envoycon

Dynamic Request Routing with Envoy

Ben Plotnick - @benplotnick

About Me

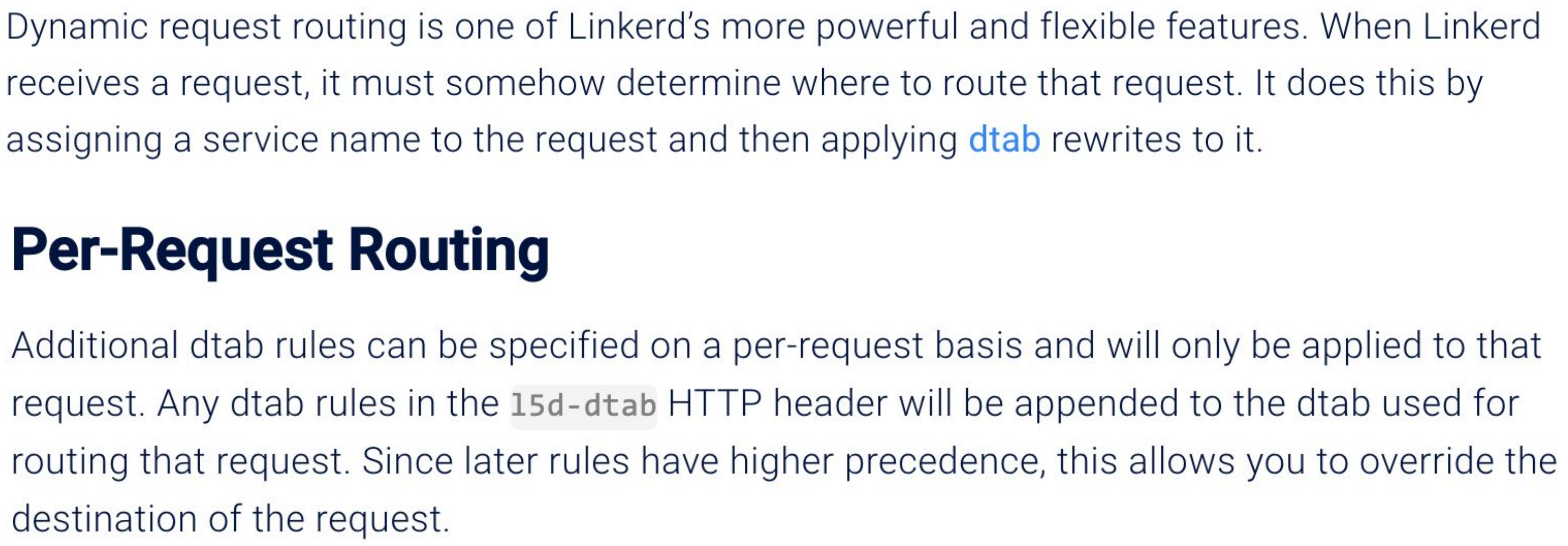
envoycon

Currently PaaS @Cruise

Formerly Service Infra @Yelp

Dynamic Request Routing

envoycon



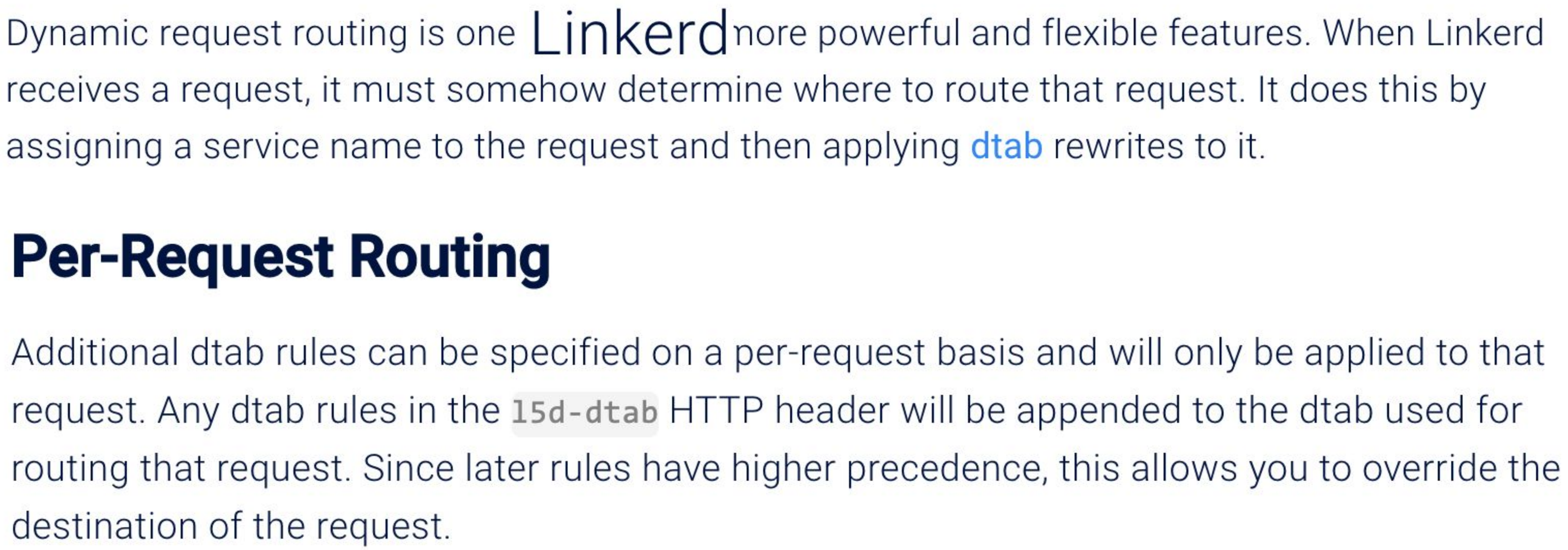
Dynamic request routing is one of Linkerd's more powerful and flexible features. When Linkerd receives a request, it must somehow determine where to route that request. It does this by assigning a service name to the request and then applying dtab rewrites to it.

