LINKERD

MULTICLUSTER DEEP DIVE

ZAHARI DICHEV

AGENDA

- Service Mesh Overview
- Multicluster Concepts
- Architecture
- Demo
- The Life Of a Request Across Clusters
- Q&A

AGENDA

- Service Mesh Overview
- Multicluster Concepts
- Architecture
- Demo
- The Life Of a Request Across Clusters
- Q&A

SERVICE MESH OVERVIEW

“A and B can exchange packets”

“A and B can exchange packets”

VS

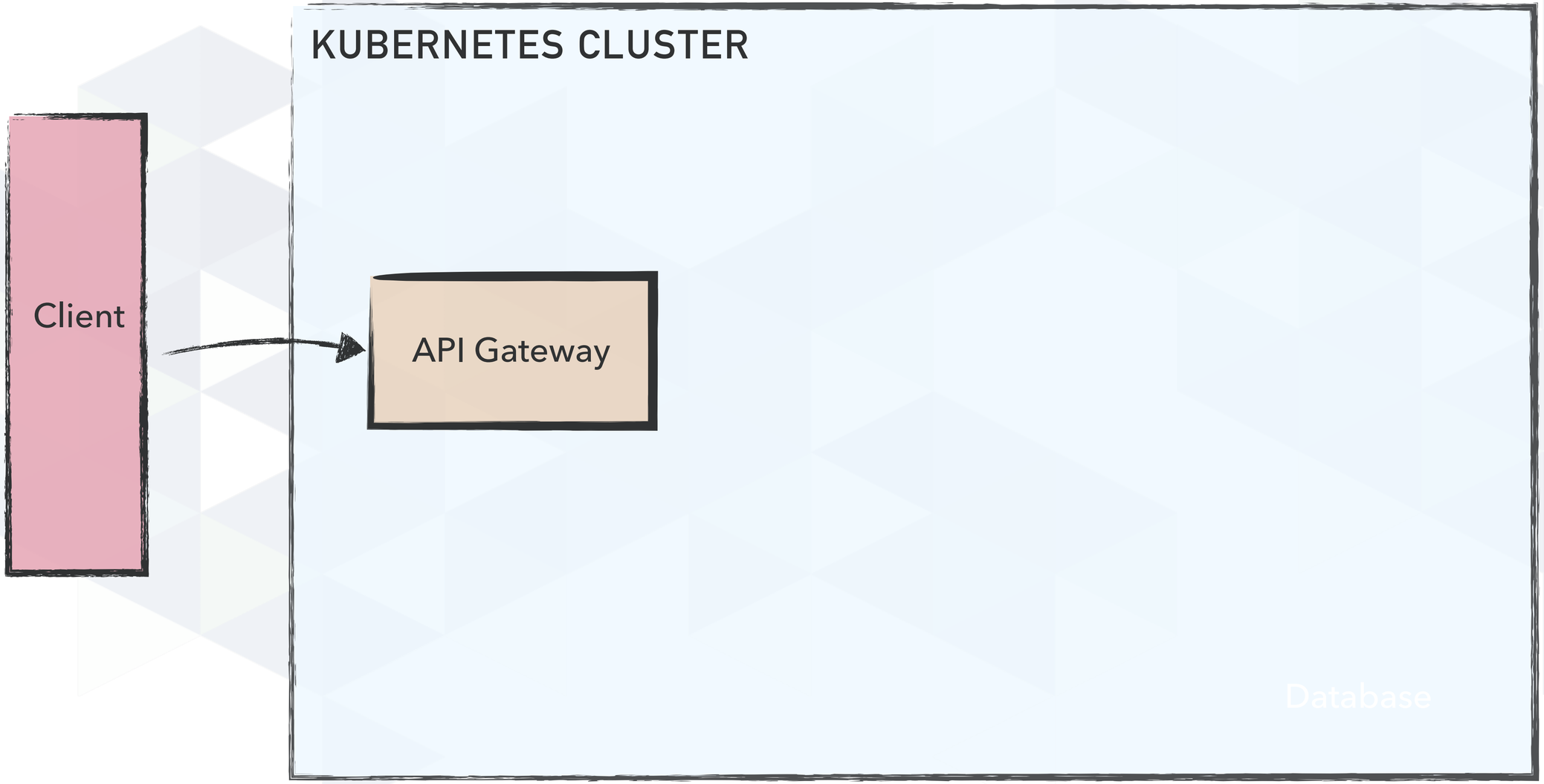
“A and B can exchange packets”

VS

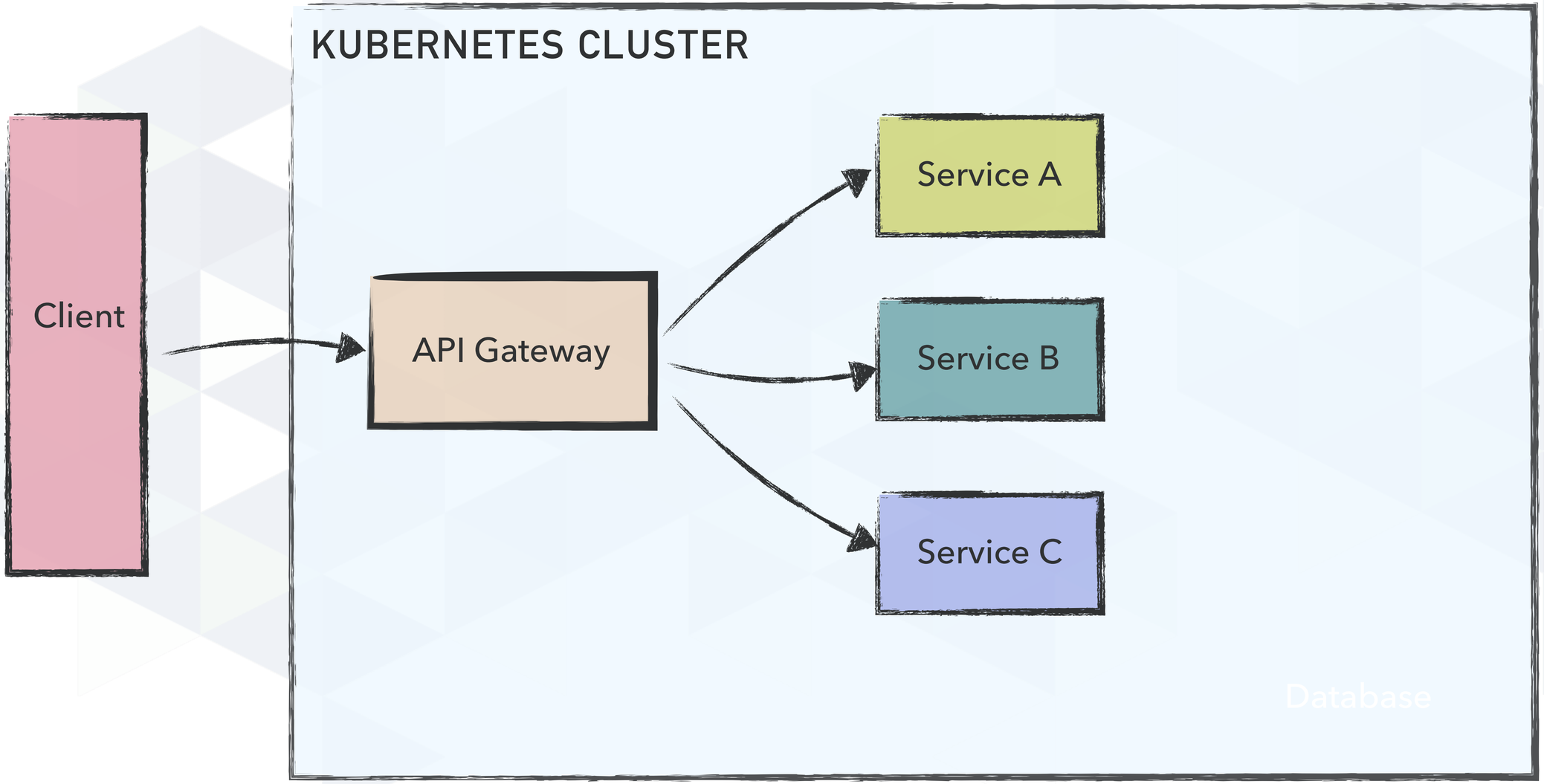
“A and B can exchange packets in a way that validates the identity on both sides; has clear authorization semantics; is confidential to third parties; and is measurable and inspectable”

