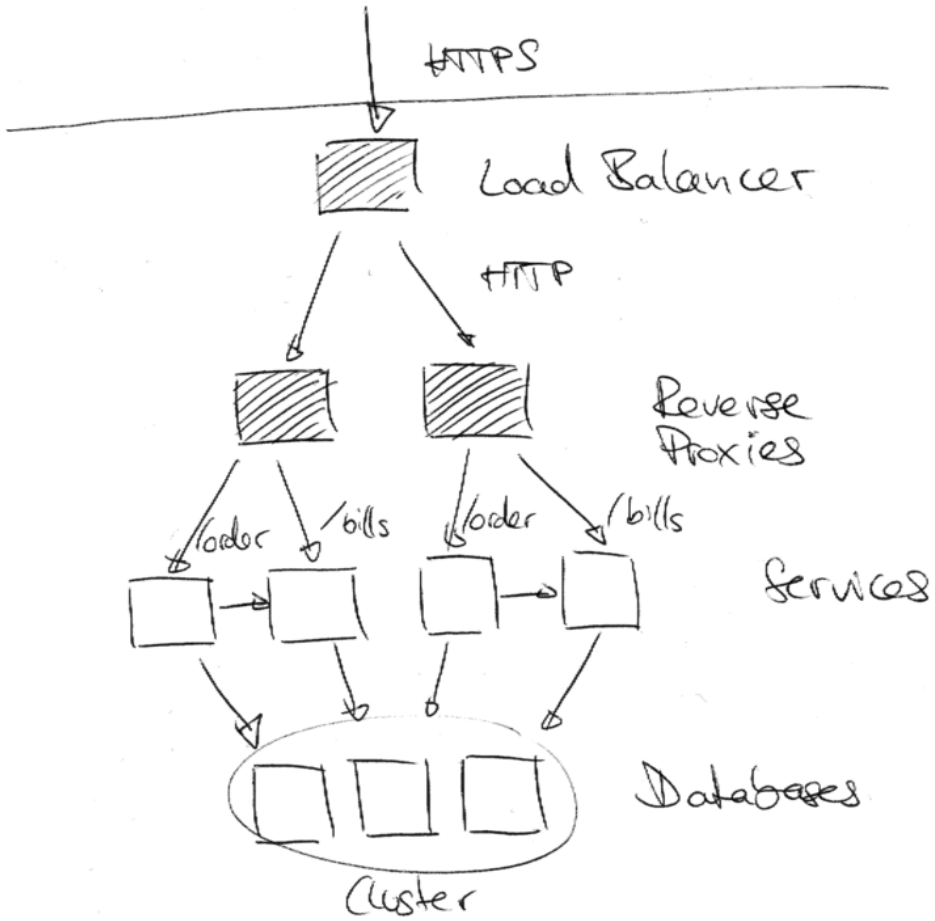
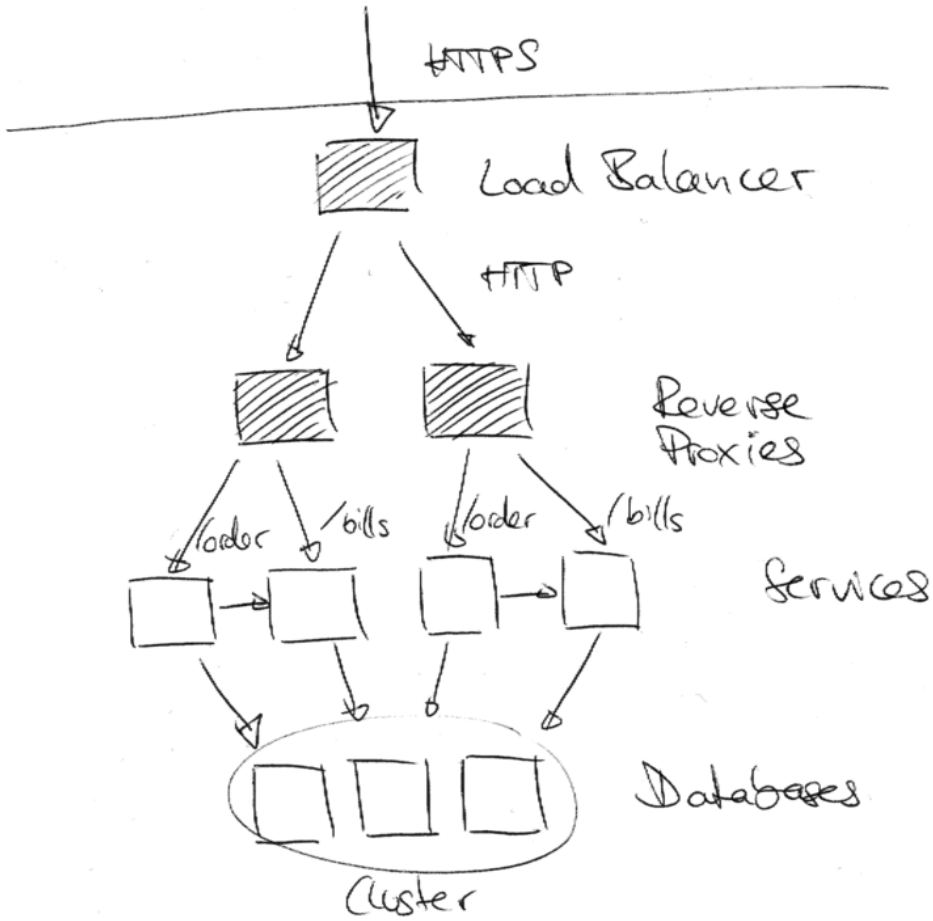
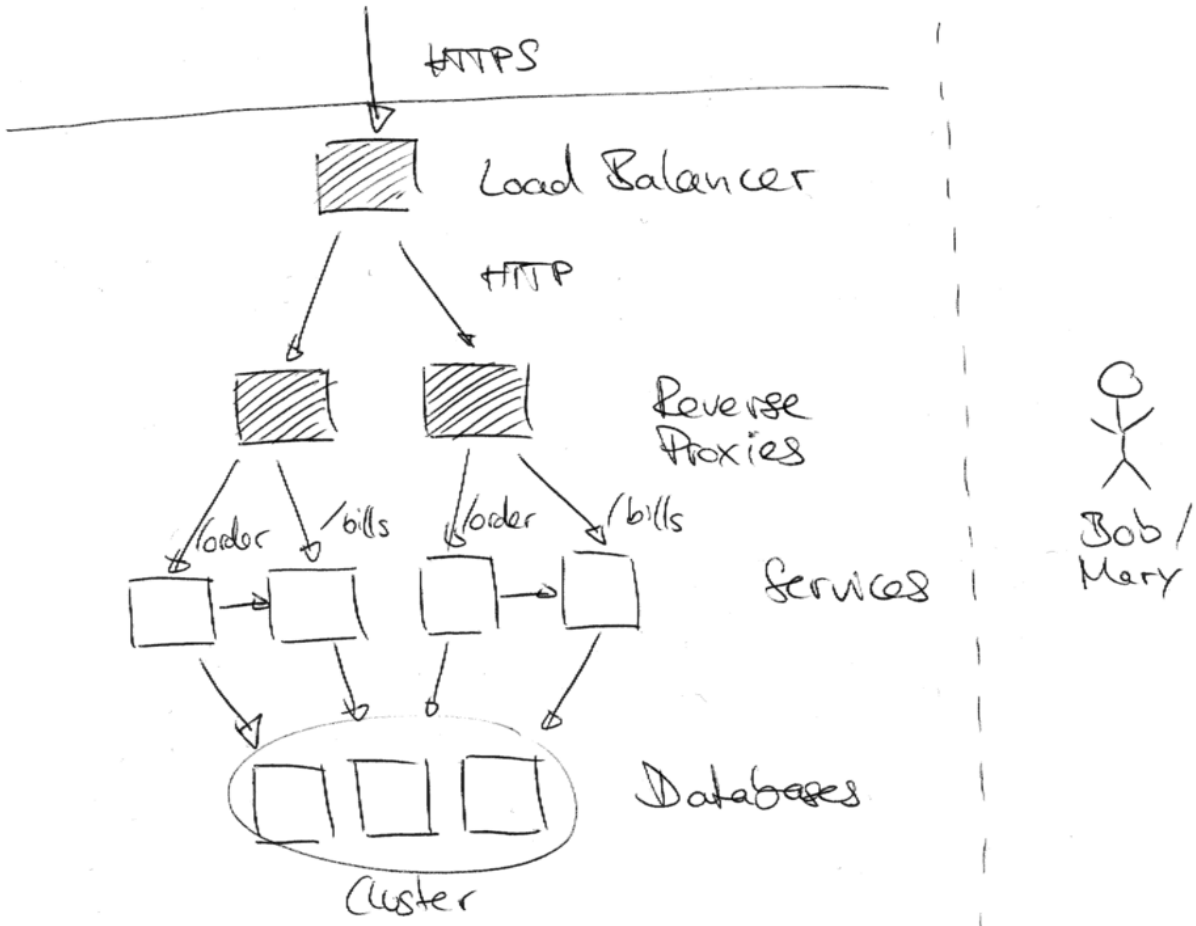
Dynamic Service Meshes for Microservices using Envoy Proxy, Java & Spring

