Google Cloud

Next '24

Ingress traffic management for your fleet using Google Kubernetes Engine Enterprise

Nick Eberts

Product Manager,
Google Cloud

Pierre-Louis Gingembre

Senior Product Manager,
Google Cloud

Agenda

01 Ingress Traffic to GKE Services

02 Platform Engineering with GKE Enterprise

03 GKE Fleet Networking

04 Ingress to your GKE Fleet

Ingress Traffic to GKE Services

With GKE Enterprise

GKE Networking

Kubernetes Core

GKE Networking is K8s networking built on OSS foundation

Consistent with GKE, Google Distributed Cloud

VPC Native

Complete out of the box integrations with Google Cloud Networking

VPC features & scale

Security & Services

Rich Network Policy

GKE Gateway for unified access to rich network services

Integrated GCP services with Cloud DNS, Cloud Armor, Cloud Load Balancing and more

GKE Services

A Service is a method for exposing a network application that is running as one or more Pods in your cluster.

Accessible from within the cluster

ClusterIP

Collection of backend Pods with a unique Service IP address

Headless

Collection of backend Pods with no Service IP address

Multi-cluster Services

Collection of backend Pods with a unique Service IP address across clusters

Accessible from outside the cluster

NodePort

Collection of nodes listening on the same host port with no Service IP address*

LoadBalancer

Collection of nodes or Pods with a unique Network Load Balancer IP address*

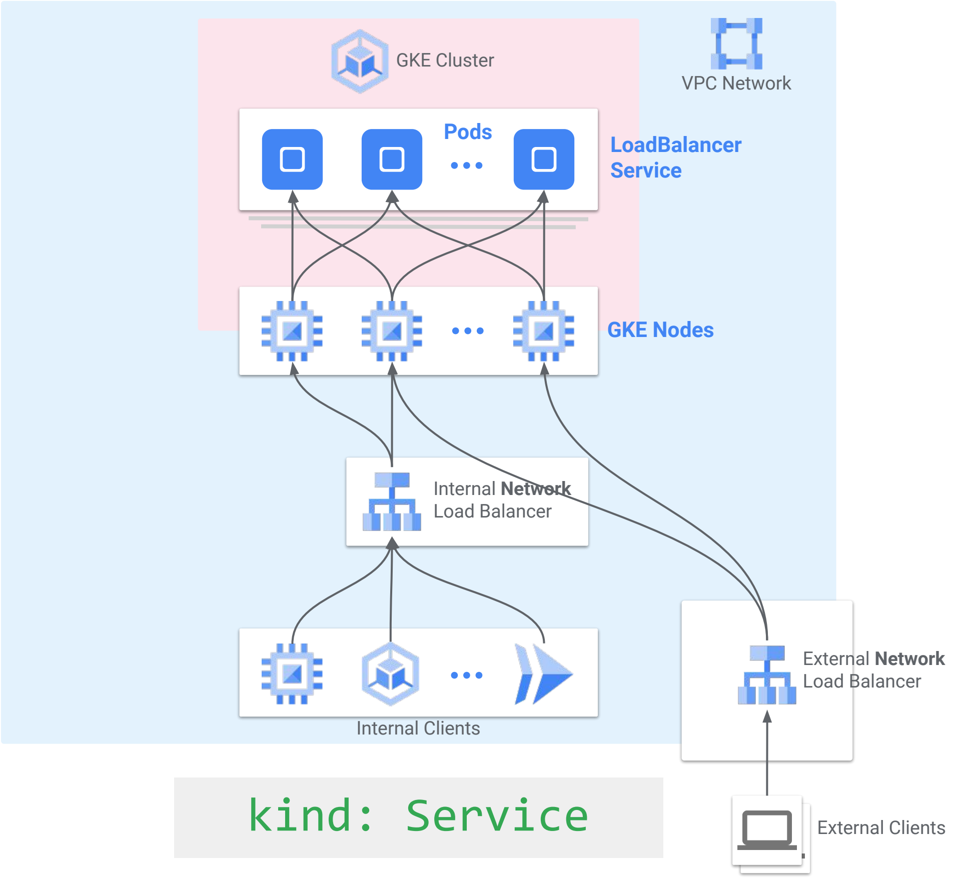
(Multi-cluster) Gateway/Ingress

Collection of nodes or Pods with a unique Application Load Balancer IP address*

* nodes = VM instances

Routing Traffic from Outside to GKE Services

LoadBalancer



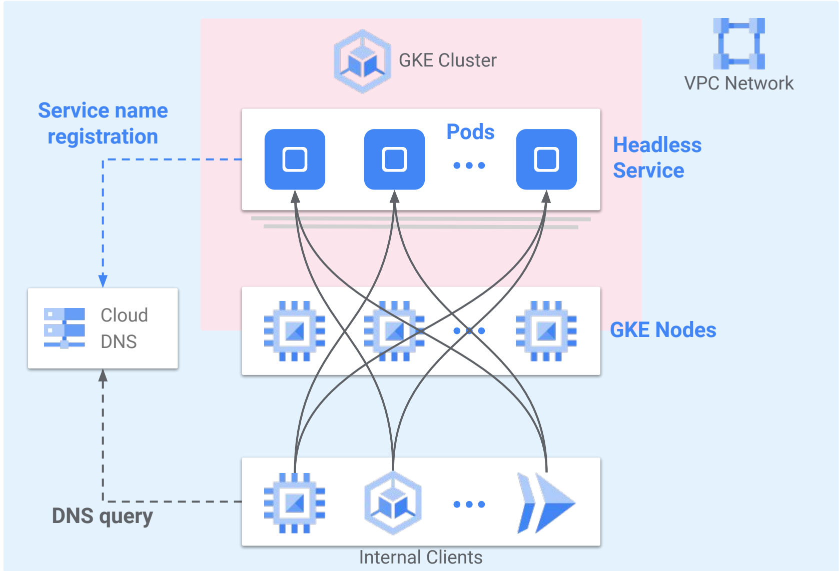
```
kind: Service
```

Service exposed with an “external” IP (i.e. outside of the GKE space)

Traffic forwarded to the GKE nodes first, then to the pods

Managed passthrough network load balancer (with Direct Server Return)

Headless



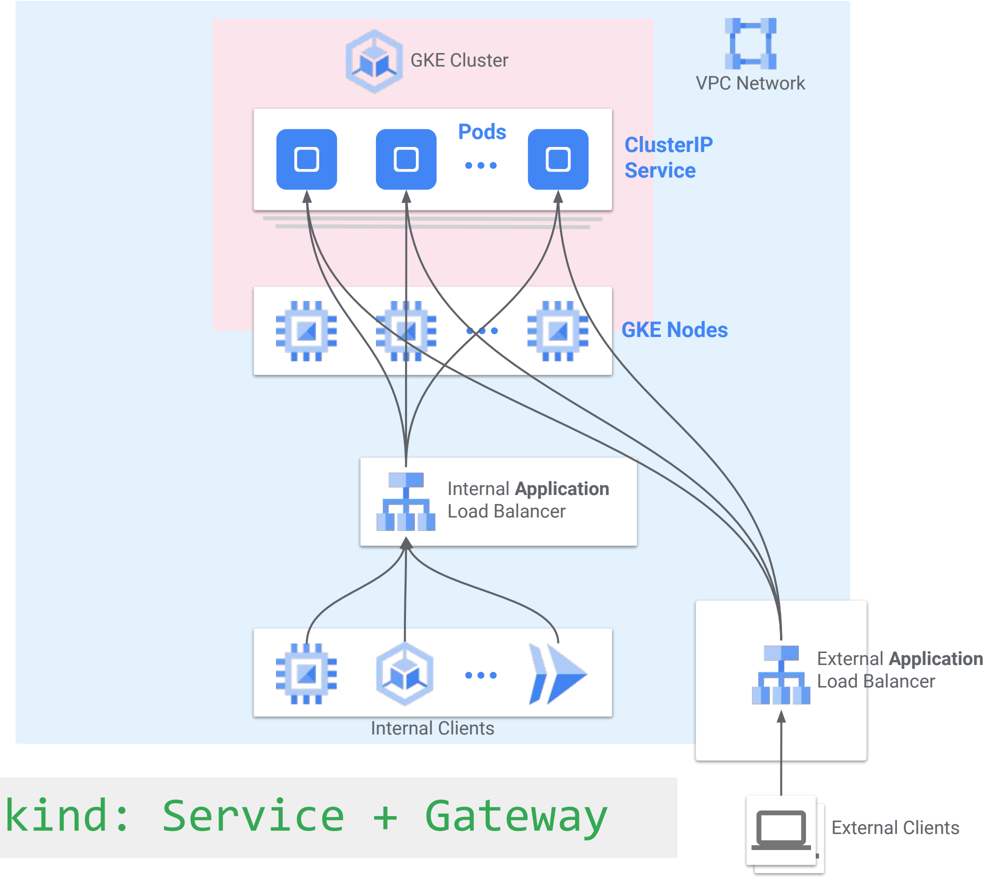
```
kind: Service
```

Service discovery & resolution required for pod names/IP (Cloud DNS VPC Scope or other)

