Introducing MetaProtocol Proxy: A Layer-7 Proxy Framework Powered by Envoy

Huabing Zhao - Tetrate.io

envoycon

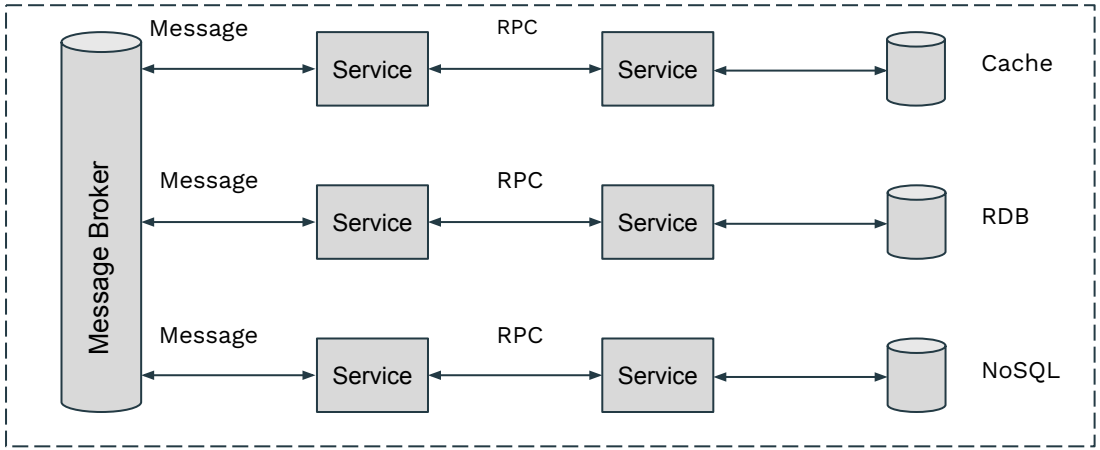
NORTH AMERICA

Overview

- Background
- MetaProtocol Proxy
- Use Cases
- Demo

Protocols Used in Microservices

- There are various types of layer-7 traffic in microservices besides HTTP/gRPC
  - RPC：Thrift, Dubbo, proprietary/Private RPC Protocols ...
  - Messaging: Kafka, RabbitMQ ...
  - Cache: Redis, Memcached ...
  - Database: mySQL, PostgreSQL, MongoDB ...
  - Other Layer-7 Protocols: ...
- Most of(All?) the current sidecar/edge proxies don't understand these protocols
  - They mainly focus on HTTP
  - Other protocols are treated as plain TCP



What We Want vs What We Get

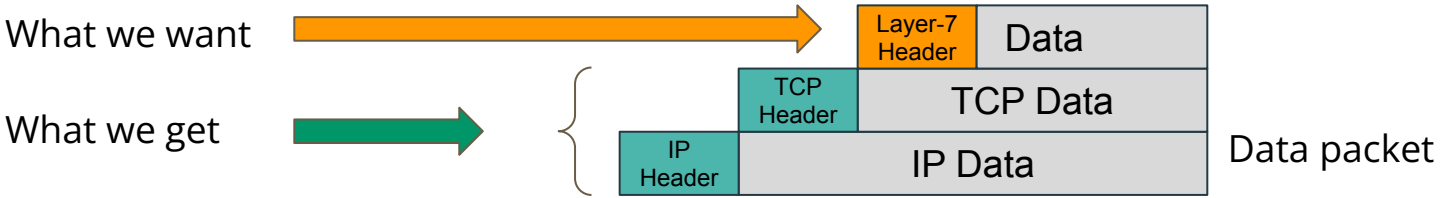
envoycon NORTH AMERICA

What do we want?

- Traffic management based on layer-7 headers
  - Load balancing at requet level
  - Rate limiting at request level
  - Retry at request level
  - Routing based on layer-7 headers (Thrift service name/method name, Dubbo Interface/method/attachment etc.)
  - Fault Injection with application layer error codes
  - ...
- Application layer observability
  - Stats: request latency/response status
  - Request access log
  - Tracing
- Application layer security
  - User Authentication & Authorization
  - ...

What do we get?

