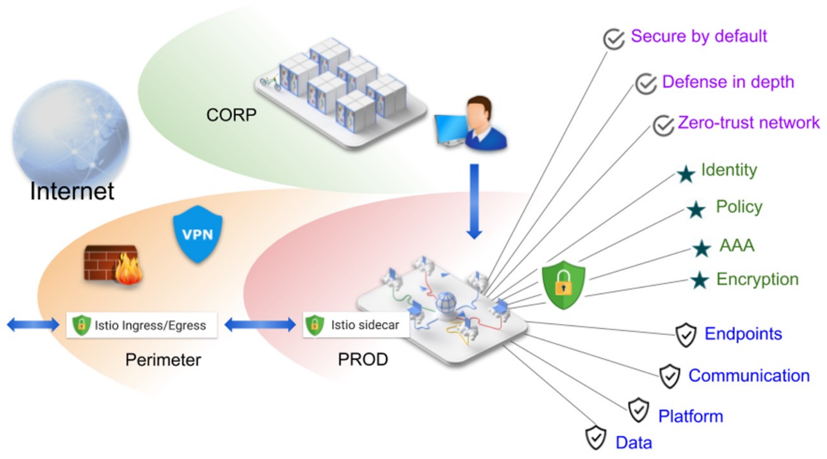
Istio and Envoy Proxy

Creating a Centrally-Controlled Service Mesh

Defense: Service Meshes

Service meshes bring encryption, service authentication, traffic control and observation to Kubernetes, among other features.

Istio is one example, wherein each pod is given a sidecar proxy through which all network traffic will flow.

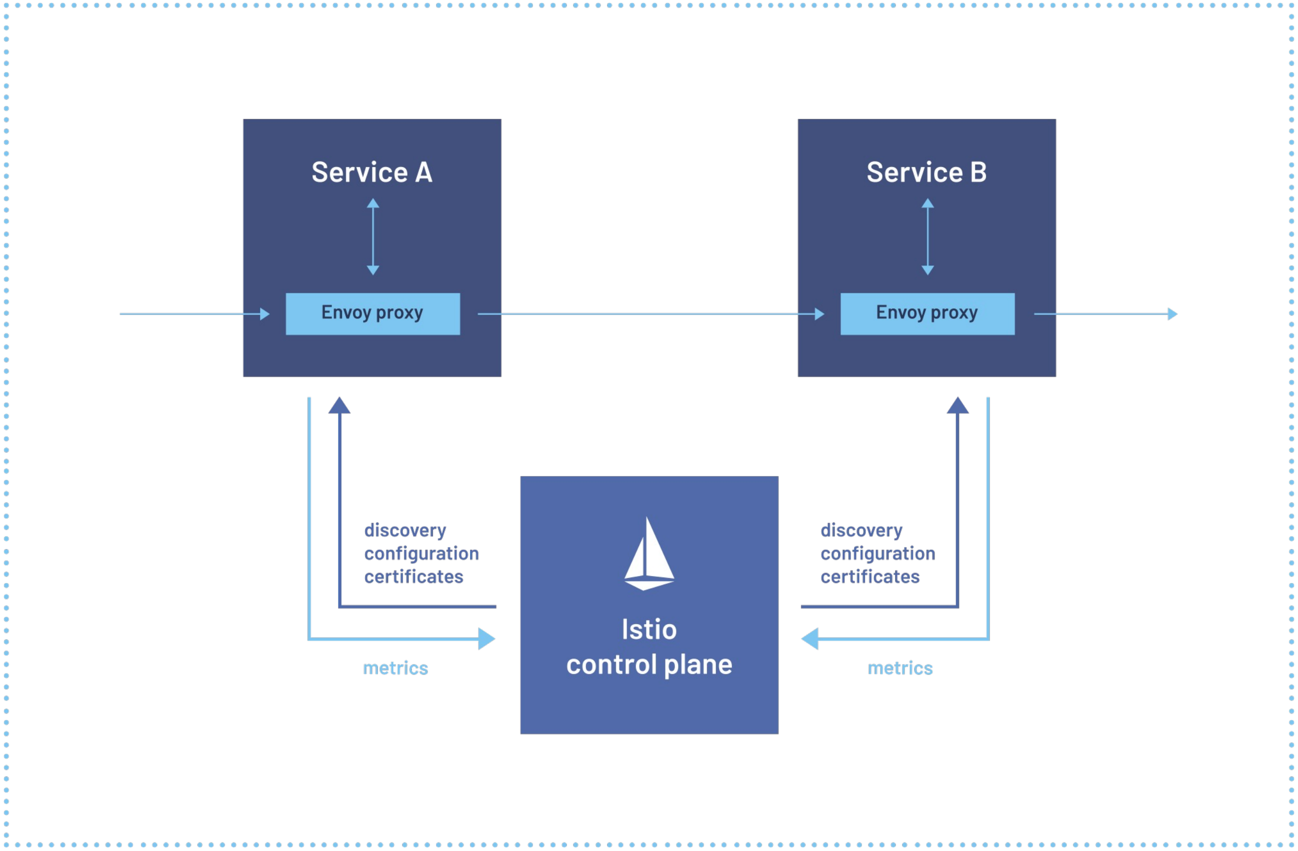


Reference and Image credit:
https://istio.io/docs/concepts/security/

Istio

- Istio is one of the prime service meshes.
- Created by Google
- Leverages the Envoy sidecar proxy, which is its own Open Source project
- Istio and Envoy could be their own four-day course – we cover some of the security benefits here.

