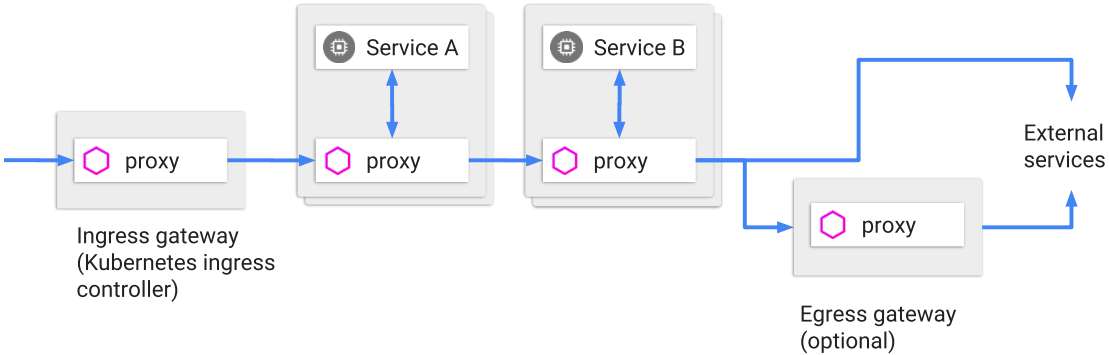
Istio Pilot – Traffic Management



- Maintains a canonical model of all the services in the mesh
- Rules are provided to route traffic between Envoy proxies (e.g: versions (v1, v2), environment (staging, prod))
- Configuration done for failure recovery features such as timeouts, retries, and circuit breakers
- Envoy's perform periodic service instance health-checks

https://istio.io/docs/concepts/traffic-management/

Istio Pilot - Traffic Management – Ingress and Egress



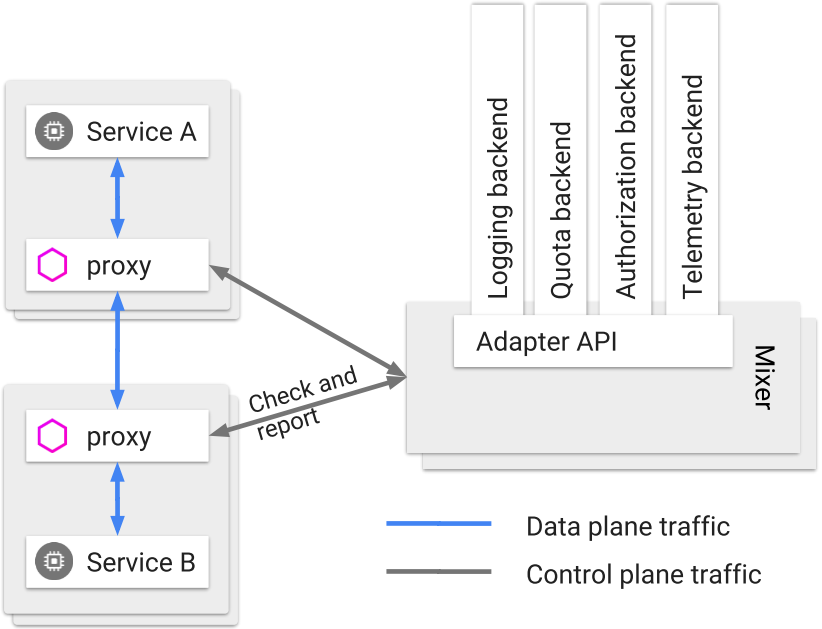
- Ingress Envoy's enable A/B testing, deploy canary services, etc. for user-facing services
- Egress failures gives failure recovery features such as timeouts, retries, circuit breakers, etc., and obtain detailed metrics on the connections to these services.