DISTRIBUTED SYSTEM

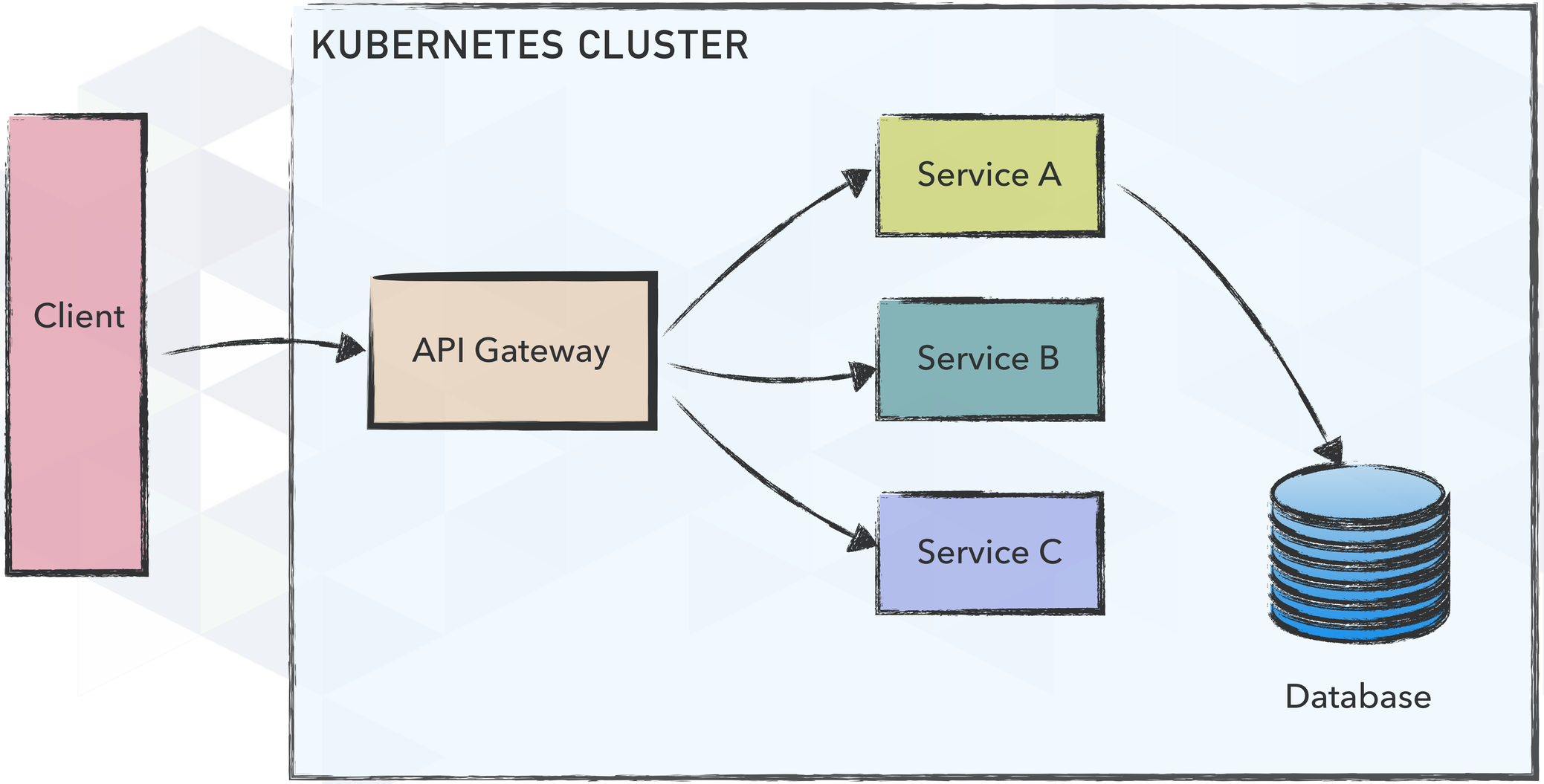
DISTRIBUTED SYSTEM



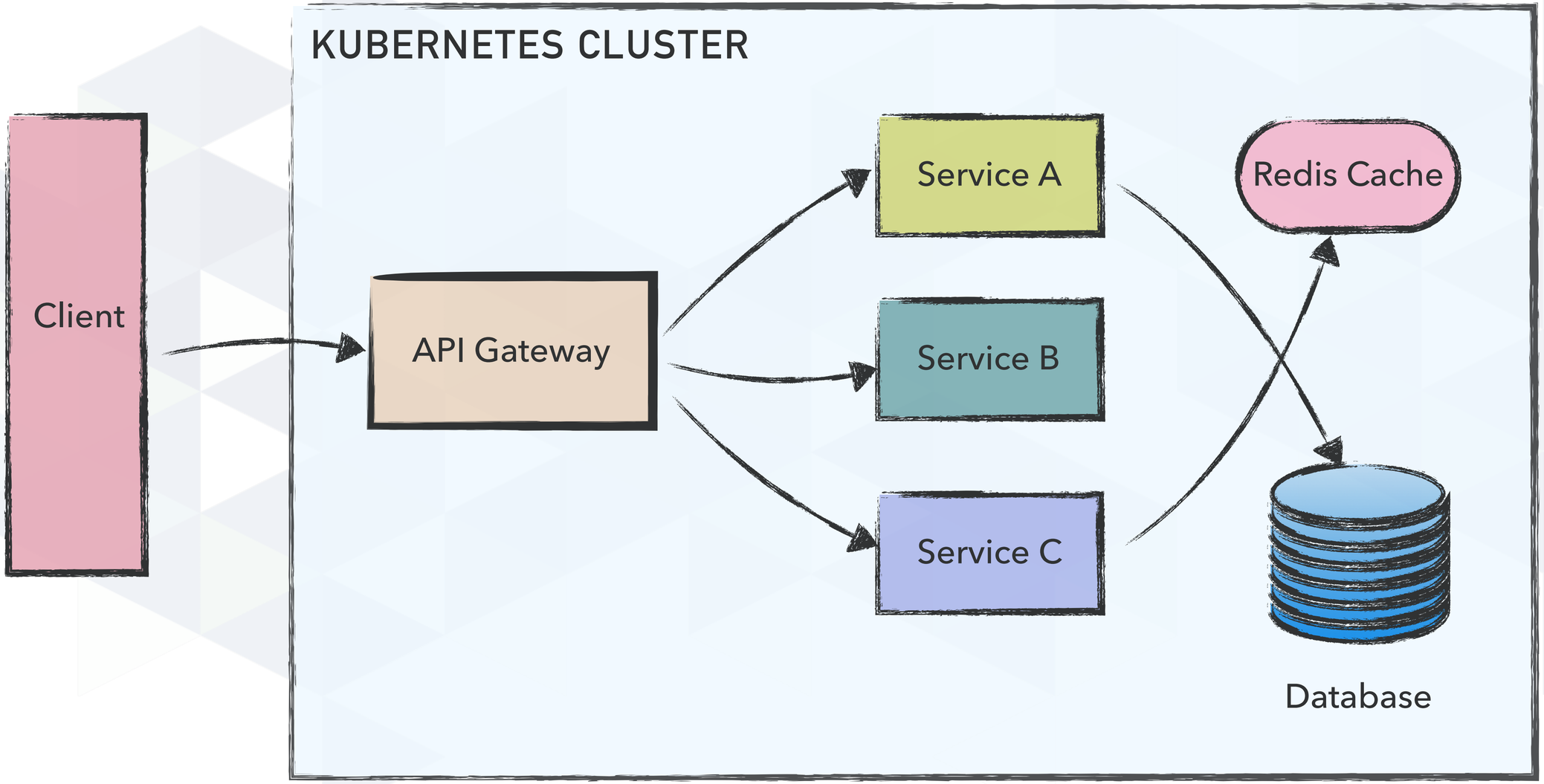
DISTRIBUTED SYSTEM



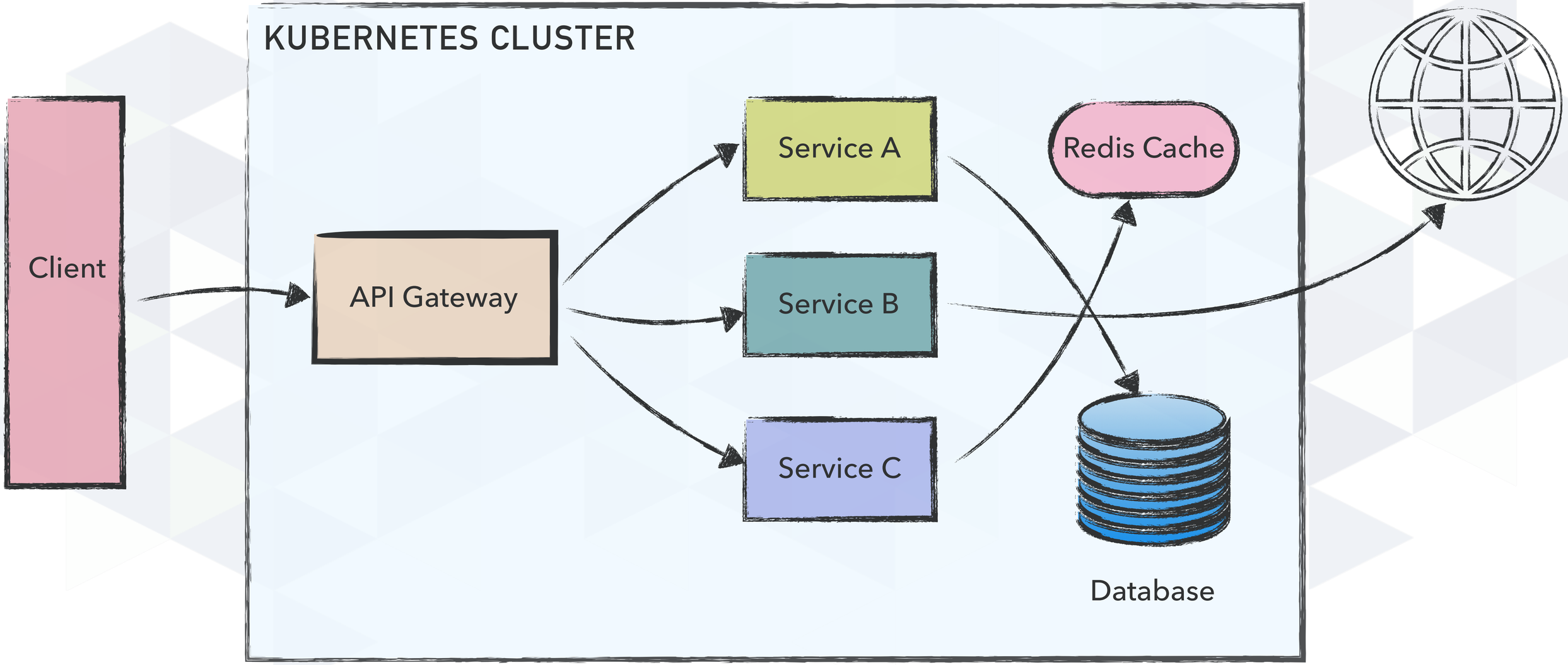
DISTRIBUTED SYSTEM



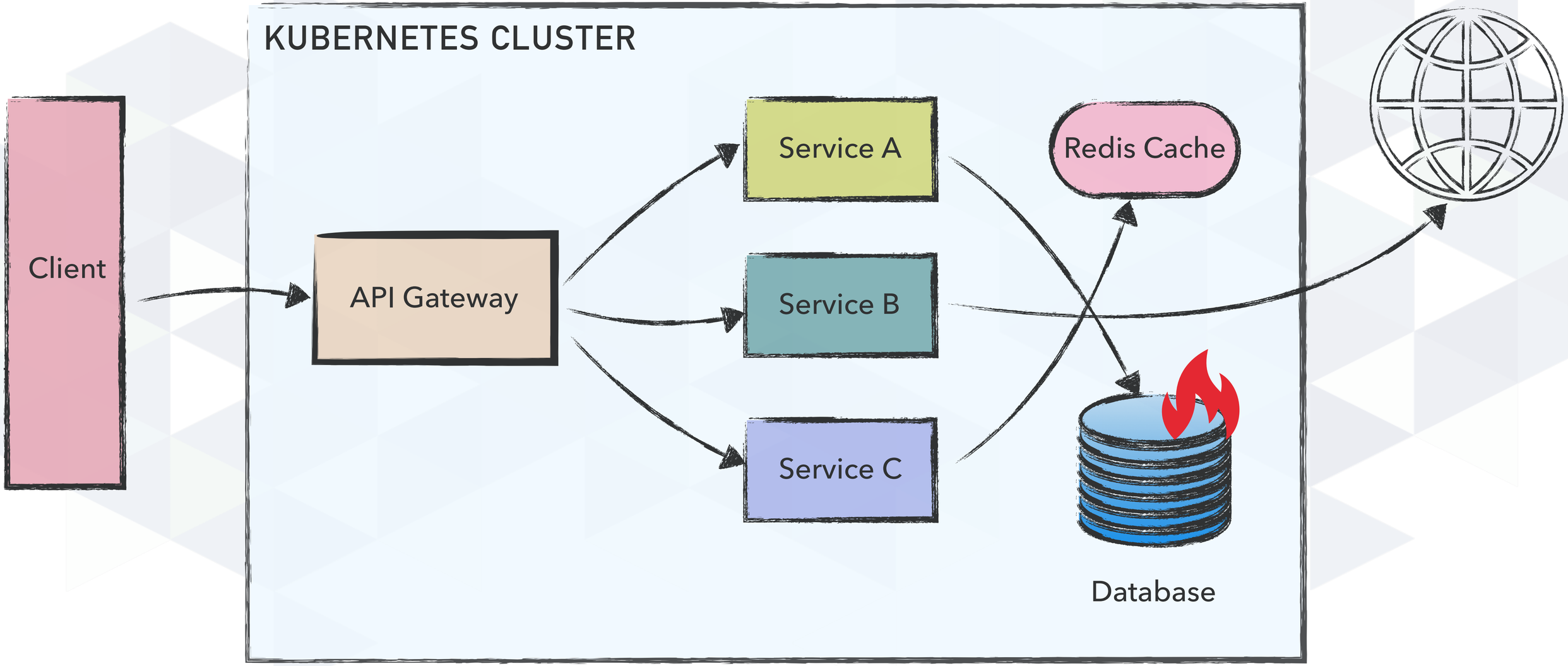
DISTRIBUTED SYSTEM



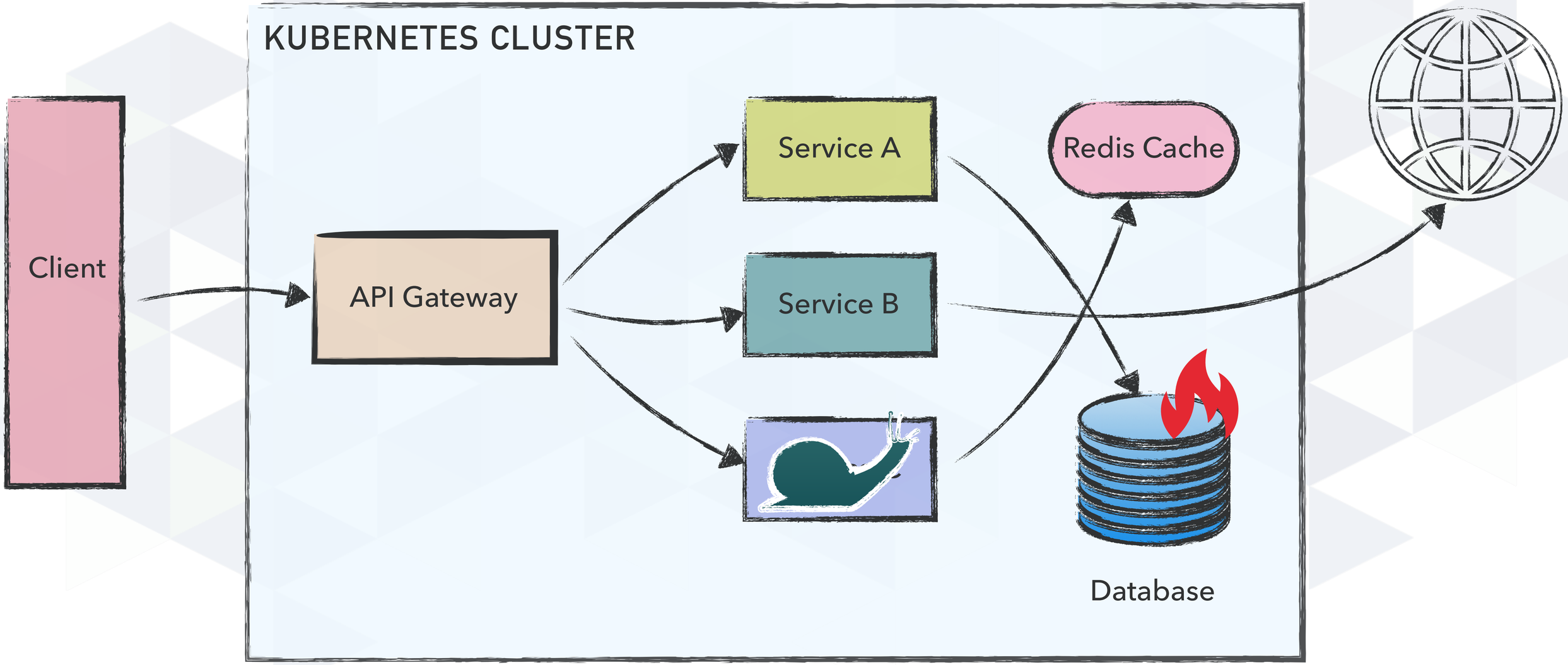
DISTRIBUTED SYSTEM



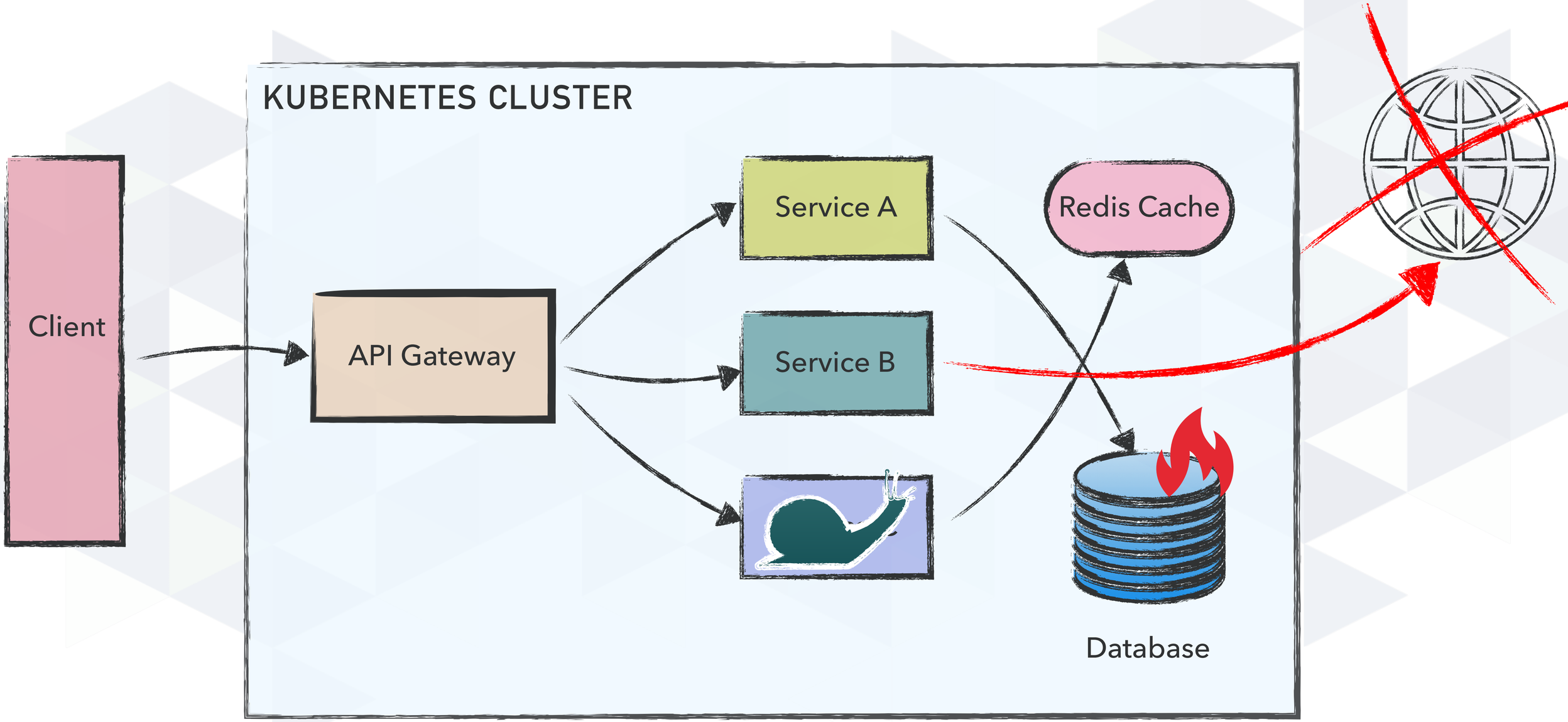
DISTRIBUTED SYSTEM



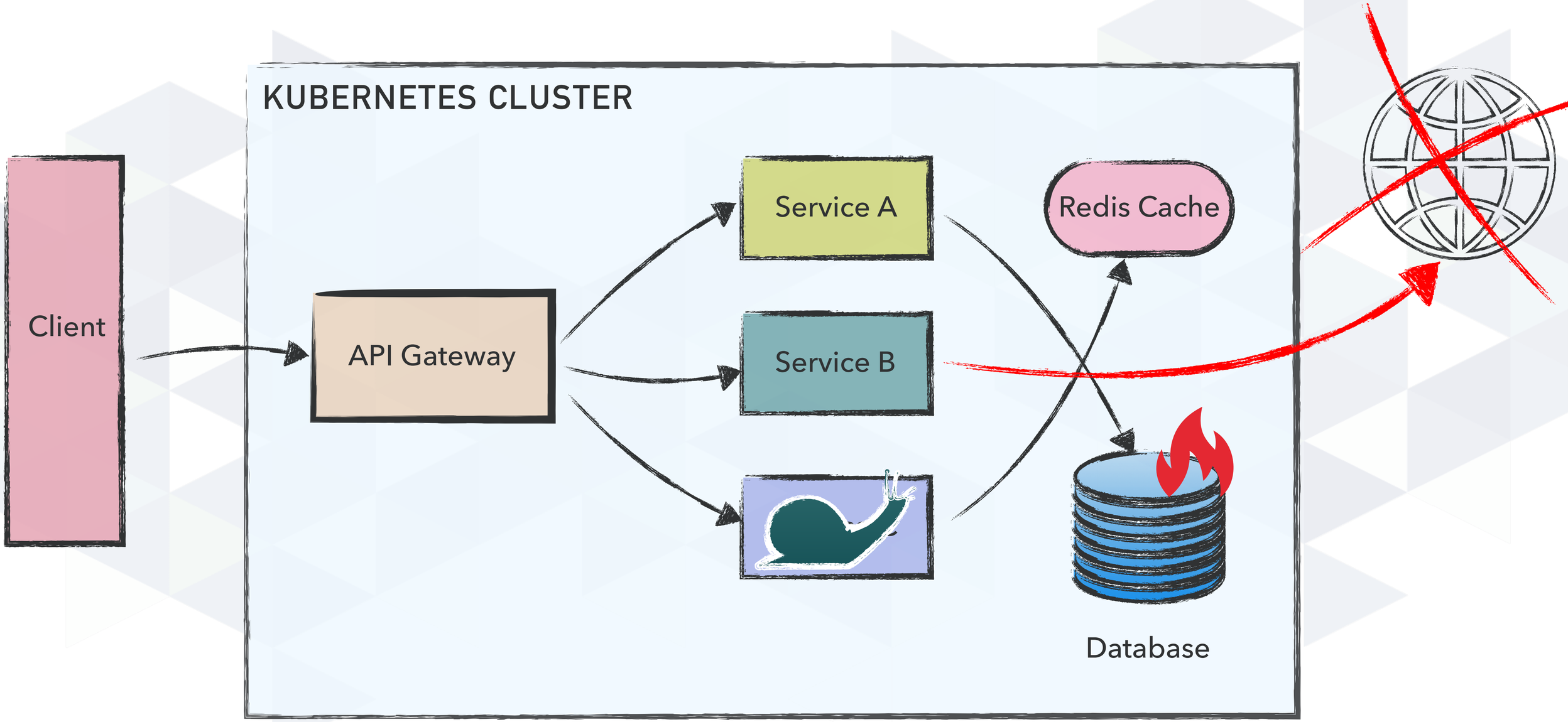
DISTRIBUTED SYSTEM



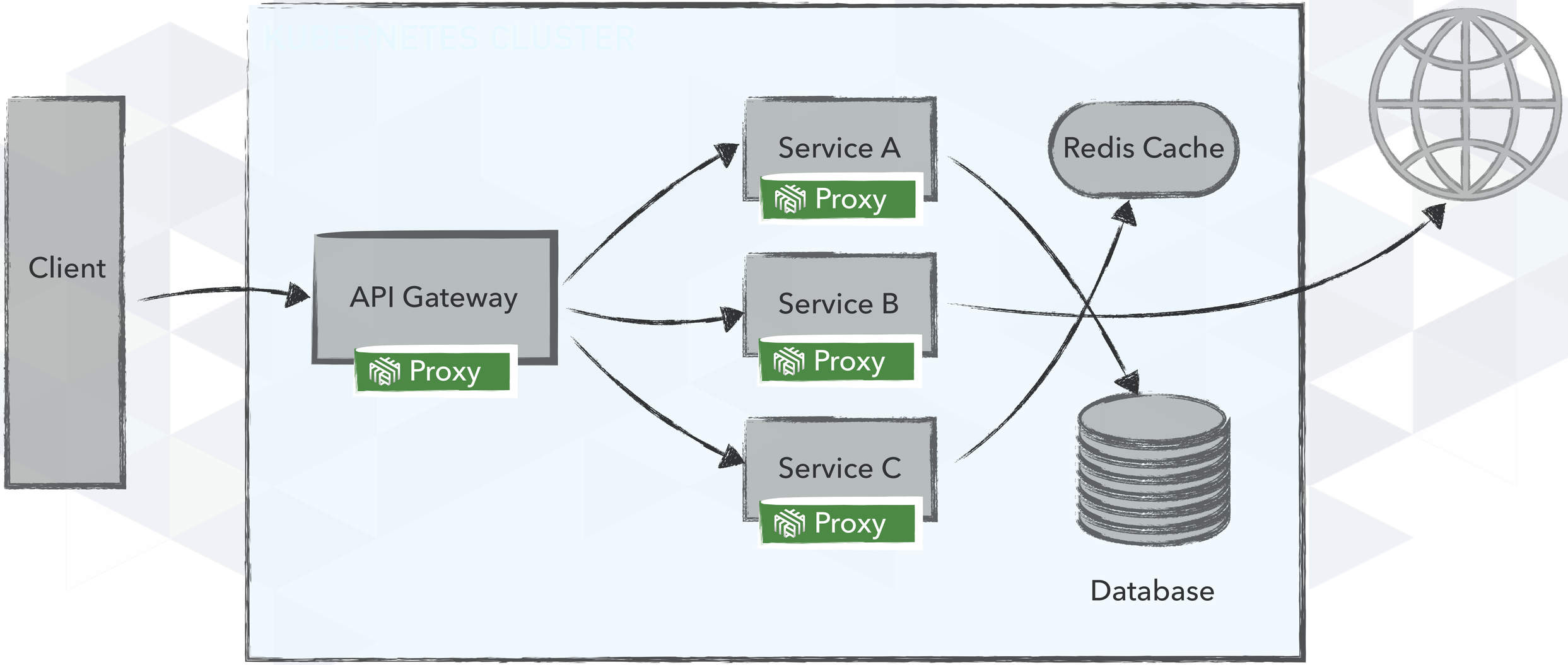
DISTRIBUTED SYSTEM



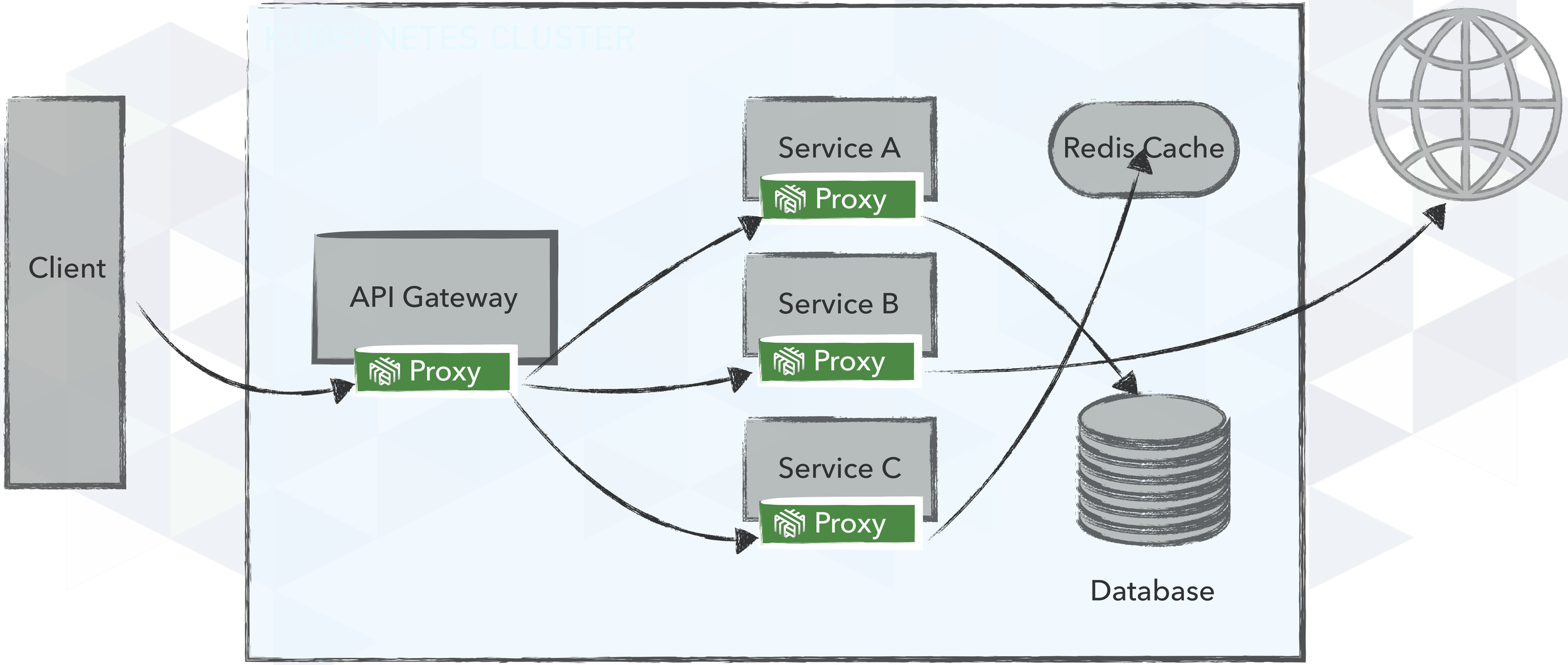
