Teleport

User Journey:
Scaling Multi-cluster Kubernetes
@ Teleport Cloud

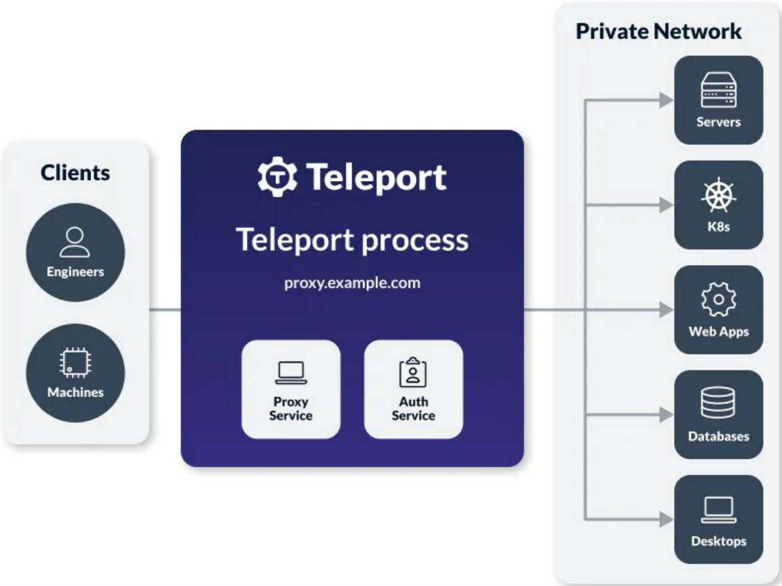
Stephen Levine

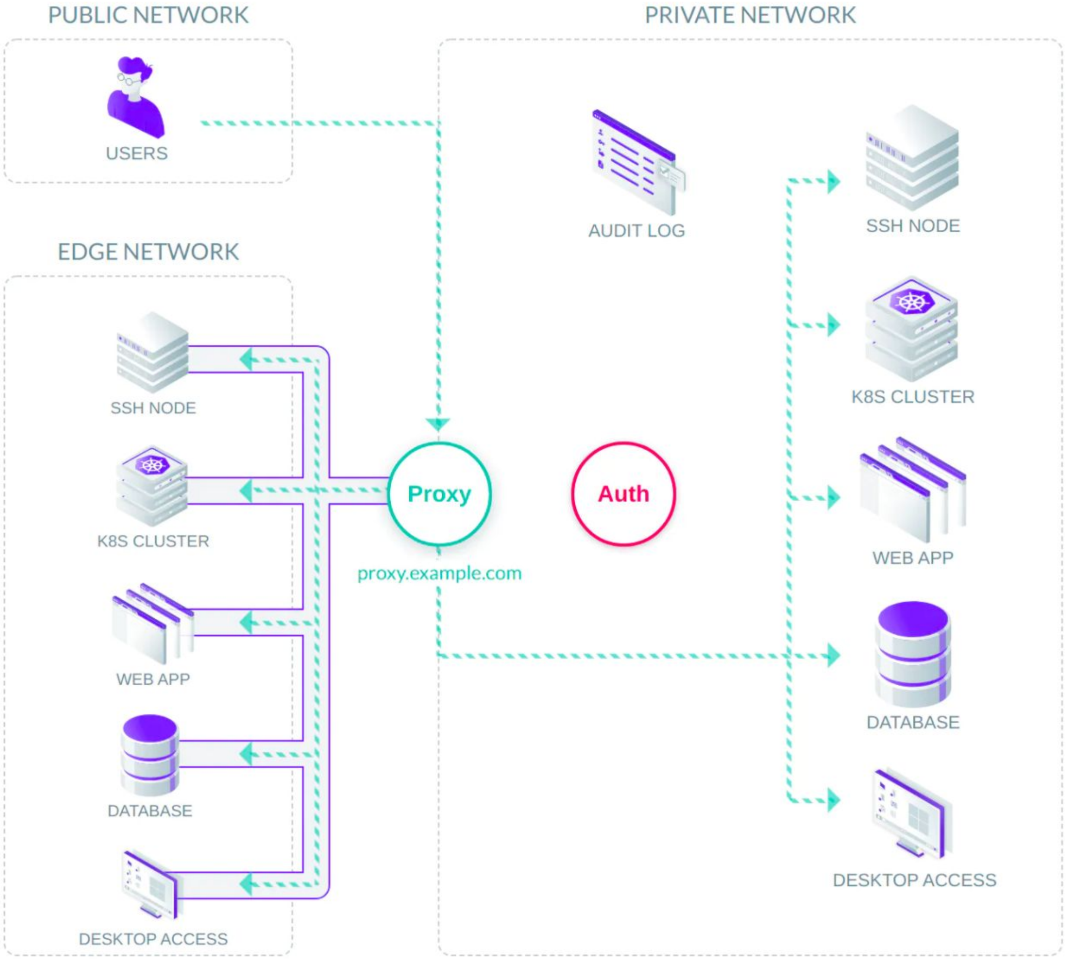
Software Engineer, Teleport

Teleport

What is Teleport?

Teleport is an open source infrastructure access platform that makes use of reverse tunneling to provide audited access to infrastructure (Kubernetes, SSH, DB, Web, Desktop).



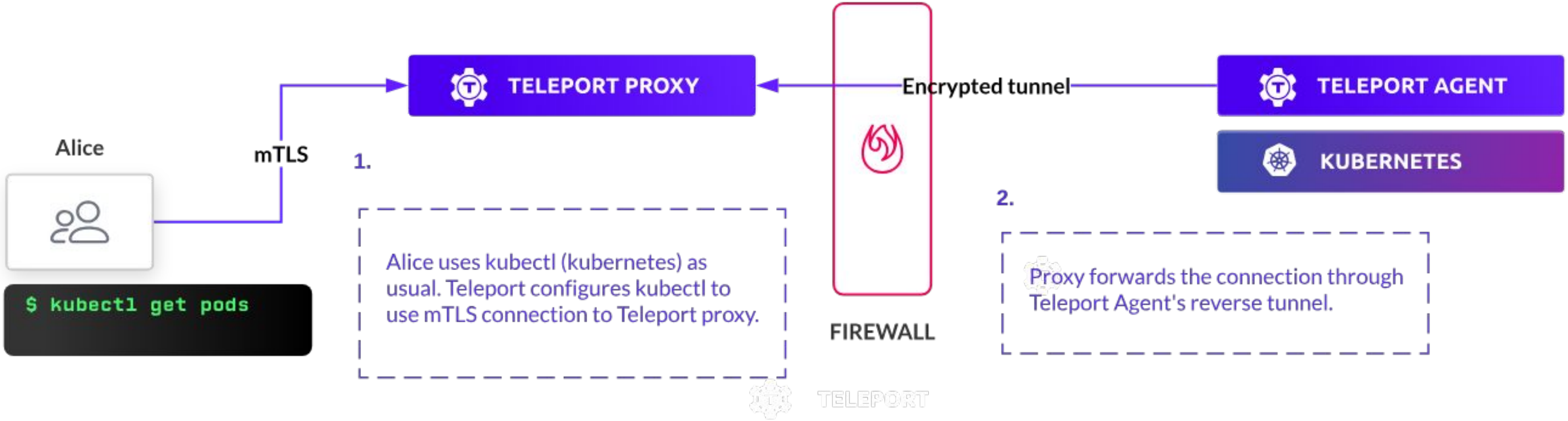


PROXY SERVICE

The proxy is the only service in a cluster visible to the outside world. All user connections for all supported protocols go through the proxy. The proxy also serves the Web UI and allows remote IoT nodes to establish reverse tunnels. Several proxies can be set up for high availability.

🟩 USER CONNECTIONS

🟥 AUDIT INFORMATION



```
$ kubectl get pods
```

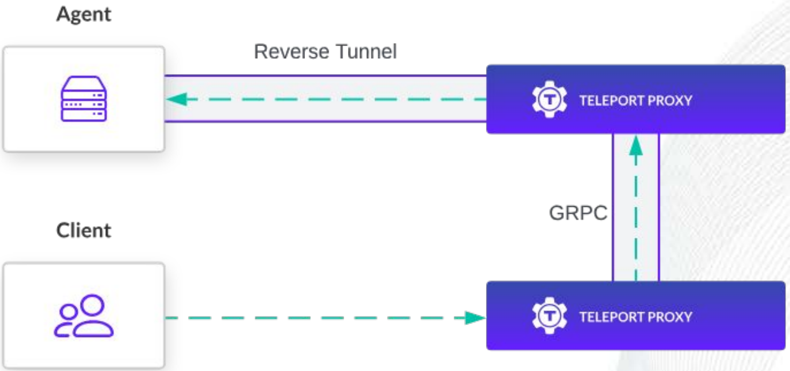
1. Alice uses kubectl (kubernetes) as usual. Teleport configures kubectl to use mTLS connection to Teleport proxy.

2. Proxy forwards the connection through Teleport Agent's reverse tunnel.

Teleport

Teleport Cloud

- Dedicated instance of Teleport **per customer**
  - **10k+** pods
  - **100k+** reverse tunnels
  - Tunnel disconnect → disrupted access
  - Globally available: clusters in **6 regions**
- Proxies peer to provide connectivity:



Gravity
MADE BY GRAVITATIONAL
↓
EKS

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
- **Deployment:** coordinated rollouts across regional clusters
- **Container networking:** proxy peering and auth traffic sent between regions

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels

- **Deployment:** coordinated rollouts across regional clusters

- **Container networking:** proxy peering and auth traffic sent between regions

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - ↳ Anycast - Global routing
  - ↳ NGINX (OSS) - Cluster ingress routing
- **Deployment:** coordinated rollouts across regional clusters
- **Container networking:** proxy peering and auth traffic sent between regions

Teleport

Needs

- **Ingress:** highly-available, **ultra-long-lived** reverse tunnels
  - ↳ Anycast?
    - Fewer issues with DNS — but **routing not stable enough**
- **Deployment:** coordinated rollouts across regional clusters
- **Container networking:** proxy peering and auth traffic sent between regions

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - ↳ ~~Anycast~~
  - ↳ NGINX (OSS)?
- **Deployment:** coordinated rollouts across regional clusters
- **Container networking:** proxy peering and auth traffic sent between regions

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - ↳ NGINX (OSS)?
    - ■ Supports ALPN routing – but **no in-process config reloading**
- **Deployment:** coordinated rollouts across regional clusters
- **Container networking:** proxy peering and auth traffic sent between regions

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - ↳ ~~Anycast~~
  - ↳ ~~NGINX (OSS)~~
- **Deployment:** coordinated rollouts across regional clusters
- **Container networking:** proxy peering and auth traffic sent between regions

Ingress

- Clients and agents must connect to closest Proxy pods
  - ↳ Route53 Latency Records
  - ↳ ExternalDNS Operator