https://istio.io/docs/concepts/traffic-management/#ingress-and-egress

Mixer

Alpha: https://github.com/istio/istio/wiki/Mixerless-HTTP-Telemetry

- Policy engine for Istio
- Tracing, quotas, telemetry, access control, etc.
- Attribute processing:
  - Generated by Envoy
  - Call sent to backends via adapters
  - Policy actions (if/then actions)



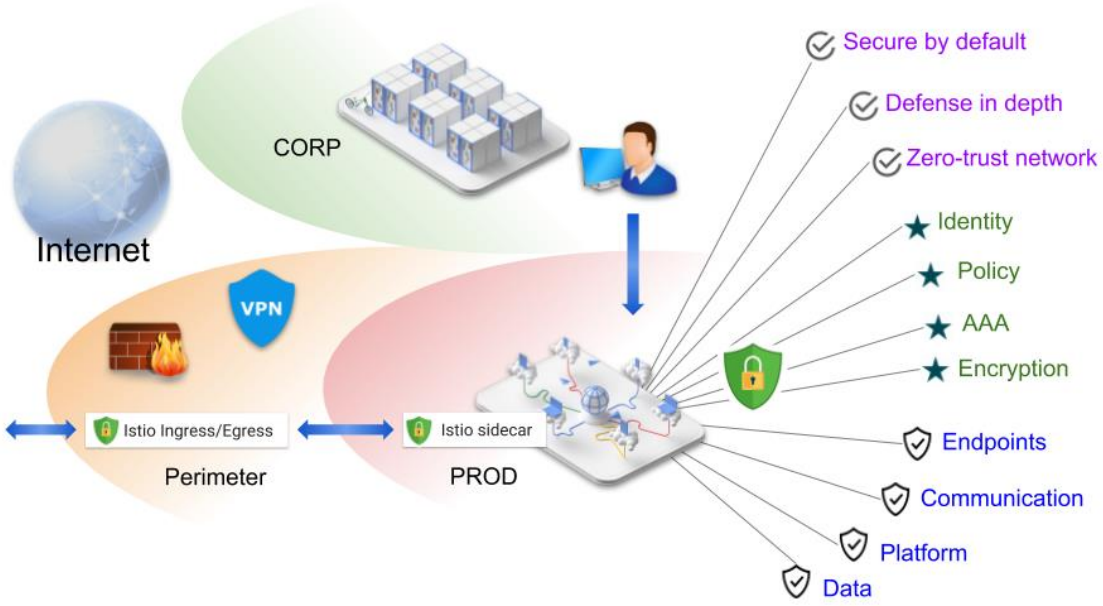
https://istio.io/docs/concepts/policies-and-telemetry/