- Traffic management based on layer-3/4 headers
  - Routing based on IP address, TCP Port and SNI
- Connection layer observability
  - Stats: TCP sent/received bytes/ opened/closed connections
  - Connection layer access log
- Security
  - Connection level authentication: mTLS
  - Connection level authorization: Source IP/ Dest Port/Subjects in Certs



Similarity of Layer-7 Protocols

The layer-7 processing of a proxy for different protocols is quite similar:

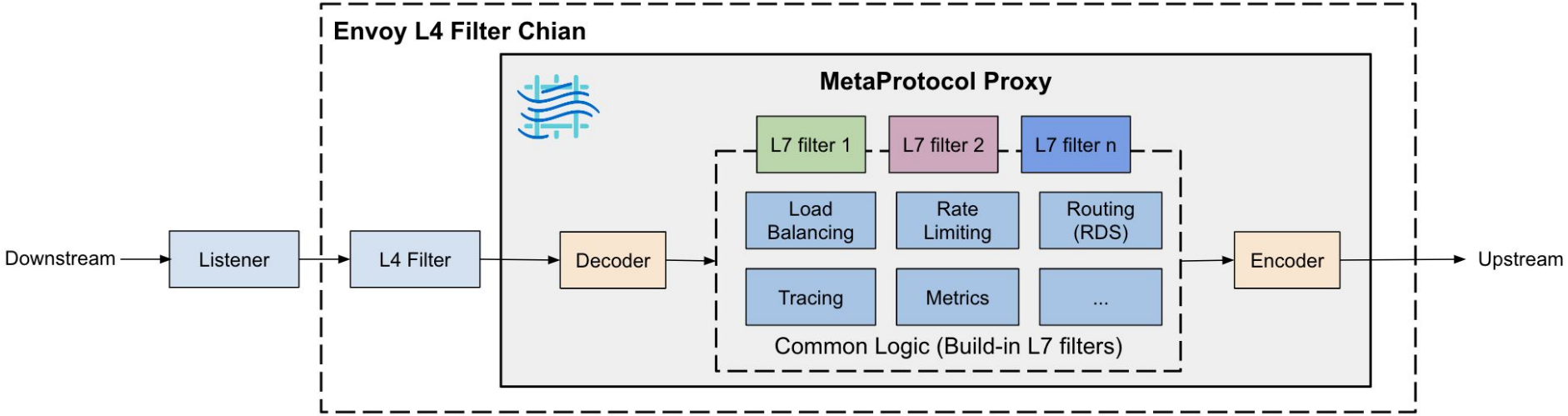
- Extract layer-7 headers form the tcp stream
- Service Discovery and Routing based on layer-7 headers
- Other processing based on layer-7 headers: Load Balancing, Rate Limiting, Observability, etc.

Protocol	Destination service	Parameters can be used for routing
HTTP 1.1	host	host, path, method headers
HTTP 2	pseudo header: authority	pseudo header: authority, path, method, headers
gRPC	HTTP 2 path	Request-Headers(Delivered as HTTP2 headers)
TARS	ServantName	ServantName, FuncName, Context
Dubbo	service name	service name, service version, service method
Any RPC Protocol	service name in message header	some key:value pairs in message header

MetaProtocol Proxy: A Layer-7 Proxy Framework

In the world of layer-7 protocols, managing traffic is usually done in a similar way. Instead of building a separate Envoy filter for each protocol, we can gather all the common functions of a layer-7 protocol proxy in one place - the MetaProtocol Proxy filter.