Ingress

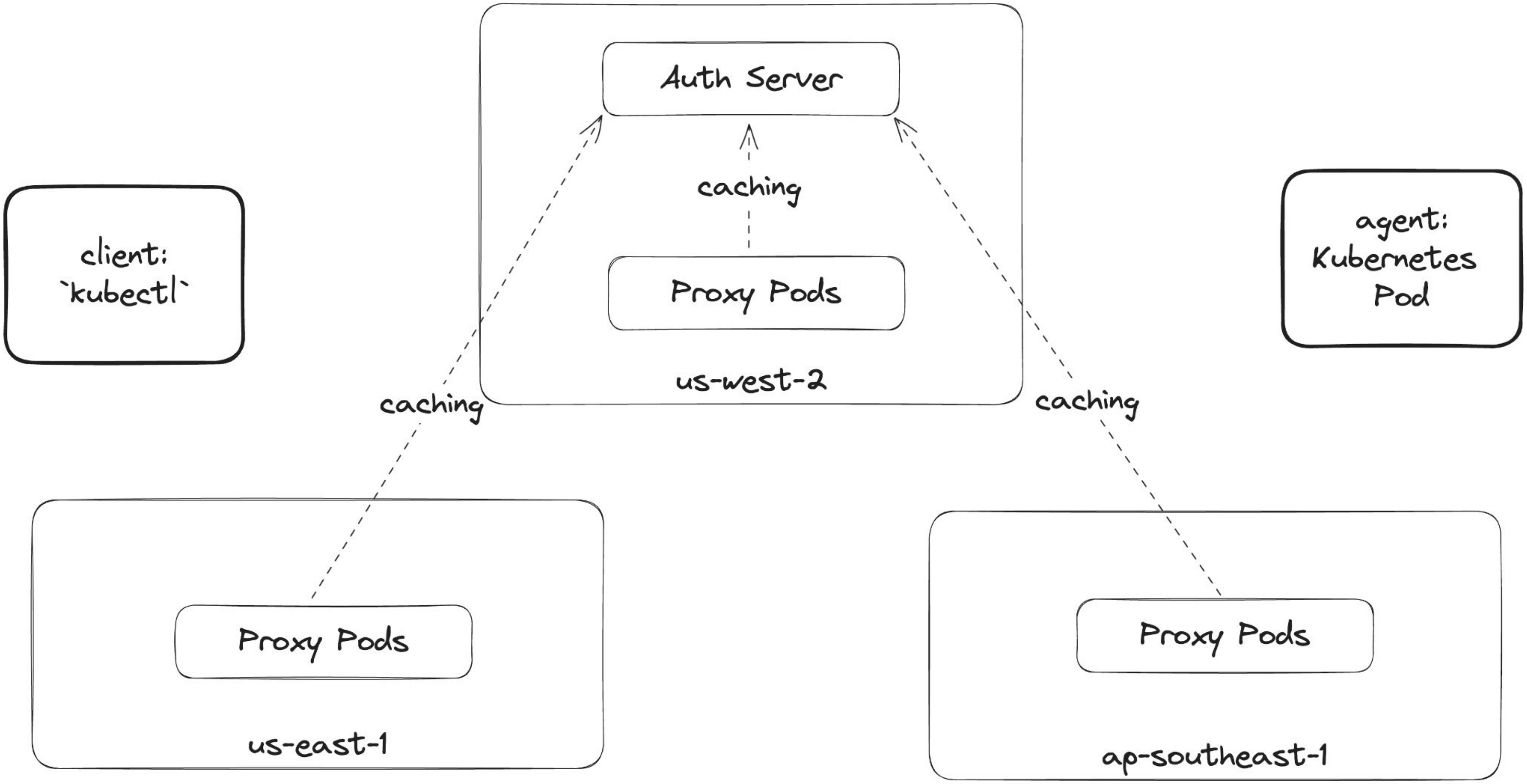
- Clients and agents must connect to closest Proxy pods
  - ↳ Route53 Latency Records
  - ↳ ExternalDNS Operator
- Reverse tunnels must not be interrupted
  - ↳ Stateless Network Load Balancers

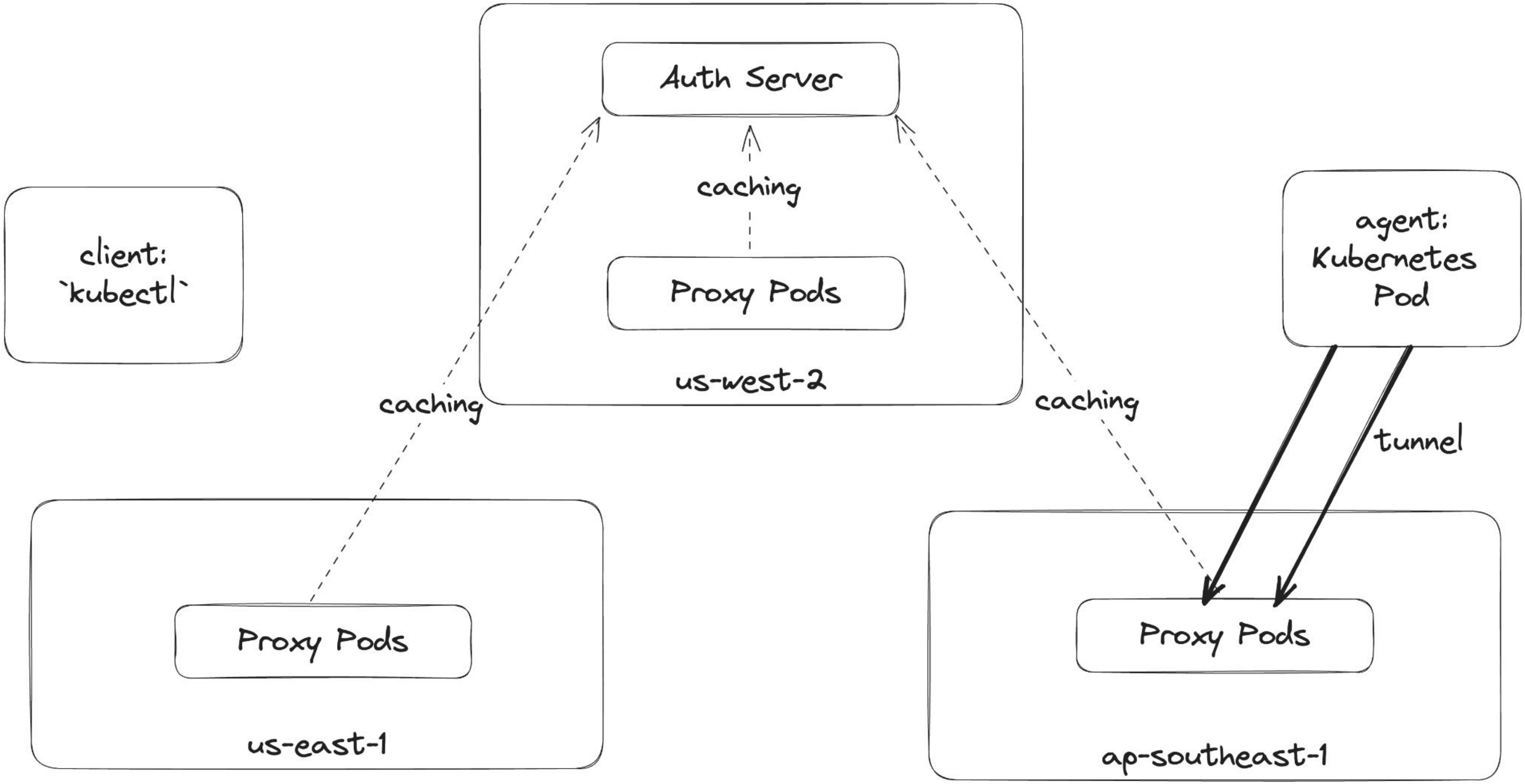
Ingress

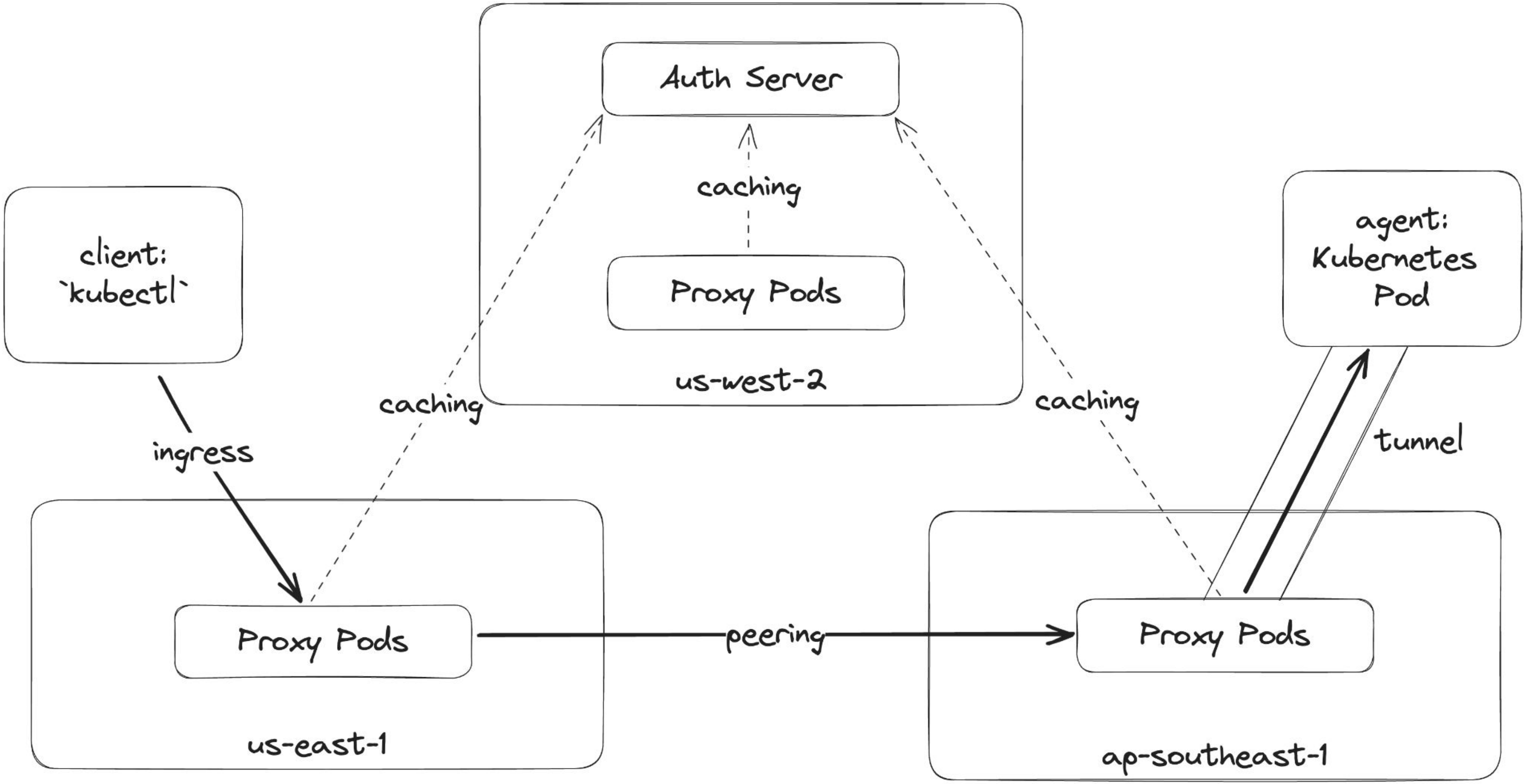
- Clients and agents must connect to closest Proxy pods
  - ↳ Route53 Latency Records
  - ↳ ExternalDNS Operator
- Reverse tunnels must not be interrupted
  - ↳ Stateless Network Load Balancers
- Proxy upgrades must not create downtime
  - ↳ Envoy Proxy with ALPN routing
  - ↳ Gateway API + Envoy Gateway
  - ↳ Clever hack with `minReadySeconds`