Superior Visibility

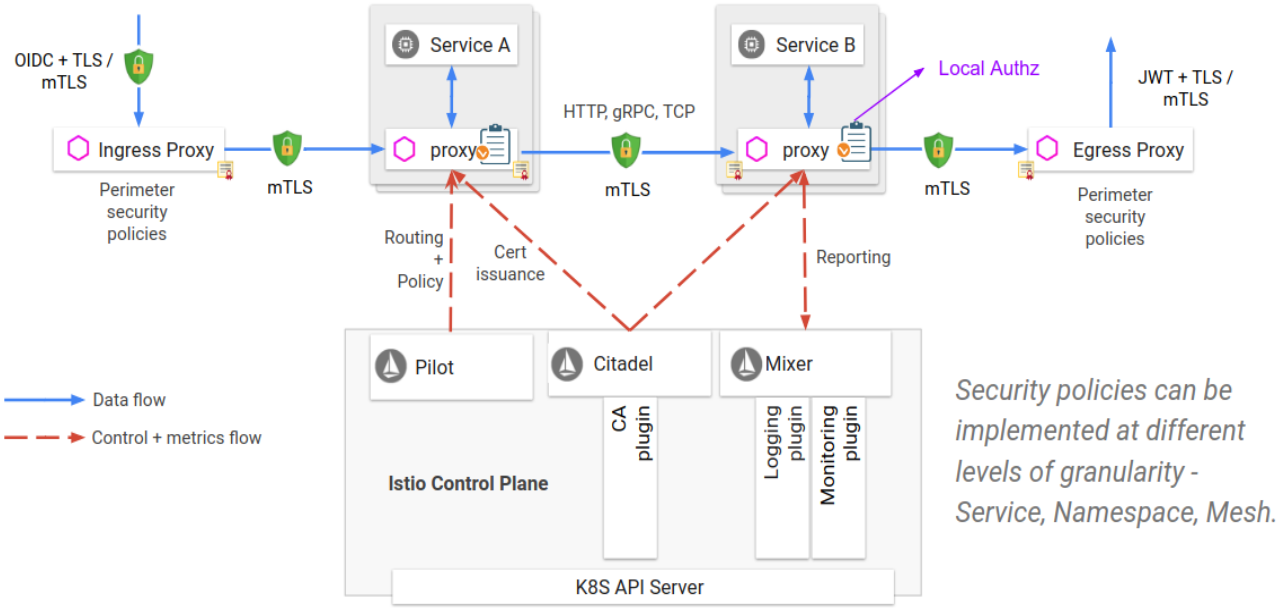
- Uniform set of metrics are collected across all microservices
- Tracing and collection is done by the proxy sidecar
  - On behalf of the microservice
  - Directly on the API transactions
- Passed to Mixer which has numerous integrations with cutting edge tools for analysis, logging and visibility.
- Maintainer of the application gets complete and uniform view across all microservices regardless of function.
- Developer of microservice is unburdened from developing individual tracing and metric module.

Istio Security Overview

https://istio.io/docs/concepts/security



Istio Security Architecture



Security policies can be implemented at different levels of granularity - Service, Namespace, Mesh.

Turbocharged Security for Applications

- Offers a huge shift in the security paradigm for applications
- Less emphasis on the enterprise firewall as the primary security vehicle
- Access control – Primarily layer 7 but also available at layer 3 and 4
  - Network policy much less important. Application policy much more
- Application security and policy definition moves more toward application owner versus network admin or IT admin
- Enables more granular admission control between the microservices that make up an application
- Less emphasis on in-band security enforcement and more capability to trigger out-of-band admission checks.
  - Allowing much higher level of intelligence and sophistication in the admission checks
  - Possibly significantly increasing latency and scale/performance concerns

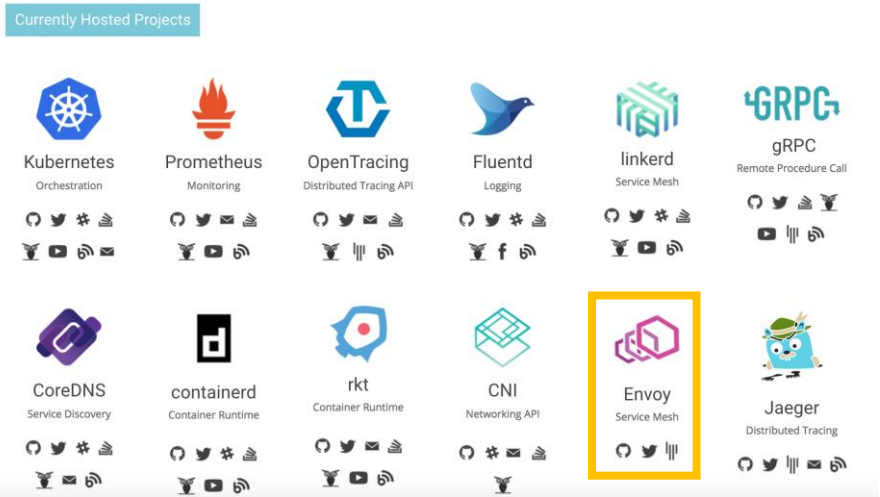
Istio Data Plane

Sidecar Proxy

- A sidecar container is companion to one or several other containers and it is meant to extend and enhance the main service container
- A sidecar proxy is a container as defined above only one that acts on behalf of the main service container for one/many uses
- A few of the many sidecar proxies:
  - Envoy, NGINX, haproxy, Linkerd

Envoy
https://www.envoyproxy.io

- Implemented by Lyft
- A C++ based L4/L7 proxy
- Can be used independently of any service mesh (Istio)



- API driven
- Traffic routing and splitting
- Transparent proxying
- Health checks, circuit breakers, etc.