Per-Request Routing

Additional dtab rules can be specified on a per-request basis and will only be applied to that request. Any dtab rules in the `l5d-dtab` HTTP header will be appended to the dtab used for routing that request. Since later rules have higher precedence, this allows you to override the destination of the request.

Dynamic Request Routing

envoycon

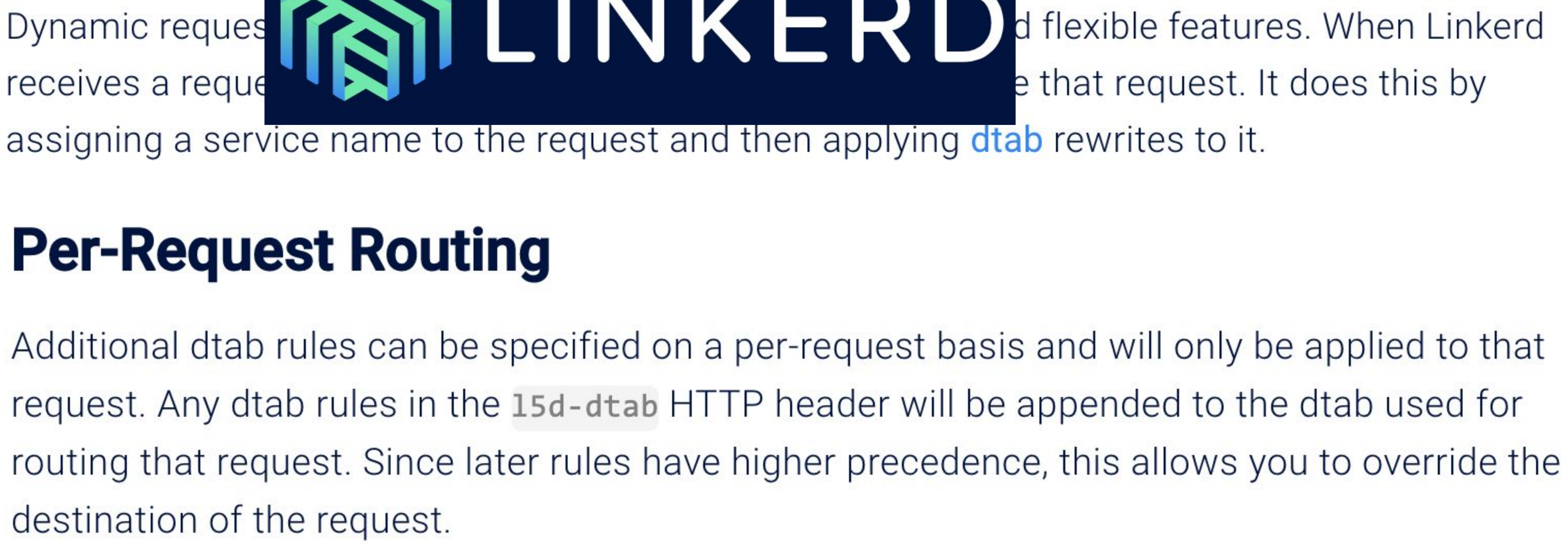


Dynamic request routing is one Linkerd nore powerful and flexible features. When Linkerd receives a request, it must somehow determine where to route that request. It does this by assigning a service name to the request and then applying dtab rewrites to it.

Per-Request Routing

Additional dtab rules can be specified on a per-request basis and will only be applied to that request. Any dtab rules in the `l5d-dtab` HTTP header will be appended to the dtab used for routing that request. Since later rules have higher precedence, this allows you to override the destination of the request.

Dynamic Request Routing



Dynamic reques d flexible features. When Linkerd receives a reque e that request. It does this by assigning a service name to the request and then applying dtab rewrites to it.

Per-Request Routing

Additional dtab rules can be specified on a per-request basis and will only be applied to that request. Any dtab rules in the `l5d-dtab` HTTP header will be appended to the dtab used for routing that request. Since later rules have higher precedence, this allows you to override the destination of the request.

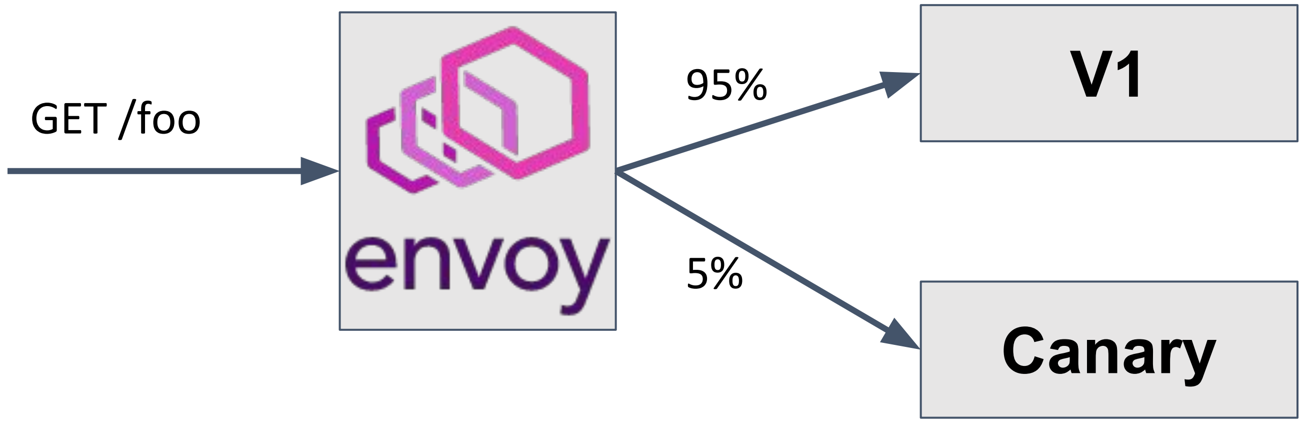
What is Dynamic Request Routing?