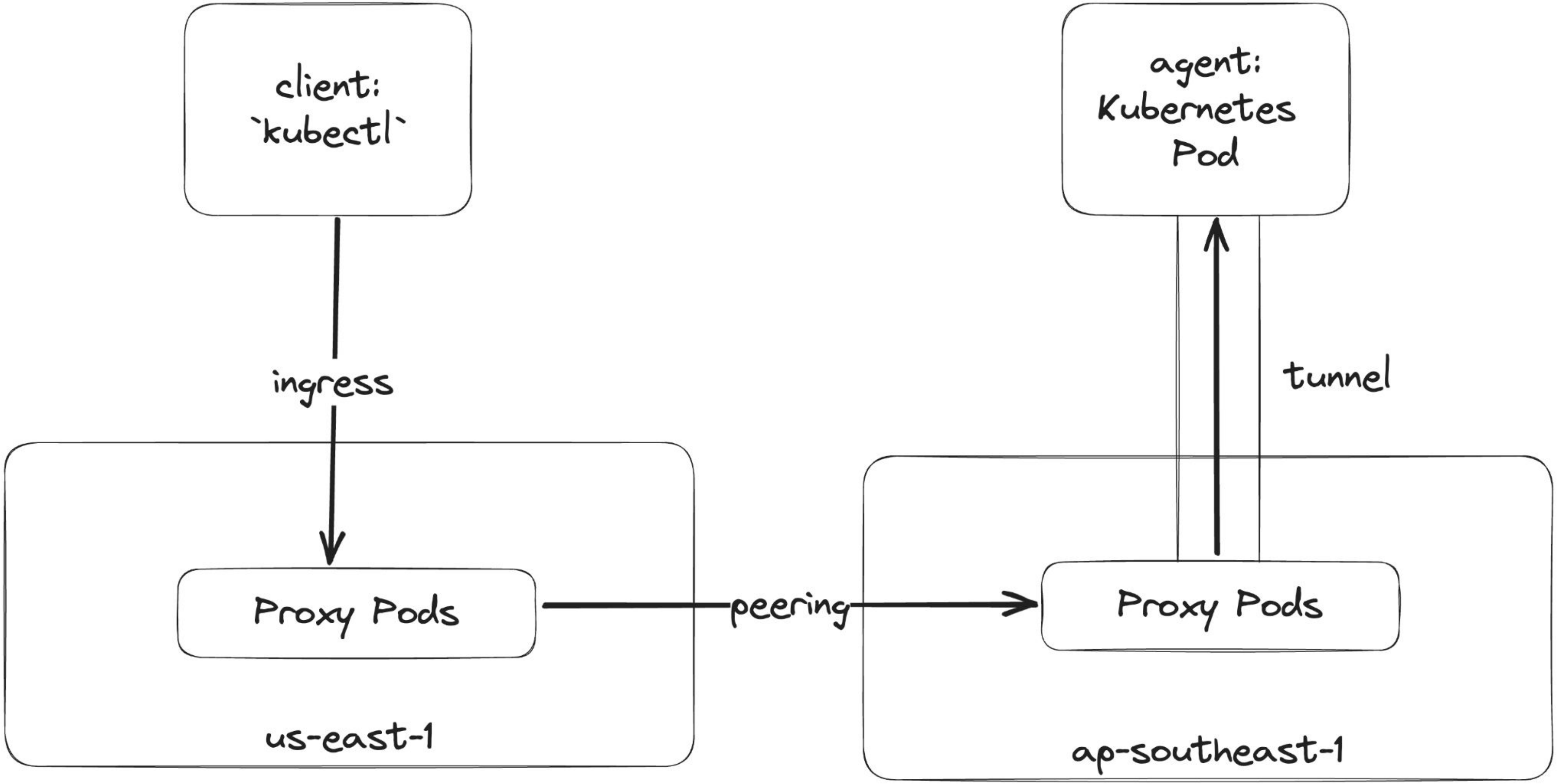
client: `kubectl`

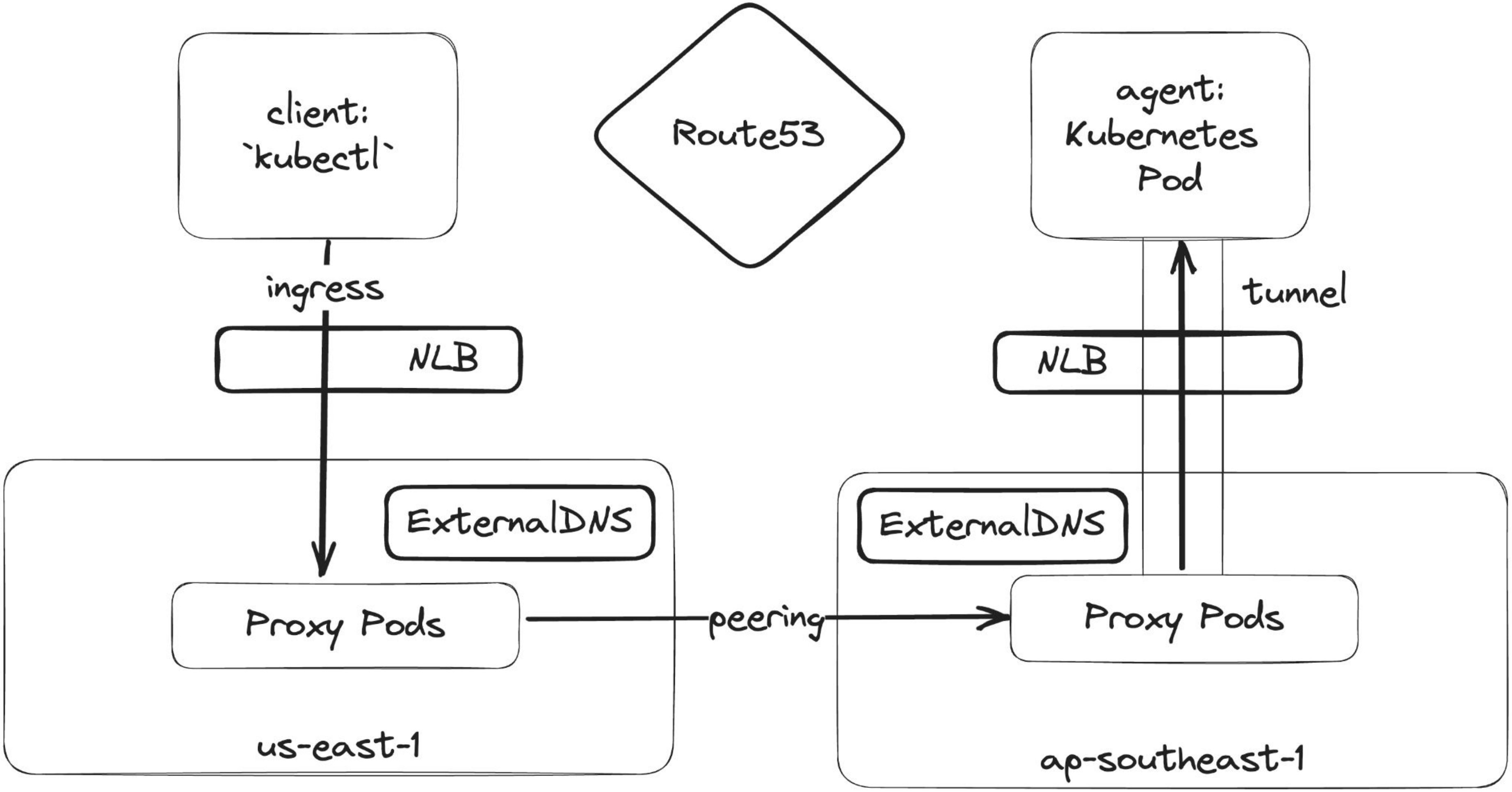
agent: Kubernetes Pod

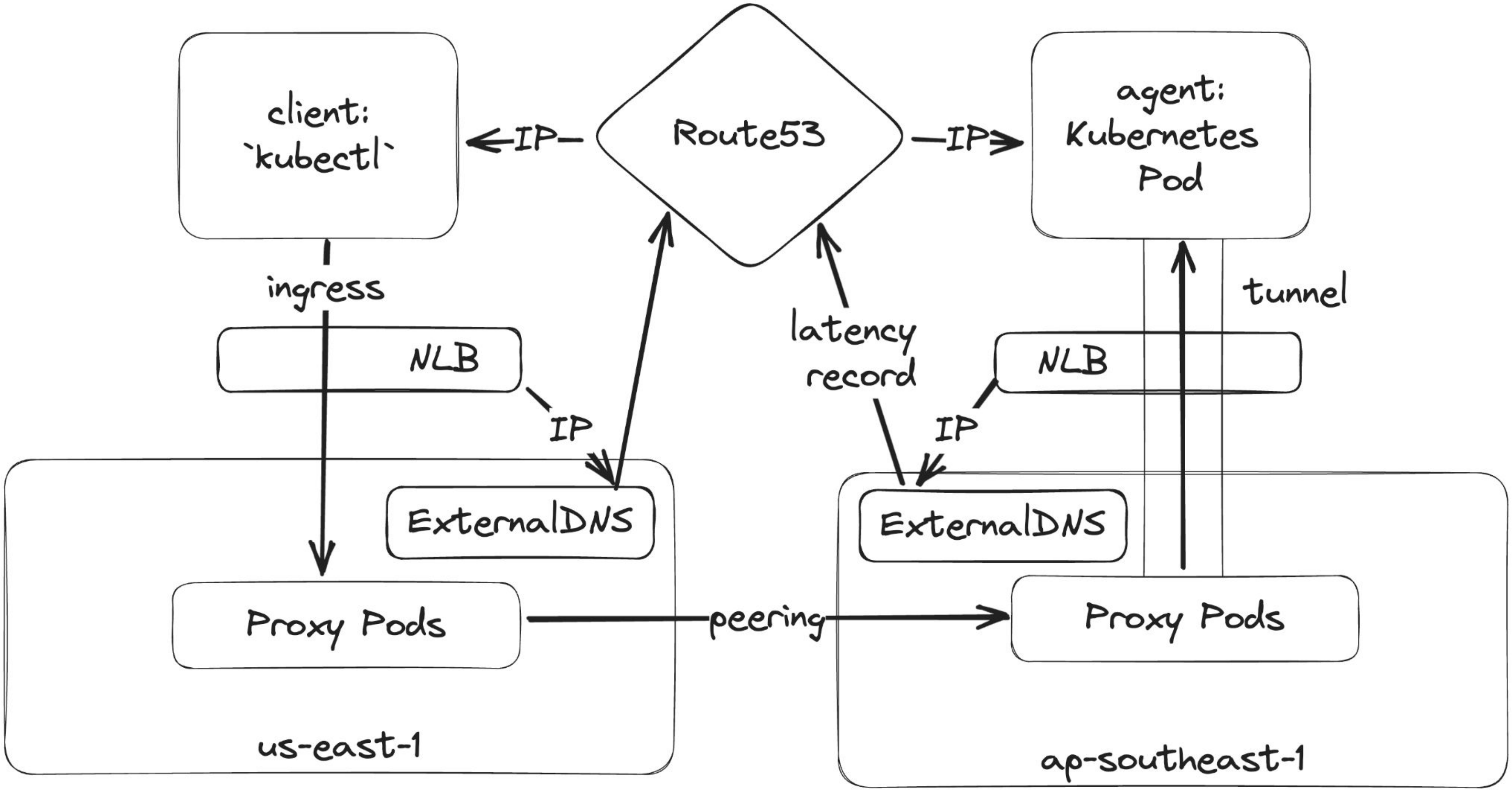

Auth Server
caching
Proxy Pods
us-west-2
client:
`kubectl`
agent:
Kubernetes
Pod
caching
caching
Proxy Pods
us-east-1
Proxy Pods
ap-southeast-1


client:
`kubectl`
Auth Server
caching
Proxy Pods
us-west-2
caching
caching
agent:
Kubernetes
Pod
tunnel
Proxy Pods
us-east-1
Proxy Pods
ap-southeast-1

