

Your laptop as part of the service mesh

by Lorenzo Fundaró
SRE @ Omio



#IstioCon

What's on the menu today

- EnvoyFilter in practice
- Demo
- Inspiration

#IstioCon



Questions



#istiocon @ istio.slack.com



@lfundaro

#IstioCon



How it all started

Omio is

- Flights + Buses + trains: all under a single transaction
- Integrates 800+ transportation providers across Europe and North America



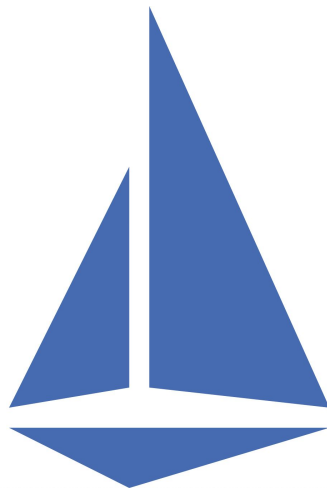
Our developer flow

Develop -> PR -> master -> deploy QA -> deploy Production

#IstioCon



Our clusters



#IstioCon



The problem

Running end-to-end tests at Omio is both not **efficient** and **cost-effective**



How tests are run

- On QA (dev -> PR -> master -> deploy QA)
- On standalone VMs running all services needed for the test



**Define efficient
and
cost-effective**

#IstioCon



1. Minimize time to bug detection

Dev -> PR -> master -> **QA** -> prod

3 steps away to find a problem



2. Allow simultaneous tests

Only one commit at a time from your microservice



3. Reuse existing infrastructure

- Minimize costs
- Reuse existing infrastructure to run tests



Why don't you ?

- Mock ?
- Contract testing ?



Why don't you ?

~~Mock ?~~

~~Contract testing ?~~

At a scale of 800+ providers ?

Mocks are like any other software:

- Bugs
- Maintenance

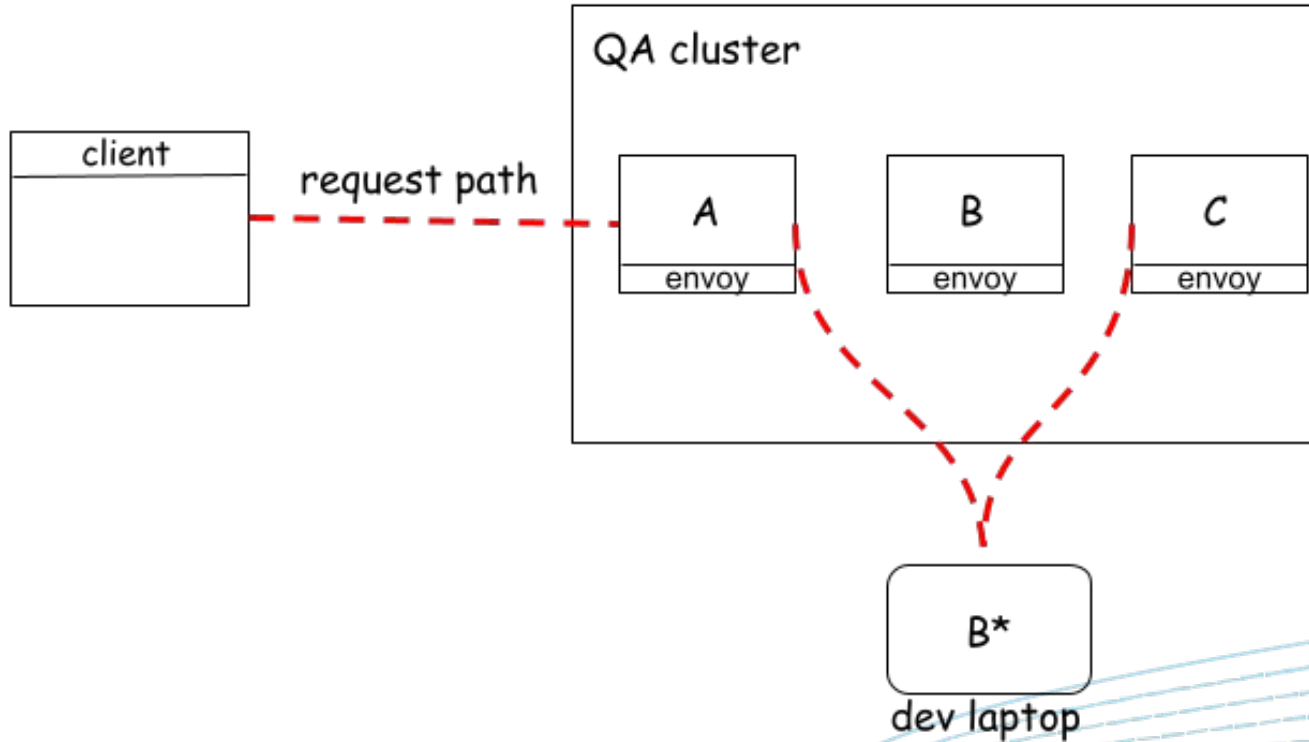


Can we do better ?

#IstioCon



What if ?



EnvoyFilter

#IstioCon



Envoy HTTP LuaFilter

```
function envoy_on_request(request_handle)
function envoy_on_response(request_handle)
```



Who and where to reroute ?