Weight-based routing

Header-based routing

What is Dynamic Request Routing?

Weight-based routing

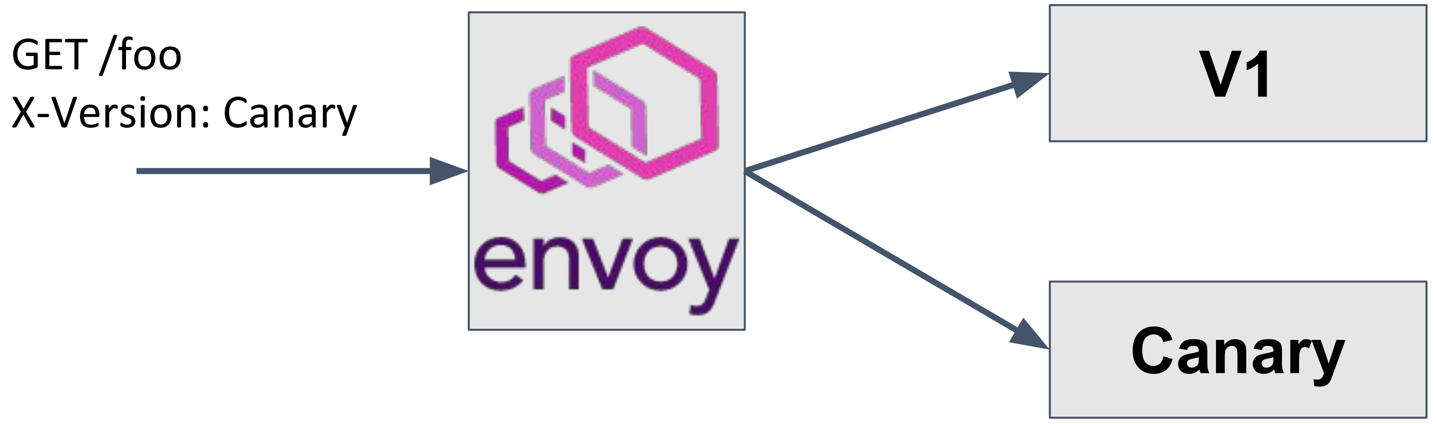


Header-based routing

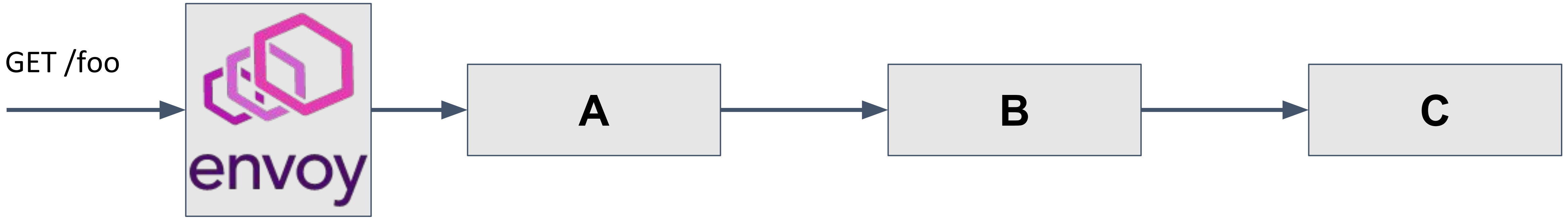
What is Dynamic Request Routing?