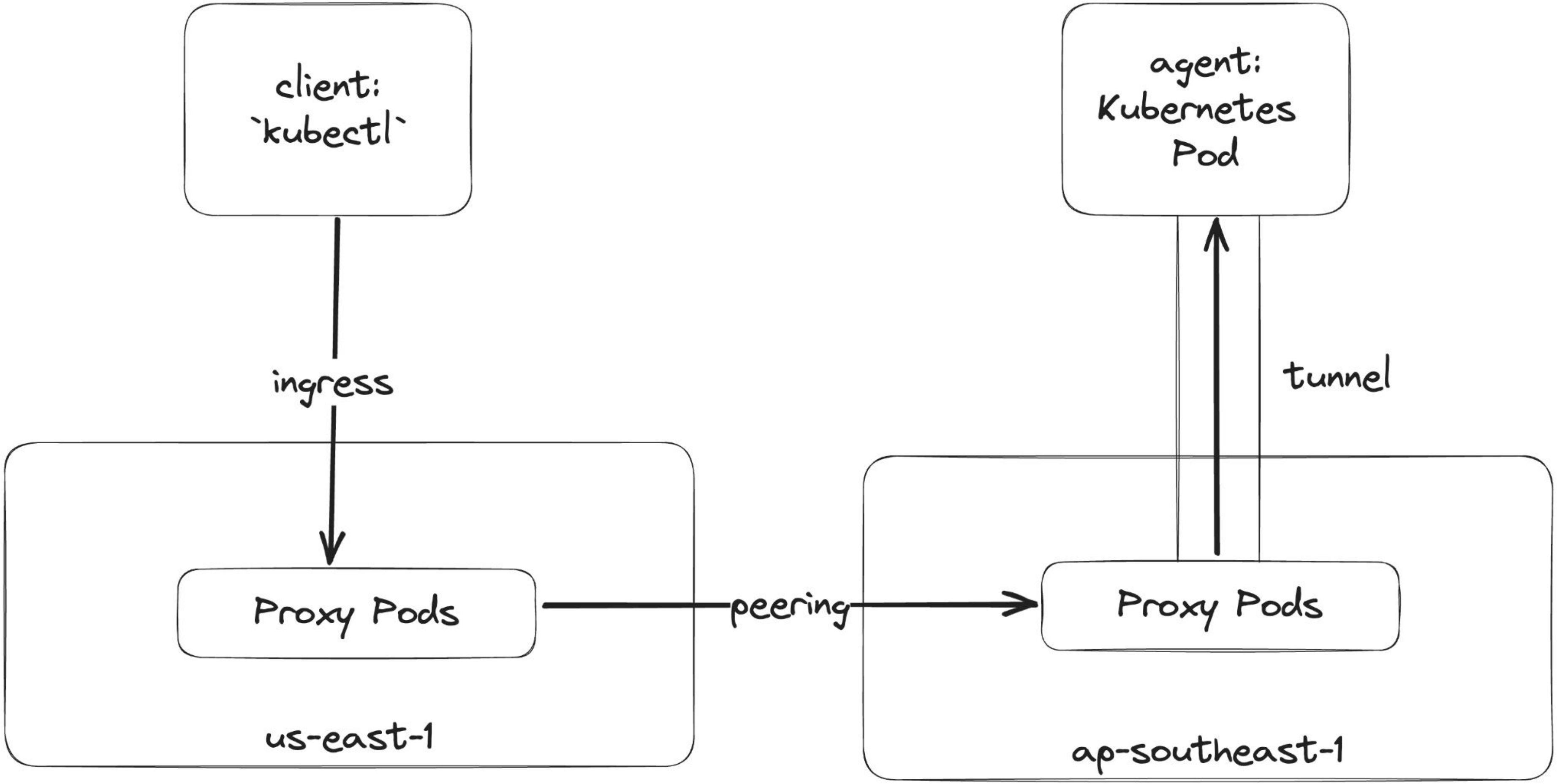

Auth Server
caching
Proxy Pods
us-west-2
client:
`kubectl`
agent:
Kubernetes
Pod
caching
caching
ingress
tunnel
Proxy Pods
peering
Proxy Pods
us-east-1
ap-southeast-1

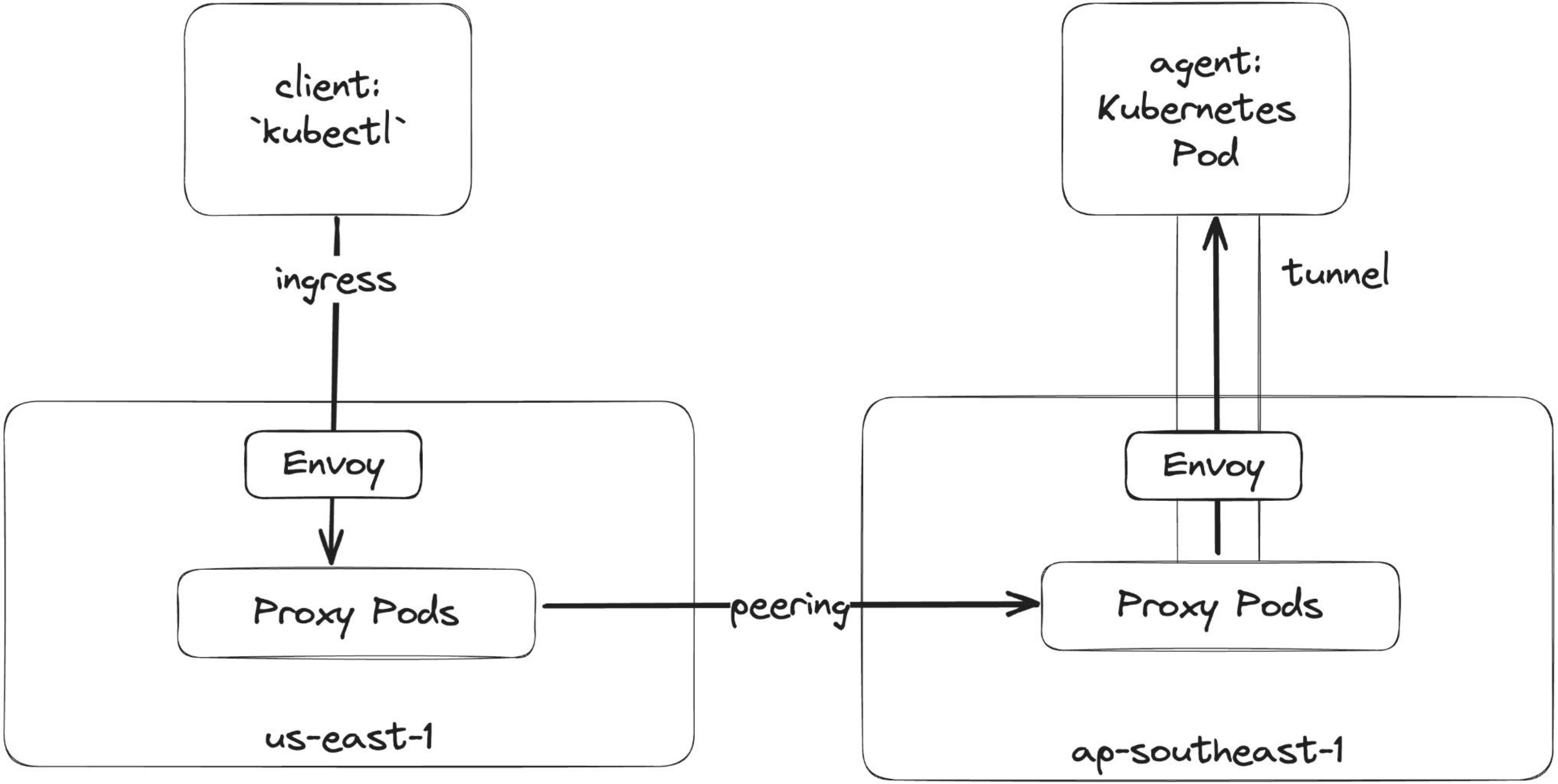

client:
`kubectl`
agent:
Kubernetes
Pod
ingress
tunnel
Proxy Pods
peering
Proxy Pods
us-east-1
ap-southeast-1

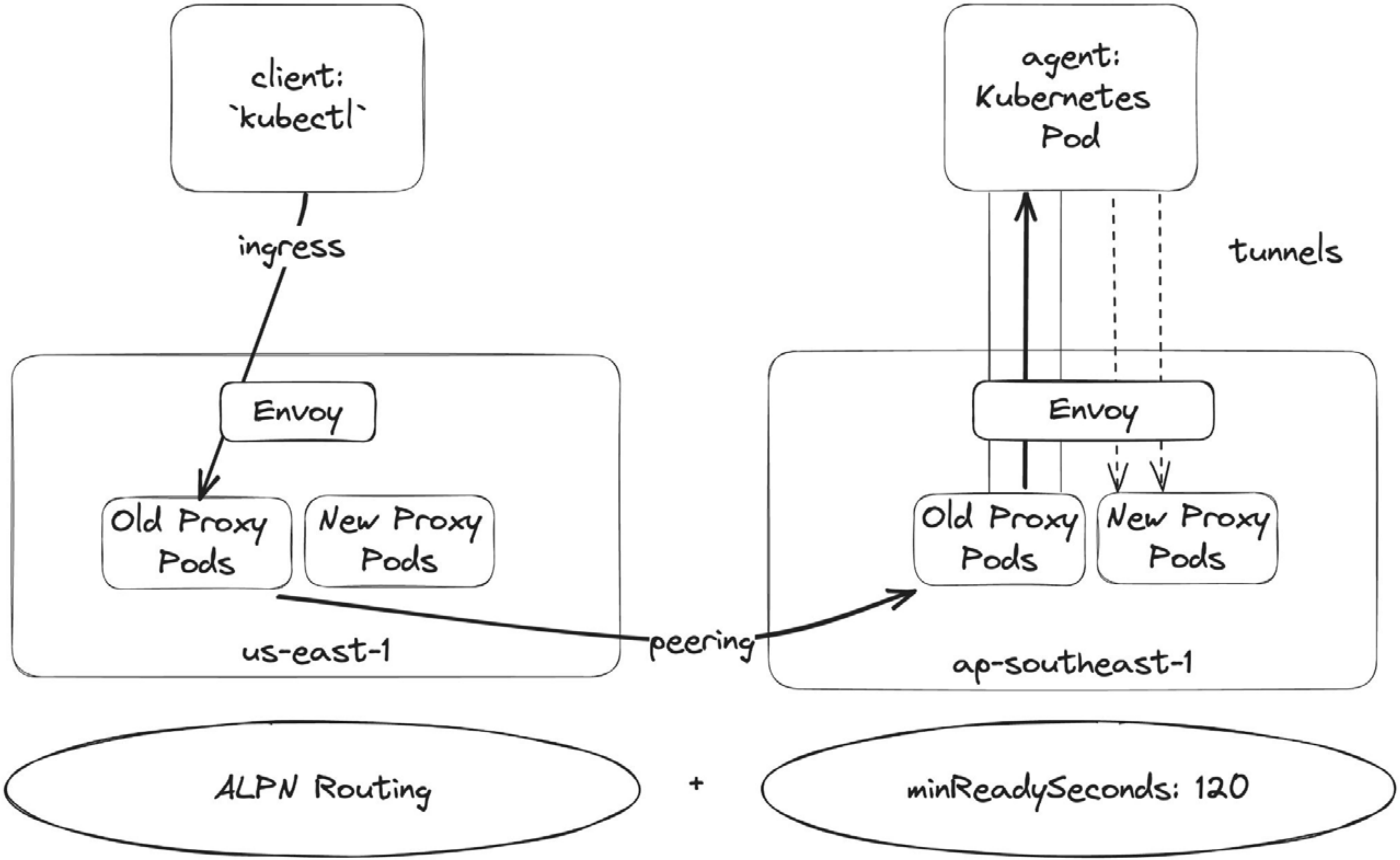

client:
`kubectl`
Route53
agent:
Kubernetes
Pod
ingress
tunnel
NLB
NLB
ExternalDNS
ExternalDNS
Proxy Pods
peering
Proxy Pods
us-east-1
ap-southeast-1