Traffic forwarded to the GKE pods directly

No load balancer required

Gateway & Ingress



```
kind: Service + Gateway
```

Service exposed with an “external” IP (i.e. outside of the GKE space)

Traffic forwarded to the GKE pods directly

Managed proxy application load balancer (with session termination)

GKE-Managed Load Balancers

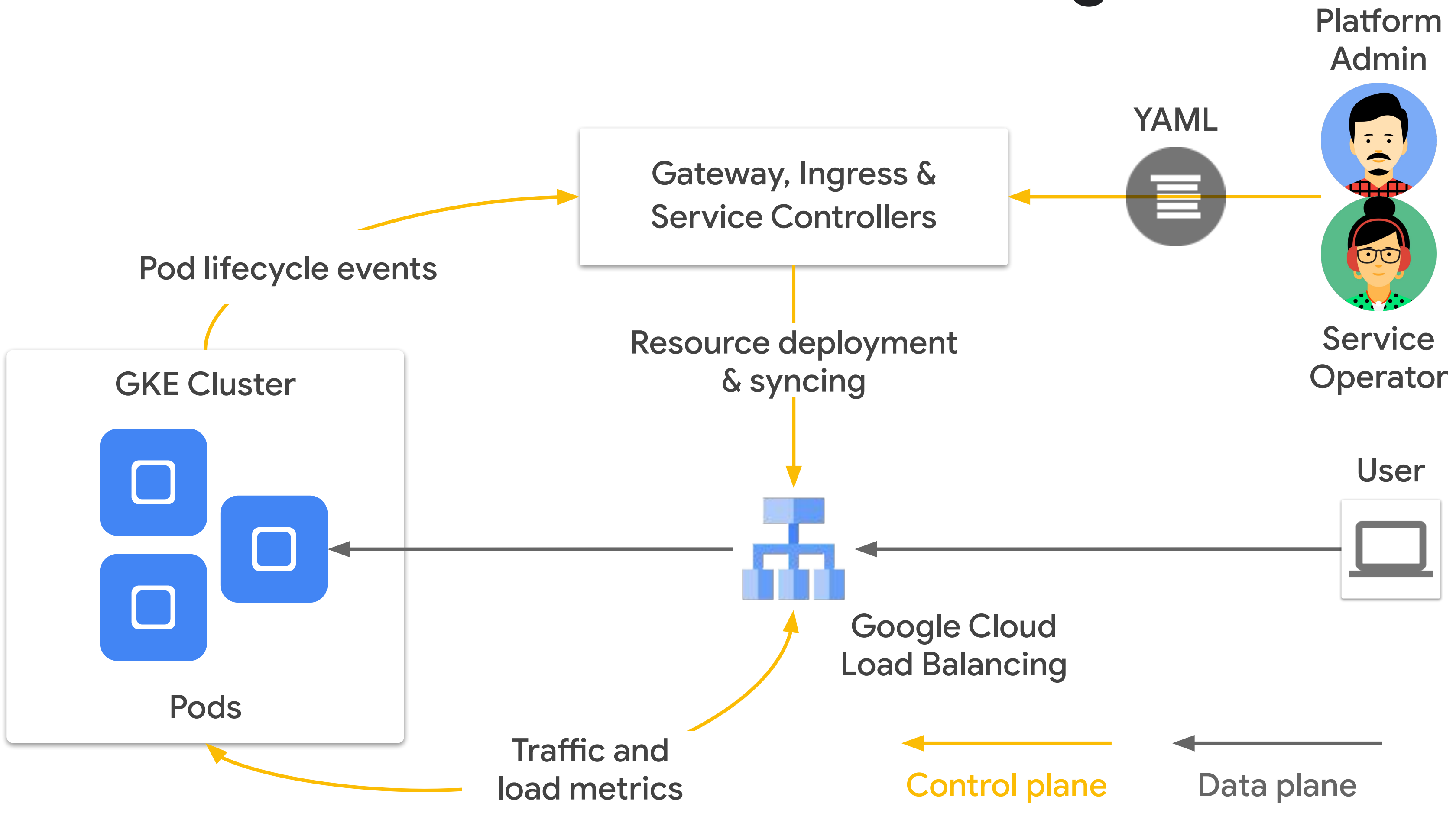
A Kubernetes-native API

```
kind: Service
```

```
kind: Ingress
```

gateway api

Google-hosted Kubernetes controllers & Cloud Load Balancing



> “We are building a Platform for our application teams and we want them to focus on the business logic, not the Cloud infrastructure.”

Jane Doe, Platform Engineer, *The Company*

Platform Engineering

With GKE Enterprise

GKE Enterprise Fleets key capabilities make Platform building Simpler

Scale easier with multi-cluster management

- Fleet-based multi-cluster management
- Fleet wide Security and governance with Policy Manager
- Cost and performance visibility across Fleets and Teams
- Multi-Cluster Networking, Load balancing & Service Meshing
- GitOps for automated infrastructure management

Provide self service multi-tenancy with teams

- Fleet-based multi-team management
- Self-service developer environments
- Private Access to Clusters w/ Connect Gateway
- Cost and Performance dashboards and recommendation for each team