Weight-based routing

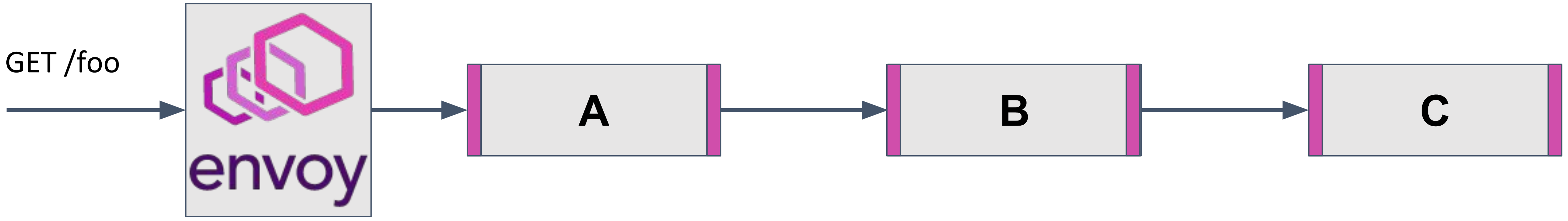
Header-based routing



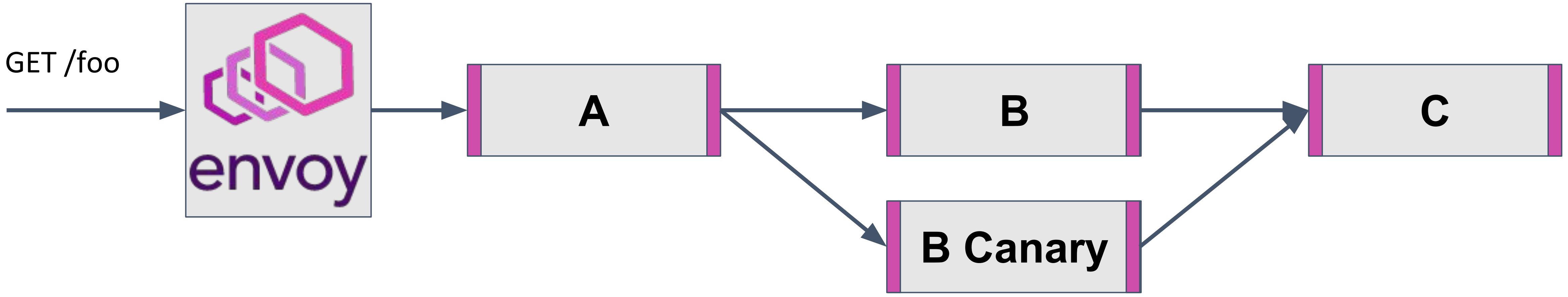
Deeper call graphs



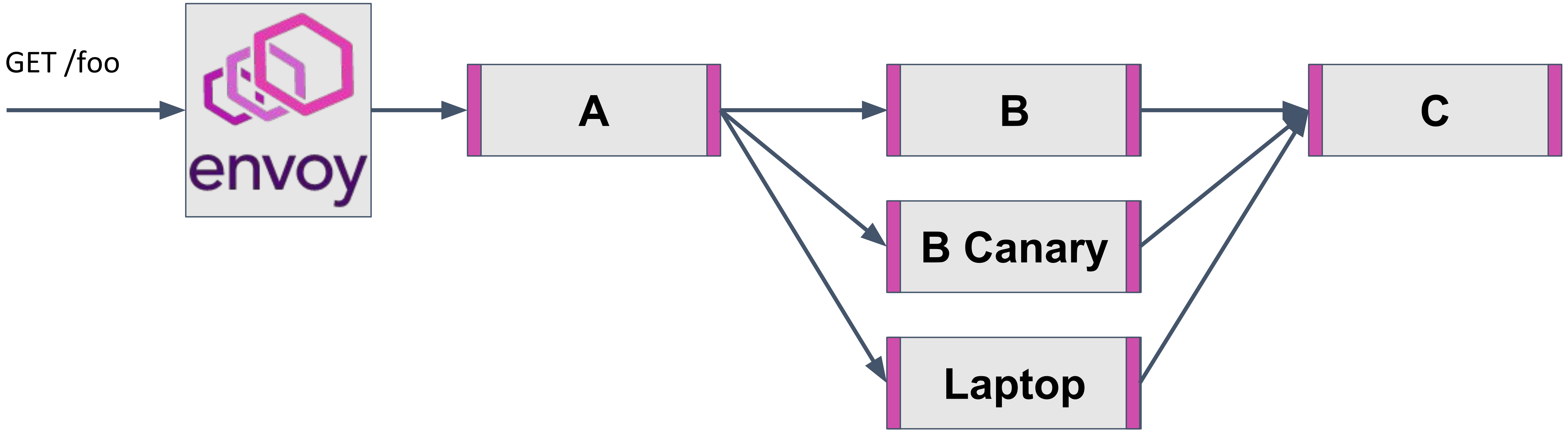
Deeper call graphs



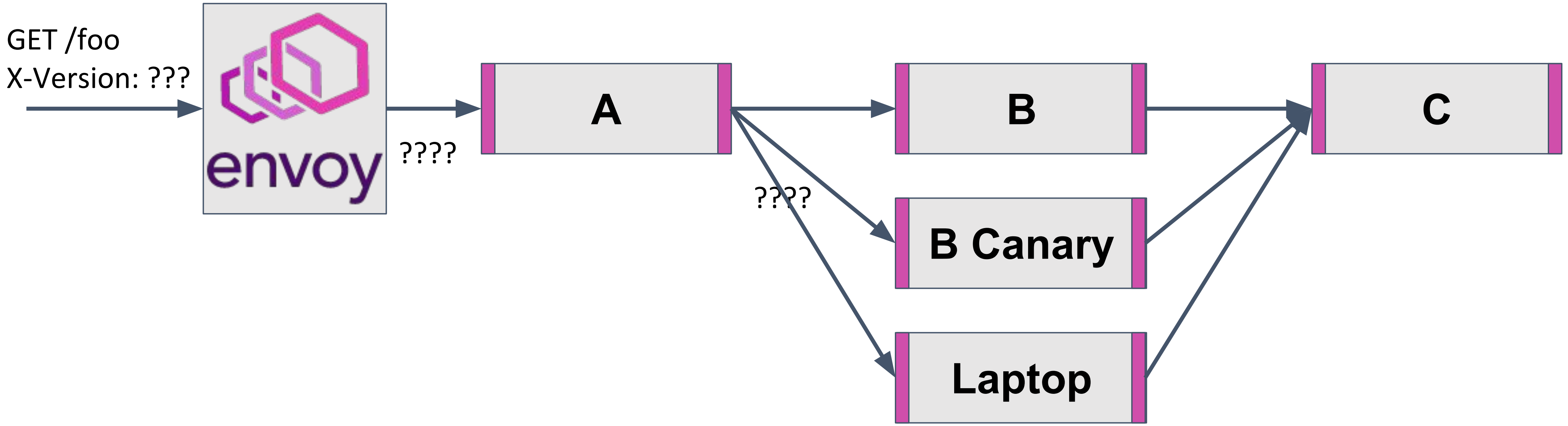
Deeper call graphs



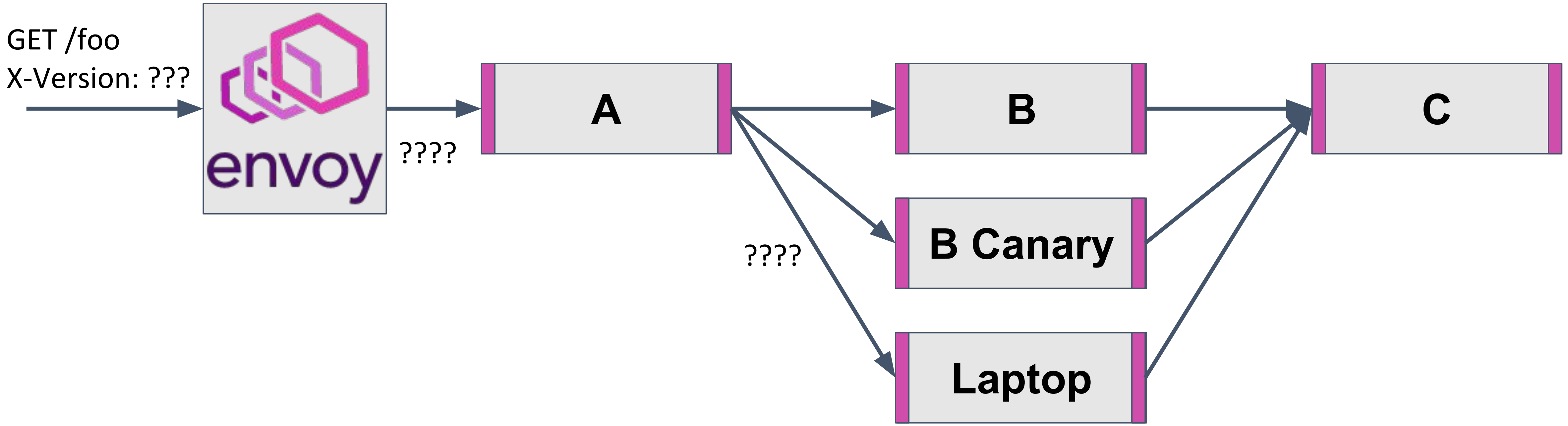
Deeper call graphs



Deeper call graphs



Deeper call graphs



1) How do we “shuffle” data around to make routing decisions?
2) How do we make routing decisions on those data?

Distributed Context Propagation

Used mainly for tracing
- e.g. X-B3-TraceId

“You must propagate tracing headers across your application”

