Introduction to Envoy

A Programmable Proxy for Cloud Native Applications

Amitosh Swain Mahapatra
Product Engineer @ Gojek

$ whoami

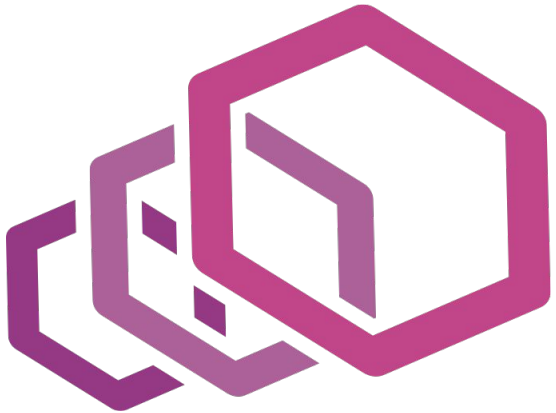
- Product Engineer @ Gojek
- Contributes to Open Source, makes cool things
- Find my code on Github @recrsn
- Catch me on Twitter @recrsn
- Lives in Bangalore, India

Agenda

- What is Envoy
- What is a “programmable proxy”
- Envoy xDS control plane
- What can envoy do for you
- Envoy as a smart load balancer
- Envoy as a proxy
- Envoy as an API gateway
- Monitoring protocol level stats with Envoy
- Envoy with Kubernetes
- Use cases

What is Envoy?

- Created by Lyft, now a graduated CNCF project
- Open-source level 4 and level 7 proxy
- Fully programmable and monitorable
- Supports many protocols such as HTTP, gRPC, MongoDB, Postgres, Redis, TCP and UDP
- Has a plug-in architecture for extensibility

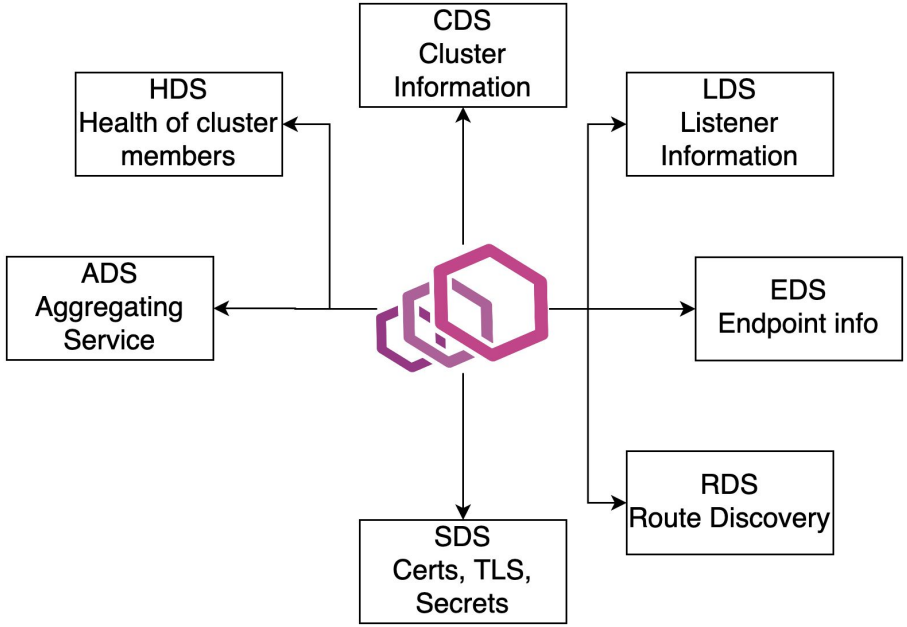


A “Programmable Proxy”

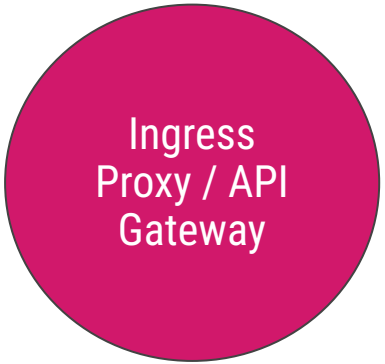
- Fully controllable by APIs
- All configuration can be fetched dynamically using xDS APIs
- You still have static FS based config, you are not forced to use APIs only
- API-style configuration makes it ideal for dynamic environments
- Authentication, Rate-limiting behaviors can also be made dynamic through authz APIs
- If it still doesn’t suit you then write a Lua/C++ filter

Envoy xDS control plane

- Envoy exposes a set of APIs called xDS to dynamically configure it
- Can be implemented as JSON-RPC or gRPC streams for even high-performance



What Envoy can do for you?



Control traffic entering your microservice clusters, perform authn and authz

A normal forward proxy, infront of ext services providing circuit-breaking, mutual authentication etc,

Quite a smart load balancer, both internal and external

Allows visibility into protocol level stats of L4/L7 traffic flowing through Envoy

Envoy as a Load Balancer

- Envoy supports dynamic configuration of LB cluster members (CDS, EDS & RDS)
- Supports active health-checks and adaptive load balancing
- Full support for HTTP 1 and 2
- Also supports MongoDB, Postgres, Redis protocols
- Can also load-balance any protocol which can be load-balanced by distributing packets in L4 level
- Can perform load balancing in server as well as client-side

Envoy as a Proxy

- Converts HTTP 1 to 2 and vice versa
- gRPC transcoding from JSON to Protobuf, gRPC web
- Reverse proxy for microservices to handle TLS, rate-limiting, authentication
- Forward proxy for external service calls to handle client-side TLS, authentication, bulkhead, circuit-breaking etc.
- As a Redis proxy for sharding keys, easy Redis cluster configuration
- TLS termination (SSL offloading)

As an API Gateway

- Programmable, hence adding routes and clusters dynamically is easy
- Block and allow traffic over routes using the xDS apis
- Authorizing clients can be taken care by plugging in a service implementing the Envoy authz protocol
- Rate-limit based on clients and routes

Many projects exist which extend Envoy capabilities as an API gateway

- Gloo
- Ambassador API gateway
- Heptio

Monitoring Traffic with Envoy

- Envoy exposes metrics through StatsD and Prometheus protocol
- These include success and failure counts, timings etc.
- Extremely helpful to debug network issues between microservices
- Stats for other protocols such as MongoDB, Postgres etc are also available
- By reading stats from a proxy, you have no additional software extension in your DB or expensive DB queries by monitoring agents

Envoy with Kubernetes

- The programmability of Envoy makes it easy to integrate with Kubernetes
- Envoy is used as a sidecar proxy with several service mesh implementations like Istio and Consul connect.
- You can use Envoy as a L7 load balancer getting its info from Kubernetes
- Envoy as a Ingress controller by the Gloo project or Heptio

Use Cases

- Istio is a full featured service-mesh built using Envoy for mesh load-balancing. Envoys are used as a lightweight sidecars for all incoming and outgoing traffic
- Envoy easily handles authentication requirements with third-parties, like payment APIs. Acting as a fronting proxy, security headers or TLS can be added to every request to the party.

Use Cases

- As a proxy for Redis sharding and high-availability
- As a proxy for Postgres failover
- As gRPC API gateway - exposing REST endpoints to external parties over a gRPC endpoint
- Client-side and server-side load-balancing with data from custom services
- Transparently associating distributed tracing information to existing services without modification

Questions?