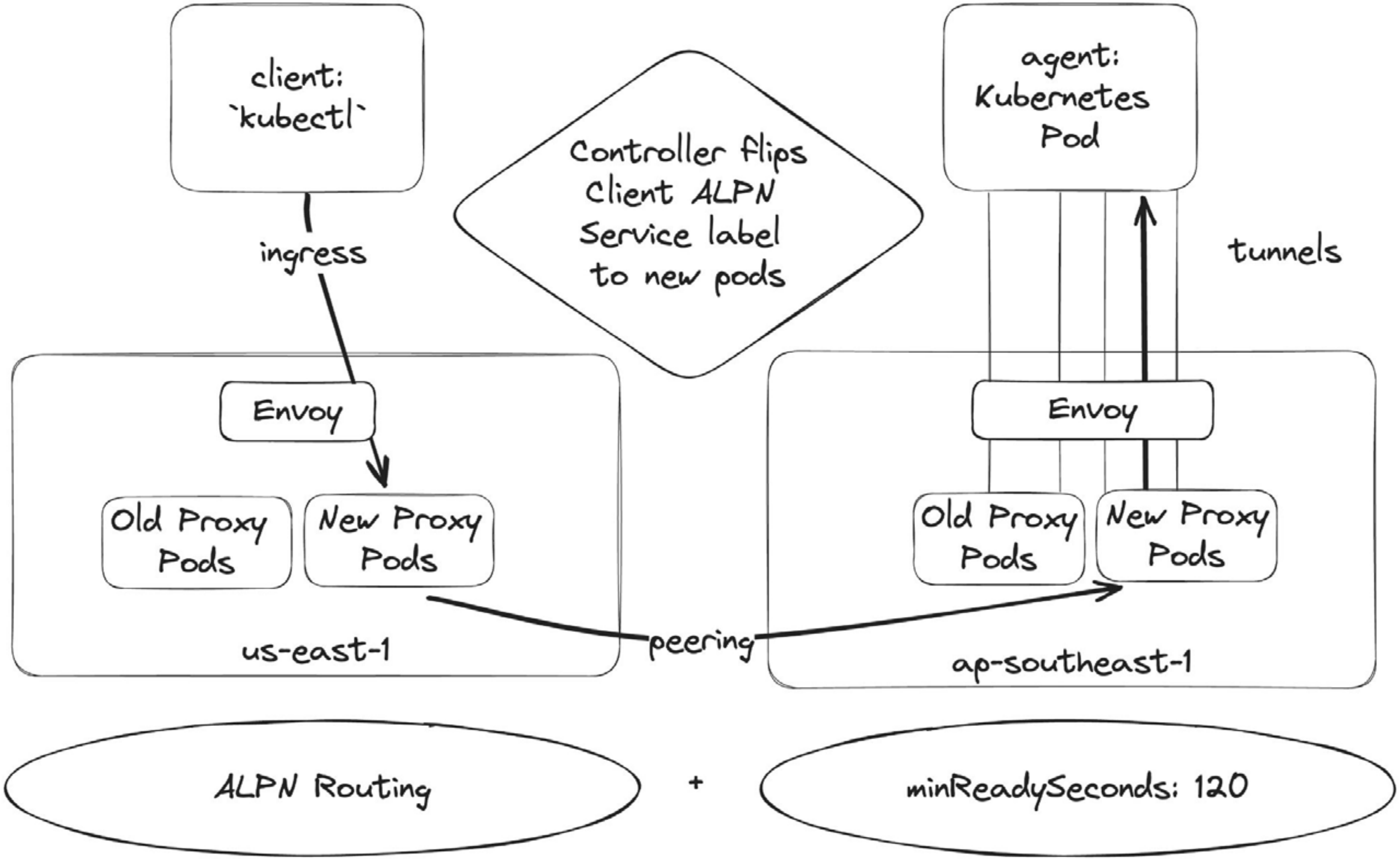

client:
`kubectl`
IP
Route53
IP
agent:
Kubernetes
Pod
ingress
latency
record
tunnel
NLB
NLB
IP
IP
ExternalDNS
ExternalDNS
Proxy Pods
peering
Proxy Pods
us-east-1
ap-southeast-1

Zero-Downtime


client:
`kubectl`
agent:
Kubernetes
Pod
ingress
tunnel
Proxy Pods
peering
Proxy Pods
us-east-1
ap-southeast-1


client:
`kubectl`
agent:
Kubernetes
Pod
ingress
tunnel
Envoy
Envoy
Proxy Pods
peering
Proxy Pods
us-east-1
ap-southeast-1


client:
`kubectl`
ingress
agent:
Kubernetes
Pod
tunnels
Envoy
Old Proxy Pods
New Proxy Pods
us-east-1
peering
Envoy
Old Proxy Pods
New Proxy Pods
ap-southeast-1
ALPN Routing
+
minReadySeconds: 120


client:
`kubectl`
ingress
Controller flips
Client ALPN
Service label
to new pods
agent:
Kubernetes
Pod
tunnels
Envoy
Old Proxy
Pods
New Proxy
Pods
us-east-1
peering
Envoy
Old Proxy
Pods
New Proxy
Pods
ap-southeast-1
ALPN Routing
+
minReadySeconds: 120

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
- **Container networking:** proxy peering and auth traffic sent between regions

Teleport

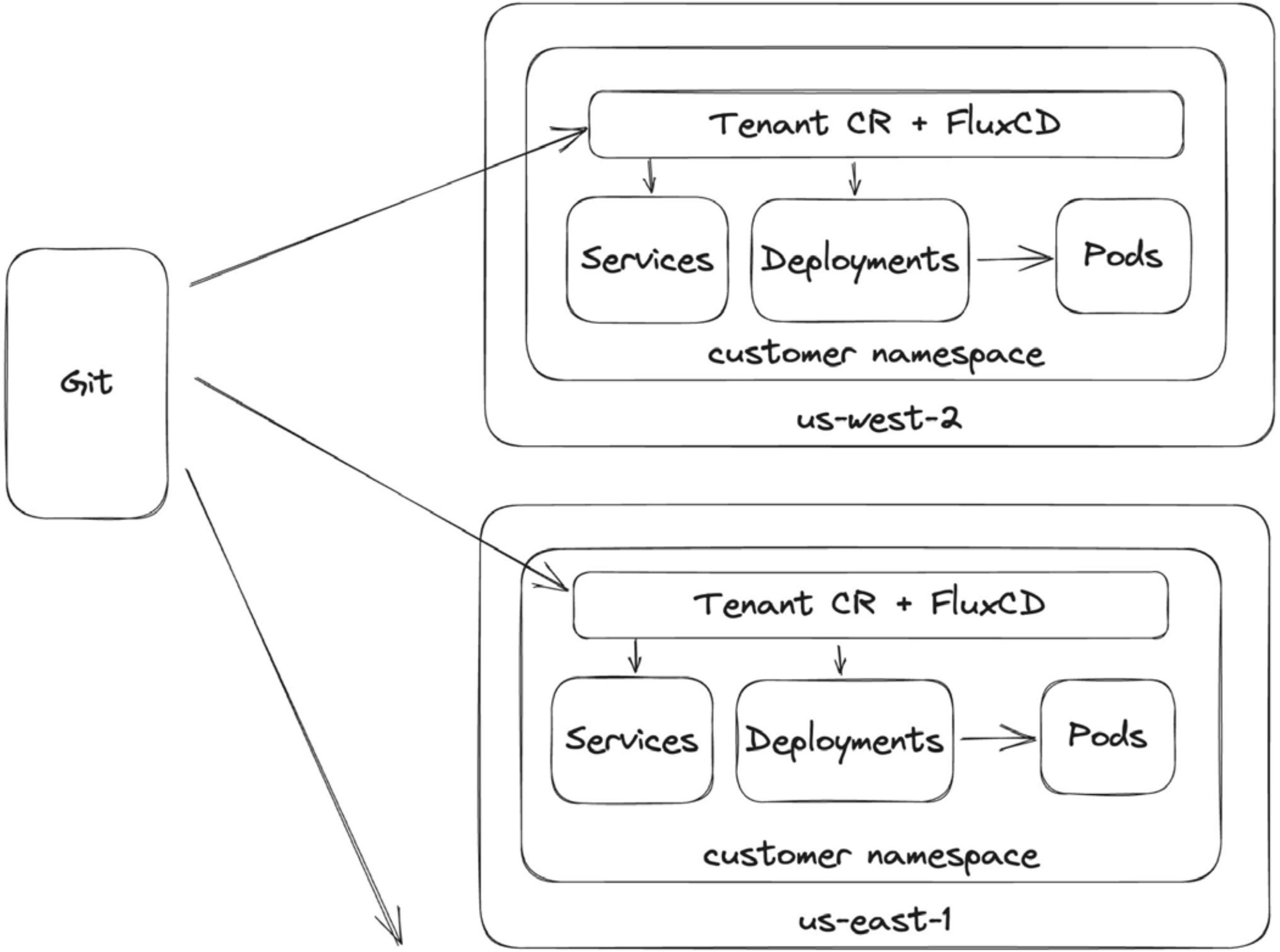
Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
- **Container networking:** proxy peering and auth traffic sent between regions

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - ↳ Gitops (e.g., FluxCD)?
- **Container networking:** proxy peering and auth traffic sent between regions


Git
Tenant CR + FluxCD
Services
Deployments
Pods
customer namespace
us-west-2
Tenant CR + FluxCD
Services
Deployments
Pods
customer namespace
us-east-1

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  ↳ Gitops (e.g., FluxCD)?
- **Container networking:** proxy peering and auth traffic sent between regions

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - ↳ Gitops (e.g., FluxCD)?
    - Prefer Postgres (RDS) for customer data
- **Container networking:** proxy peering and auth traffic sent between regions