Save with a managed container platform

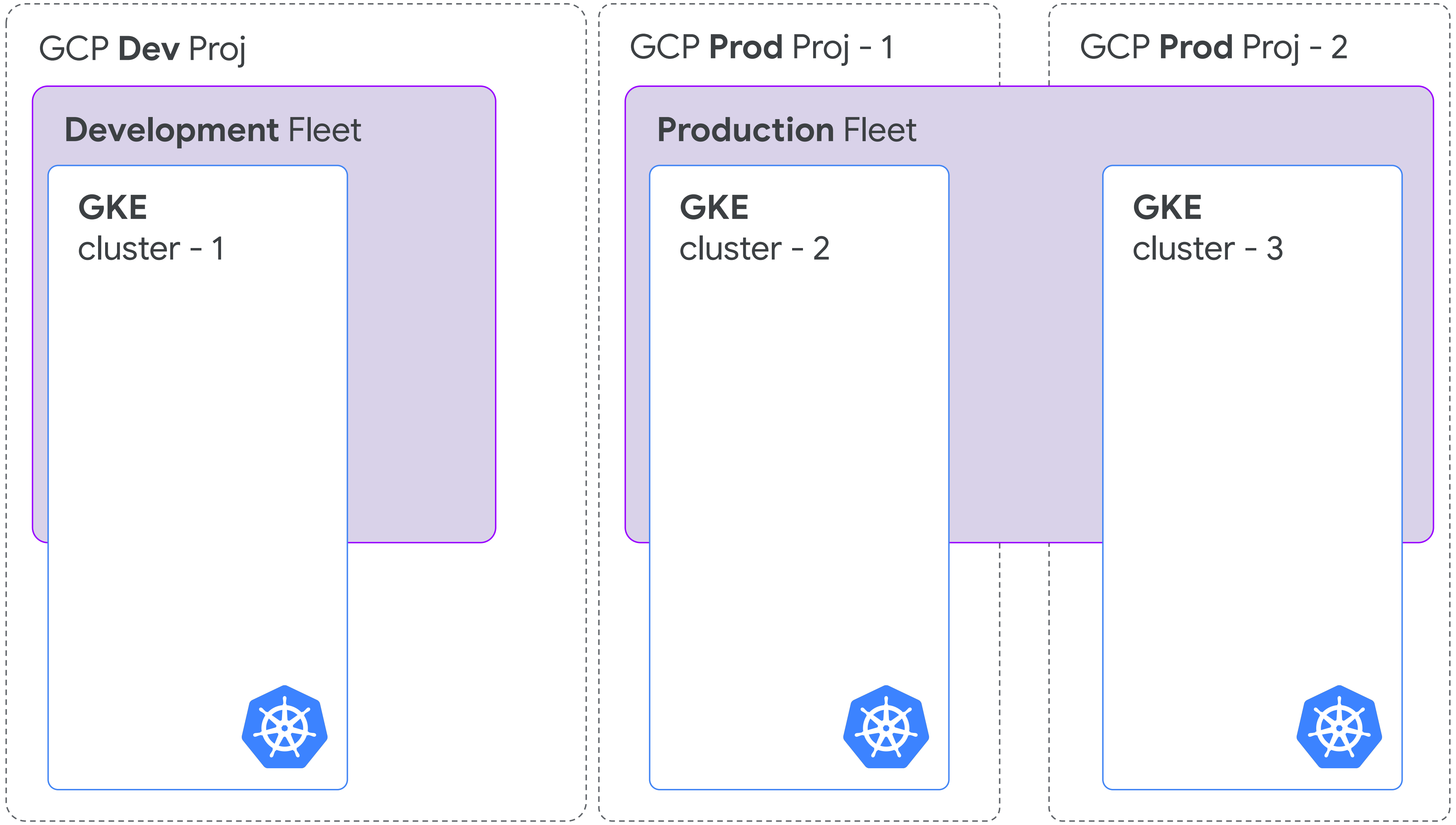
- Run on multi-cloud, on-prem, and edge based workloads with one Platform
- OSS tools turned into Managed Services (Config Sync, OPA, Gateway Controllers, Istio)
- Unified operations console for visibility across fleets

Fleets

A **Fleet is a collection** of Kubernetes clusters, **that function together to serve an Environment.** (Example: my production infrastructure for a line-of-business)

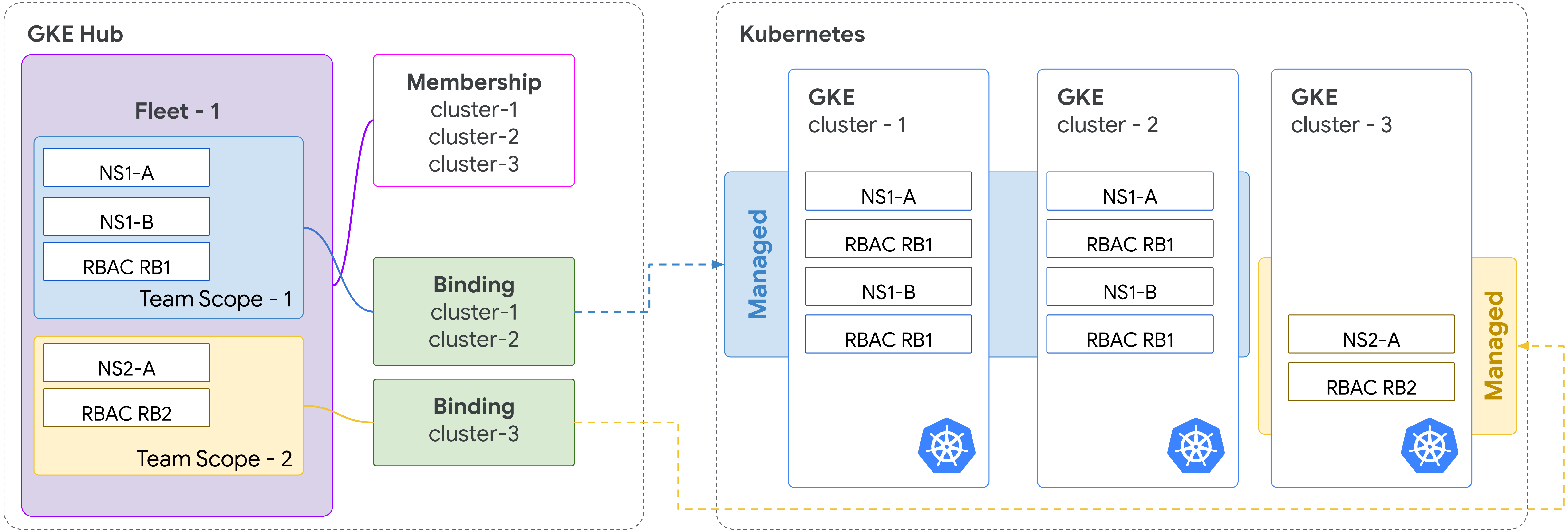
A fleet is a **visible entity** that has an identity, ownership, and permissions. **A fleet is hosted in a GCP project,** but can contain resources from **multiple GCP Projects, public clouds,** and **on-premises environments.**

You can have multiple fleets per GCP organization.



States of Kubernetes Namespaces: managed

Kubernetes namespaces (KNS) which have the same name as an FNS and there is a binding from their scope to their cluster are called “managed” KNS. These are legit namespaces which are approved to exist on their clusters.



Centrally Managed Fleet Features

- **Workload identity pools**
  Common workload identity pool that can be used to authenticate and authorize workloads uniformly.
- **Anthos Service Mesh**
  Form a service mesh across the resources within the fleet.
- **Multi Cluster Gateway**
  Load balance the traffic across services within the fleet.
- **Policy Controller**
  Managed OPA/Gatekeeper with prebuilt constraints
- **Config Management**
  Deploy declarative configuration and policies within the fleet.
- **Kubernetes Security Posture**
  Policy Controller, Advanced Vulnerability Insights,