Michael Hartle
(mhartle@hartle-klug.com)

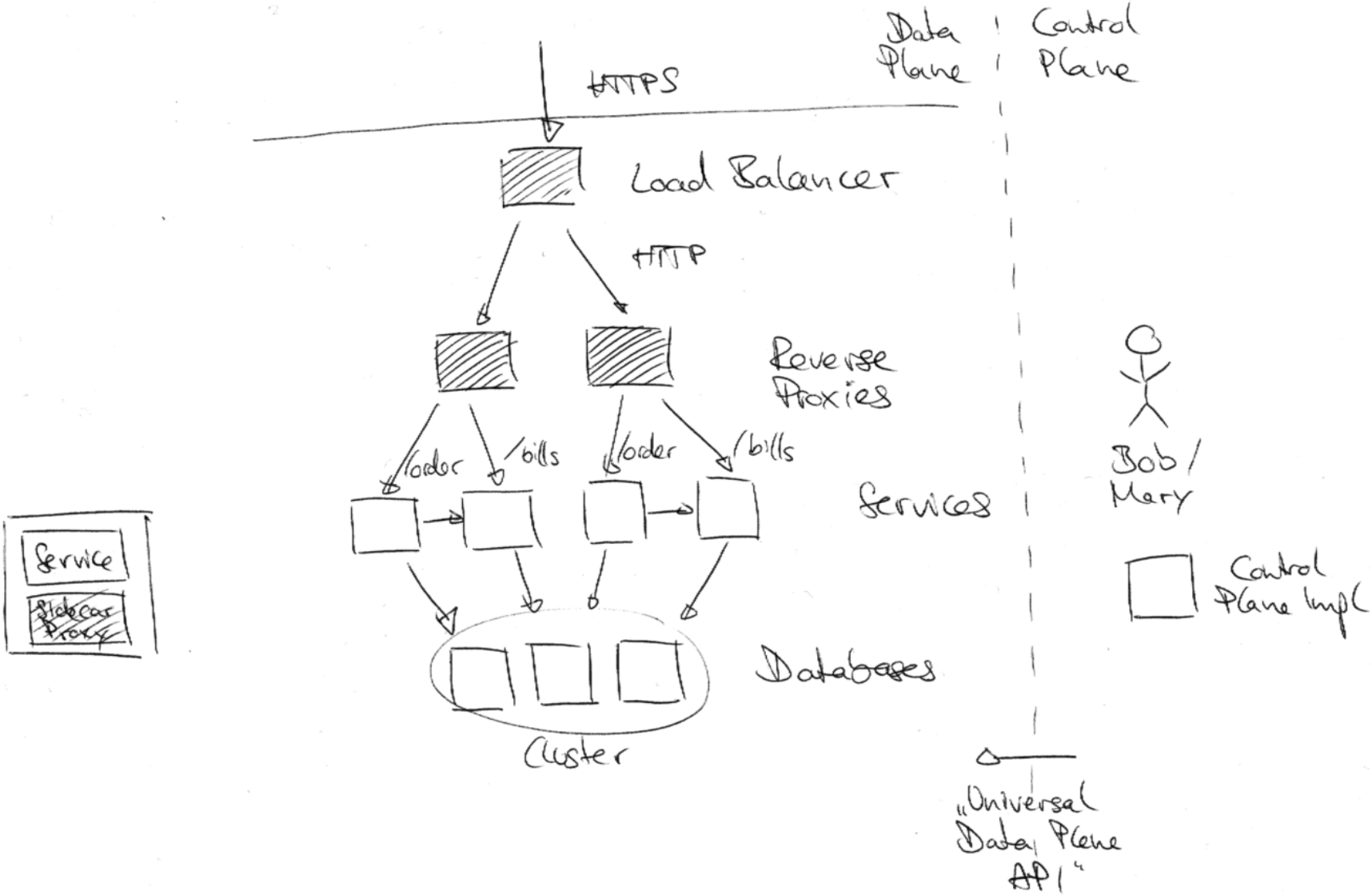

HTTPS
Load Balancer
HTTP
Reverse Proxies
/order
/bills
/order
/bills
Services
Databases
Cluster