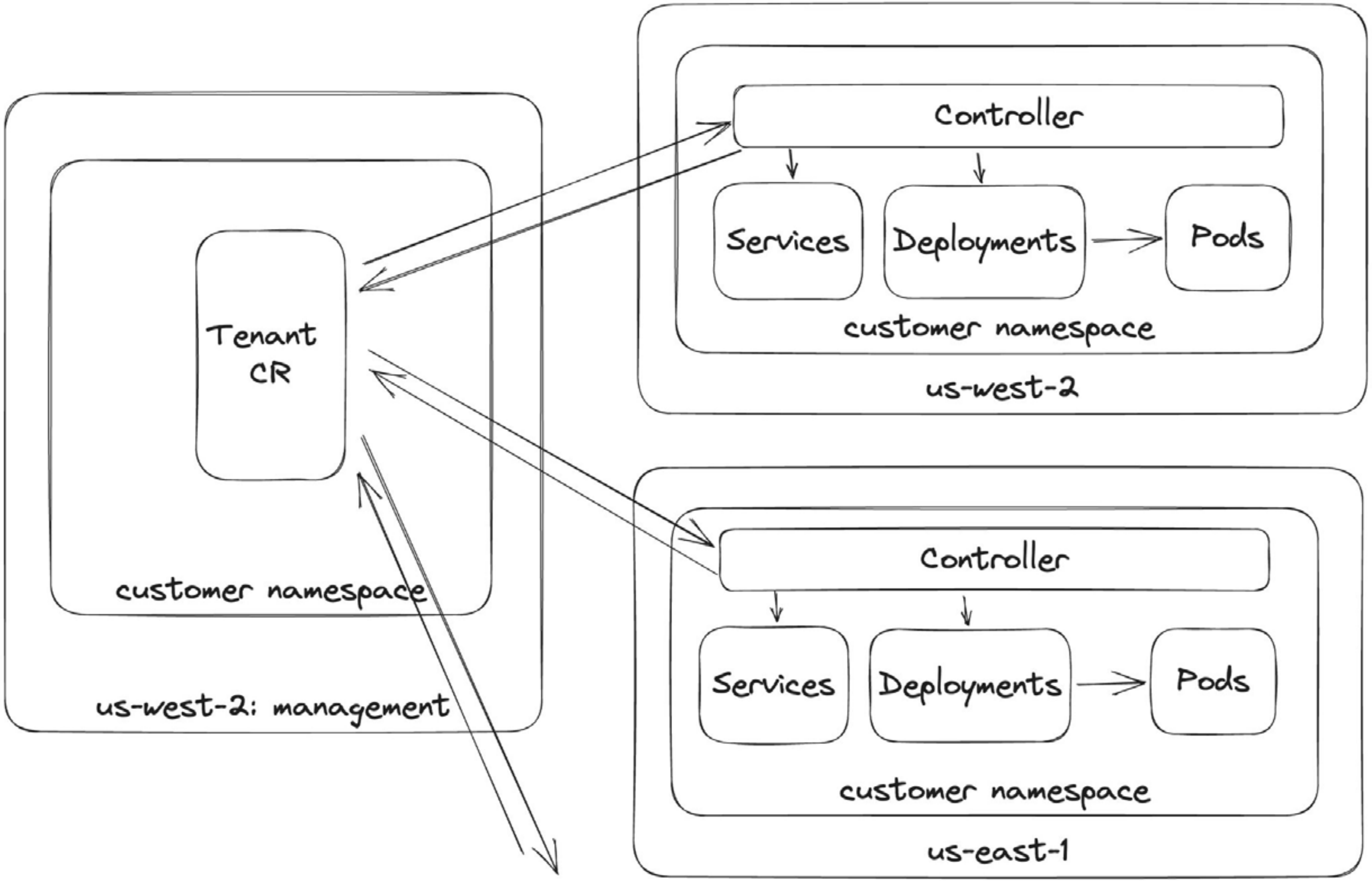
Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - ↳ Gitops (e.g., FluxCD)?
    - Unidirectional, difficult to orchestrate multi-region deployment
- **Container networking:** proxy peering and auth traffic sent between regions

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - ↳ Cross-cluster reconcilers?
- **Container networking:** proxy peering and auth traffic sent between regions


Controller
Services
Deployments
Pods
customer namespace
us-west-2
Tenant CR
customer namespace
us-west-2: management
Controller
Services
Deployments
Pods
customer namespace
us-east-1

Teleport

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - ↳ Cross-cluster reconcilers + shared custom resource?
    - Single point-of-failure, multiple writers, etc.
- **Container networking:** proxy peering and auth traffic sent between regions