Workload identity pools

Anthos Service Mesh

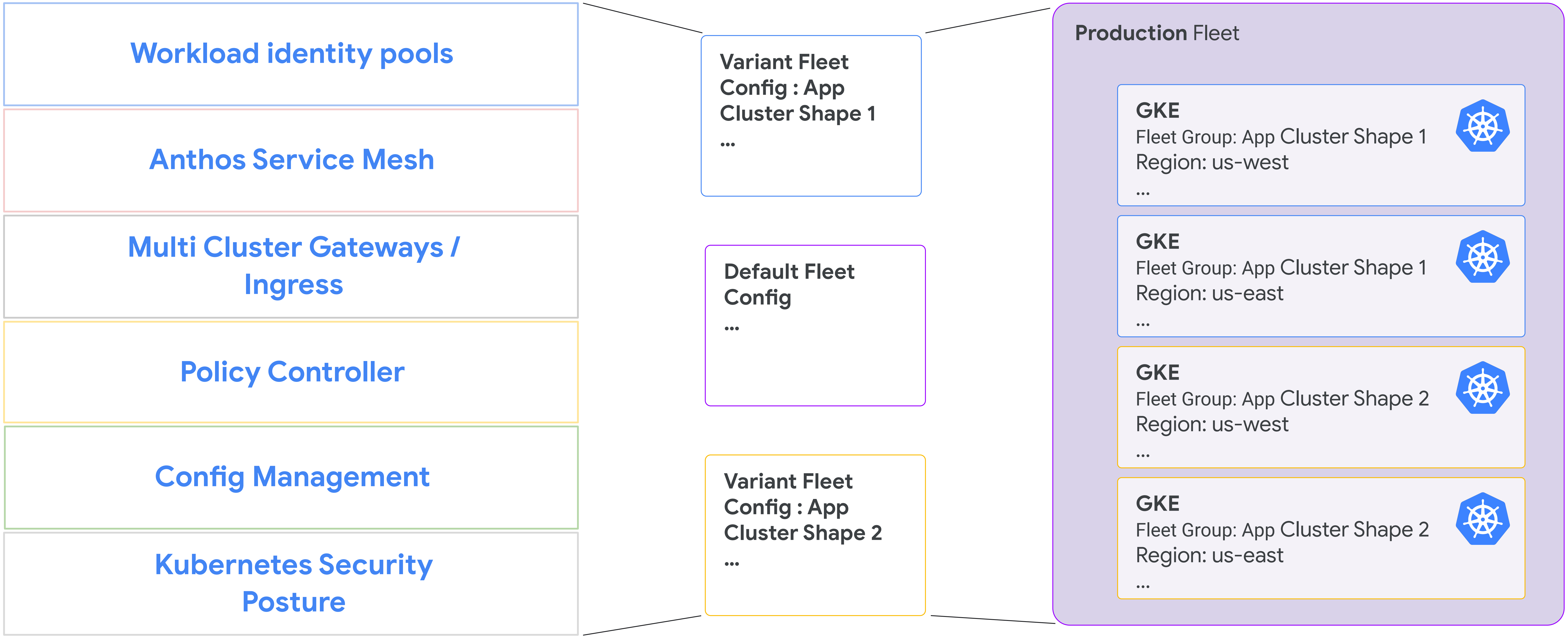
Multi Cluster Gateways / Ingress

Policy Controller

Config Management

Kubernetes Security Posture

Centrally Managed Fleet Features for Shapes of App Clusters



Workload identity pools

Anthos Service Mesh

Multi Cluster Gateways / Ingress

Policy Controller

Config Management

Kubernetes Security Posture

Variant Fleet Config : App Cluster Shape 1
…

Default Fleet Config
…

Variant Fleet Config : App Cluster Shape 2
…

Production Fleet

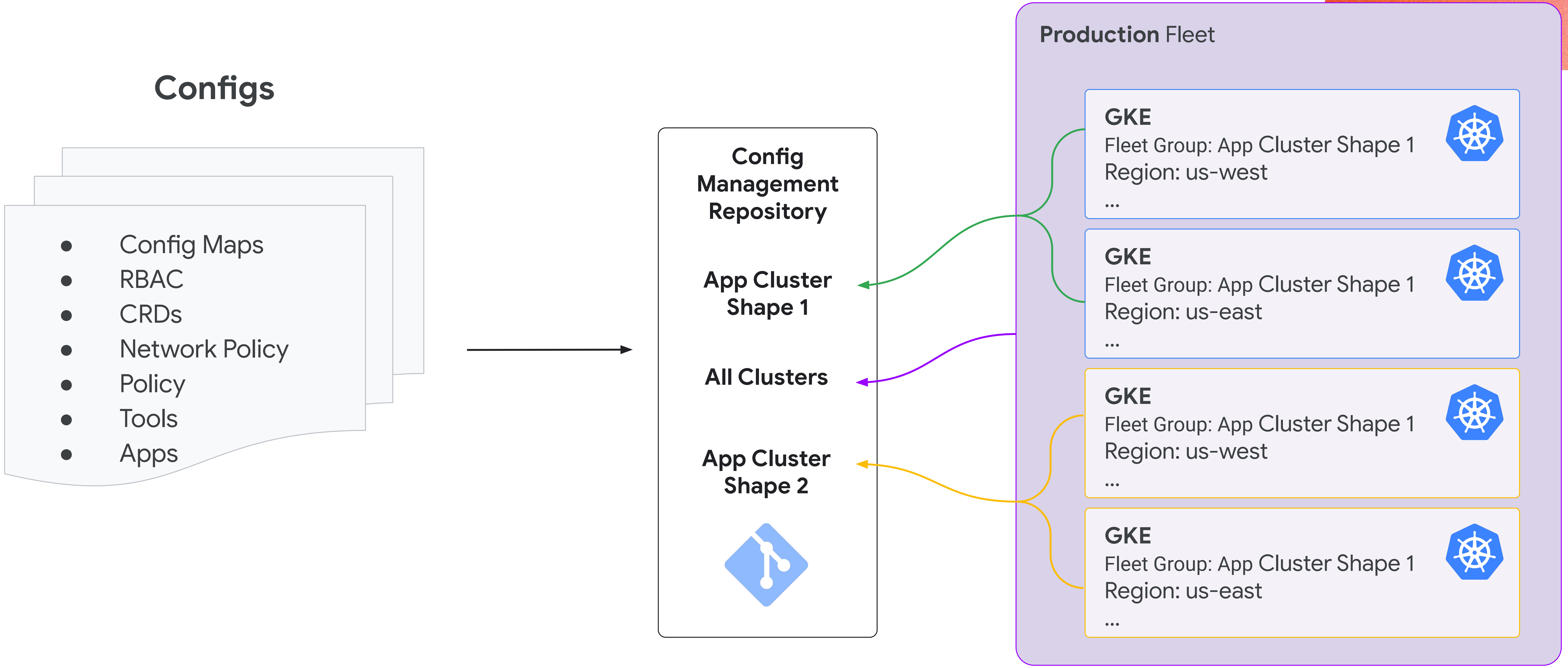
GKE
Fleet Group: App Cluster Shape 1
Region: us-west
…

GKE
Fleet Group: App Cluster Shape 1
Region: us-east
…

GKE
Fleet Group: App Cluster Shape 2
Region: us-west
…

GKE
Fleet Group: App Cluster Shape 2
Region: us-east
…

Config Management & **Policy Controller**



Configs

- Config Maps
- RBAC
- CRDs
- Network Policy
- Policy
- Tools
- Apps

Config Management Repository

App Cluster Shape 1

All Clusters

App Cluster Shape 2

Production Fleet

GKE
Fleet Group: App Cluster Shape 1
Region: us-west
...

GKE
Fleet Group: App Cluster Shape 1
Region: us-east
...

GKE
Fleet Group: App Cluster Shape 1
Region: us-west
...

GKE
Fleet Group: App Cluster Shape 1
Region: us-east
...

Fleet Multi-Cluster Networking

With GKE Enterprise

Fleet Multi-Cluster Networking

Multi-Cluster Services

Run Services across clusters for higher availability and geo-distributed applications

Multi-Cluster Gateway

Route ingress traffic to multi-cluster services with advanced traffic management and security capabilities