Istio Envoy Proxies



Istio's Main Feature Set

- Traffic Control
- Resiliency
- Chaos Injection
- Observability
- Security

Traffic Control and Resiliency

- Traffic Control
  - Routing traffic to different versions of an application, in specific percentages
  - Extremely Application-Aware Routing
- Resiliency
  - Similar to Netflix's Hystrix
  - Load Balancing
  - Timeout, where the mesh returns an error to the client when the service doesn't respond
  - Retry, with exponential backoff added to the mesh
  - Circuit Breaker, where the mesh prevents an overloaded service from receiving new connections
  - Pool Ejection

Chaos Injection and Observability

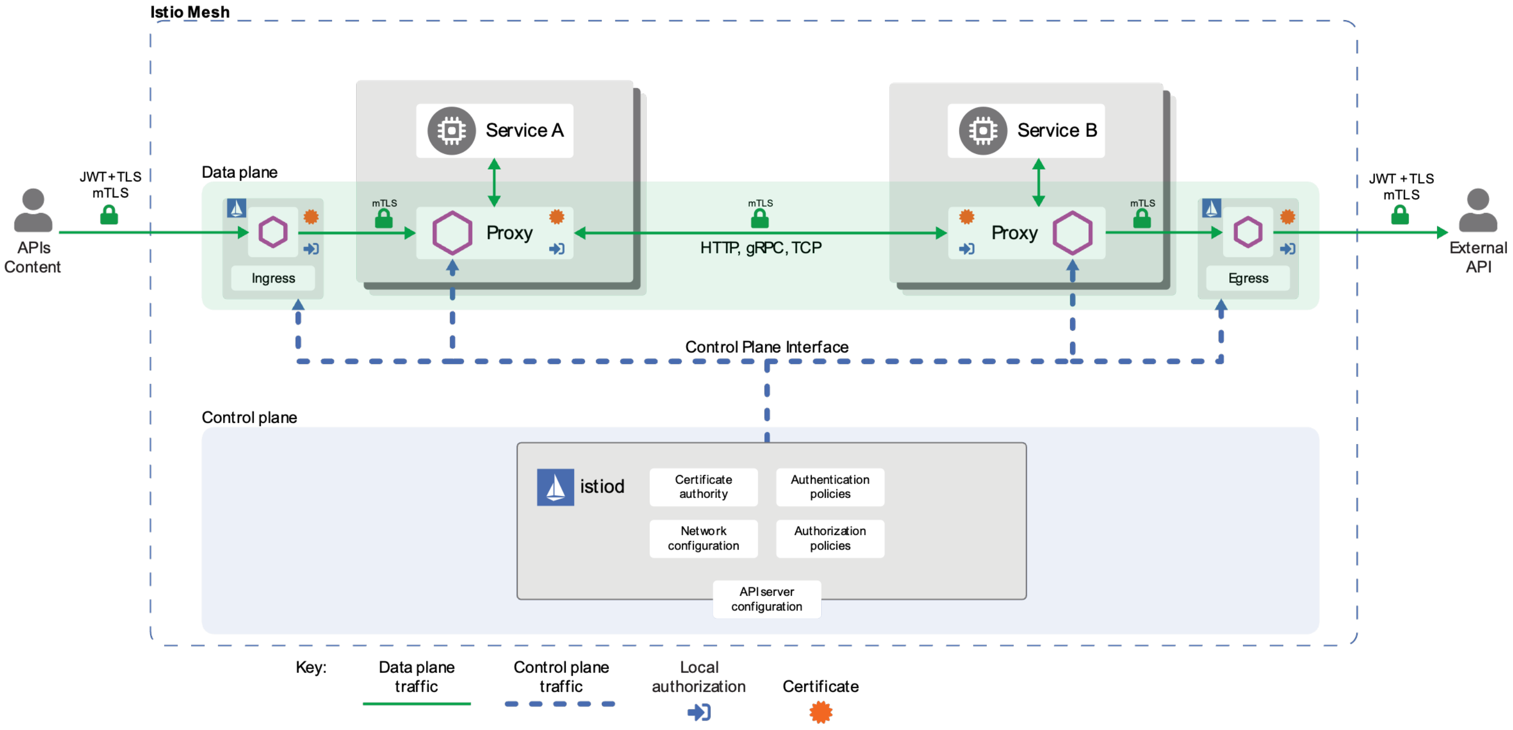
- Chaos Injection
  - Inserting HTTP Errors
  - Inserting Delays
  - Each of these allows developers to see whether a microservice-based application will fall apart if one part slows or fails.

- Observability
  - Tracing – observing the dependencies between microservices and path of execution
  - Metrics – leverages Prometheus and Grafana
  - Service Graph - visualization

Istio's Security Feature Set (ToC)

- Mutual TLS
- Encryption
- Network Segmentation
- Egress allowlisting

Istio Architecture



Reference:
https://istio.io/docs/concepts/security/architecture.svg

Mutual TLS and Encryption

Mutual TLS:

- every single pod authenticates itself to every other pod using a certificate
- istiod manages the certificates, including issuing, deploying, cycling
  - https://istio.io/docs/ops/security/keys-and-certs/

Encryption:

- every connection gains TLS encryption, making interception and modification of traffic within the cluster far more difficult.

Network Segmentation

- Network Segmentation – network access control within the cluster, via:
    - Pod name-based rules
    - Label-based rules
    - RBAC Service Account-based rules

- This is particularly useful for the "Zero Trust Networking" concept that was popularized by Google's BeyondCorp model.

Egress allowlisting

- If activated, every destination outside the cluster must be named
  - This was the default for the first few years of Istio's existence.
- All traffic leaving the cluster passes through the Egress Proxy.
- This severely hampers many of the kinds of attacks that we use in cloud-native environments.