- Two-layer filter chain architecture:
  - MetaProtocol Proxy: a filter in the Envoy L4 filter chain
  - MetaProtocol L7 filter chain:
    - Common logic built into the framework and L7 filters: Load balancing, Rate limiting, Routing(Static and dynamic), Traffic mirroring, Tracing, Metrics, Logging, etc.
    - Can be extended through custom C++, Lua, and WASM L7 filters
- To create a new layer-7 proxy, only the codec interface(decoder and encoder) needs to implemented (a few hundred lines of code)



Request Path and Response Path

envoycon NORTH AMERICA

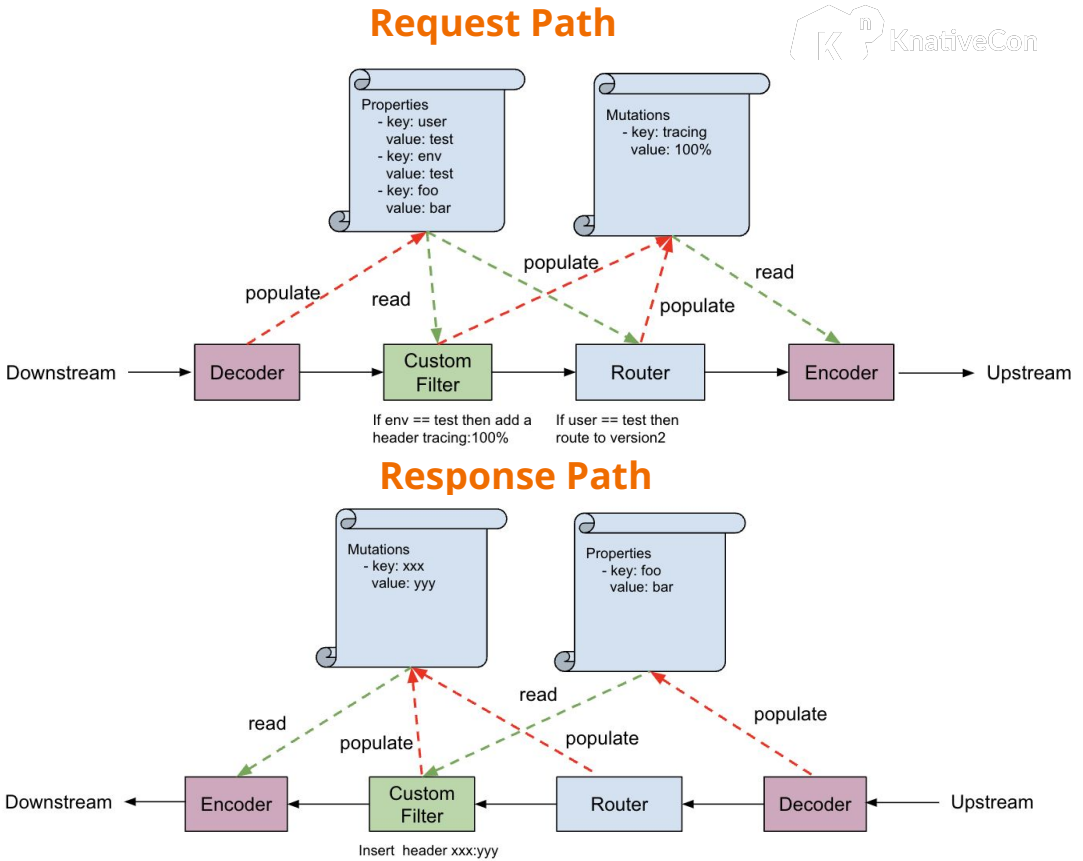
Two important data structures:

Metadata

- Extracted from data packet by the decoder
- Can be any arbitrary key-value pairs
- Used by L7 filters: routing match, rate limiting match, etc.

Mutation

- Populated by the L7 filters
- Can be any arbitrary key-value pairs
- Used by the encoder to mutate data packet



Adding a New Protocol

envoycon NORTH AMERICA

1. Implement codec interface

```
class Codec {
public:
  virtual ~Codec() = default;

  /*
   * decodes the protocol message.
   *
   * @param buffer the currently buffered data.
   * @param metadata saves the meta data of the current message.
   * @return DecodeStatus::DONE if a complete message was successfully consumed,
   * DecodeStatus::WaitForData if more data is required.
   * @throws EnvoyException if the data is not valid for this protocol.
   */
  virtual DecodeStatus decode(Buffer::Instance& buffer, Metadata& metadata) PURE;

  /*
   * encodes the protocol message.
   *
   * @param metadata the meta data produced in the decoding phase.
   * @param mutation the mutation that needs to be encoded to the message.
   * @param buffer save the encoded message.
   * @throws EnvoyException if the metadata or mutation is not valid for this protocol.
   */
  virtual void encode(const Metadata& metadata, const Mutation& mutation,
                      Buffer::Instance& buffer) PURE;

  /*
   * encodes an error message. The encoded error message is used for local reply, for example, envoy
   * can't find the specified cluster, or there is no healthy endpoint.
   *
   * @param metadata the meta data produced in the decoding phase.
   * @param error the error that needs to be encoded in the message.
   * @param buffer save the encoded message.
   * @throws EnvoyException if the metadata is not valid for this protocol.
   */
  virtual void onError(const Metadata& metadata, const Error& error, Buffer::Instance& buffer) PURE;
};
```