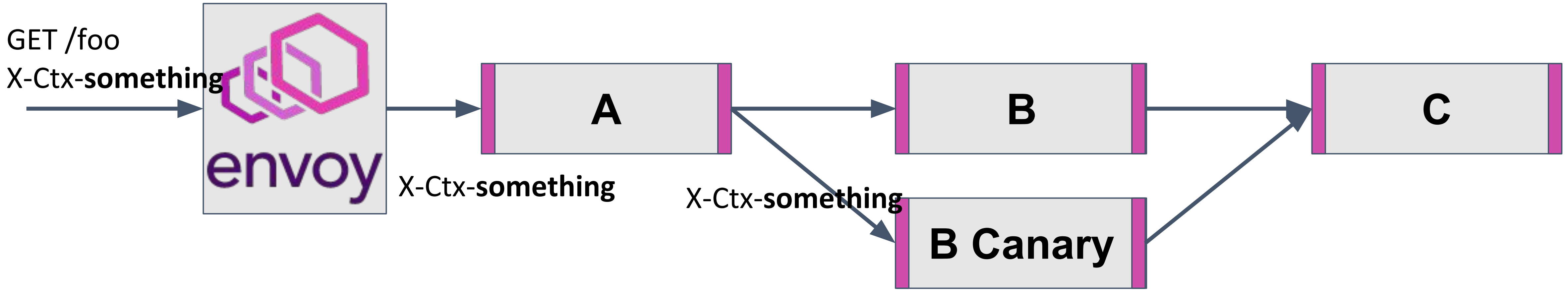
Distributed Context Propagation

envoycon

Requirement 1:
Have mechanism to propagate arbitrary context across processes

e.g. “Any header with X-Ctx-* must be propagated”

Deeper call graphs



✔️ How do we “shuffle” data around to make routing decisions?
❓ How do we make routing decisions on those data?

Header Format

Data will be “instead of routing to X, route to Y”

Something like l5d-dtab

```
X-Ctx-Override: foo=foo.v2;bar=10.11.12.13
```

Dynamic Request Routing Filter

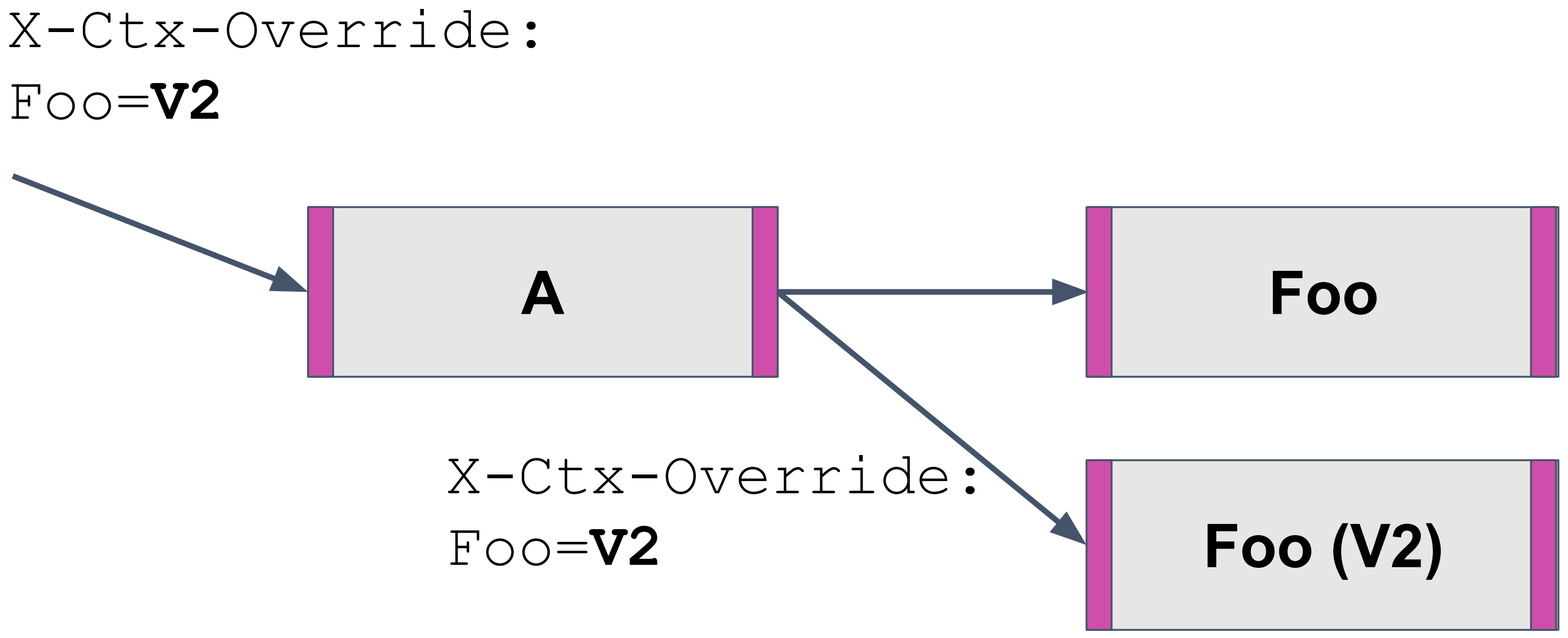
Requirement 2:
Parse X-Ctx-* header and redirect request to desired endpoint