DISTRIBUTED SYSTEM



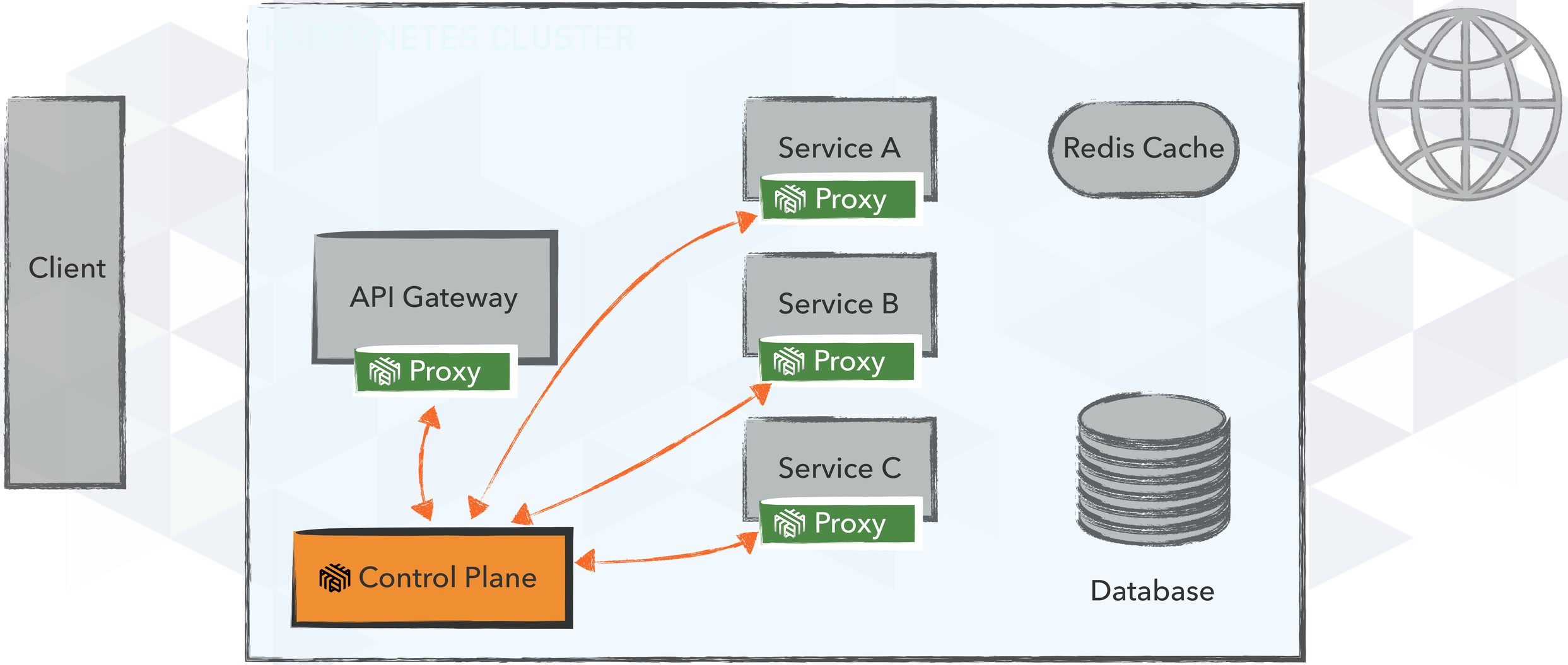
DISTRIBUTED SYSTEM



DISTRIBUTED SYSTEM



DISTRIBUTED SYSTEM



CONTROL PLANE

- TLS certificates for the proxy
- Service Discovery
- Service Profiles
- Automatic Proxy Injection
- Dashboard + Metrics
- API interface for CLI commands (tap, stat, etc…)

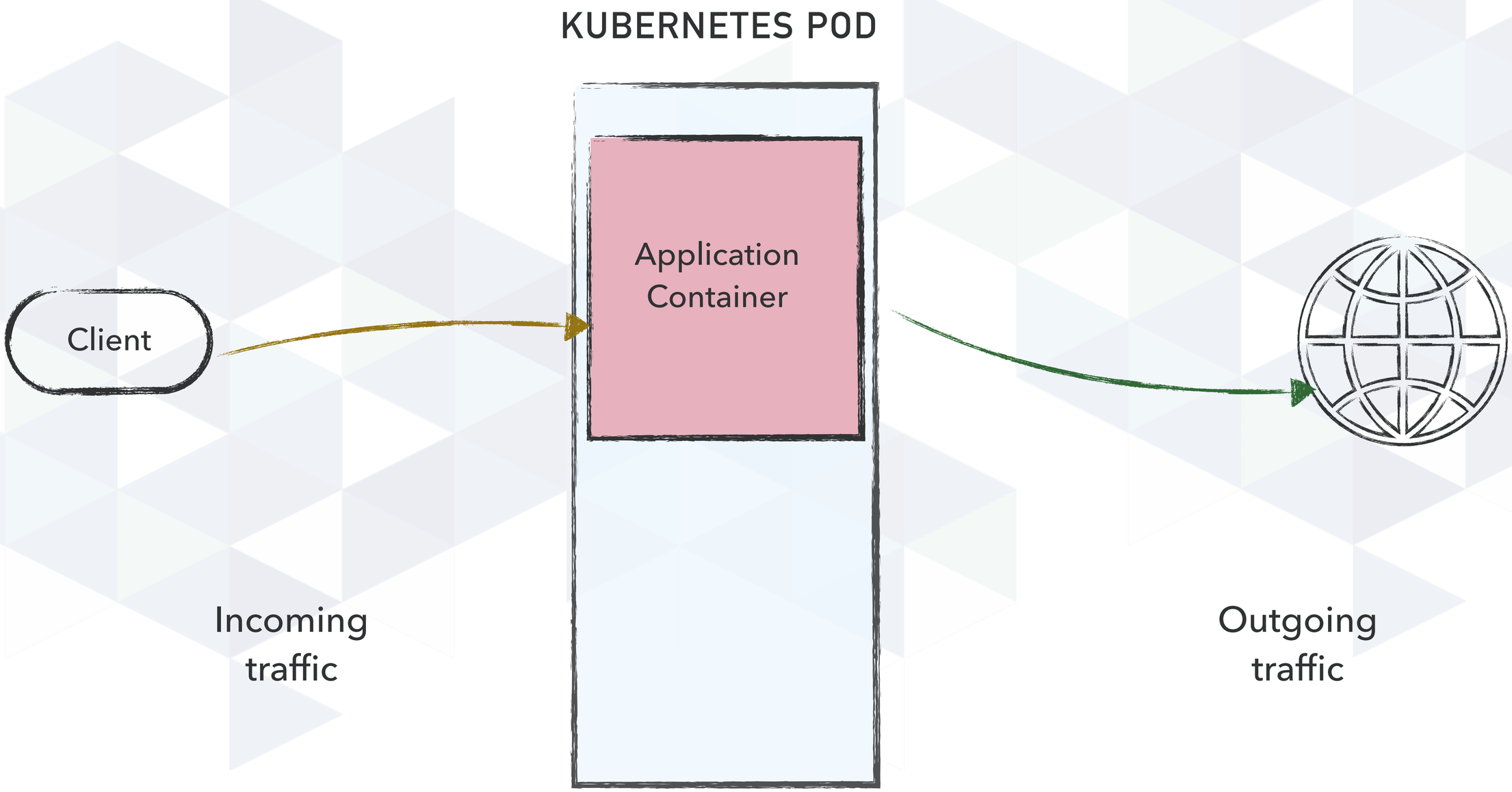
PROXY (DATA PLANE)

- Ultralight transparent proxy written in Rust
- Automatic Prometheus metrics export for HTTP and TCP traffic.
- Latency-aware, layer-7 load balancing
- Automatic TLS
- An on-demand diagnostic tap API