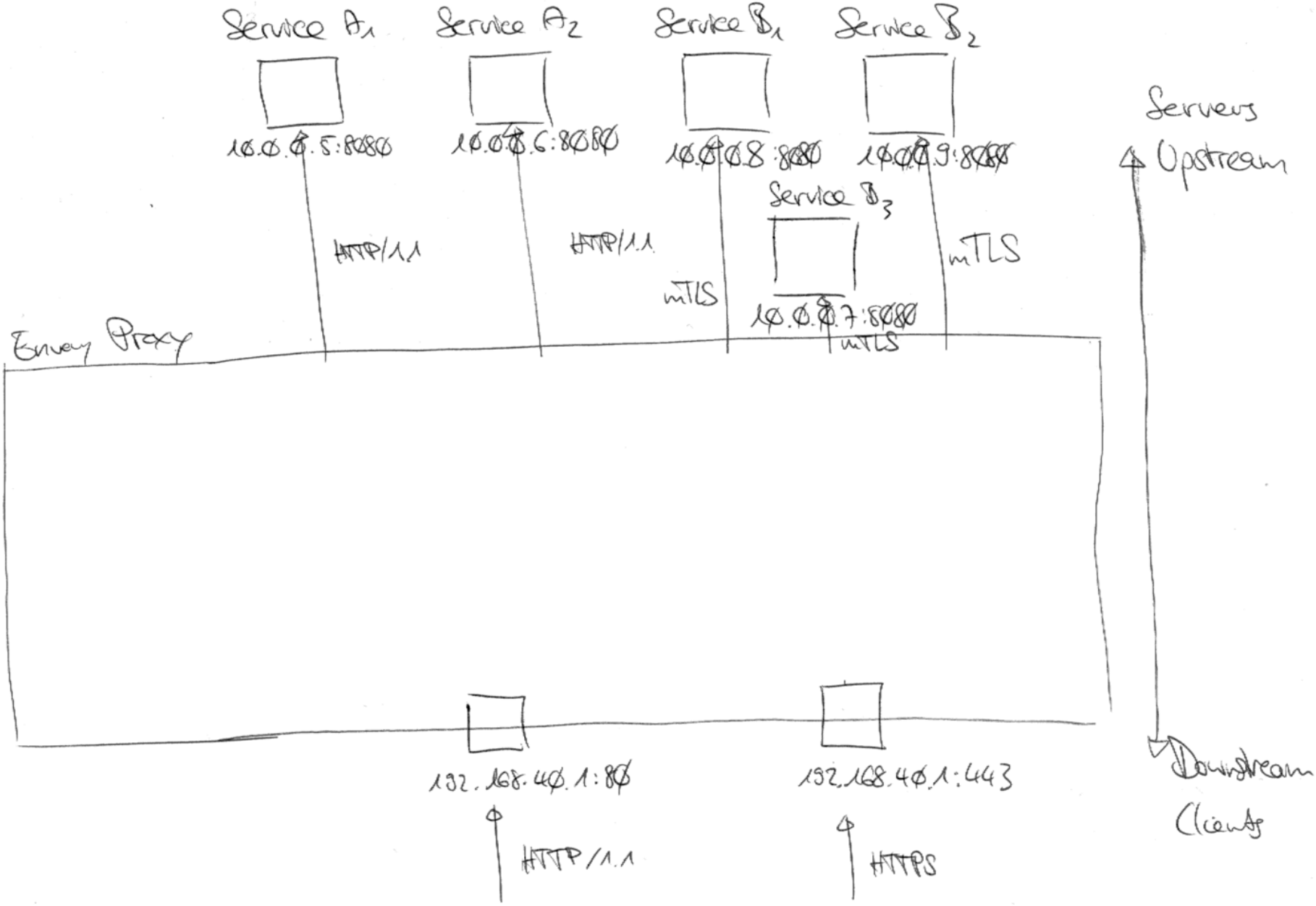

HTTPS
Load Balancer
HTTP
Reverse Proxies
/order
/bills
/order
/bills
Services
Databases
Cluster