Service Mesh

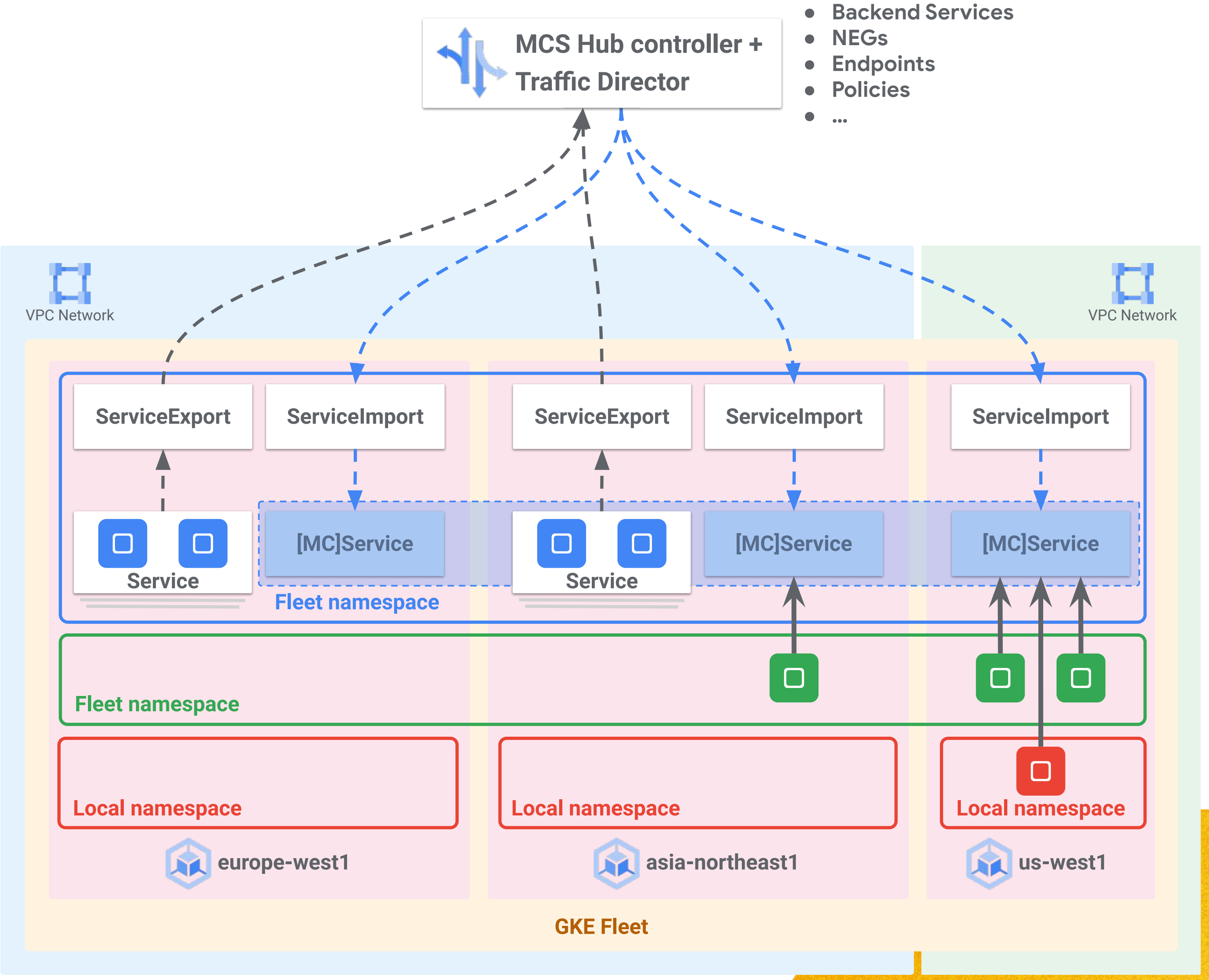
Route east-west traffic across clusters with advanced traffic management, observability and security capabilities

Topics for today’s session

Multi-Cluster Service

A Foundation for Multi-cluster Networking based on an open source API standard, powered by Google infrastructure

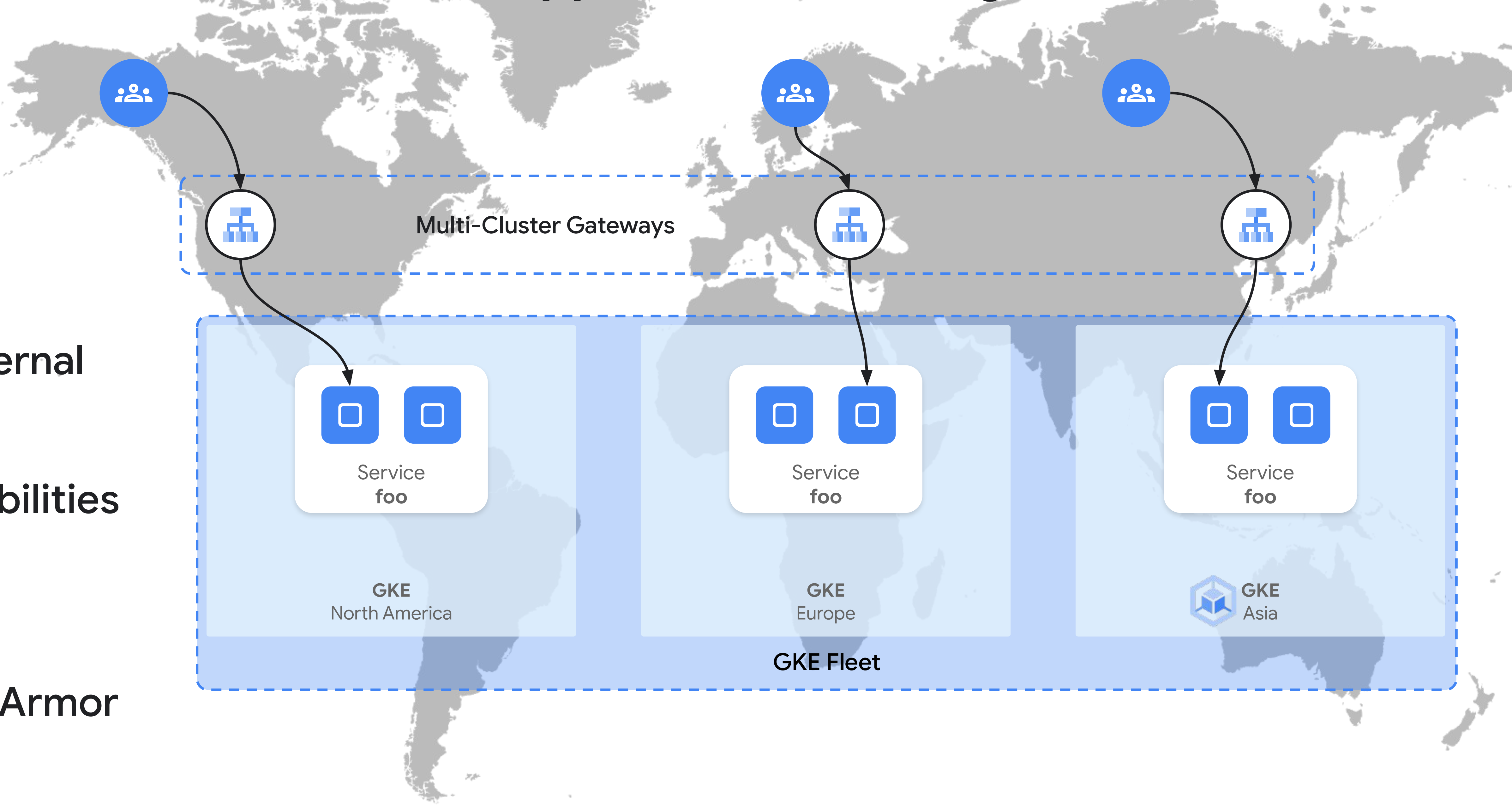
- Kubernetes APIs for producing and consuming Services that select endpoints across clusters
- Cross-cluster ClusterIP Service for East-West Traffic management with no sidecar
- Central control plane that supports all cloud runtimes for future integrations
- Flexible deployment strategies for projects and VPC networks
- Foundational element for Multi-Cluster Gateway



Multi-Cluster Gateway

Google Cloud Load Balancers managed through GKE controllers to deliver advanced traffic management capabilities for distributed applications running on GKE

- Kubernetes native API to express your routing intent
- Single Anycast IP for your global applications in GKE
- Supports for regional internal or external deployments for compliance
- Advanced traffic management capabilities across zones and regions
- Protect services from DDoS and application layer attacks with Cloud Armor



Foundation: An open source API

Extensible support for multiple implementations, OSS + vendor specific options.