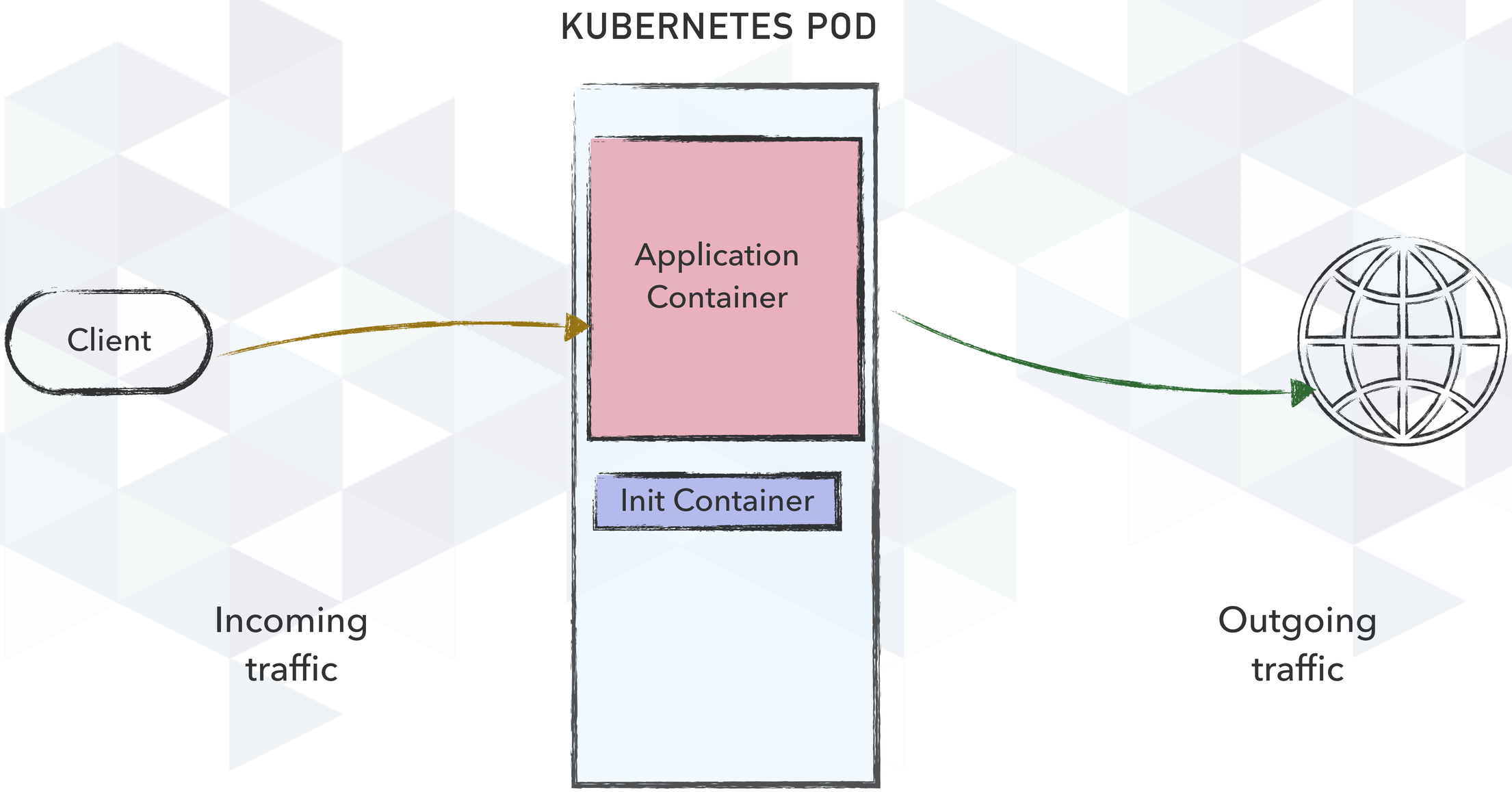
INJECTION

- Usually accomplished by the proxy-injector component
- Can be automatic or manual
- An init container added, which setups iptables rules for the pod
- A container that runs the proxy, intercepting traffic

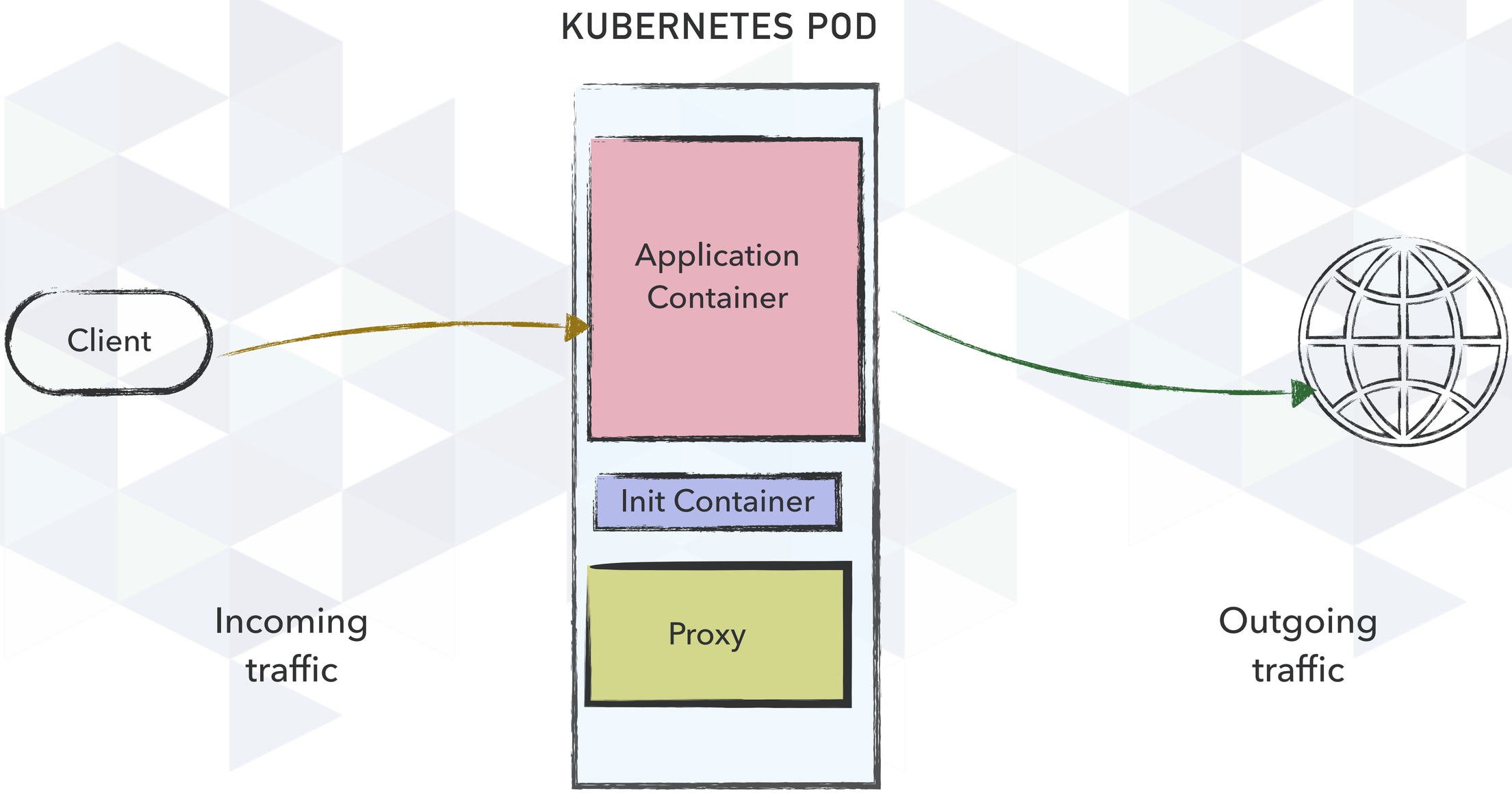
INJECTION


KUBERNETES POD
Application Container
Client
Incoming traffic
Outgoing traffic

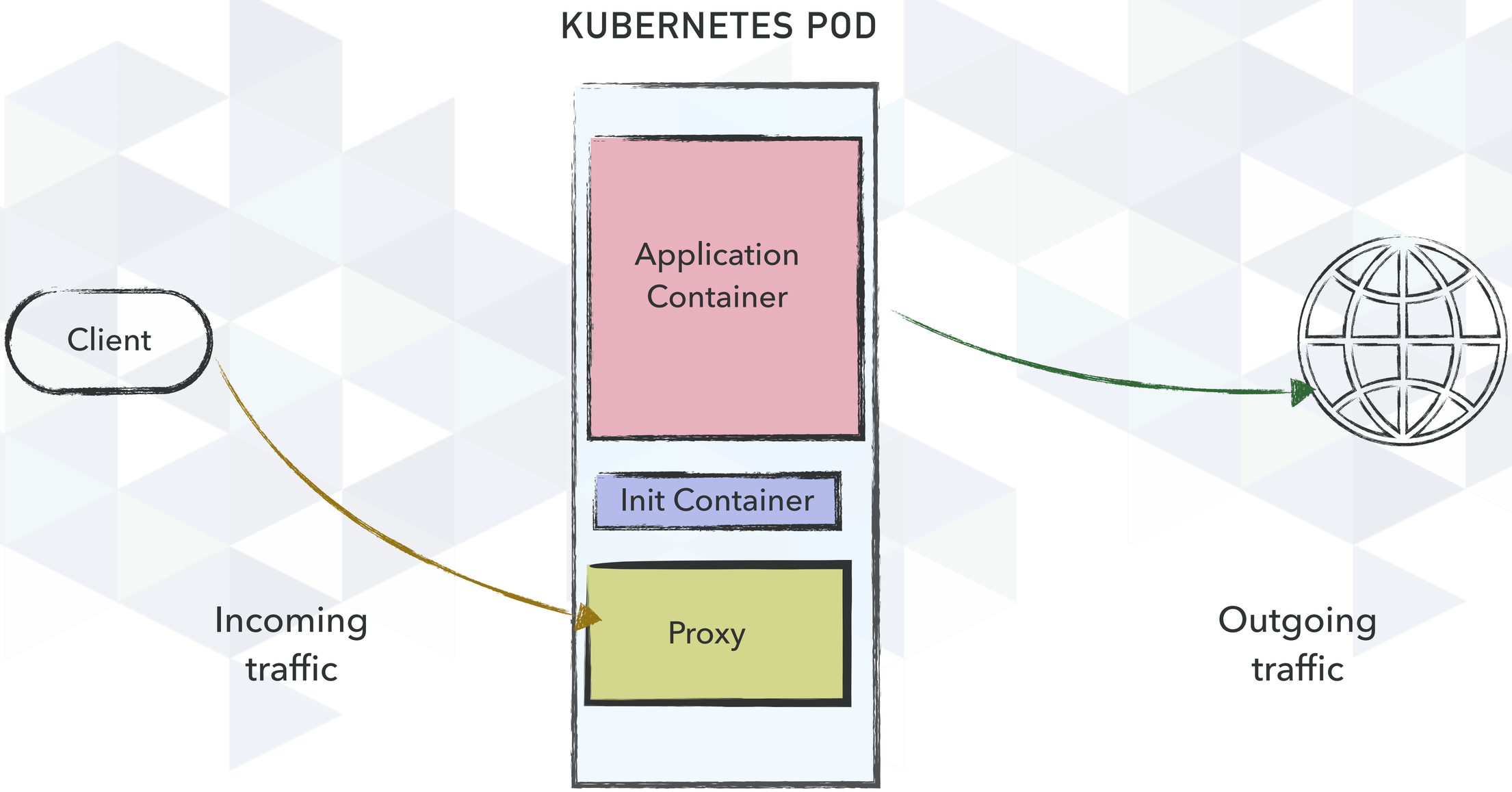
INJECTION


KUBERNETES POD
Application Container
Init Container
Client
Incoming traffic
Outgoing traffic

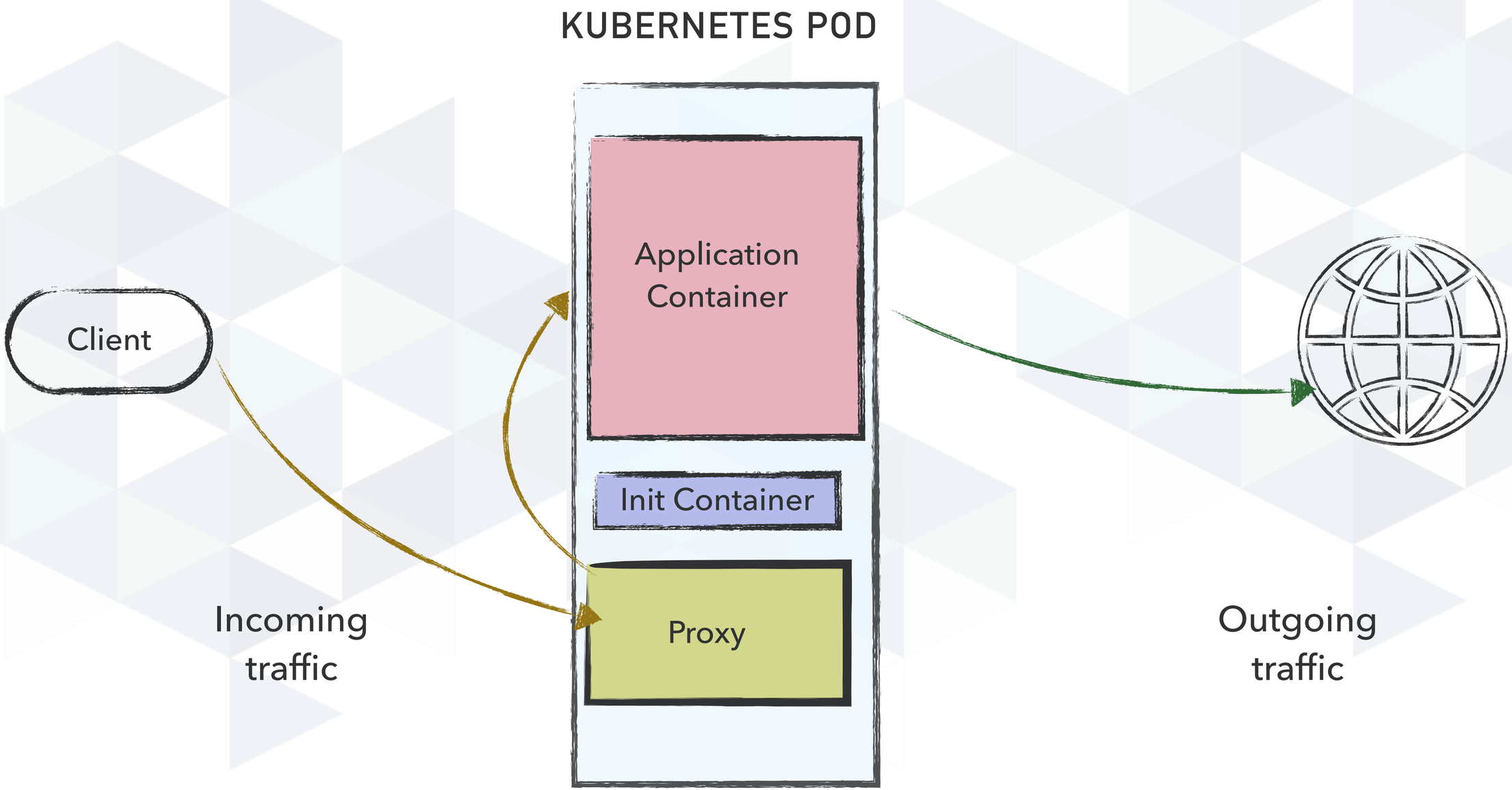
INJECTION


KUBERNETES POD
Client
Application Container
Init Container
Proxy
Incoming traffic
Outgoing traffic