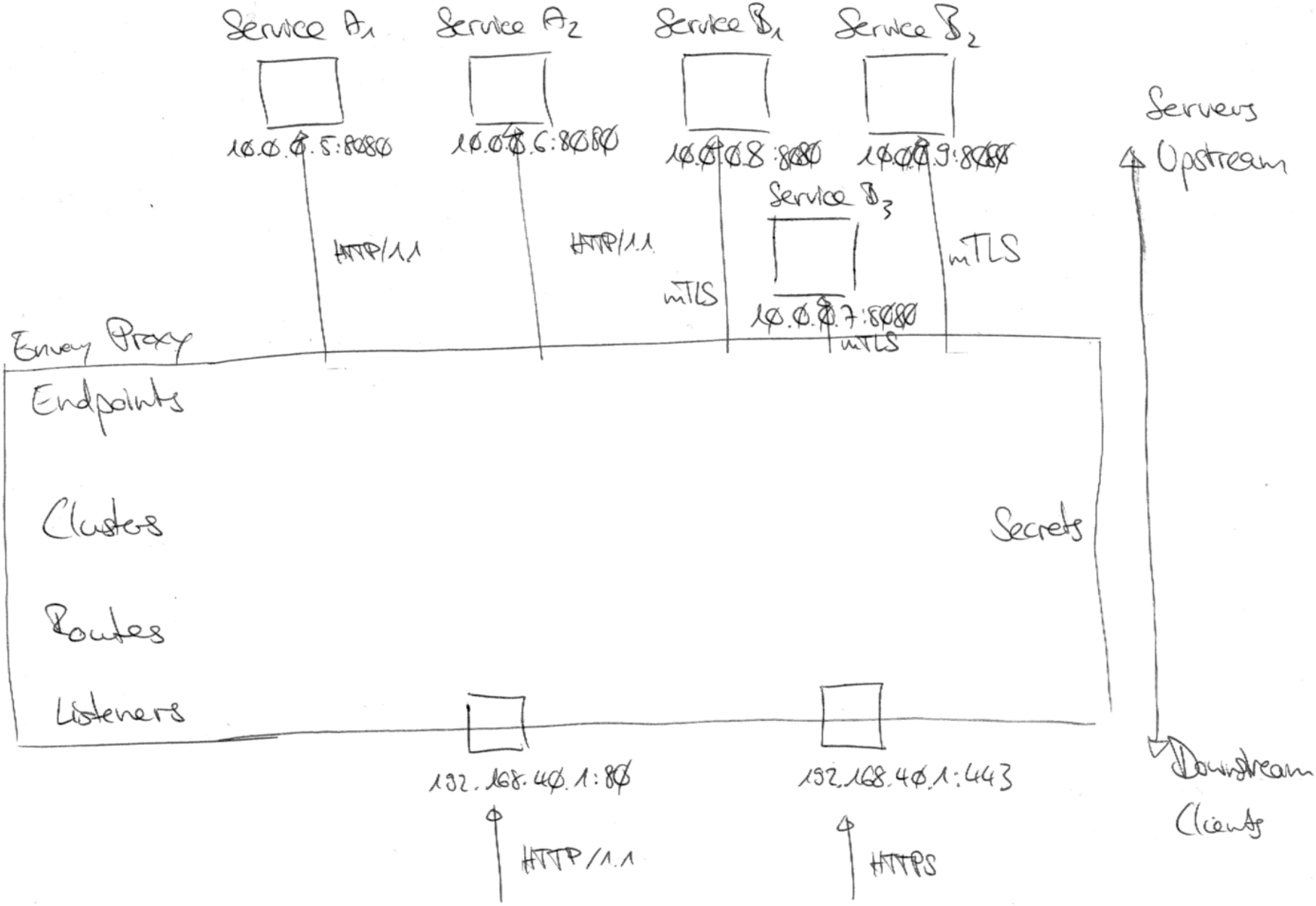
Service
Sidecar Proxy

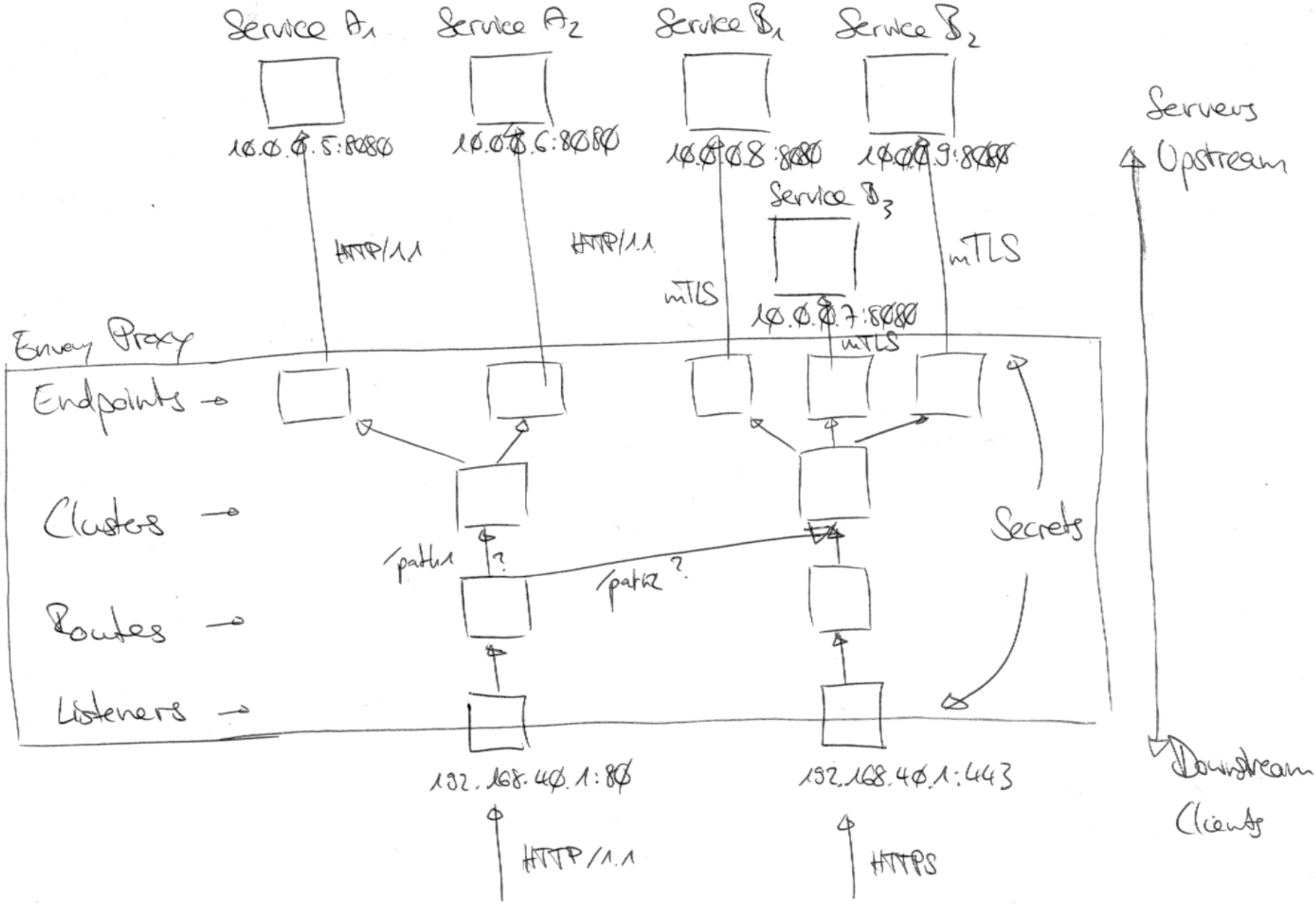

HTTPS
Load Balancer
HTTP
Reverse Proxies
/order
/bills
/order
/bills
Services
Databases
Cluster
Bob / Mary