USED BY

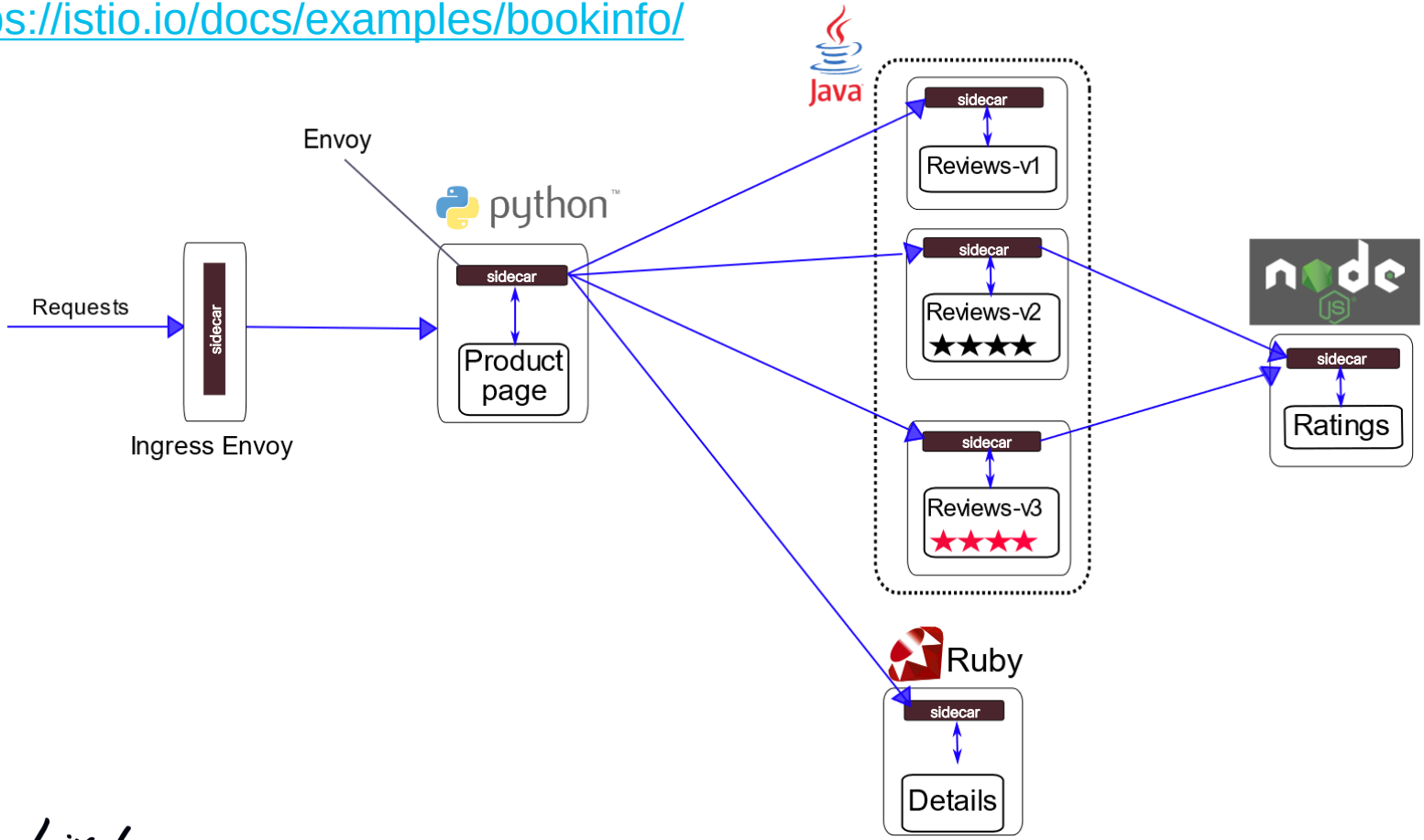
https://github.com/envoyproxy/envoy

Istio: How Does It Work?

- Istio Configuration Time
  - Injection of the Envoy proxy sidecar during application deployment
  - Envoy proxies are provided API endpoints of Pilot (to get the mesh state and routing rules) and Mixer (for telemetry reporting and dynamic access control)
  - Istio-Auth provides certificates and keys to the envoy proxies to both provide authentication and authorization capabilities to the proxies
- Service Deployment Time
  - Service Discovery – Pilot watches the Kubernetes APIs for creation and deletion of services
  - Mesh Creation – Pilot provides Envoy with xDS (service, cluster, endpoint and route Discovery Service (DS) database for mesh topology
- Active State and Policy Time
  - Security Model – Provided via Ingress policy, Route rules in the Mesh, and Attribute checking in Mixer
  - Route Rules – when operator configures route rules Pilot will update the mesh route rules
  - Server-side access control provided by Mixer by comparing incoming attributes against configured policy or propagating to 3rd party systems for access
  - Telemetry collection – Mixer collects the Envoy data and can provide it to 3rd Party add-ons (Prometheus, Grafana, Zipkin, etc.) for logging and analysis

Bookinfo

https://istio.io/docs/examples/bookinfo/



Other Stuff

Exciting New Stuff – You Can Help!

- Istio Feature Status: https://istio.io/about/feature-stages/
- Multicluster: https://istio.io/docs/setup/install/multicluster/
- IPv6 Support – Dual Stack is now supported!
- Istio and Network Service Mesh (NSM): https://networkservicemesh.io/docs/concepts/what-is-nsm
- Multitenancy
- Real world use cases

Istio: How Do I Get It?

Istio: How Do I Get It?

- Where to get it:
  - Istio currently is available directly from the Istio community at: https://istio.io/about/community/join/