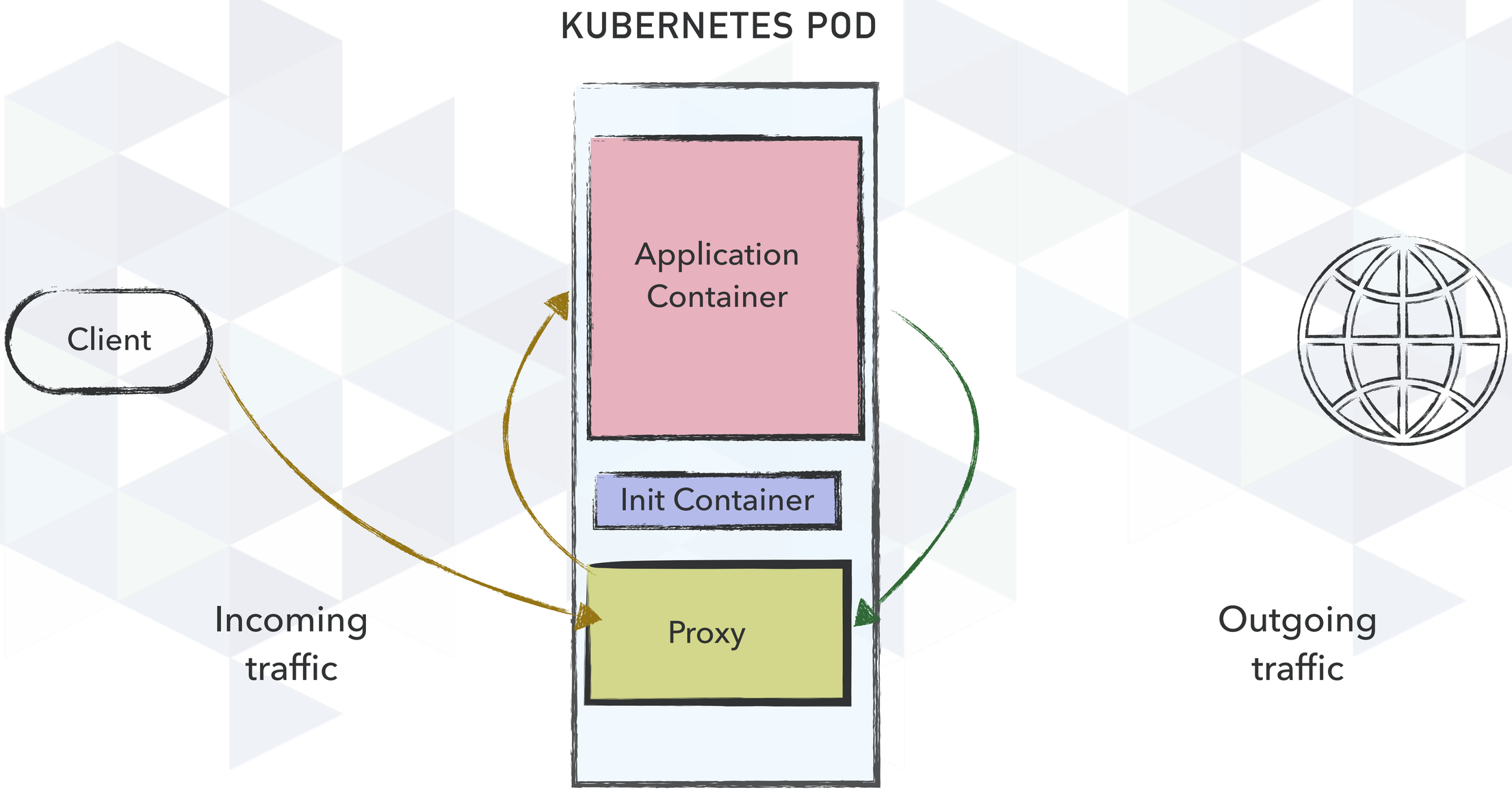
INJECTION


KUBERNETES POD
Application Container
Init Container
Proxy
Client
Incoming traffic
Outgoing traffic

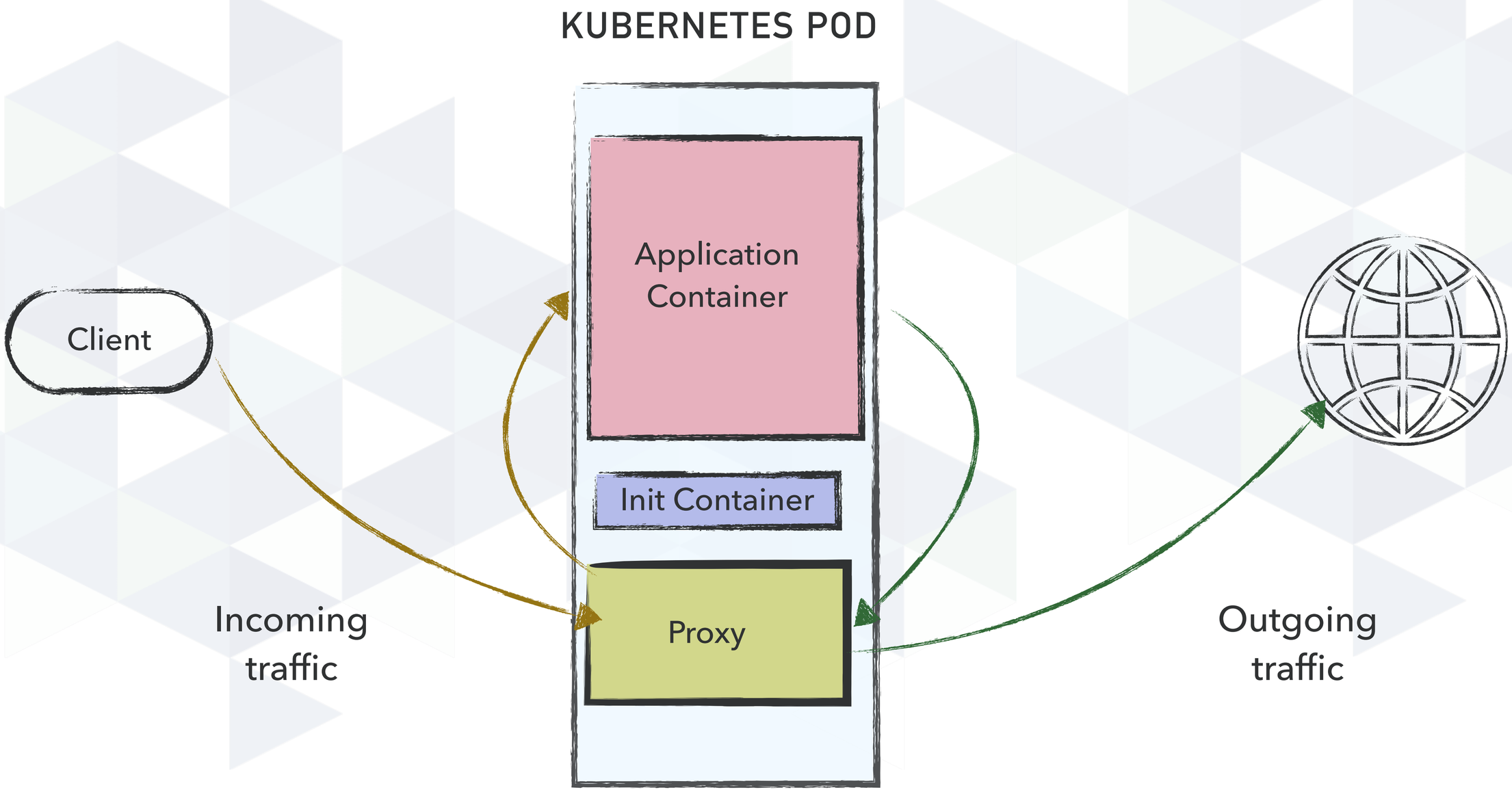
INJECTION


KUBERNETES POD
Application Container
Init Container
Proxy
Client
Incoming traffic
Outgoing traffic

INJECTION


KUBERNETES POD
Application Container
Init Container
Proxy
Client
Incoming traffic
Outgoing traffic

INJECTION


KUBERNETES POD
Application Container
Init Container
Proxy
Client
Incoming traffic
Outgoing traffic

CORE CONCEPTS

- Observability: Collecting actionable traffic metrics
- Security: Encrypting traffic between services
- Reliability: Ensuring services are available
- Traffic Management: Routing traffic to services

MULTICLUSTER DEEPDIVE

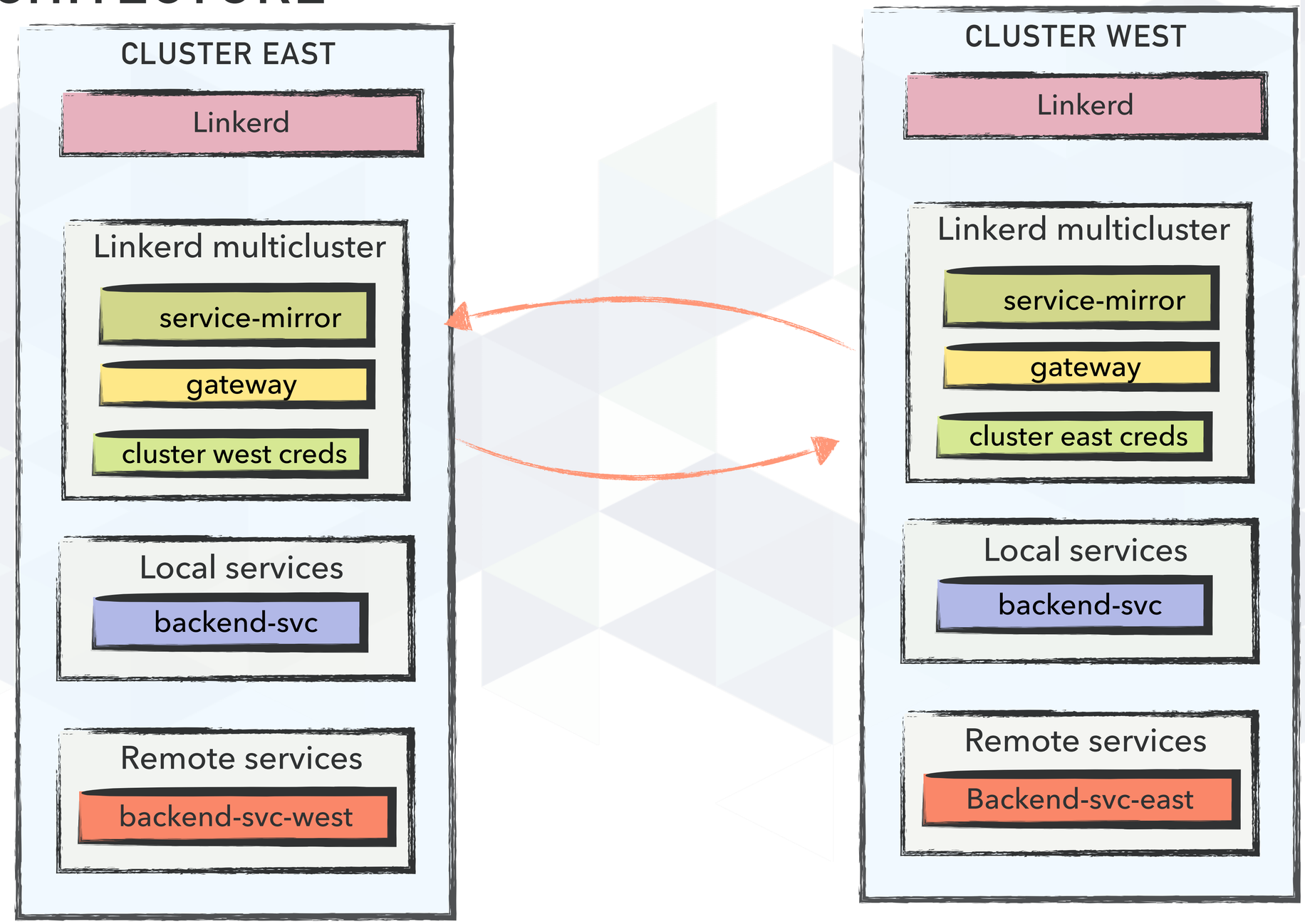
WHY MULTIPLE CLUSTERS ?

- Traffic Migration
- Canary Deployments
- Different Environments
- Failover

CORE CONCEPTS

- **Secure**: everything happens over mTLS
- **Kubernetes-first:** remote services should appear as K8s services
- **No SPOF:** no single cluster is blessed or magical
- **Transparent:** applications do not need to know whether a service is remote or local
- **Network independent:** only requirement is gateway connectivity

ARCHITECTURE



ARCHITECTURE

- Service mirror - monitors the exported state of the target cluster and replicating it
- Gateway - responsible for routing incoming traffic to the appropriate target services
- Credentials - service account (target cluster) and a secret containing k8s api config (source cluster)

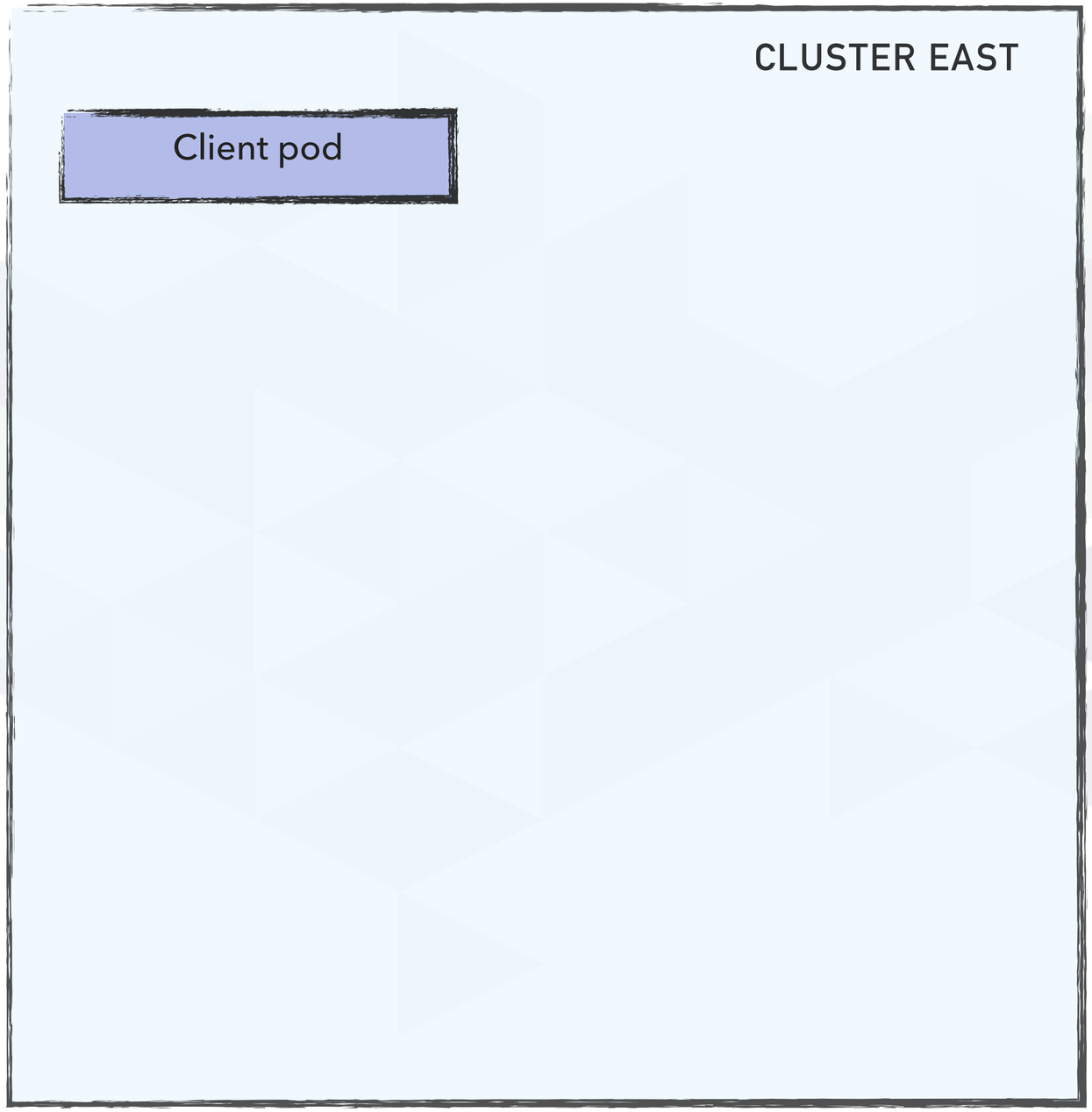
DEMO TIME

- Two clusters - east and west
- Each have a backend-svc installed
- A test client deployed on cluster east
- We want to split the traffic to backend-svc between east and west