Scope: All Kubernetes services: L4 and L7 load-balancing, service mesh (Istio)

gateway api

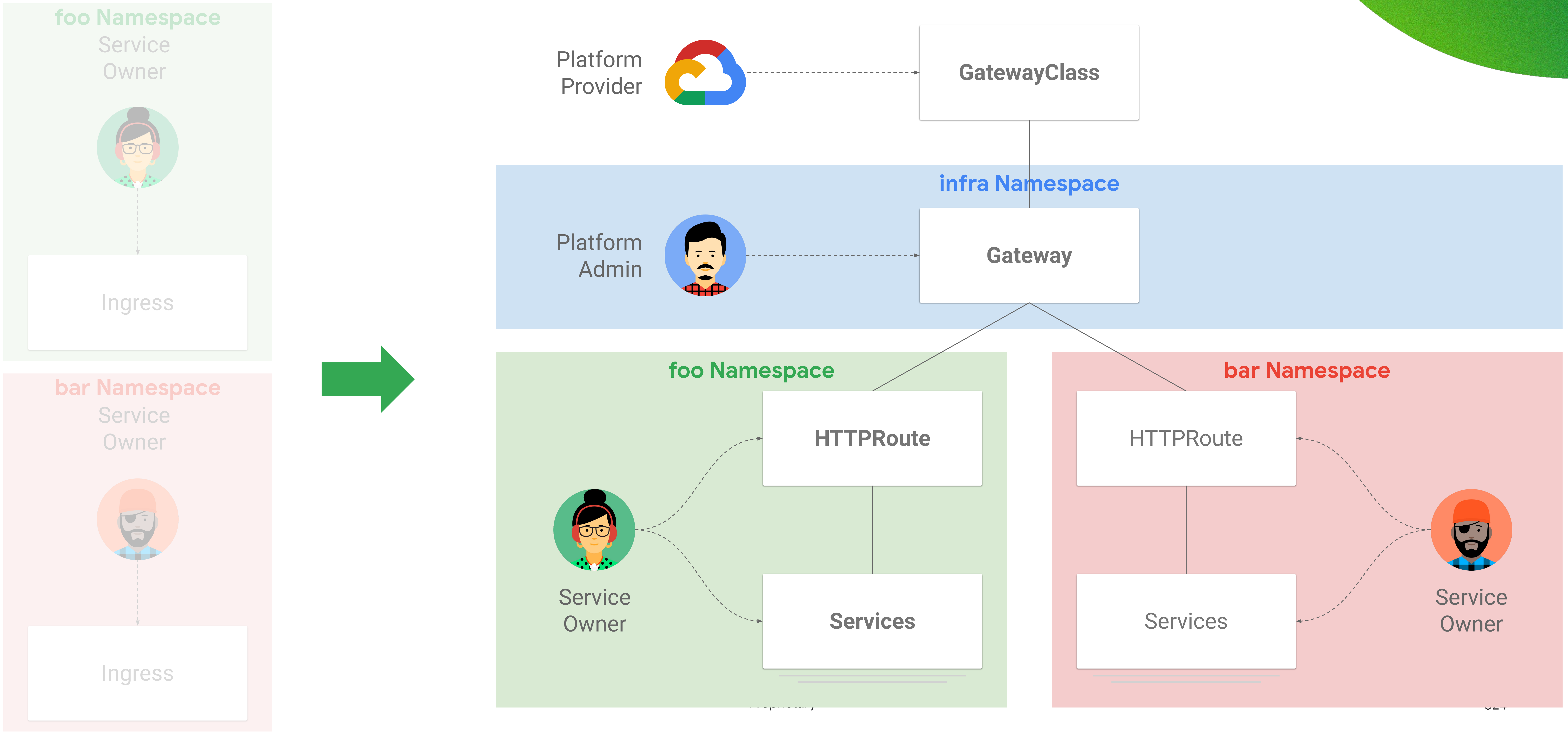
Gateway API: a single, unified, extensible, role-oriented API for Kubernetes Service Networking.

Kubernetes-native, open-source, portable with conformance testing.

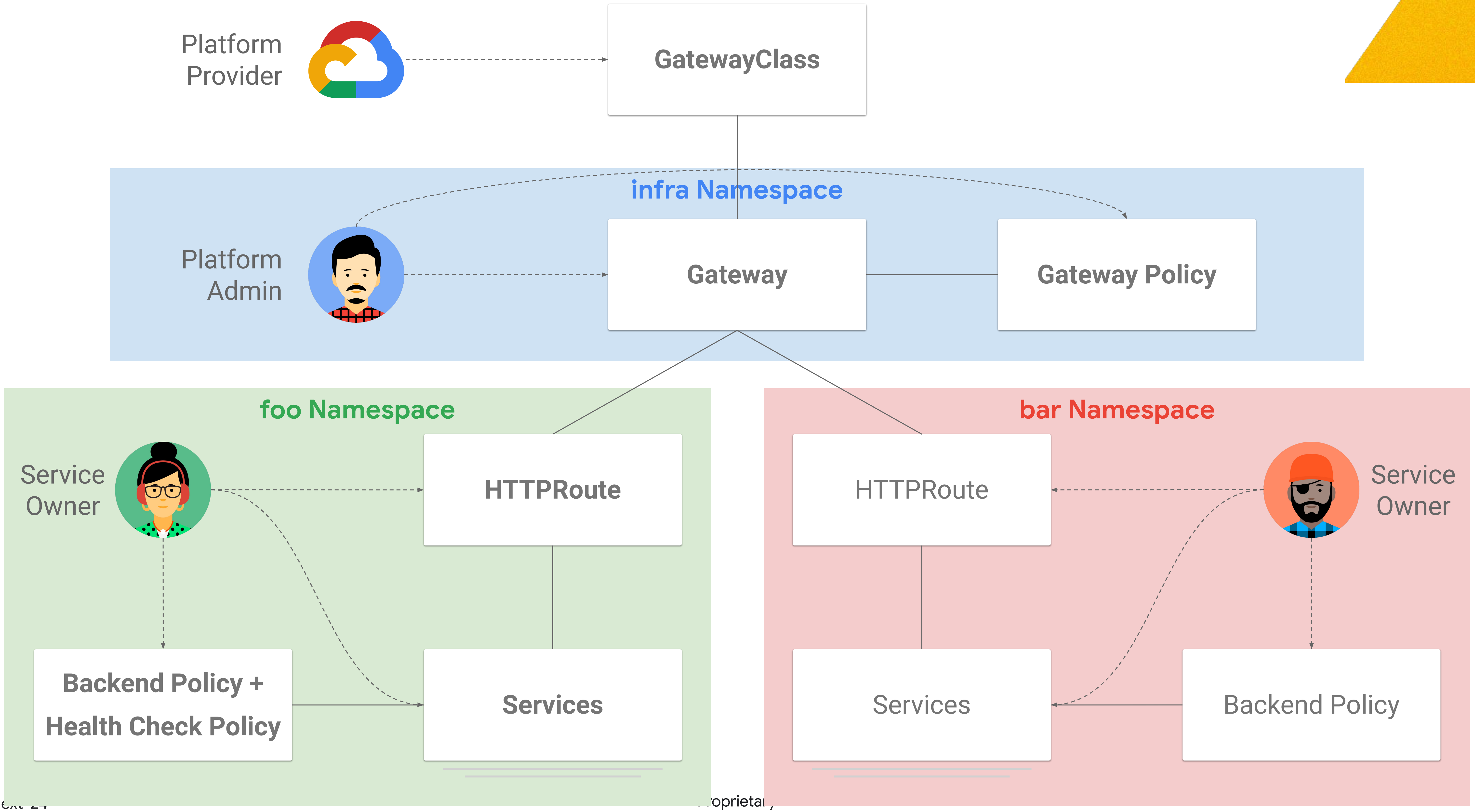
Persona-based resource design to support complex deployments.