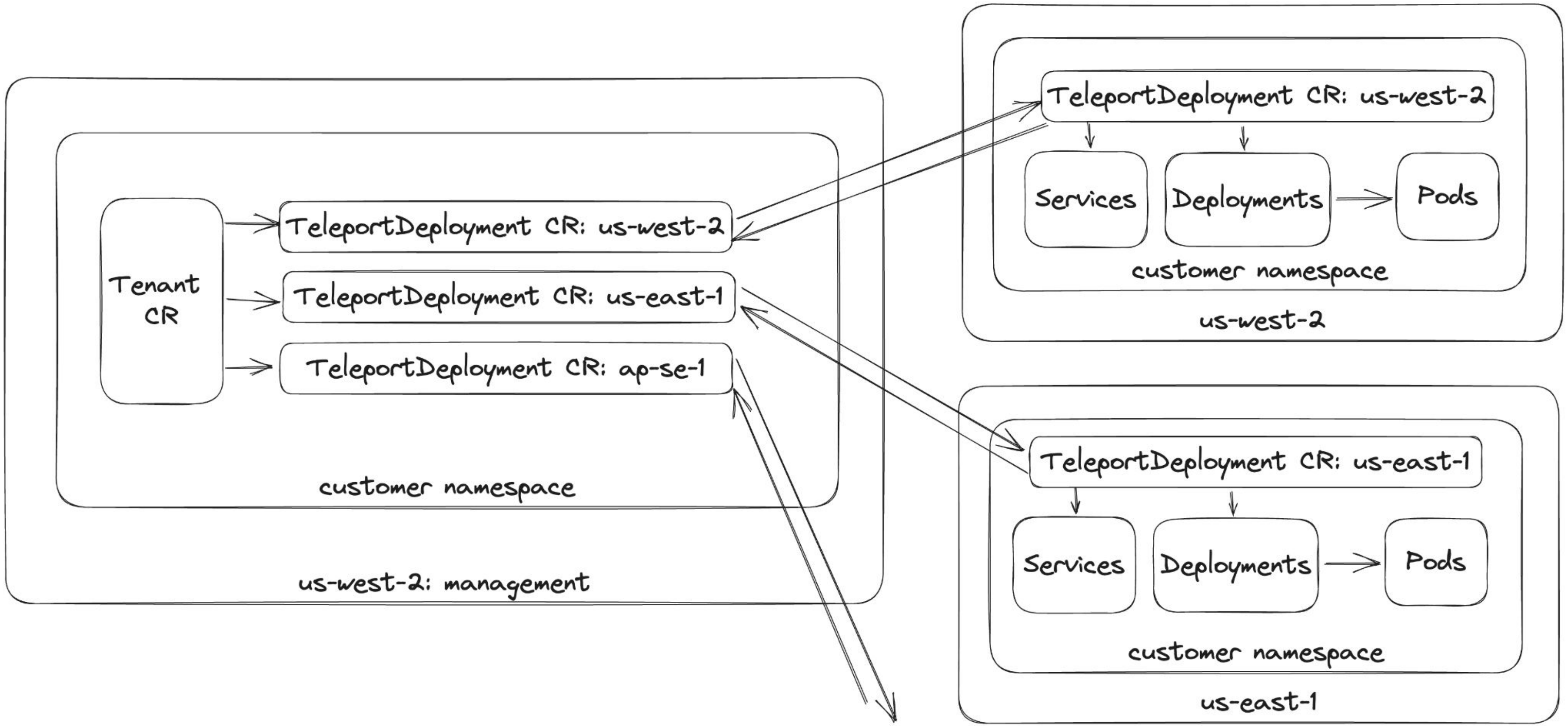
Teleport

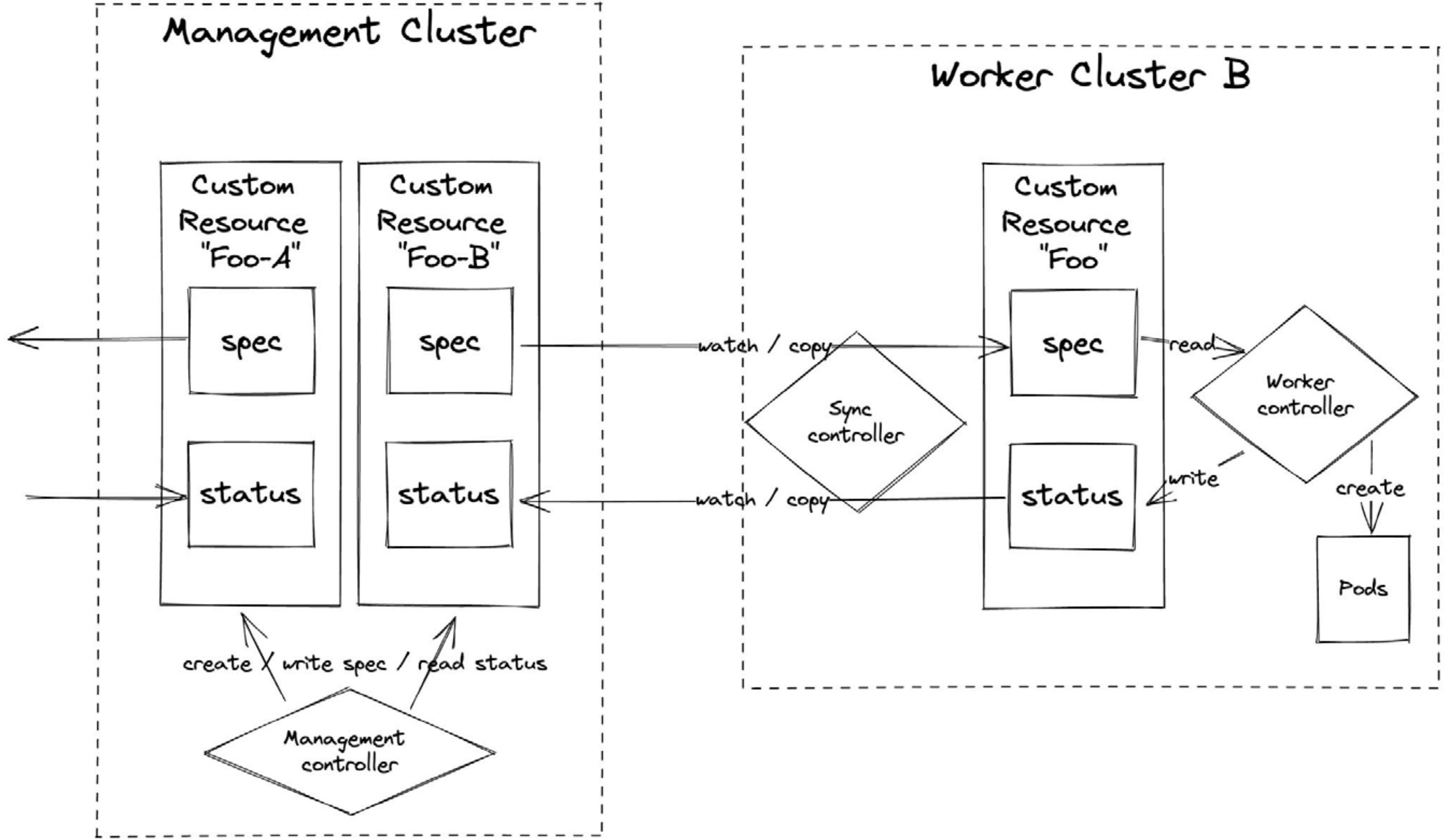
Needs

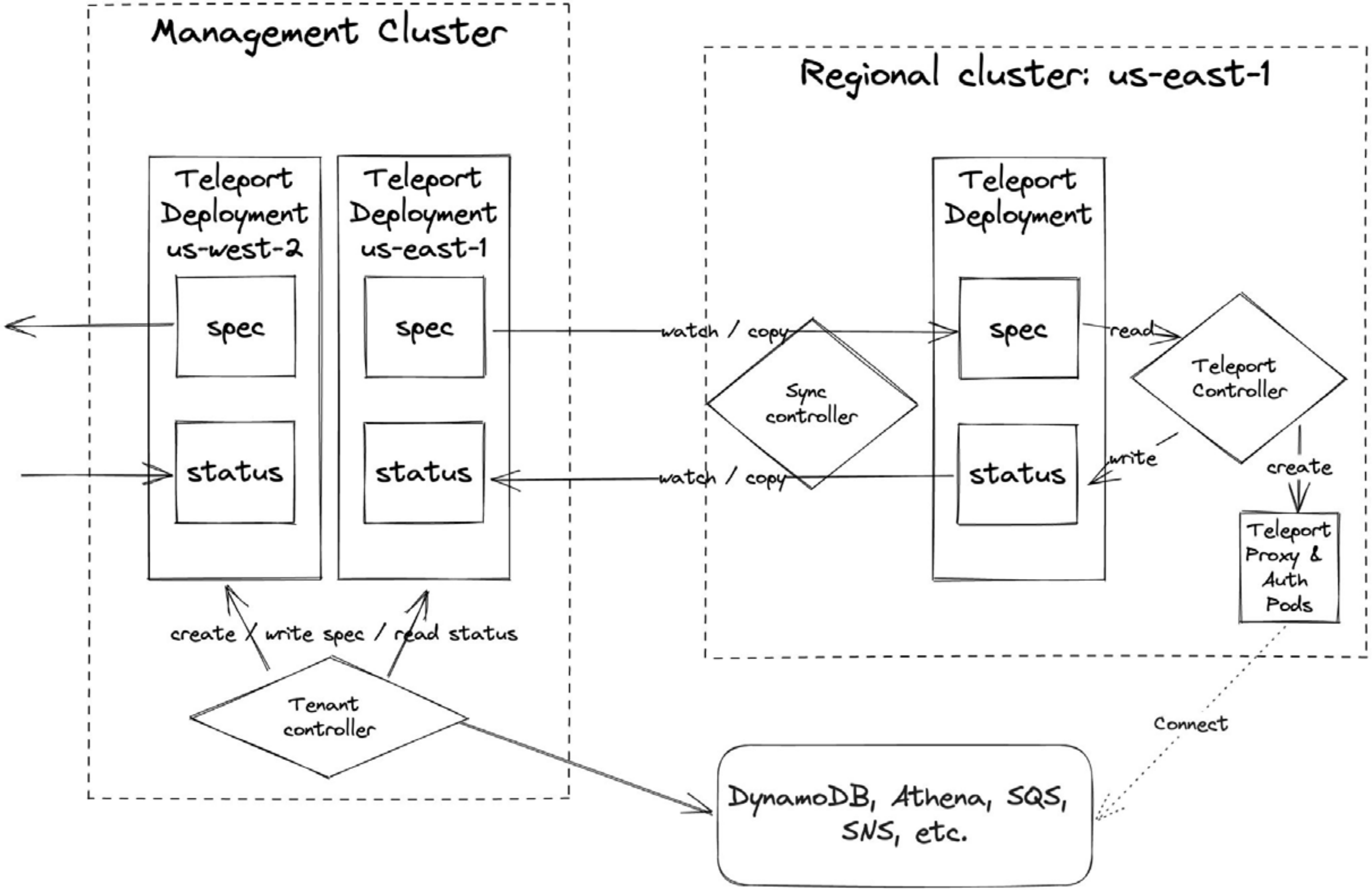
- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - ↳ Kubefed?
    - Ideal model, but no longer active…
- **Container networking:** proxy peering and auth traffic sent between regions

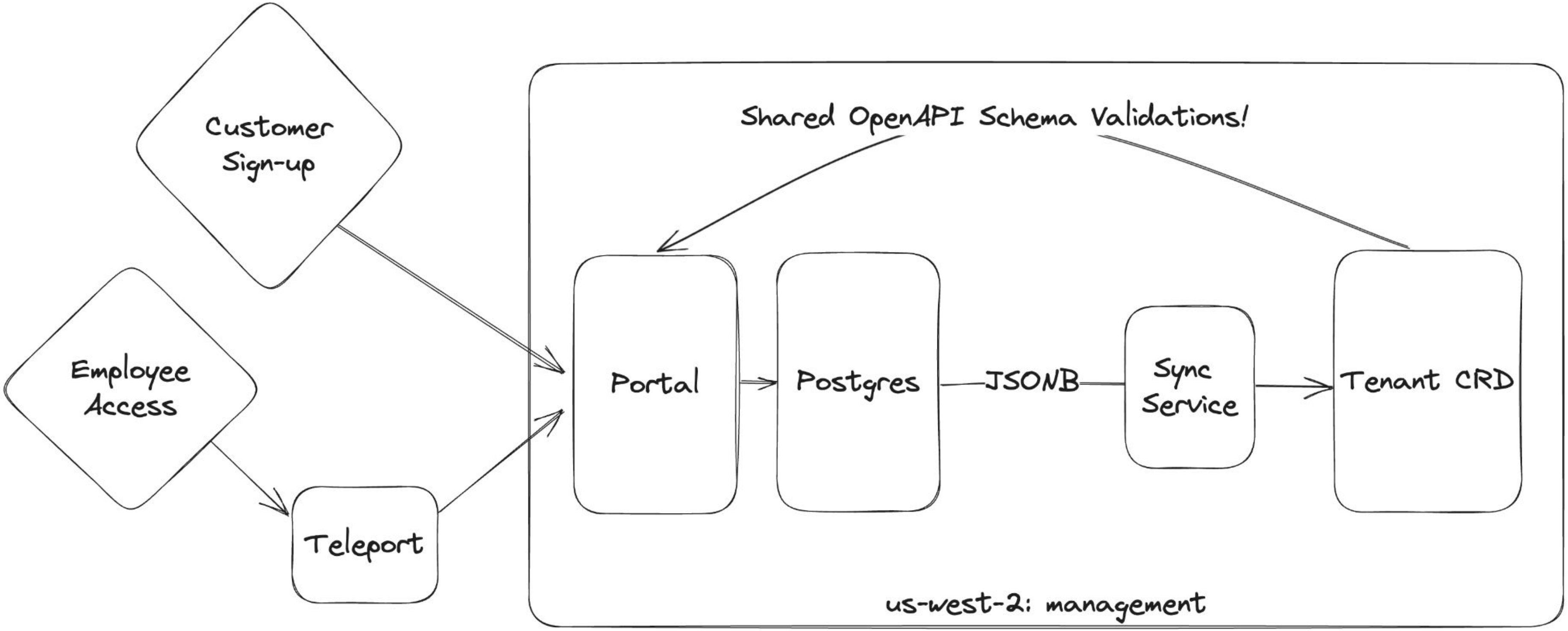
Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  ↳ Sync Controller (now OSS)!
- **Container networking:** proxy peering and auth traffic sent between regions


TeleportDeployment CR: us-west-2
Services
Deployments
Pods
customer namespace
us-west-2
Tenant CR
TeleportDeployment CR: us-west-2
TeleportDeployment CR: us-east-1
TeleportDeployment CR: ap-se-1
customer namespace
us-west-2: management
TeleportDeployment CR: us-east-1
Services
Deployments
Pods
customer namespace
us-east-1


Management Cluster
Worker Cluster B
Custom Resource "Foo-A"
spec
status
Custom Resource "Foo-B"
spec
status
create / write spec / read status
Management controller
watch / copy
Sync controller
watch / copy
Custom Resource "Foo"
spec
status
read
Worker controller
write
create
Pods


Management Cluster
Teleport Deployment us-west-2
spec
status
Teleport Deployment us-east-1
spec
status
create / write spec / read status
Tenant controller
Regional cluster: us-east-1
watch / copy
Sync controller
watch / copy
Teleport Deployment
spec
status
read
Teleport Controller
write
create
Teleport Proxy & Auth Pods
Connect
DynamoDB, Athena, SQS, SNS, etc.


Customer Sign-up
Employee Access
Teleport
Shared OpenAPI Schema Validations!
Portal
Postgres
JSONB
Sync Service
Tenant CRD
us-west-2: management

Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - Configuration storage: gRPC + Postgres + JSONB
  - Cross-cluster operation: Sync Controller (inspired by Kubefed)
- **Container networking:** proxy peering and auth traffic between regions