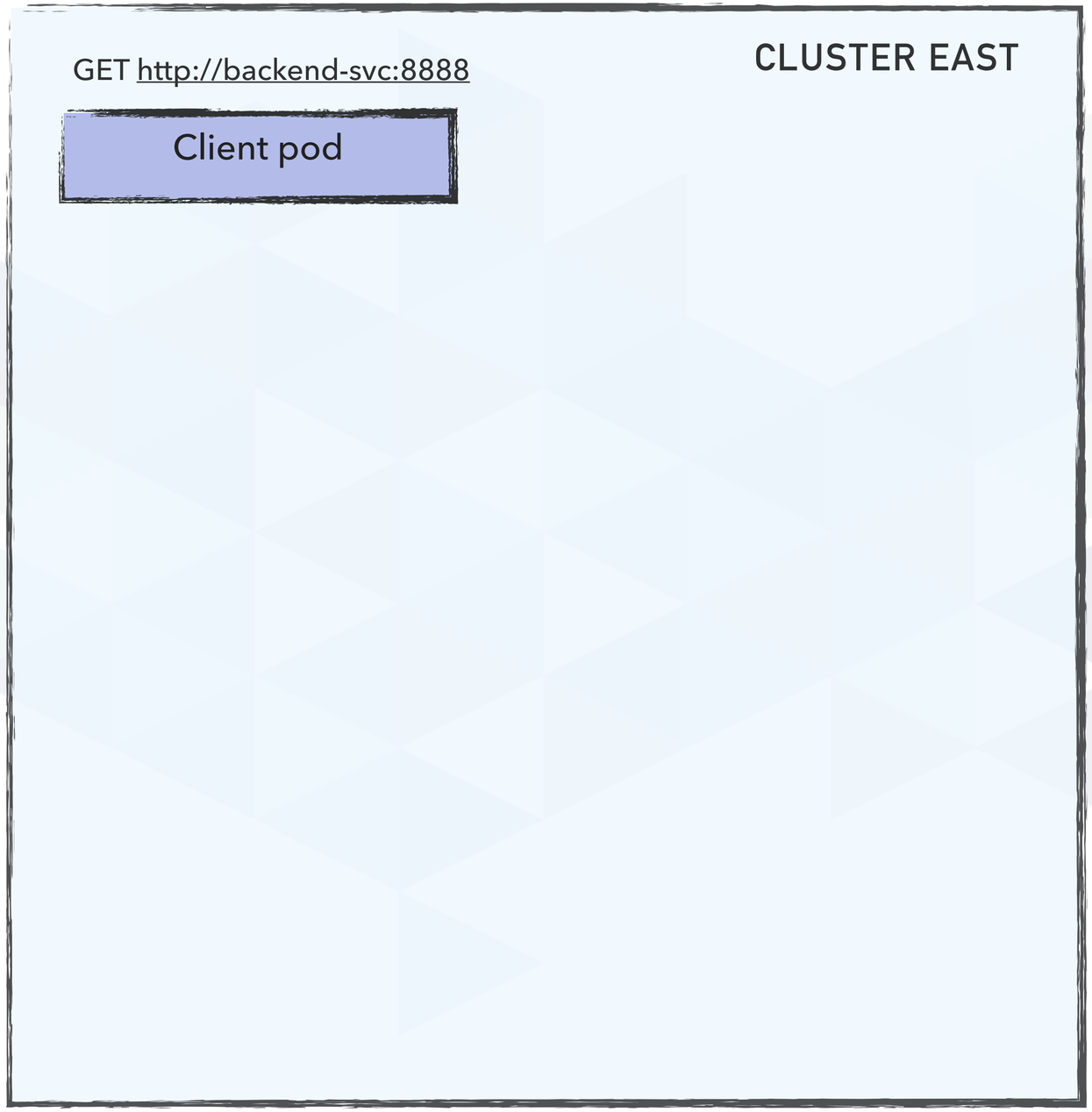
BUT… HOW DOES IT WORK?

BUT… HOW DOES IT WORK?

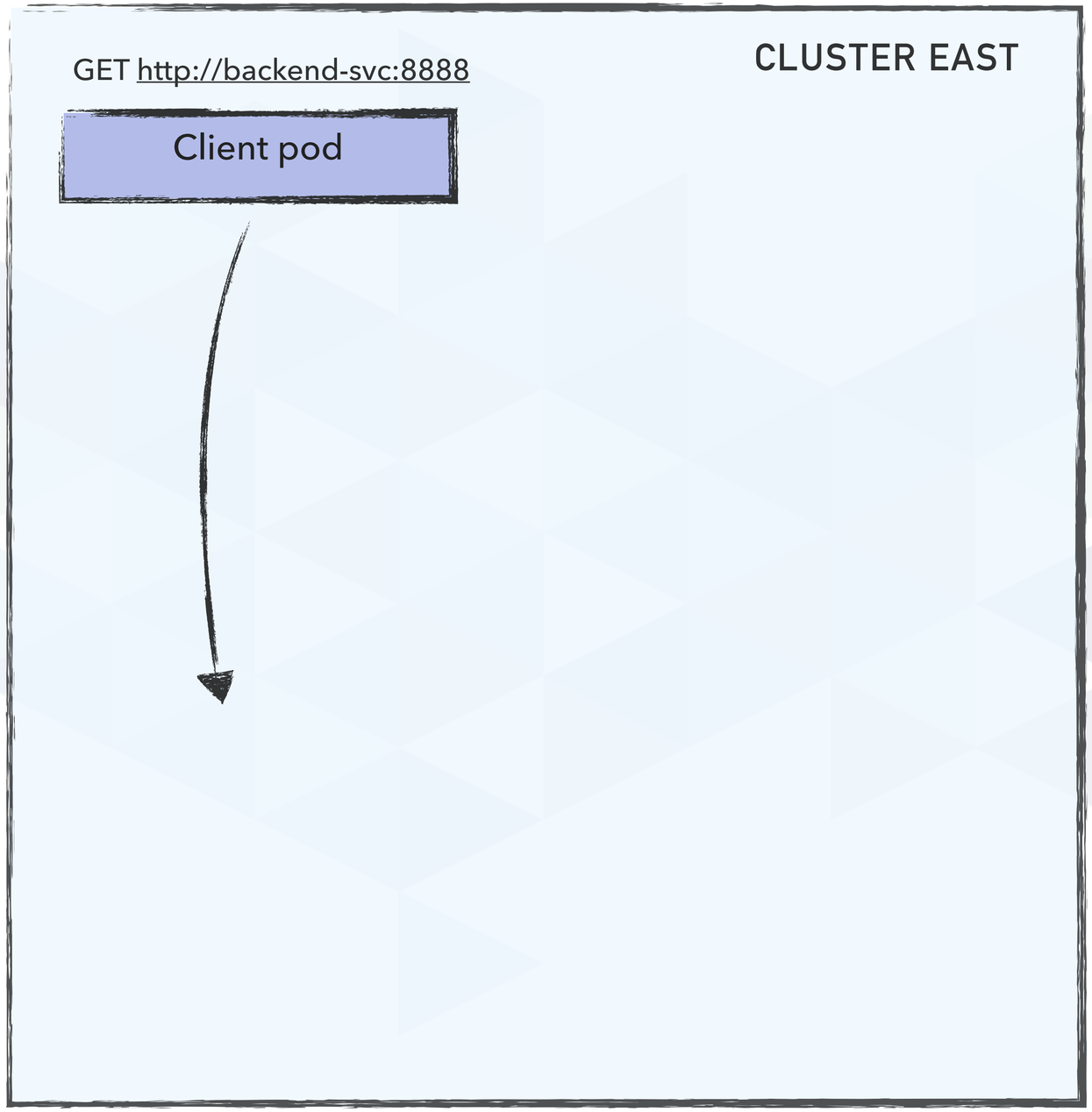
BUT... HOW DOES IT WORK?



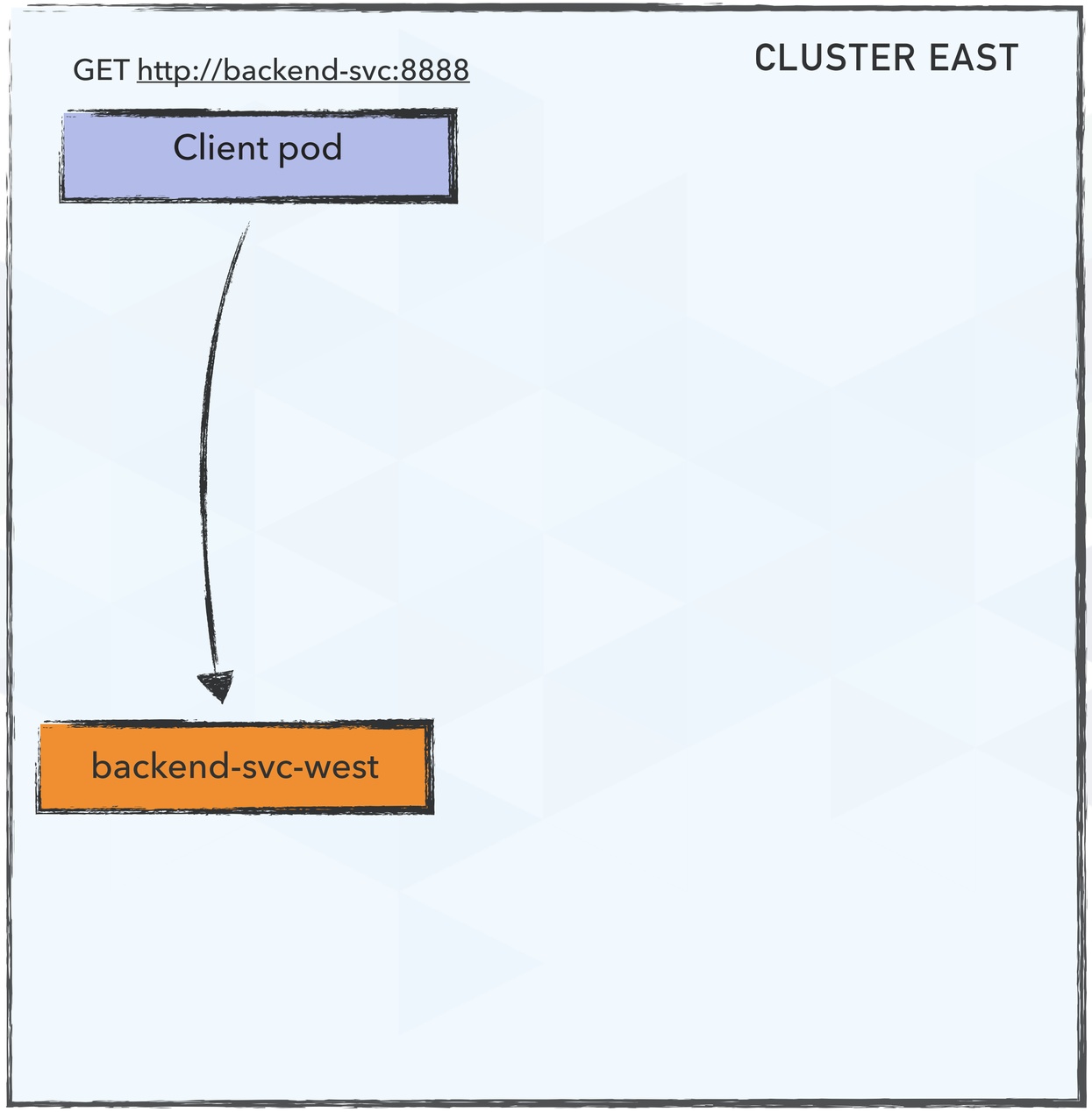
BUT… HOW DOES IT WORK?



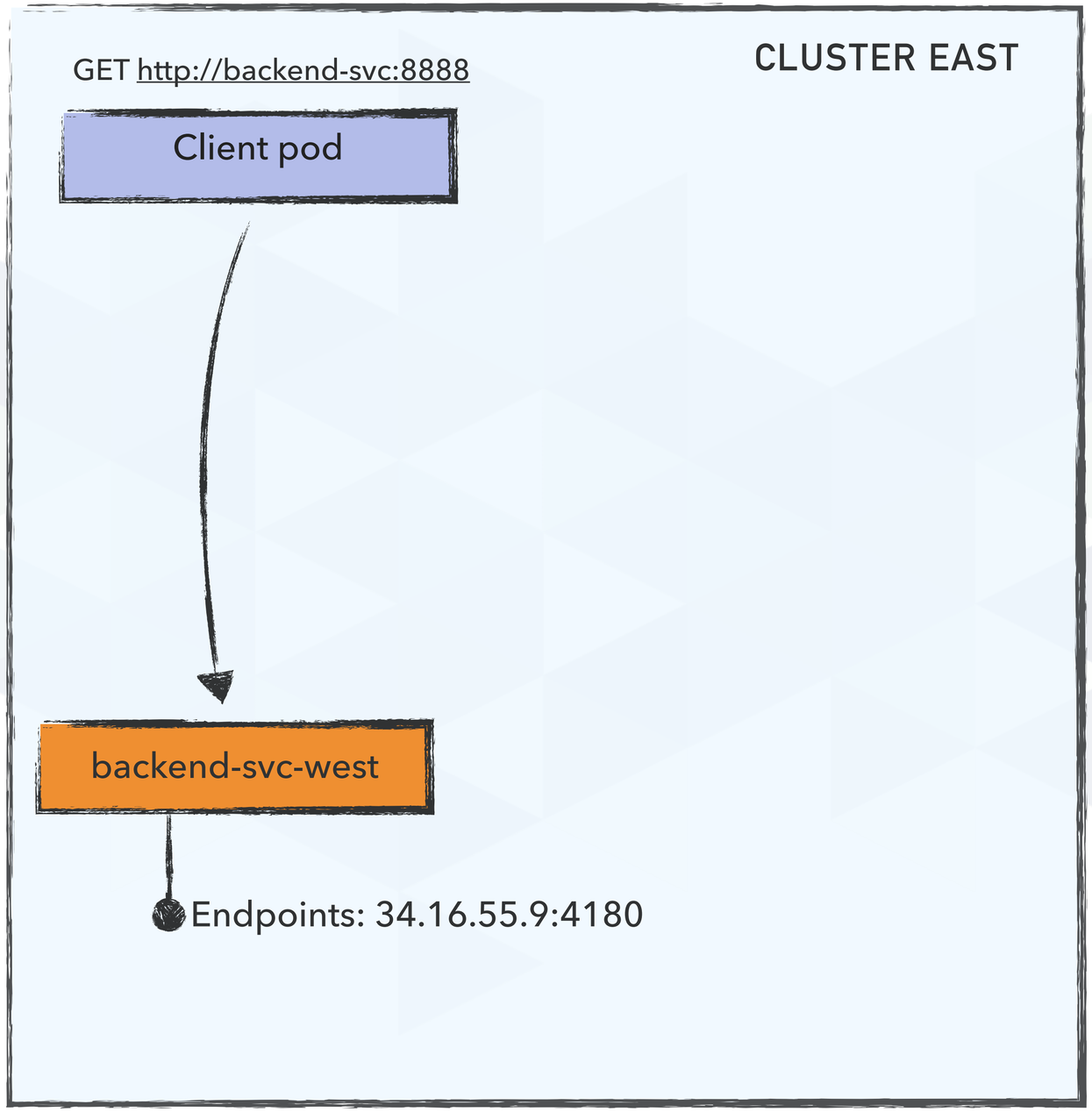
BUT… HOW DOES IT WORK?