Envoy filter time!

Dynamic Request Routing Filter

```
decodeHeaders()
  Parse header (get dst->new_dst map)
  Determine where we are about to route
  If there's an override for dst:
    overrideRoute(new_dst)
```

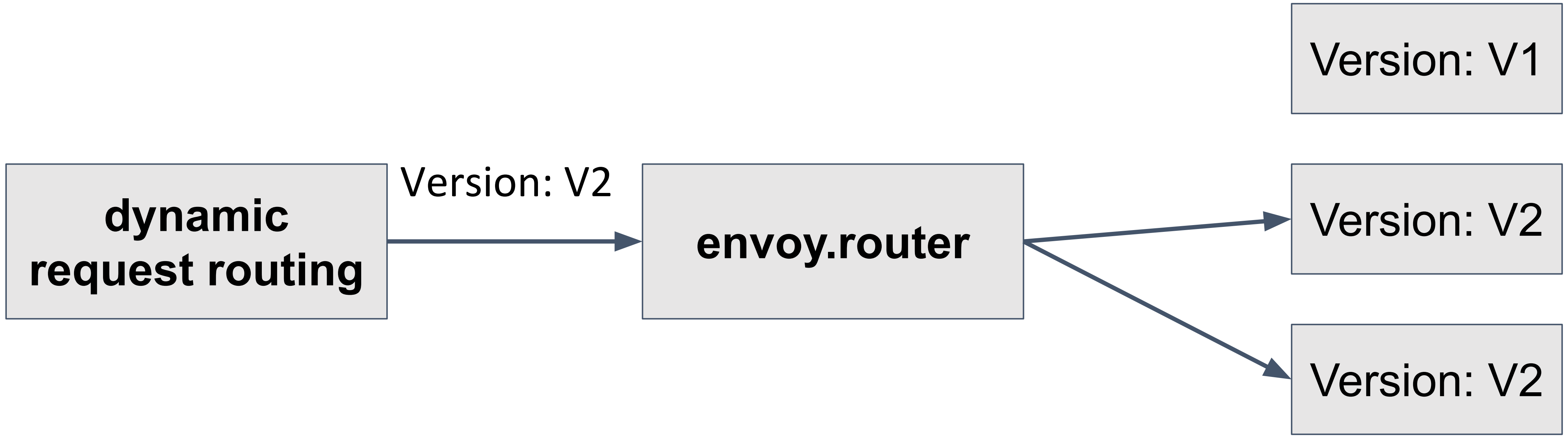
Pre-Existing Endpoint



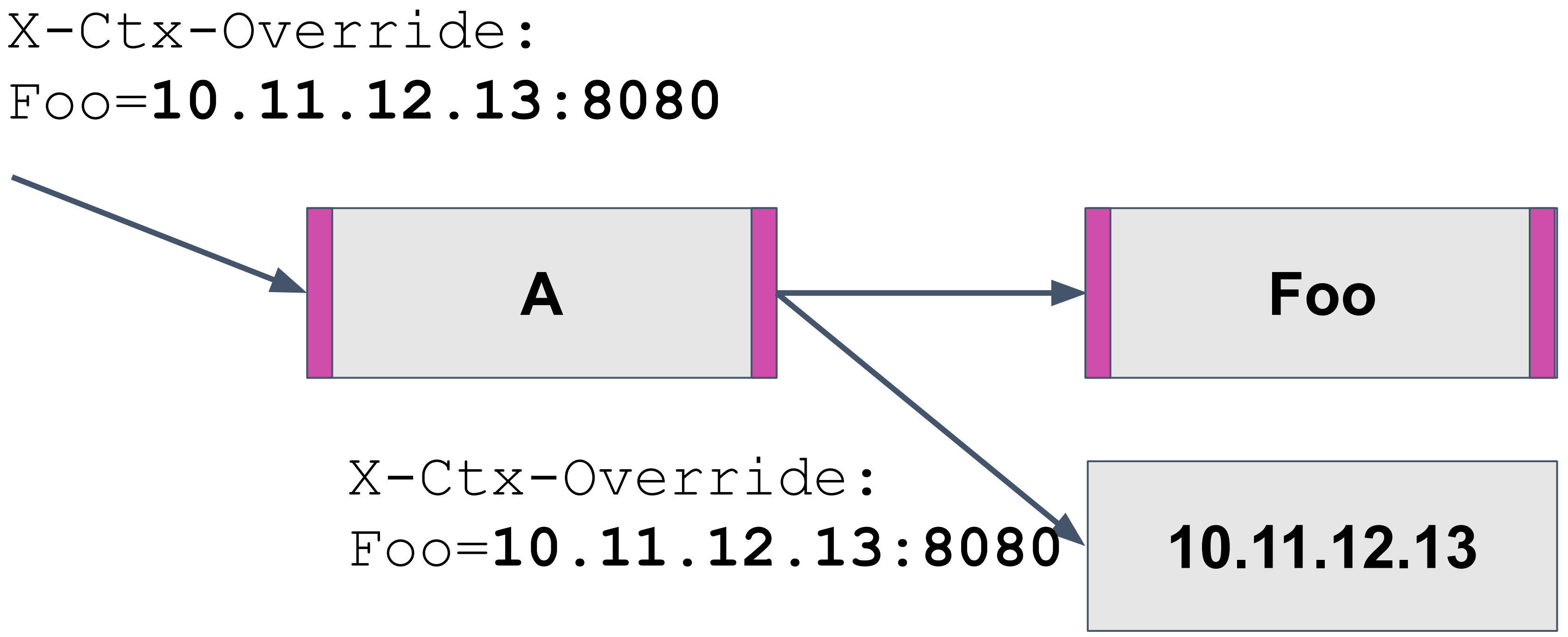
Pre-Existing Endpoint

envoycon

```
overrideRoute(new_dst)
  Inject new_dst as metadata for subsetting
```