  - It can also be built directly from Master at: https://github.com/istio/istio
  - It is likely to be available as an infrastructure option in some public clouds in the near future
  - It might be package in other distributions in the future
- How to install it (Kubernetes):
  - https://istio.io/docs/setup/getting-started/
  - Kubernetes installation is a prerequisite
  - Directly from the manifests included in the release
  - Using Helm charts included in the release

Contributing to Istio

Contribution

- Contribution Readme: https://github.com/istio/community/blob/master/CONTRIBUTING.md
- Contributing to the Docs: https://istio.io/about/contribute/
- Istio Discussion: https://discuss.istio.io/

It's About The Application... Not The Infrastructure

- Istio provides injection of proxy sidecars along with each microservice
- Injection is secure and automatic
- Allows the application developer to focus on the application
- The proxy sidecar connects cleanly to the microservice without additional development and regardless of the language used by the microservice
- Off-loads the authentication, authorization, telemetry (logs, metrics, tracing etc.), and network connectivity responsibilities
- Provides uniformity in both policy enforcement and telemetry collection
- Proxy sidecar (e.g. mesh state) updated automatically by Istio control plane

Summary

- Istio is a Service Mesh for microservices
- Istio allows you to enable security, health checking, optimized load-balancing and routing without adding code to the microservice directly
- Istio is young and maturing quickly
- Istio is a great community and you should get involved

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

Demos in the Cisco Showcase

Walk-In Labs

Meet the Engineer 1:1 meetings

Related sessions

Thank you

cisco Live!

You make **possible**