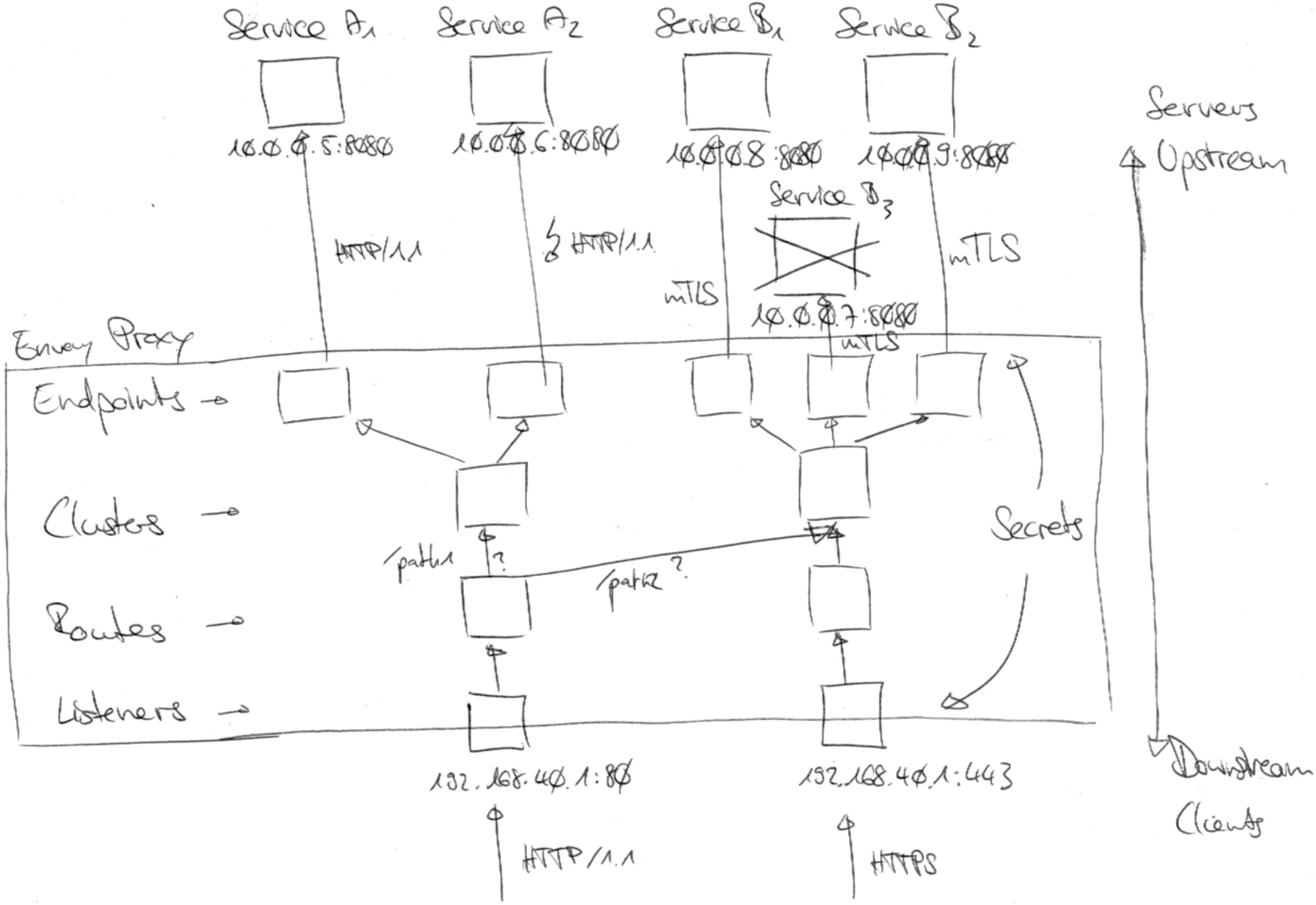
Service
Sidecar Proxy

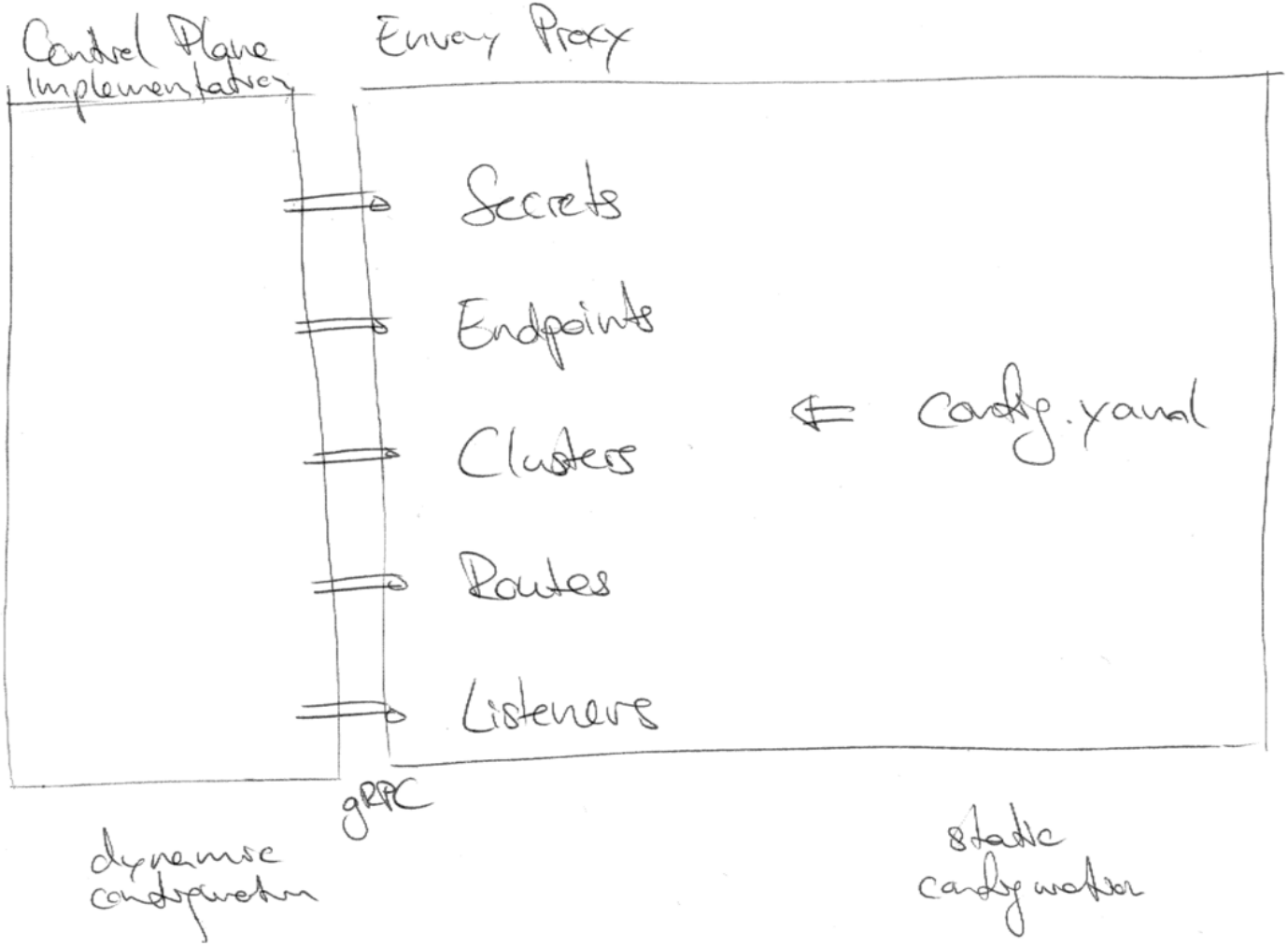

Data Plane
Control Plane
HTTPS
Load Balancer
HTTP
Reverse Proxies
/order
/bills
/order
/bills
Services
Bob / Mary
Service
Sidecar Proxy
Control Plane Impl
Databases
Cluster
„Universal Data Plane API"

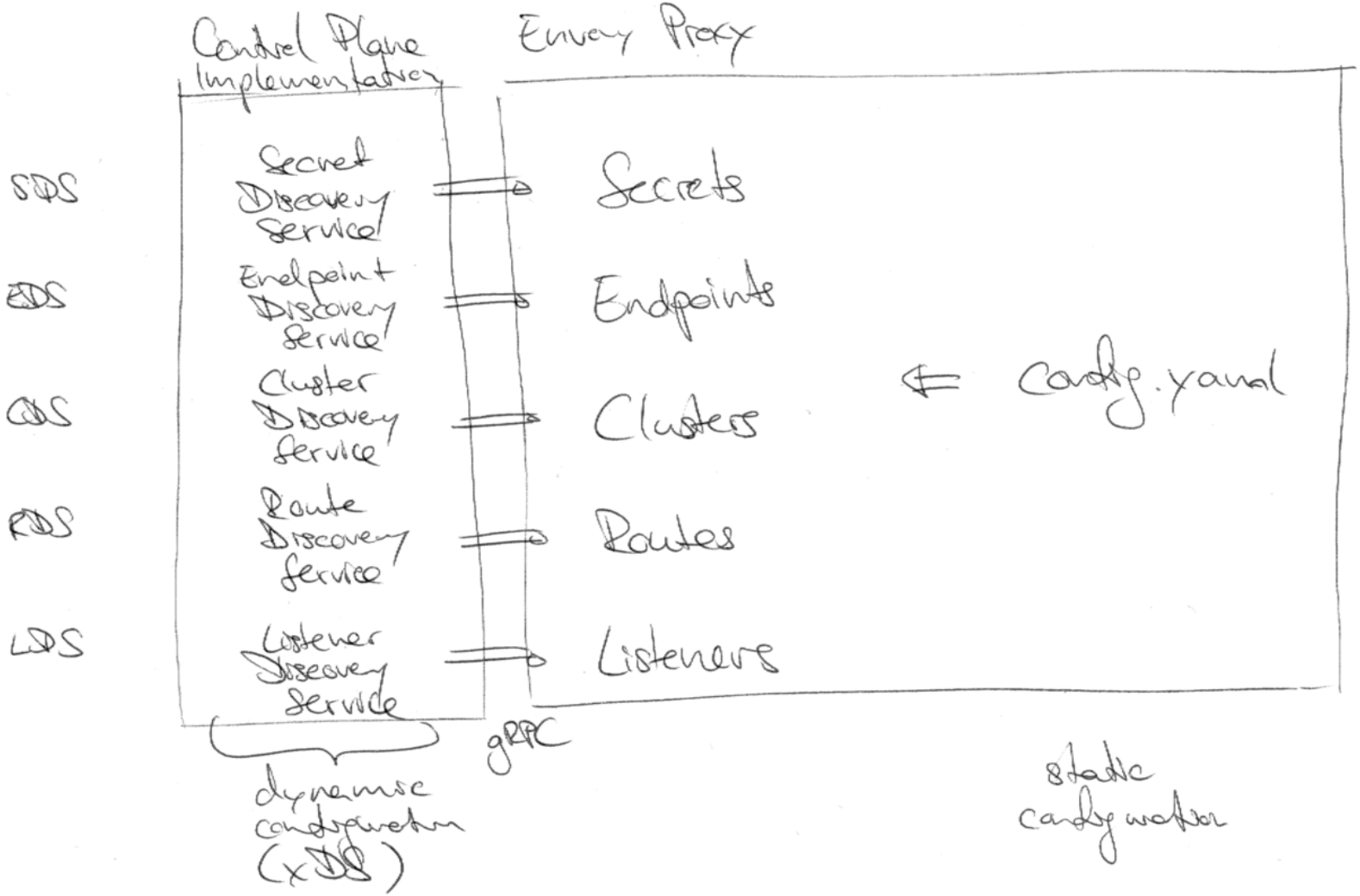

Service A1
Service A2
Service B1
Service B2
Service B3
10.0.0.5:8080
10.0.0.6:8080
10.0.0.8:8080
10.0.0.9:8080
10.0.0.7:8080
HTTP/1.1
HTTP/1.1
mTLS
mTLS
mTLS
Envoy Proxy
192.168.40.1:80
192.168.40.1:443
HTTP/1.1
HTTPS
Servers
Upstream
Downstream
Clients


Service A1
Service A2
Service B1
Service B2
Service B3
10.0.0.5:8080
10.0.0.6:8080
10.0.0.8:8080
10.0.0.9:8080
10.0.0.7:8080
HTTP/1.1
HTTP/1.1
mTLS
mTLS
mTLS
Envoy Proxy
Endpoints
Clusters
Routes
Listeners
Secrets
192.168.40.1:80
192.168.40.1:443
HTTP/1.1
HTTPS
Servers
Upstream
Downstream
Clients


Service A1
Service A2
Service B1
Service B2
Service B3
10.0.0.5:8080
10.0.0.6:8080
10.0.0.8:8080
10.0.0.9:8080
10.0.0.7:8080
HTTP/1.1
HTTP/1.1
mTLS
mTLS
mTLS
Envoy Proxy
Endpoints →
Clusters →
Routes →
Listeners →
/path1 ?
/path2 ?
Secrets
192.168.40.1:80
192.168.40.1:443
HTTP/1.1
HTTPS
Servers
Upstream
Downstream
Clients


Service A1
Service A2
Service B1
Service B2
Service B3
10.0.0.5:8080
10.0.0.6:8080
10.0.0.8:8080
10.0.0.9:8080
10.0.0.7:8080
HTTP/1.1
HTTP/1.1
mTLS
mTLS
mTLS
Envoy Proxy
Endpoints →
Clusters →
Routes →
Listeners →
/path1 ?
/path2 ?
Secrets
192.168.40.1:80
192.168.40.1:443
HTTP/1.1
HTTPS
Servers
Upstream
Downstream
Clients


Control Plane
Implementation
Envoy Proxy
Secrets
Endpoints
Clusters
Routes
Listeners
⇐ config.yaml
gRPC
dynamic
configuration
static
config watcher


Control Plane Implementation
Envoy Proxy
SDS
EDS
CDS
RDS
LDS
Secret Discovery Service
Endpoint Discovery Service
Cluster Discovery Service
Route Discovery Service
Listener Discovery Service
Secrets
Endpoints
Clusters
Routes
Listeners
⇐ config.yaml
gRPC
dynamic configuration (xDS)
static config watcher