2. Configure the application protocol codec

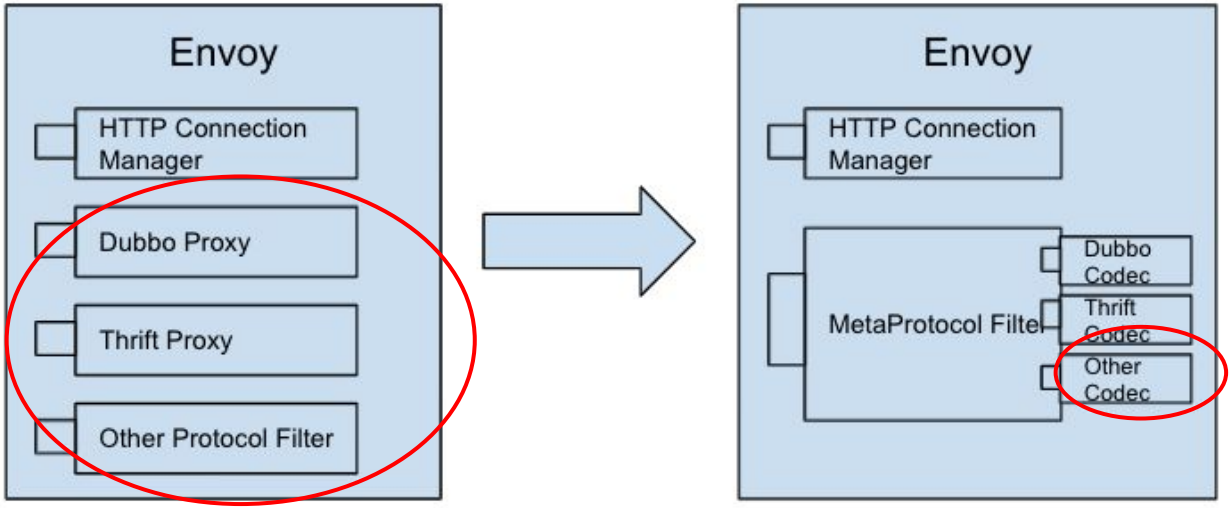
```
- filters:
  - name: aeraki.meta_protocol_proxy
    typed_config:
      '@type': type.googleapis.com/aeraki.meta_protocol_proxy.v1alpha.MetaProtocolProxy
      protocol:
        name: dubbo
        codec:
          name: aeraki.meta_protocol.codec.dubbo
```

Example code:
https://github.com/aeraki-mesh/meta-protocol-awesomerpc

Adding a New Protocol

Work comparison:

- Before: Huge; write a full-fledged l4 filter (considering the efforts of writing an Http Connection Manager filter)
- After: Small; write a codec implementation(normally a few hundred of lines, can be done in 1 week by 1 person)



