Istio Architecture:

Istio is an implementation of a service mesh. A service mesh is a network of microservices that adds additional capabilities like:

- Service Discovery
- Load balancing
- Failure recovery
- Metrics
- Monitoring
- A/B testing
- Canary releases
- Rate limiting
- Access control
- End-to-end authentication

A service mesh allows applications to offload these capabilities from application-level libraries and allow developers to focus on differentiating business logic.

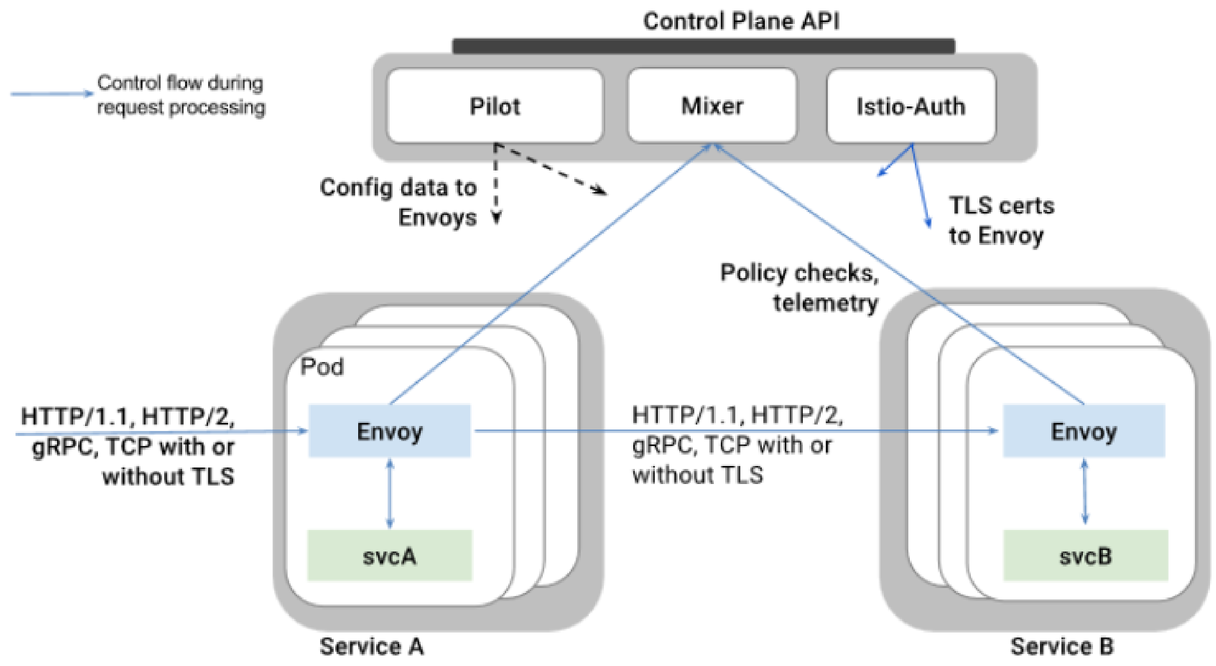
In the previous step we deployed Istio. Along with Istio many components have been deployed like:

Pilot, Mixer, Citadel, Envoy Proxy etc.

An Istio service mesh consists of two parts as, data plane and control plane.

Data plane — is composed of a set of intelligent proxies named Envoy which is deployed as a sidecar. These proxies mediate and control all the network communication between micro-services along with Mixer (a general-purpose and telemetry hub)

Control plane — manages and configures the proxies to route traffic. Plus it configures Mixers to enforce policies and collect telemetry.



Envoy: Envoy sidecar proxies serve as Istio's data plane. Built-in features such as failure handling (for example, health checks and bounded retries), dynamic service discovery, and load balancing make Envoy a powerful tool. Envoy also provides information about service requests through attributes.

Mixer: The mixer is a platform-independent component which accesses control and usage policies across the service-mesh and collects telemetry data from the envoy proxy and other services. The proxy extracts request level attributes and send them to Mixer to evaluate.

Pilot: provides services discovery for the Envoy sidecars, traffic management capabilities for intelligent routing (A/B testing, canary deployments etc) and resiliency (timeouts, retires, circuit breakers ..etc).

Pilot converts high-level routing rules that control traffic behaviour into Envoy-specific configurations and propagates them to the sidecars at runtime.

Pilot abstracts platform-specific service discovery mechanisms and synthesizes them into a standard format that any sidecar conforming with the Envoy data plane APIs can consume. This loose coupling allows Istio to run on multiple environments such as Kubernetes, Consul or Nomad while maintaining the same operator interface for traffic management.

Citadel: is also a main component in Istio service mesh. It provides strong service-to-service and end-user authentication with built-in identity and credential management. This can be used to upgrade unencrypted traffic in the service mesh. Using Citadel, operators can enforce policies based on service identity rather than on network controls.

Citadel manages keys and certifications across the mesh.

Galley: It validates user-authorised Istio API configuration on behalf of the other Istio control plane components.