Echo Service

Spring Webflux
5.1.10.RELEASE
Spring Boot
2.1.9.RELEASE

Control Plane Impl

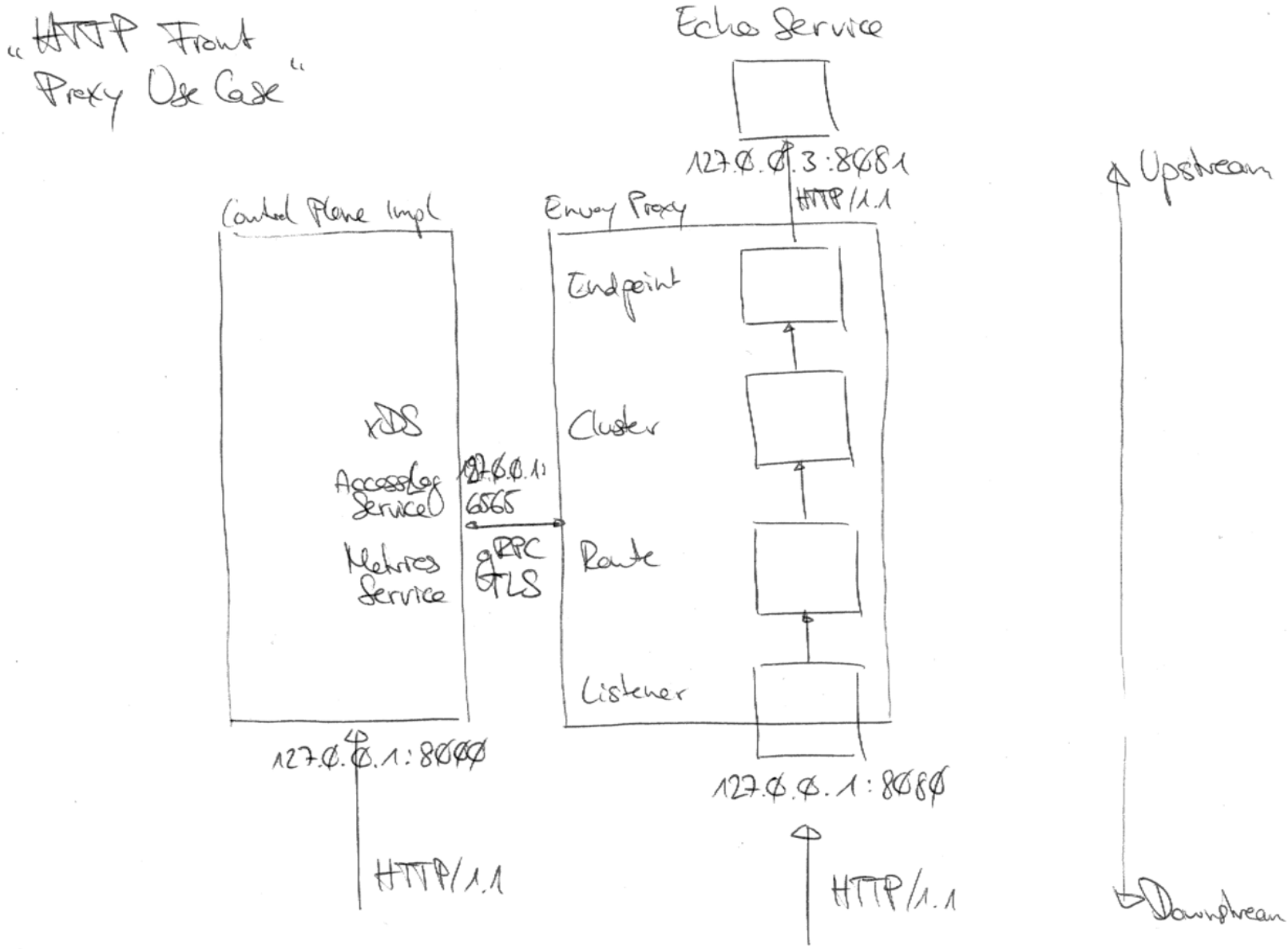
Bouncy Castle
1.63

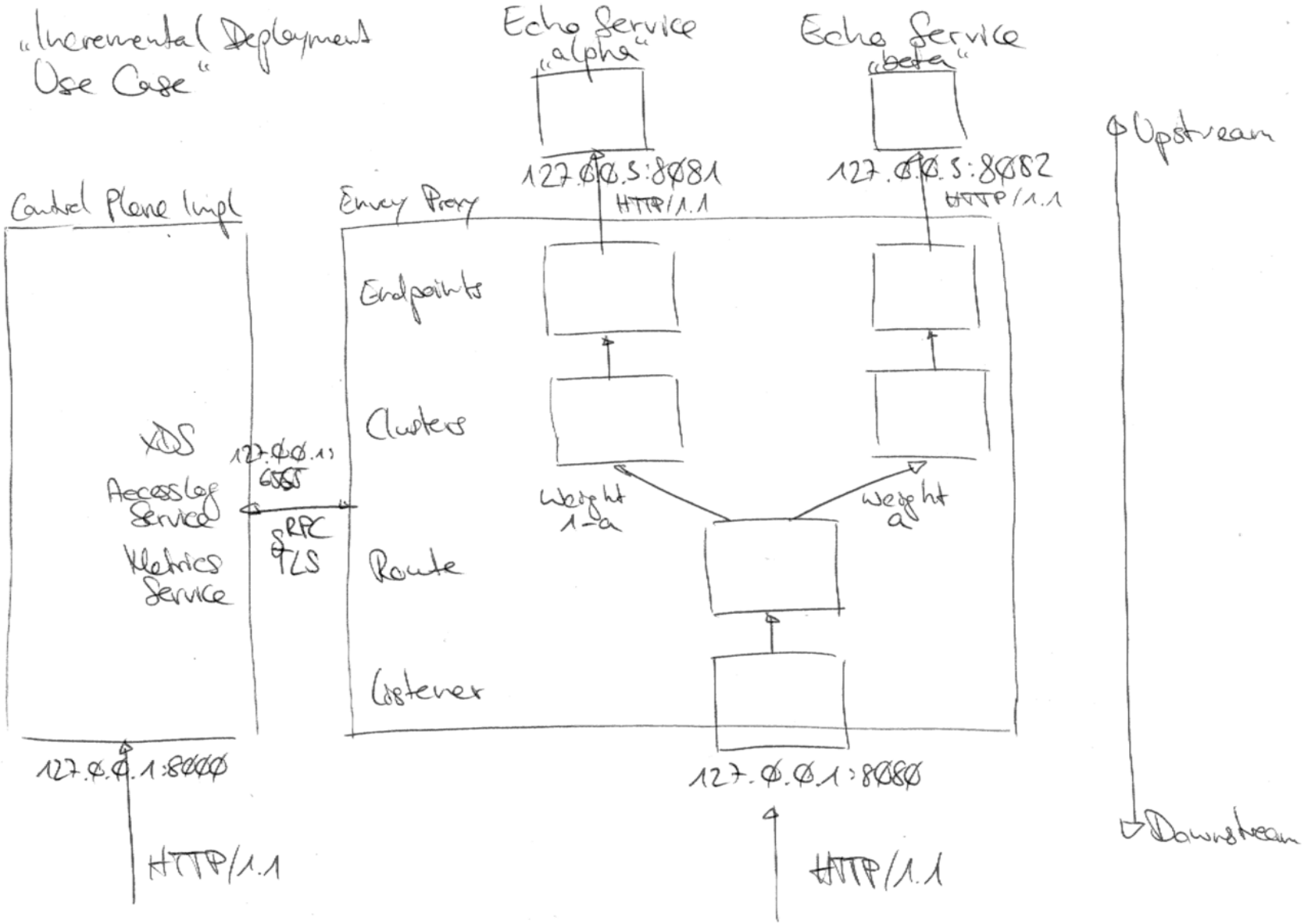
java-control-plane
0.1.19

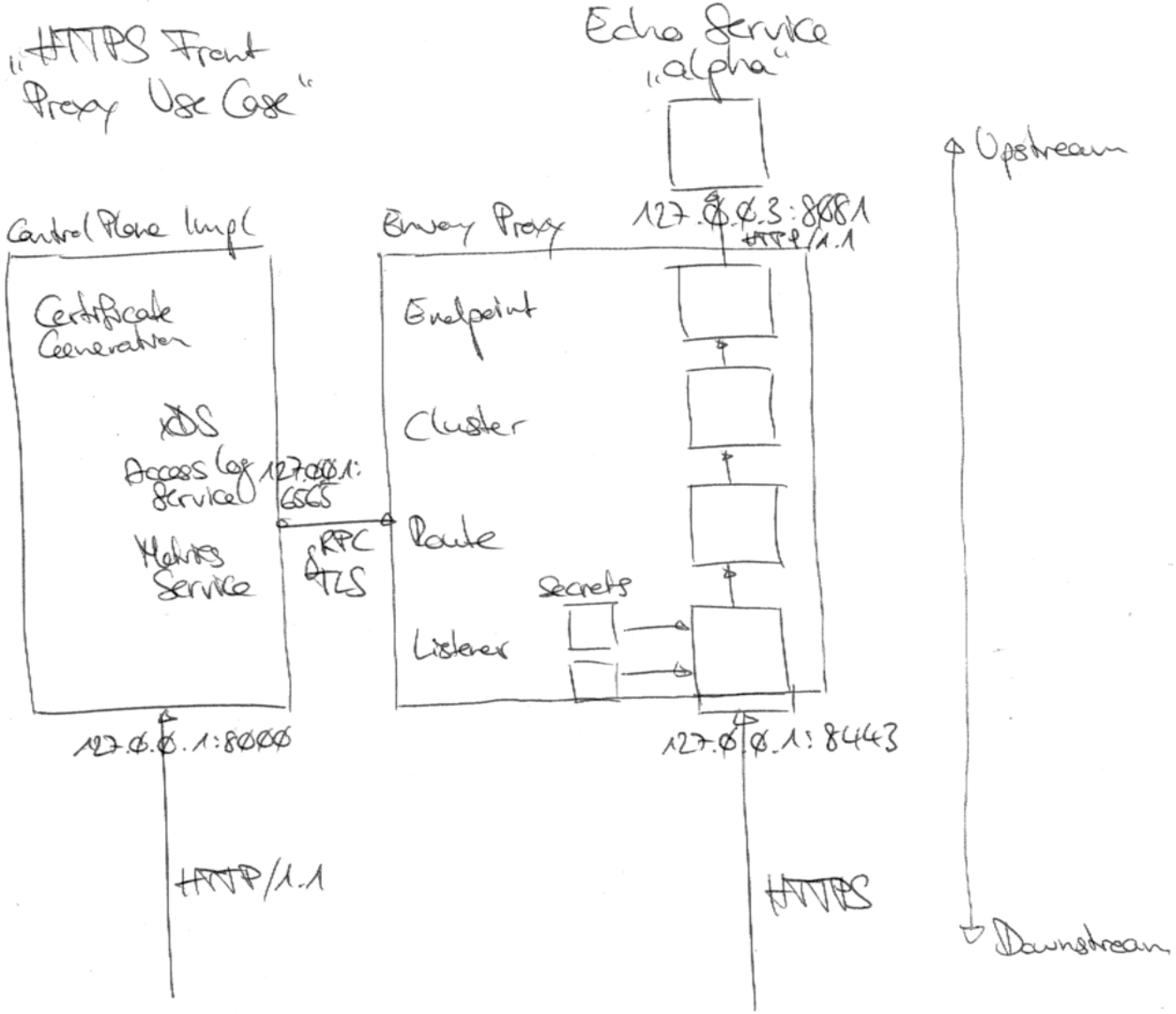
Spring Webflux
5.1.10.RELEASE
Spring Boot
2.1.8.RELEASE

Envoy Proxy 1.12.0

Java 8


"HTTP Front Proxy Use Case"
Echo Service
127.0.0.3:8081
HTTP/1.1
Upstream
Control Plane Impl
Envoy Proxy
Endpoint
xDS
Cluster
Access Log Service
127.0.0.1:6565
Metrics Service
gRPC mTLS
Route
Listener
127.0.0.1:8000
127.0.0.1:8080
HTTP/1.1
HTTP/1.1
Downstream


"Incremental Deployment Use Case"
Echo Service "alpha"
Echo Service "beta"