Supported Protocols and Use Cases

Supported Protocols：More than 10 open source and private protocols

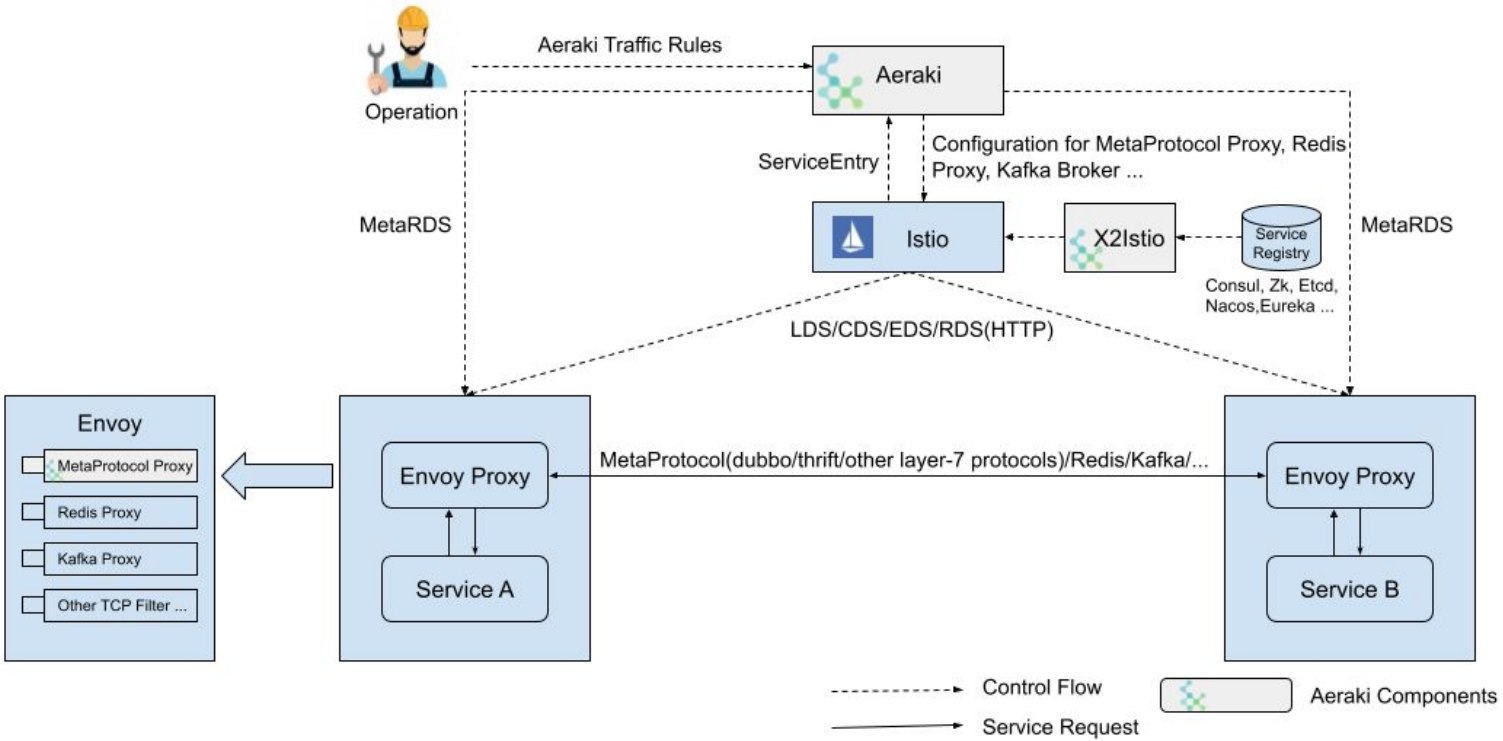
- Dubbo - open source protocol
- Thrift - open source protocol
- bRPC - open source protocol
- tRPC - private protocol used in Tencent
- xxx - private protocol used in Huawei
- xxx - private protocol used in Tencent Music
- xxx - private protocol used in Tencent Games
- ……

Use cases：

- 2022 Olympic online streaming service - private protocol
- Tencent Music - private protocol
- Boss Zhiping - Dubbo, Thrift
- Alauda cloud - Dubbo
- More use cases：https://github.com/aeraki-mesh/aeraki/issues/105

Demo - MetaProtocol Proxy as Sidecar Proxy

Control Plane: Istio + Aerkai(for none-HTTP)
Data Plane: MetaProtocol Proxy



Q&A

envoycon NORTH AMERICA

KnativeCon NORTH AMERICA

https://github.com/aeraki-mesh/meta-protocol-proxy
https://www.aeraki.net/docs/v1.x/tutorials/implement-a-custom-protocol