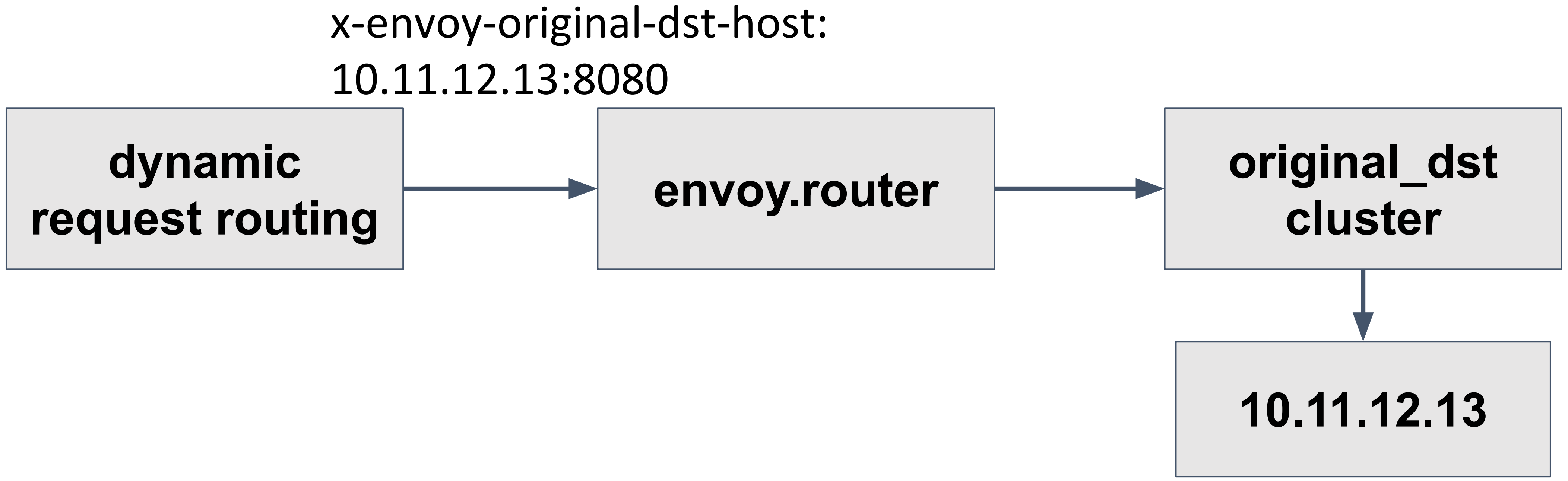


Arbitrary IP

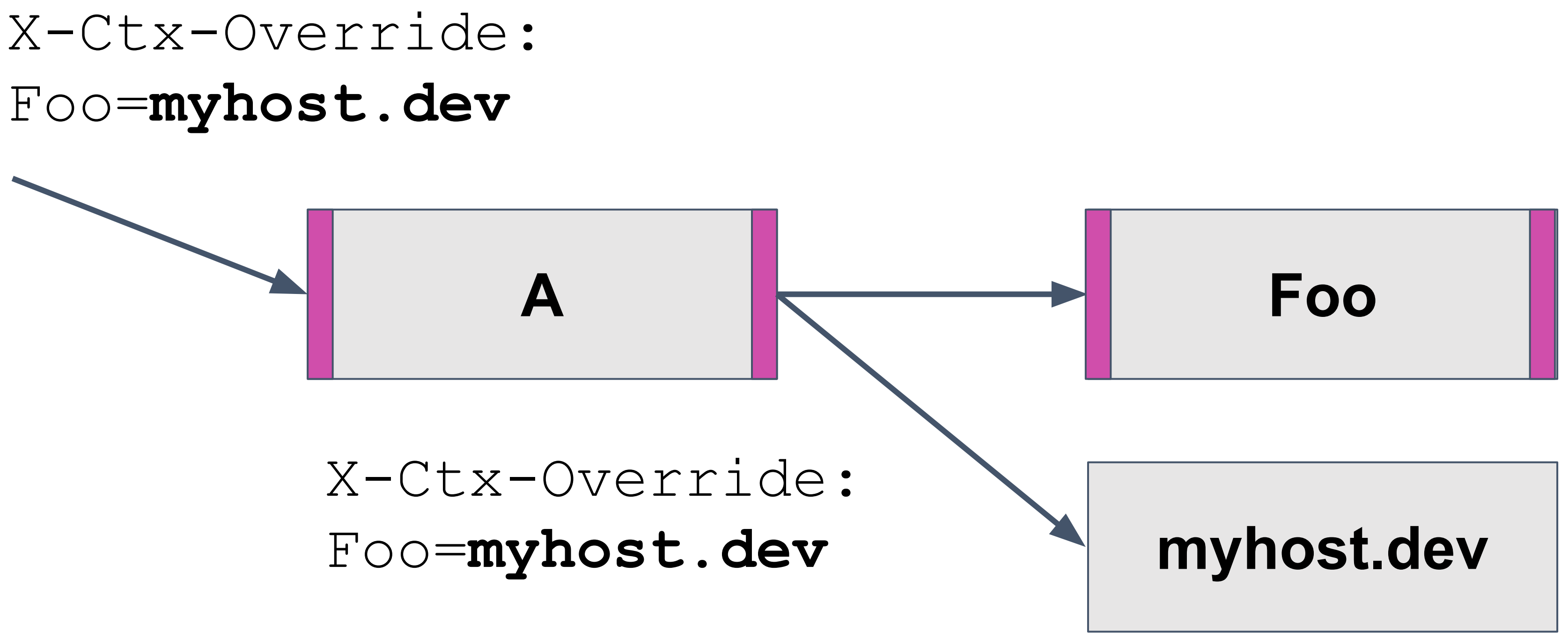


Arbitrary IP

```
overrideRoute(new_dst)
  inject x-envoy-original-dst-host header
```

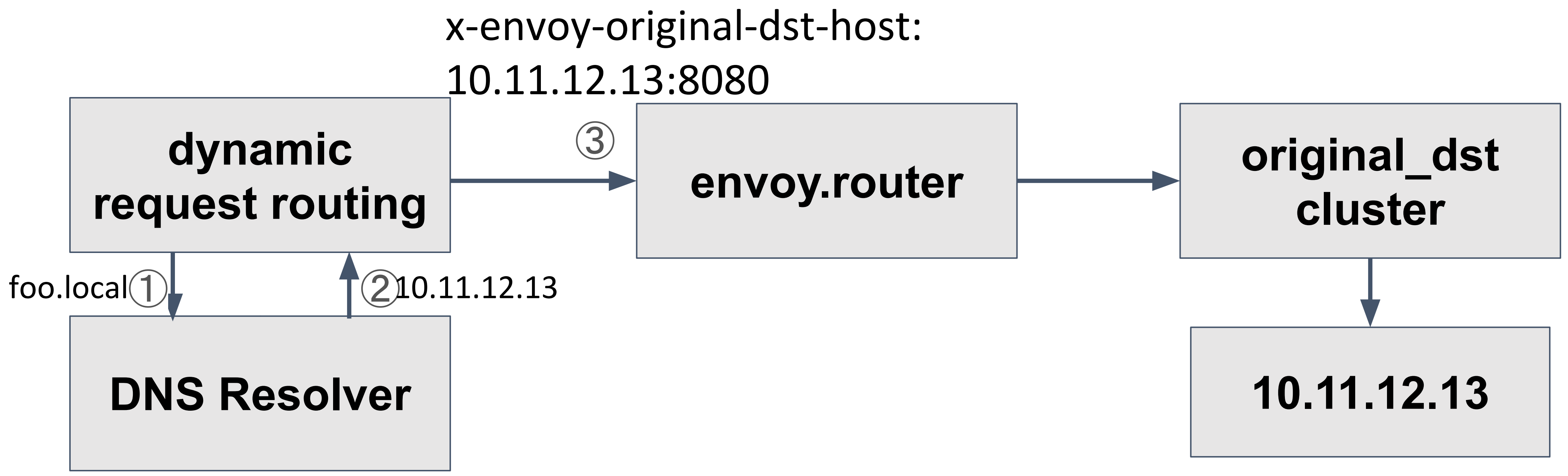


Routing decisions

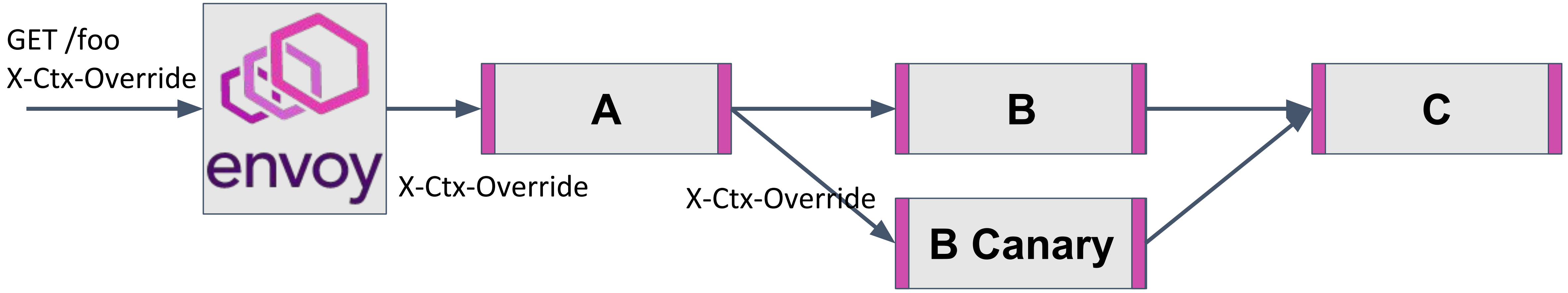


Arbitrary Address

```
overrideRoute(new_dst)
  Do async, thread-local DNS lookup
  Callback adds original dst header
```



Success



✓ How do we “shuffle” data around to make routing decisions?
✓ How do we make routing decisions on those data?

Sugar on Top

Chrome Extension to inject headers

Debug views on App

CLI support (wrapper around curl)

Considerations

Security

- Don’t allow arbitrary routing in prod!

Validation

- Ensure that you are routing to a valid endpoint

Override logic

- Not trivial to figure out where request will be routed

Thanks!

ben.plotnick@getcruise.com
@benplotnick