Upstream
127.0.0.1:8081
HTTP/1.1
127.0.0.1:8082
HTTP/1.1
Control Plane Impl
Envoy Proxy
Endpoints
Clusters
Route
Listener
xDS
Access Log Service
Metrics Service
127.0.0.1: 6565
gRPC + TLS
weight 1-a
weight a
127.0.0.1:8000
127.0.0.1:8080
HTTP/1.1
HTTP/1.1
Downstream


"HTTPS Front Proxy Use Case"
Echo Service "alpha"
Upstream
Control Plane Impl
Certificate Generation
xDS
Access Log Service
Health Service
127.0.0.1: 6565
RPC & TLS
Envoy Proxy
Endpoint
Cluster
Route
Secrets
Listener
127.0.0.3: 8081
HTTP/1.1
127.0.0.1: 8000
127.0.0.1: 8443
HTTP/1.1
HTTPS
Downstream

Roll out a service mesh?

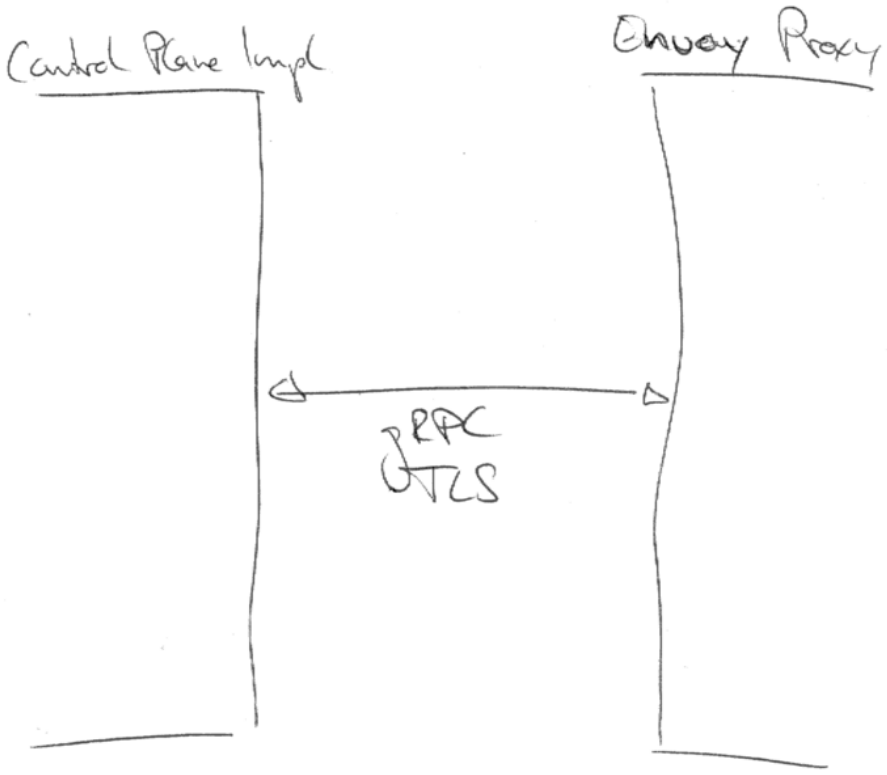
Write your own control plane?