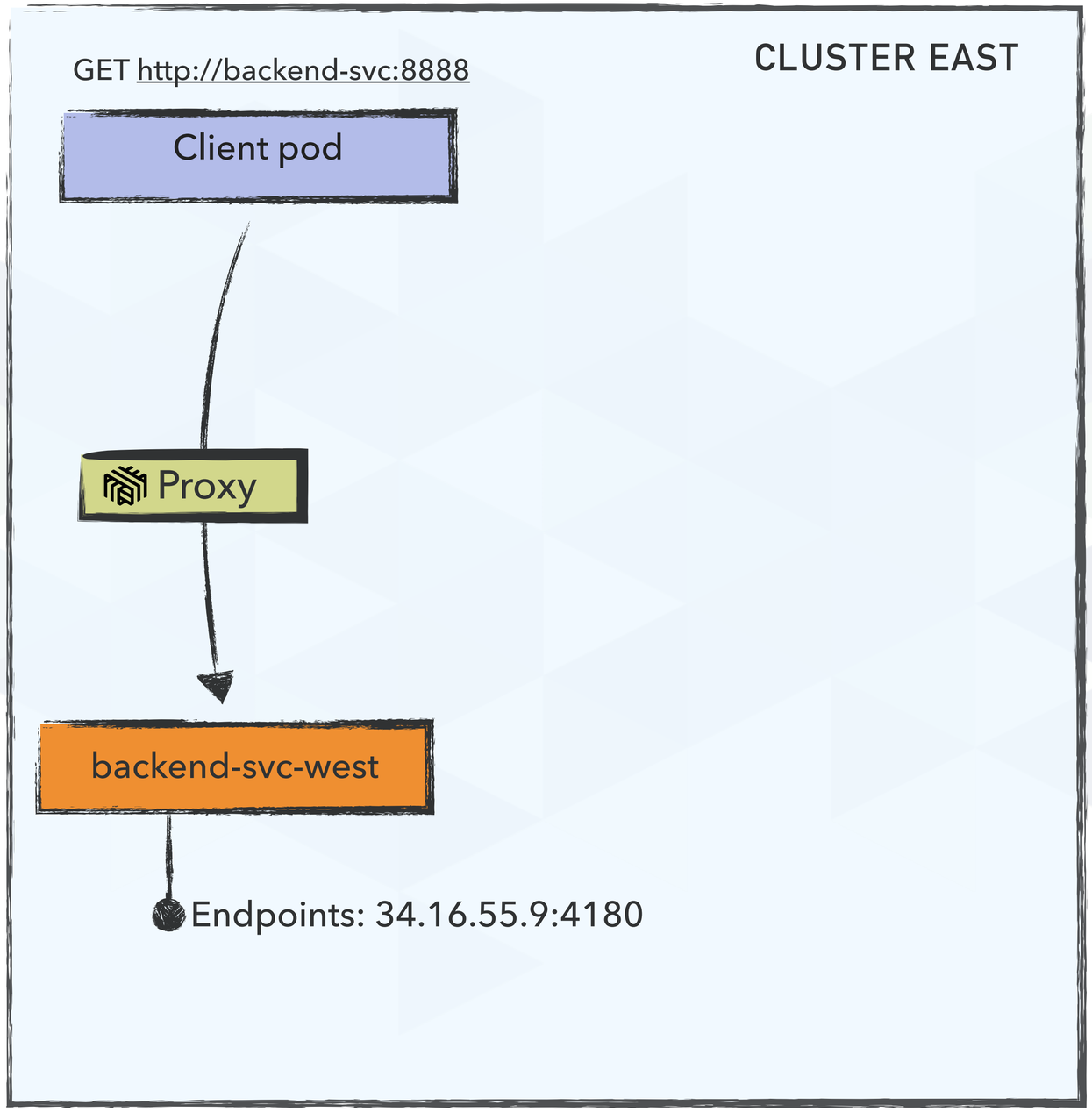
BUT… HOW DOES IT WORK?



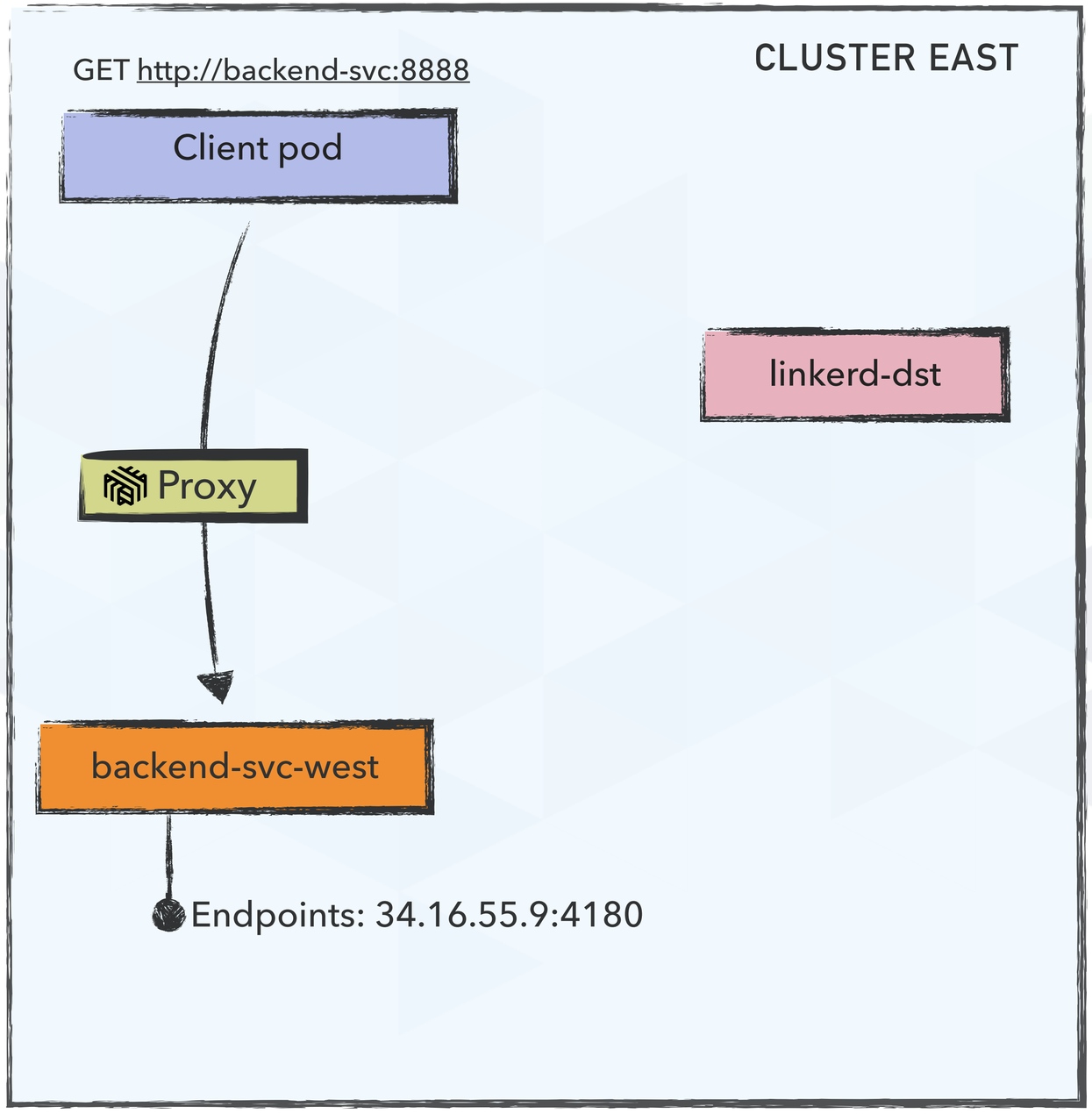
BUT… HOW DOES IT WORK?



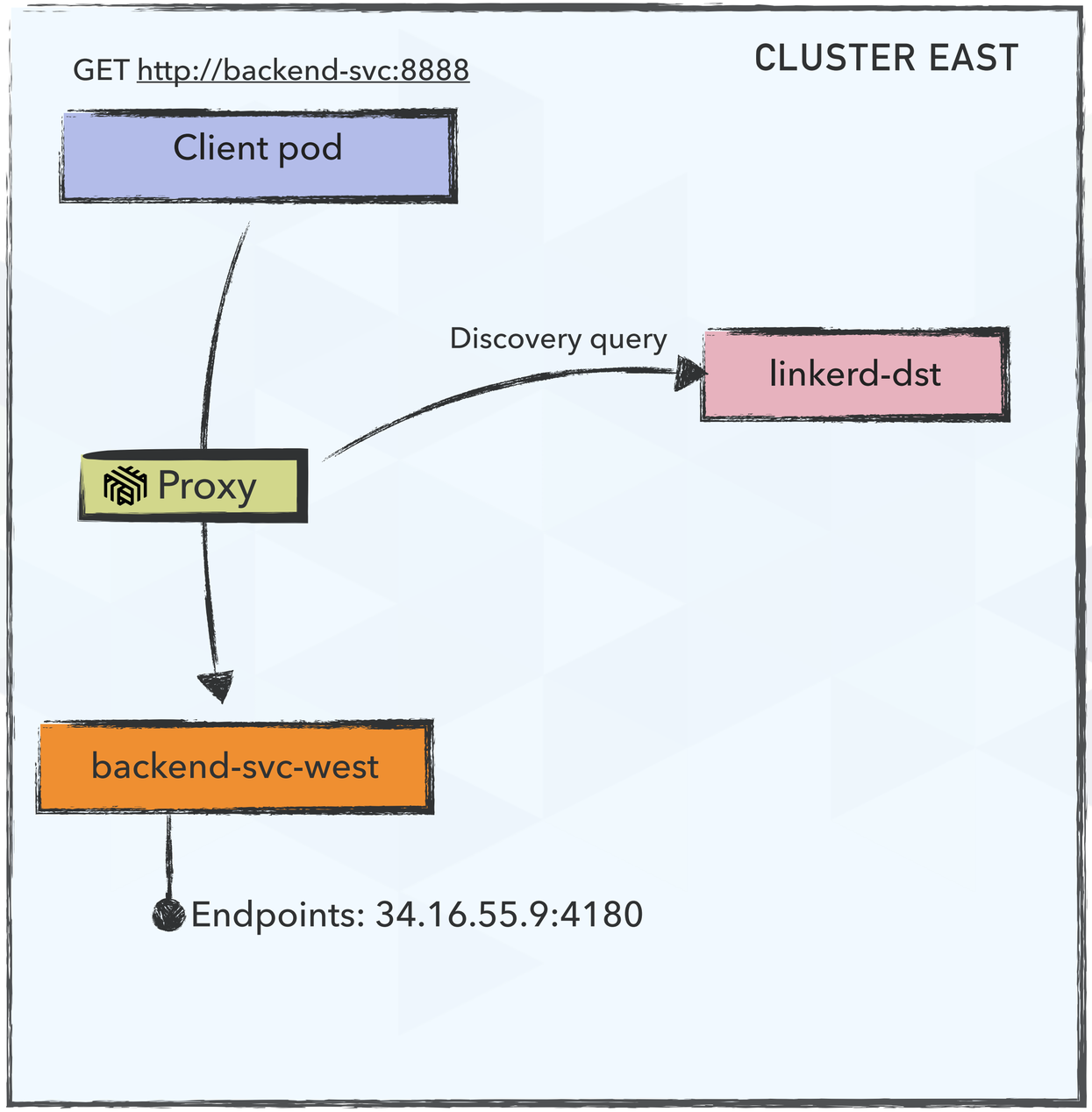
BUT… HOW DOES IT WORK?



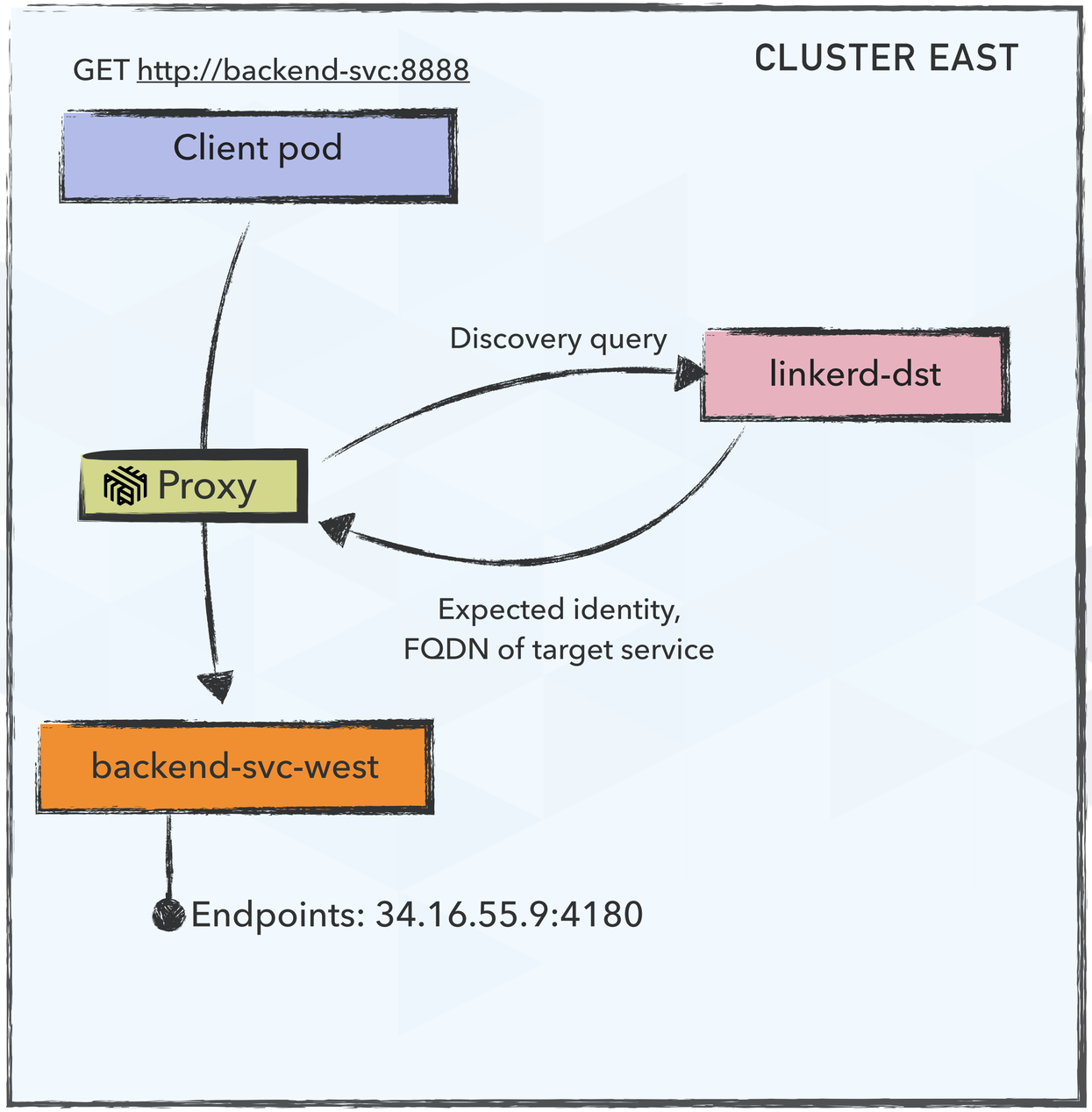
BUT… HOW DOES IT WORK?