#IstioCon



The contract

```
GET / HTTP/1.1
```

```
Host: example.com
```

```
User-Agent: curl/7.64.1
```

```
X-devroute: { "foo": "192.168.1.12:8001" }
```

```
Accept: */*
```



Pseudo implementation

```
1  function envoy_on_request(request_handle)
2      contract = request_handle:headers():get("x-devroute")
3      if string.match(contract, "foo") == nil then
4          return
5      end
```



```
6      -- we have a contract match
7      address = contract["foo"]
8      headers = request_handle:headers()
9      -- send the request somewhere else
10     response = request_handle:httpCall(address,headers,..)
11     -- respond immediately and don't proxy to original Foo
12     request_handle:respond(response)
13 end
```



Ouch !

- Your laptop is not part of the mesh club

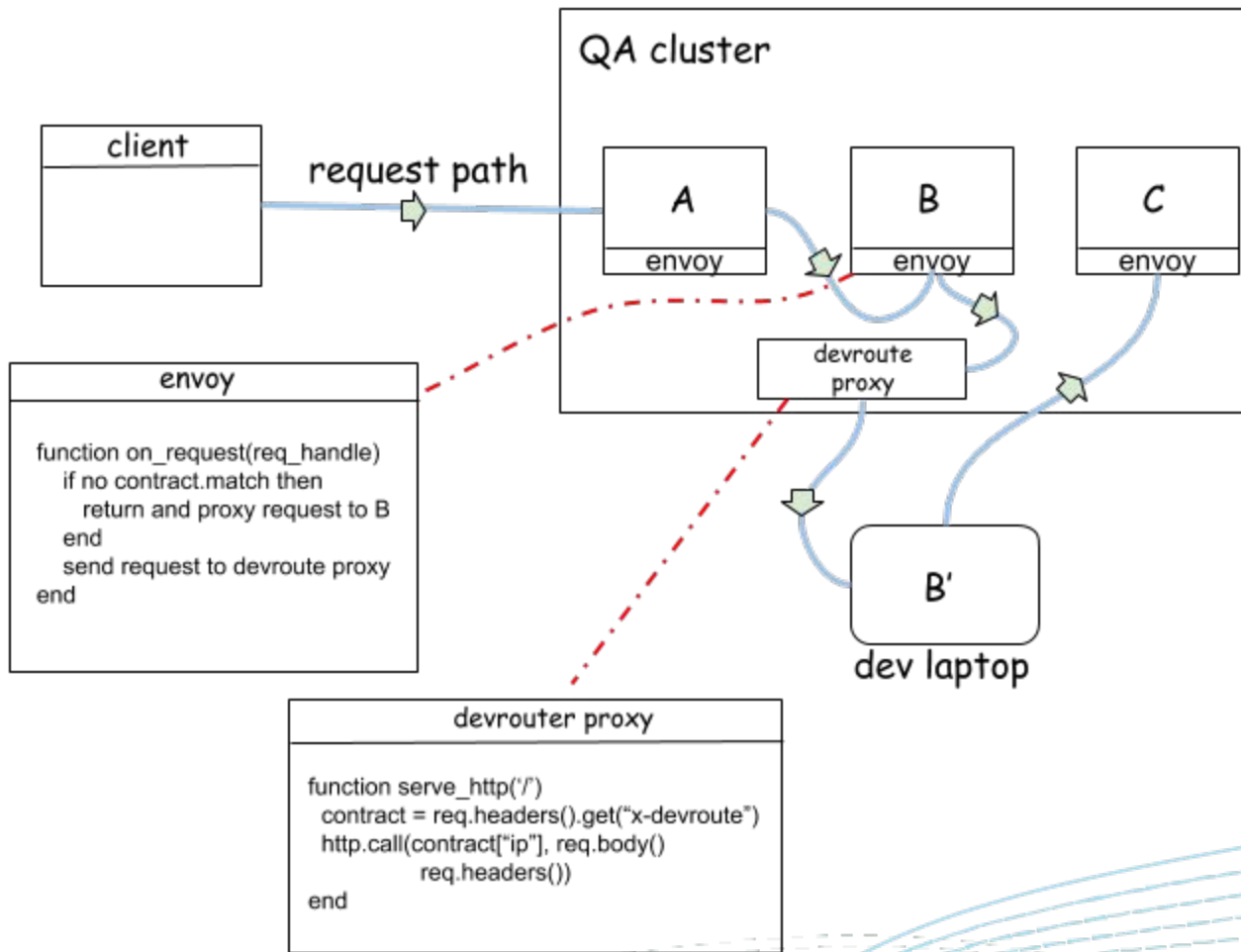
#IstioCon



A dummy proxy for the mesh

- Called by Lua code
- Parses the contract header and makes http call





Wait ... What about VirtualService ?

More work for developers to consume this contract

- Prior knowledge of Istio
- Need to create VirtualService and DestinationRule before anything happens
- VirtualService evaluation order matters



Checkpoint

1. Minimize time to bug detection: **yes**
2. Allow simultaneous tests: **yes**
3. Reuse infrastructure: **yes**



Drawbacks

Contract header needs to be preserved all the way through the call chain



Demo time

#IstioCon



Thank you !

#IstioCon

- Your laptop as part of the service mesh @ Medium
 - Reference implementation and run-it-yourself-demo at github.com/omio-labs/devrote
-