Kubernetes Gateway API is now GA

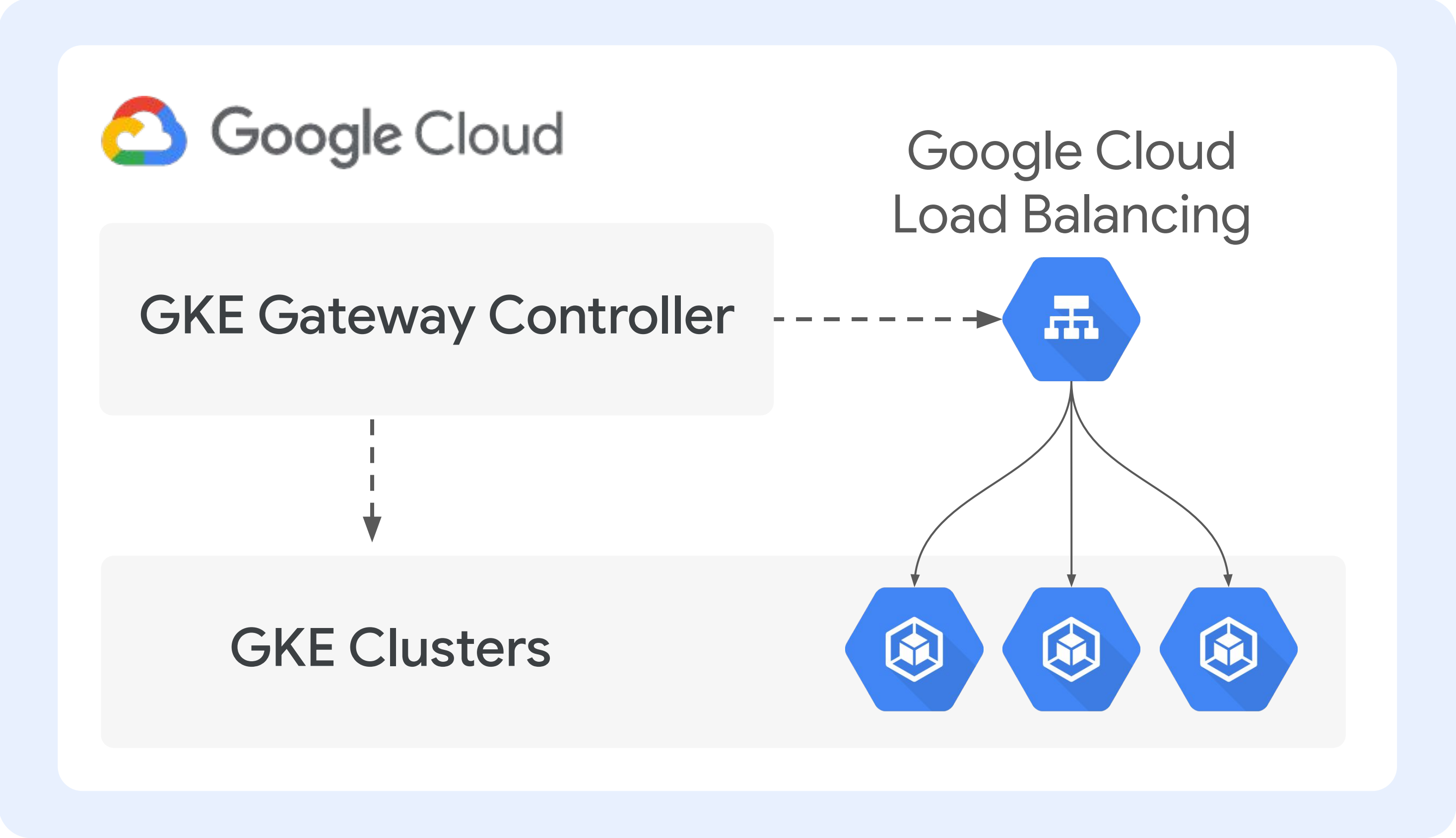
Role-oriented for today's organizations



Extensible through Policies



GKE Gateway: Google’s implementation

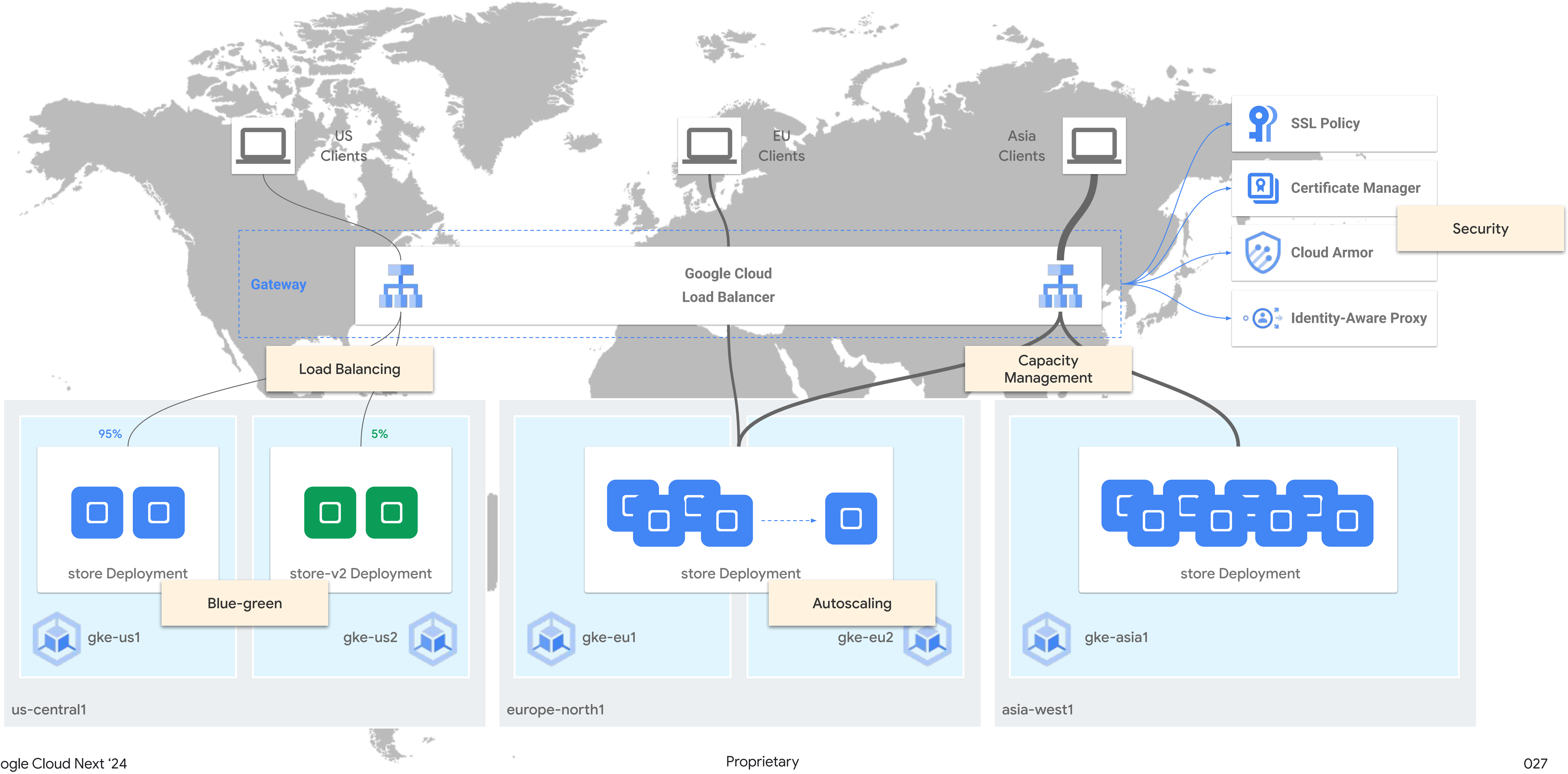


A set of Google-hosted Kubernetes controllers that orchestrate Google Cloud Load Balancers via the open source Gateway API specification.