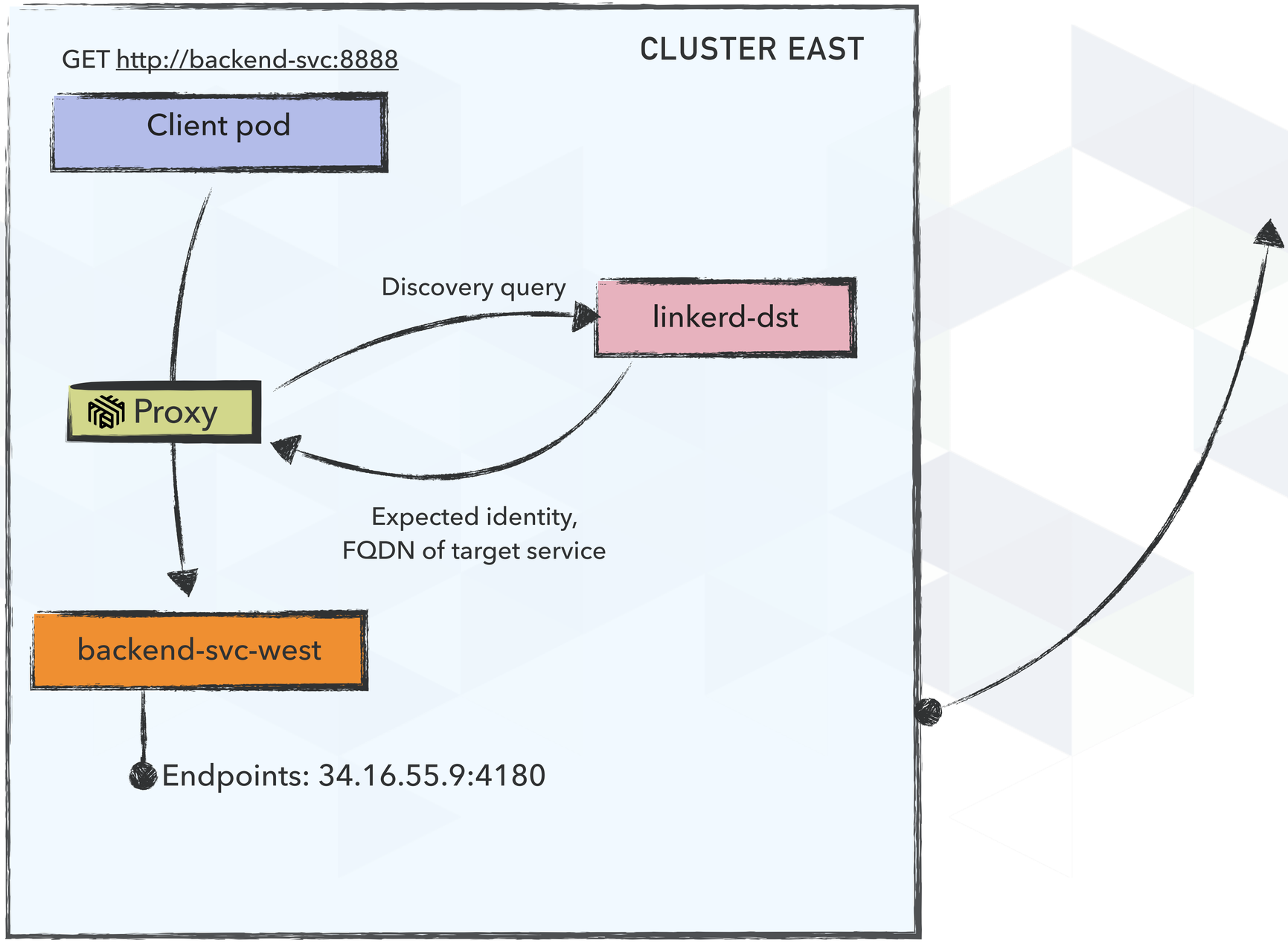
BUT… HOW DOES IT WORK?



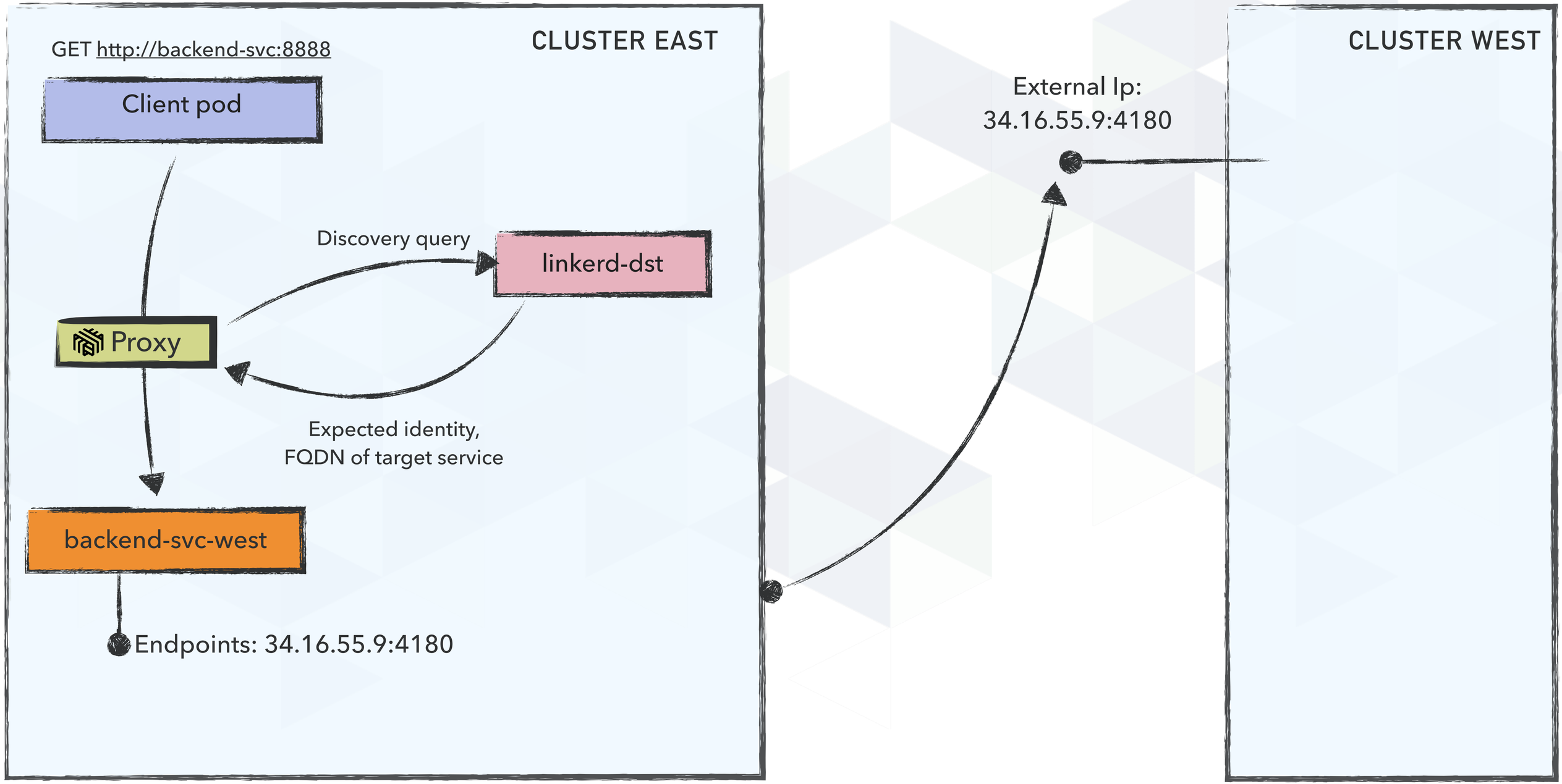
BUT… HOW DOES IT WORK?



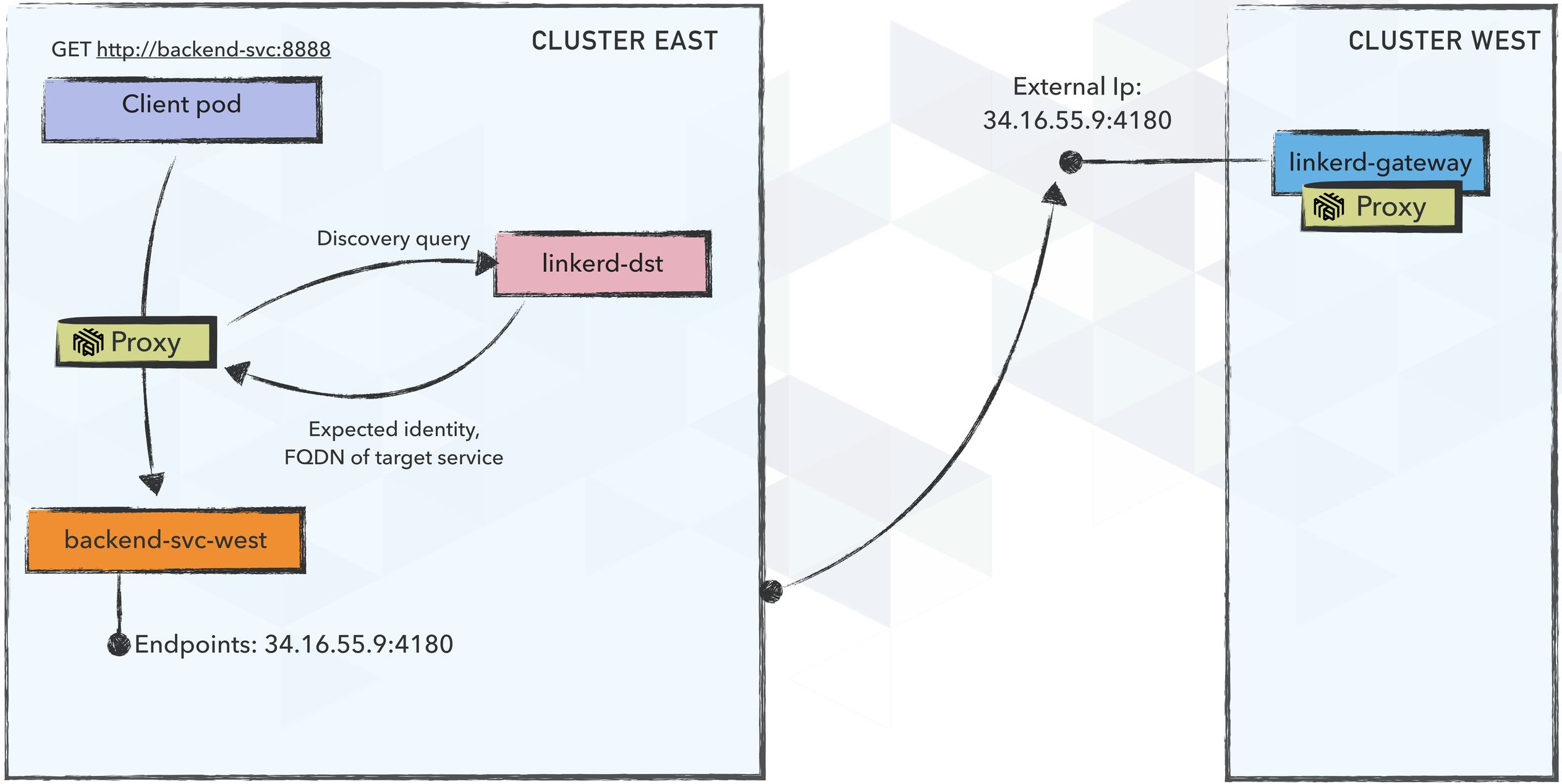
BUT… HOW DOES IT WORK?



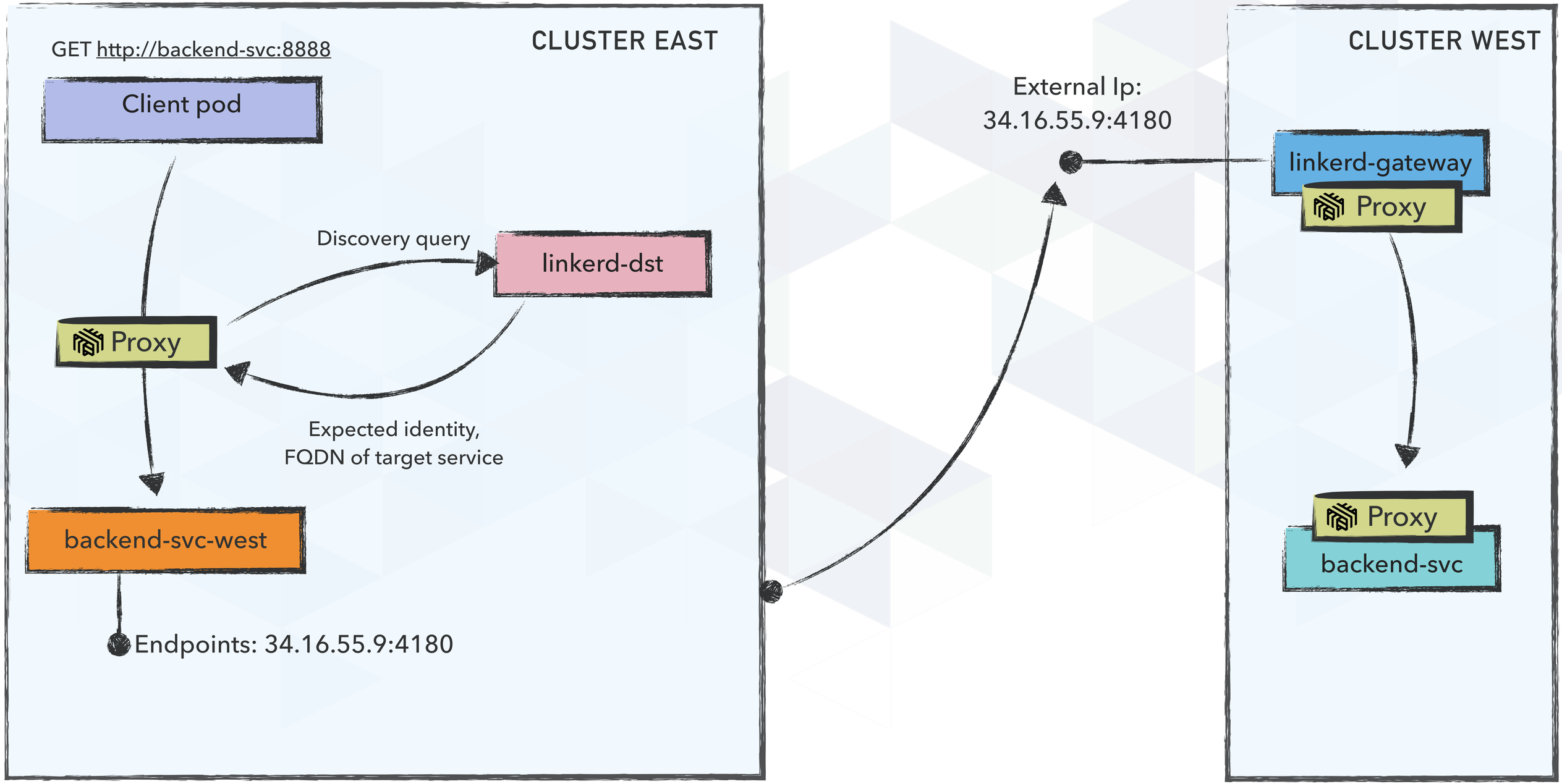
BUT… HOW DOES IT WORK?



BUT… HOW DOES IT WORK?



BUT… HOW DOES IT WORK?



FUTURE WORK

- Service mirror controller per target cluster
- Introduce a CRD to better represent target cluster information
- Support traffic policy, finer grained permissions control
- Support for TCP traffic

Q&A

https://github.com/zaharidichev/talks