Teleport

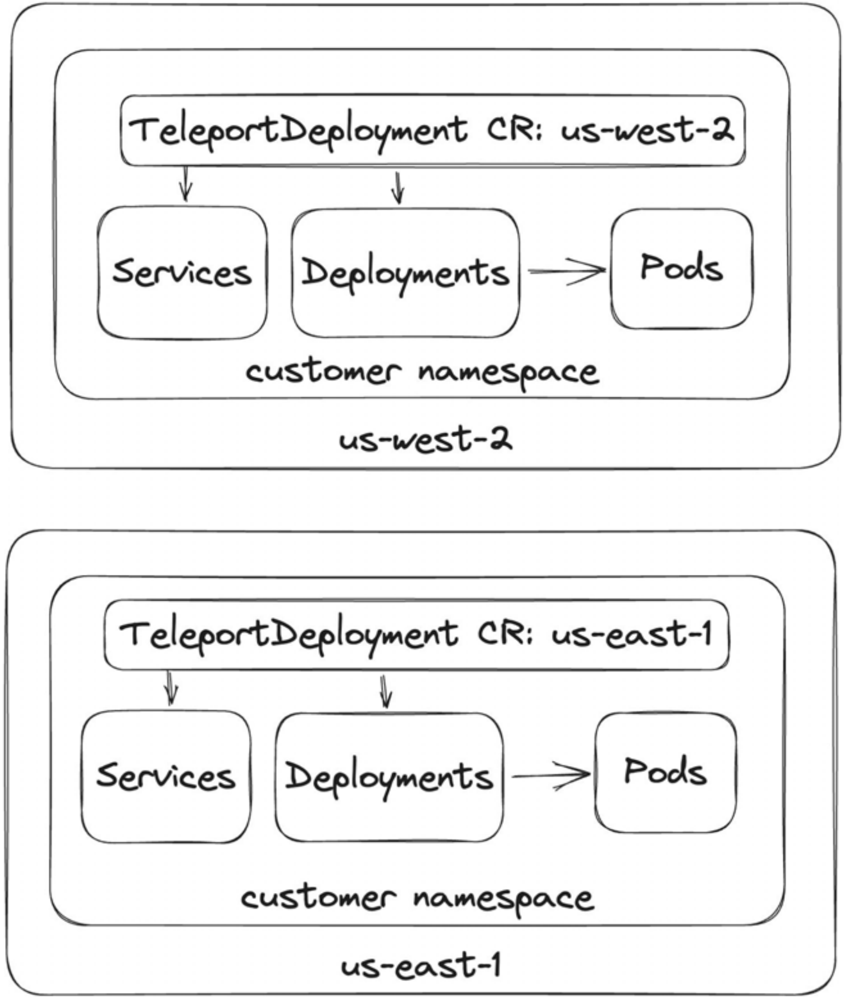
Needs

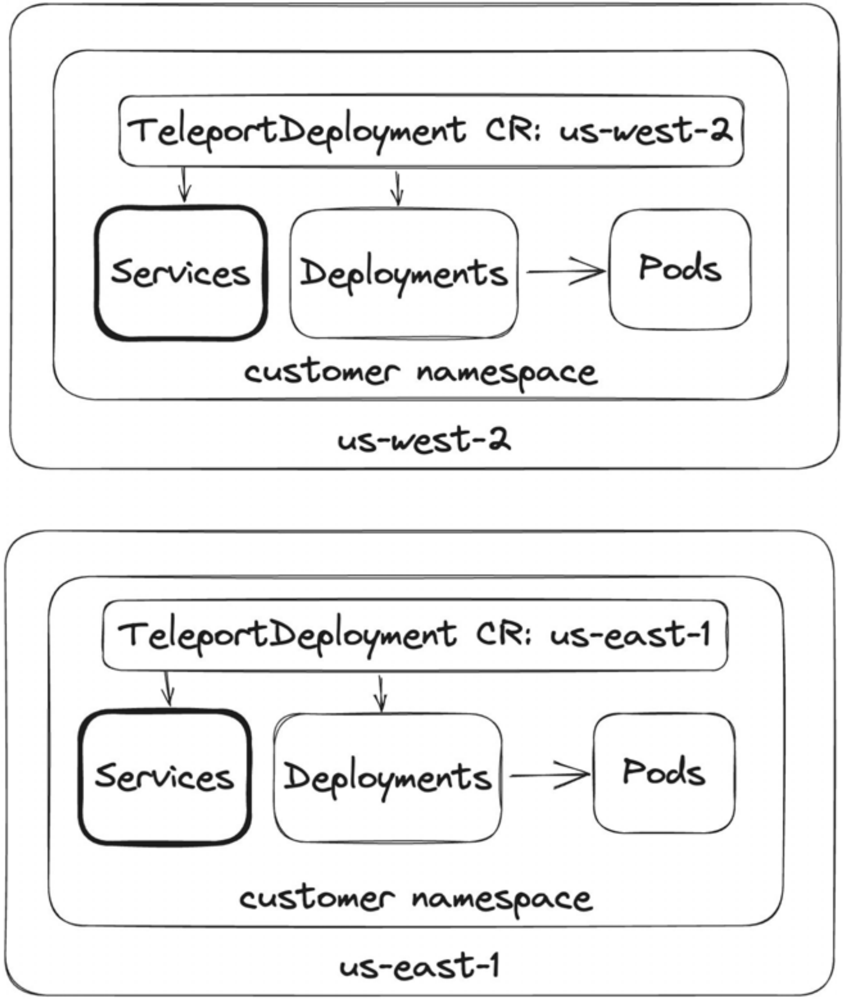
- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - Configuration storage: gRPC + Postgres + JSONB
  - Cross-cluster operation: Sync Controller (inspired by Kubefed)
- **Container networking:** proxy peering and auth traffic between regions

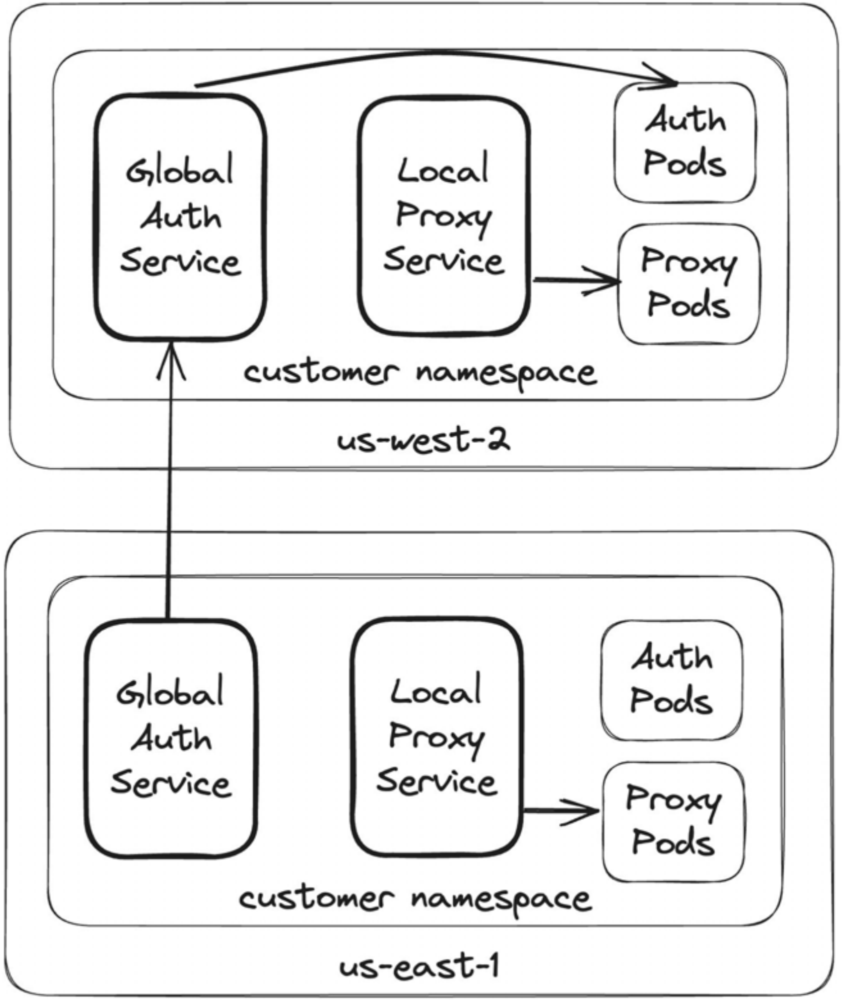
Teleport

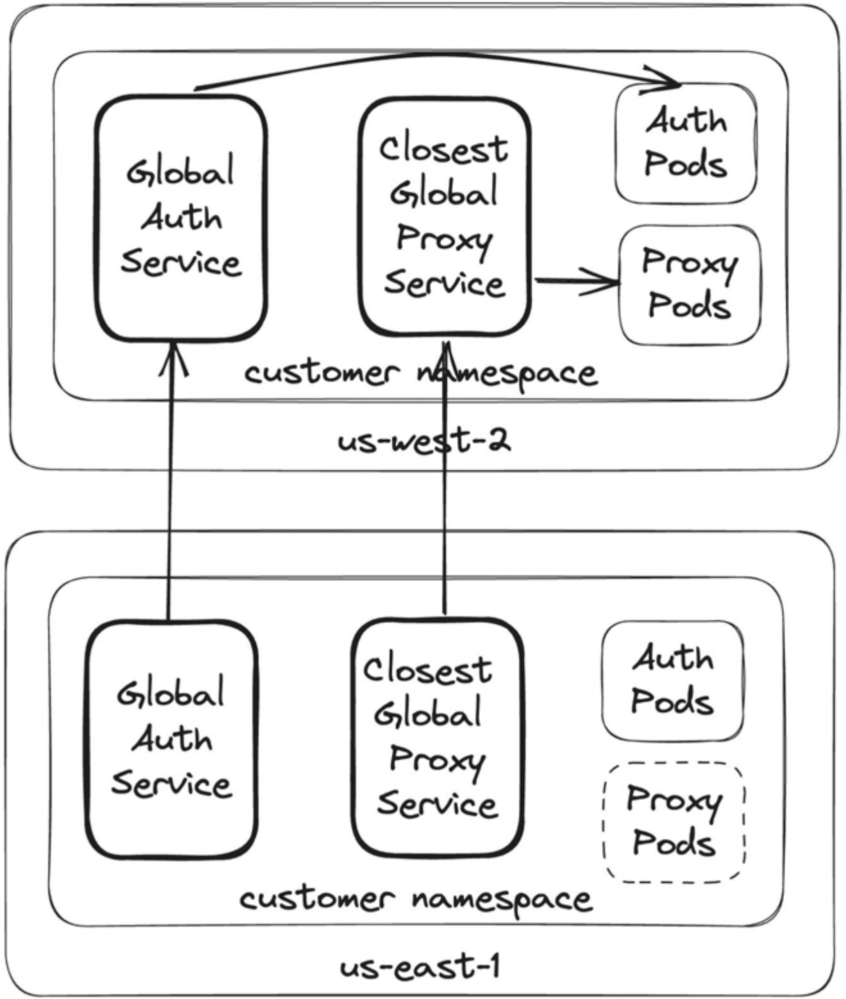
Needs

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - Configuration storage: gRPC + Postgres + JSONB
  - Cross-cluster operation: Sync Controller (inspired by Kubefed)
- **Container networking:** proxy peering and auth traffic between regions
  - Cilium Global Services with dedicated etcd


TeleportDeployment CR: us-west-2
Services
Deployments
Pods
customer namespace
us-west-2
TeleportDeployment CR: us-east-1
Services
Deployments
Pods
customer namespace
us-east-1


TeleportDeployment CR: us-west-2
Services
Deployments
Pods
customer namespace
us-west-2
TeleportDeployment CR: us-east-1
Services
Deployments
Pods
customer namespace
us-east-1


Global Auth Service
Local Proxy Service
Auth Pods
Proxy Pods
customer namespace
us-west-2
Global Auth Service
Local Proxy Service
Auth Pods
Proxy Pods
customer namespace
us-east-1


Global Auth Service
Closest Global Proxy Service
Auth Pods
Proxy Pods
customer namespace
us-west-2
Global Auth Service
Closest Global Proxy Service
Auth Pods
Proxy Pods
customer namespace
us-east-1

User Journey

- **Ingress:** highly-available, ultra-long-lived reverse tunnels
  - Envoy Proxy / Envoy Gateway / Gateway API
  - ALPN routing to separate tunnel and client connections to 443
  - minReadySeconds + Service updates => minimal downtime
- **Deployment:** coordinated rollouts across regional clusters
  - Configuration storage: gRPC + Postgres + JSONB
  - Cross-cluster operation: Sync Controller (inspired by Kubefed)
- **Container networking:** proxy peering and auth traffic between regions
  - Cilium Global Services with dedicated etcd

Teleport

Open Source

- Sync Controller:
    - github.com/gravitational/sync-controller *(Apache 2)*
- Envoy Gateway:
    - Upstream: github.com/envoyproxy/gateway *(Apache 2)*
    - Teleport Cloud fork: github.com/gravitational/gateway *(Apache 2)*
- Teleport:
    - github.com/gravitational/teleport *(Apache 2)*

Thanks!

Teleport Cloud Backend Team:

- ○ **Carson Anderson** - CNI stack, Cilium deployment
- ○ **David Boslee** - Ingress stack, Envoy Gateway fork
- ○ **Tobiasz Heller** - Teleport integrations
- ○ **Bernard Kim** - JSONB syncing & OpenAPI validations