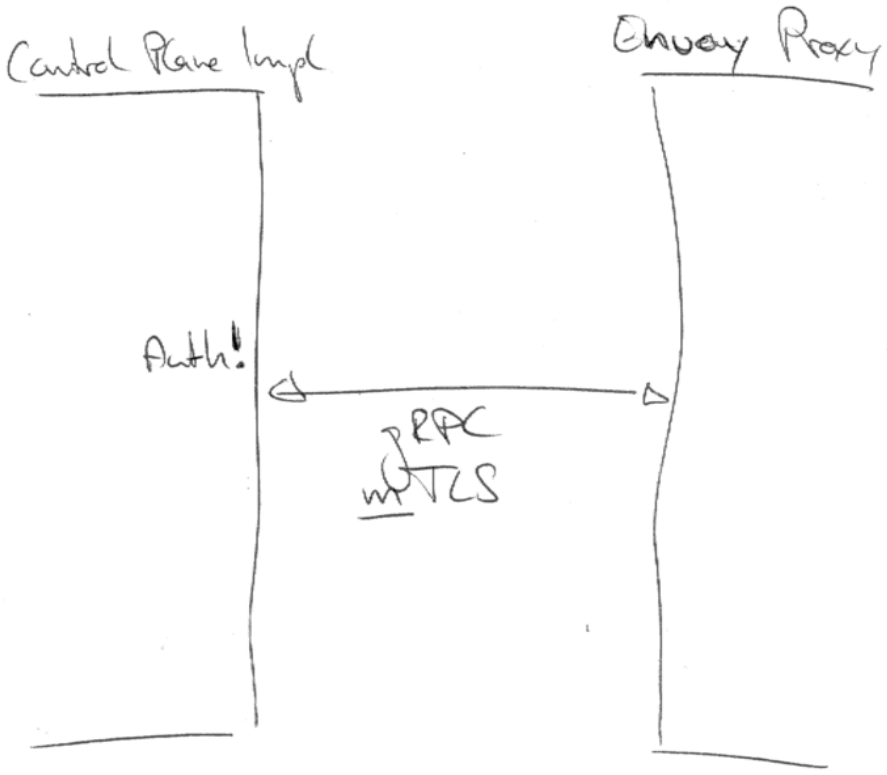
Buy a control plane impl?

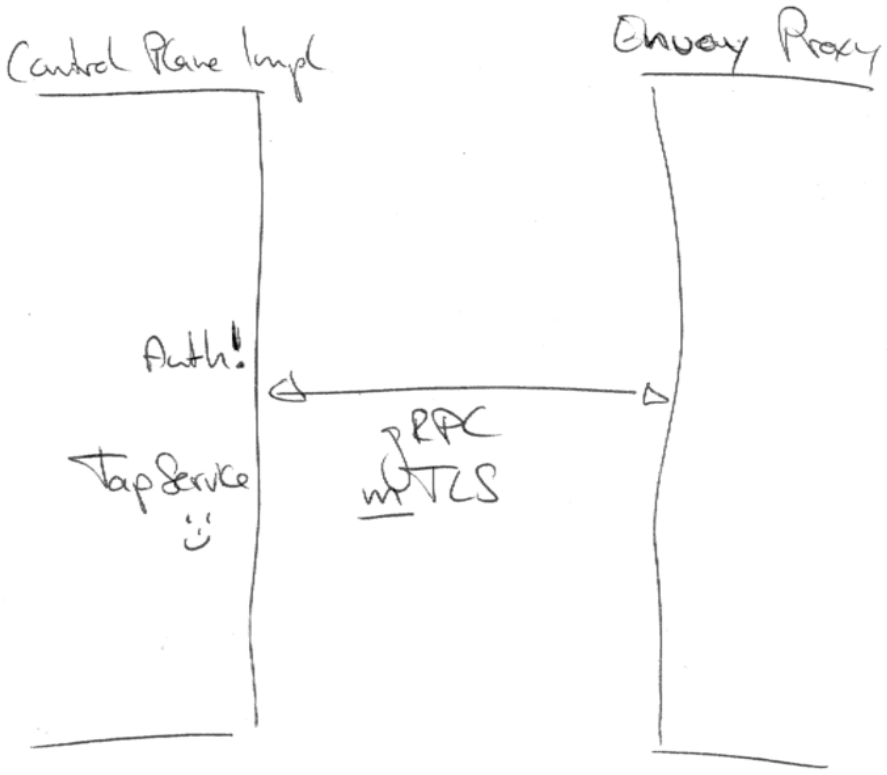
Roll out a service mesh?

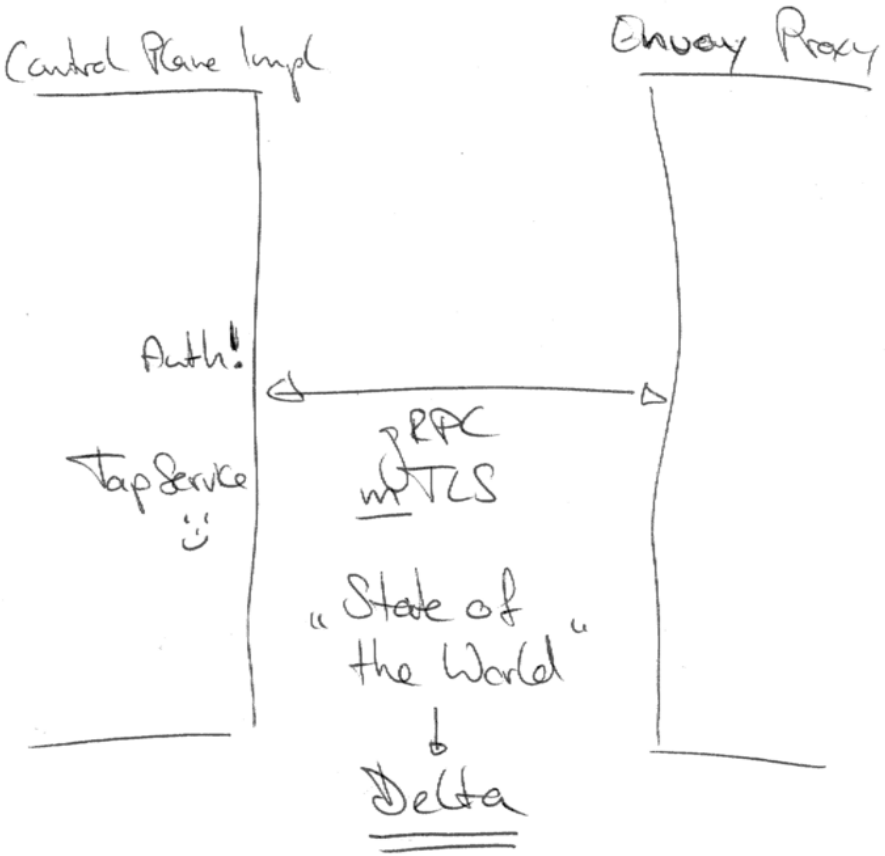
Write your own control plane? ⇒ "It depends..."

Buy a control plane impl?


Control Plane Impl
Envoy Proxy
RPC
mTLS


Control Plane Impl
Envoy Proxy
Auth!
RPC
w/ TLS


Control Plane Impl
Envoy Proxy
Auth!
TapService
:)
RPC
w/ TLS


Control Plane Impl
Envoy Proxy
Auth!
TapService
:)
RPC
w/ TLS
"State of the World"
Delta

Sources / Suggested Reading

Sources / Suggested Reading

- Matt Klein, "Introduction to modern network load balancing and proxying" (2017)
- Matt Klein, "Service mesh data plane vs. control plane" (2017)

Sources / Suggested Reading

- Matt Klein, "Introduction to modern network load balancing and proxying" (2017)
- Matt Klein, "Service mesh data plane vs. control plane" (2017)
- https://www.envoyproxy.io

Sources / Suggested Reading

- Matt Klein, "Introduction to modern network load balancing and proxying" (2017)
- Matt Klein, "Service mesh data plane vs. control plane" (2017)
- https://www.envoyproxy.io
- https://github.com/envoyproxy/java-control-plane

„Where to find the sources?"

https://github.com/mhartle/oreilly-sacon2019-envoy-proxy

Questions?

Thank you
for your attention!

Ah, I almost forgot...

...please rate this session on the conference website („Rate This Session" button) or in the O'Reilly Events App („Session Evaluation" button) :)