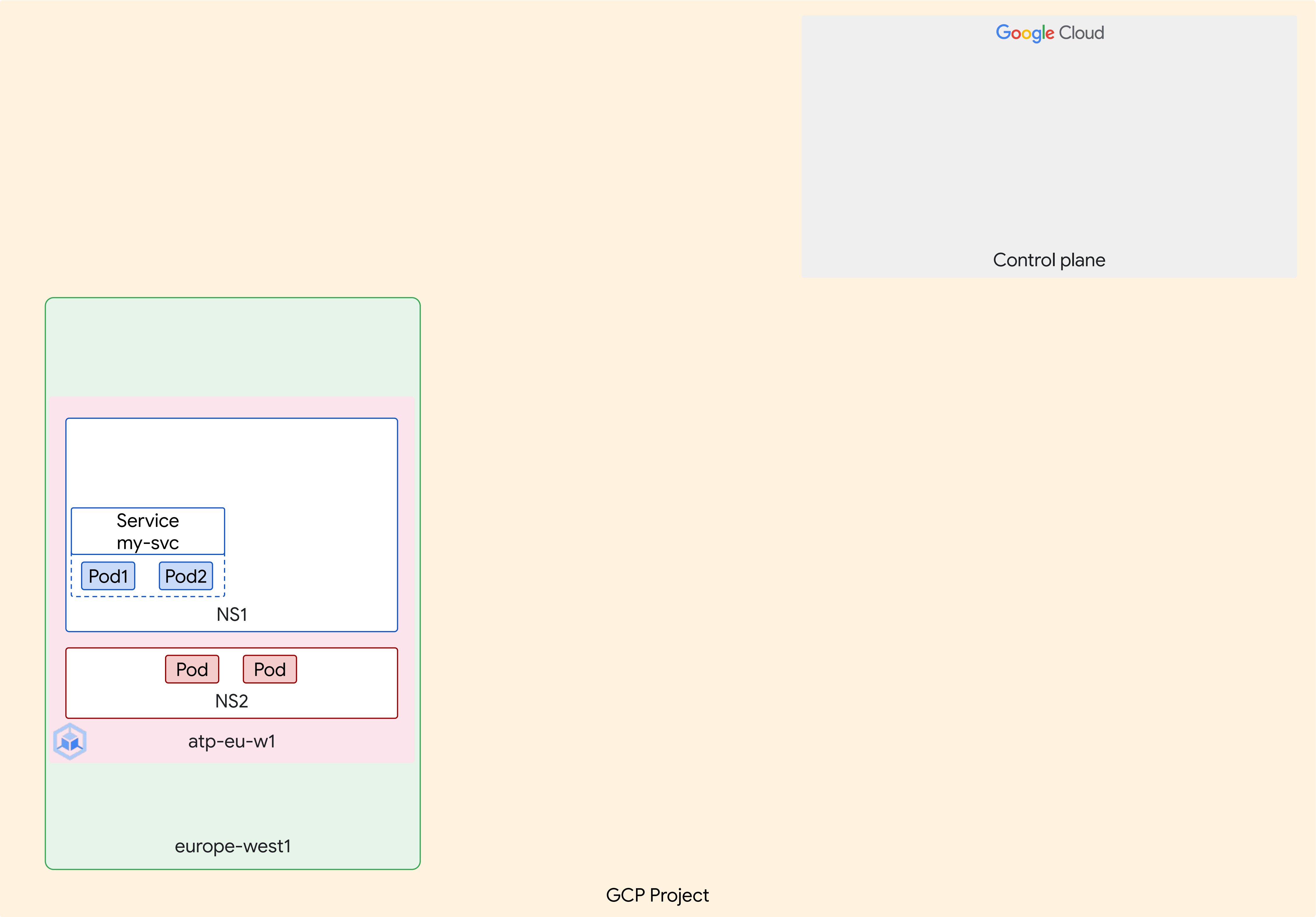
- Supports internal *and* external Application Load Balancers
- Host, path, header-based routing
- HTTP header manipulation
- Weight-based traffic splitting
- Traffic capacity-based load balancing
- Traffic mirroring

- Multi-cluster Gateways (MCGs) for internal *and* external load balancing
- Support for Cloud Armor, Identity-Aware Proxy (IAP), and Cloud CDN
- Geographic-based load balancing
- End-to-end TLS between client and backends

Multi-Cluster Gateway Use Cases



Multi-Cluster Gateway Walk-through



1 Single Cluster, “Regular” Service

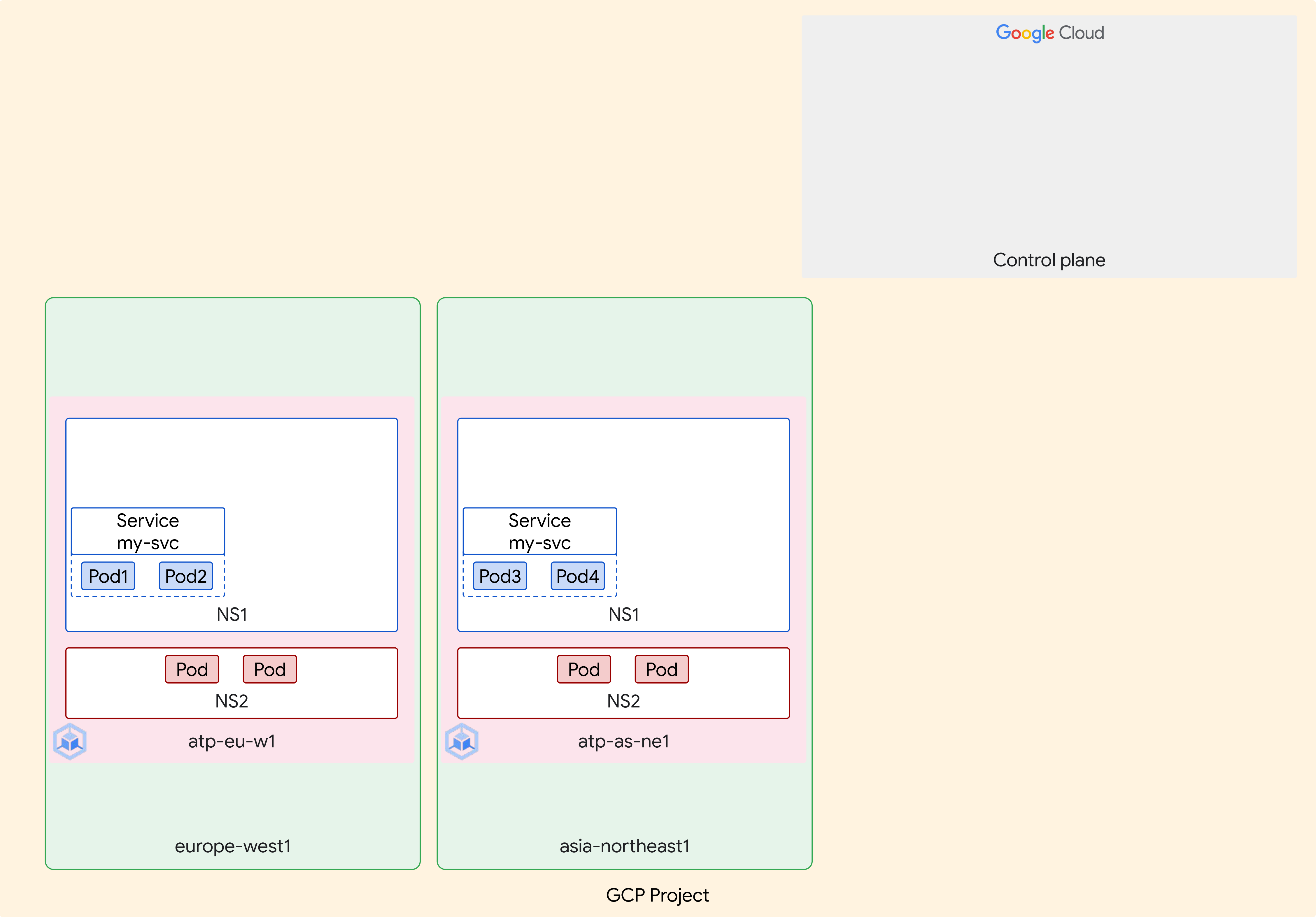
✓ ClusterIP Service with a unique IP for the Service

Service Discovery through DNS with an entry in the cluster.local. DNS zone

All other pods can communicate with the Service by name/IP

Other cloud runtimes can communicate with the service via load balancer or with the pods using their IP addresses

Multi-Cluster Gateway Walk-through



2 New region, new cluster

✓ More capacity

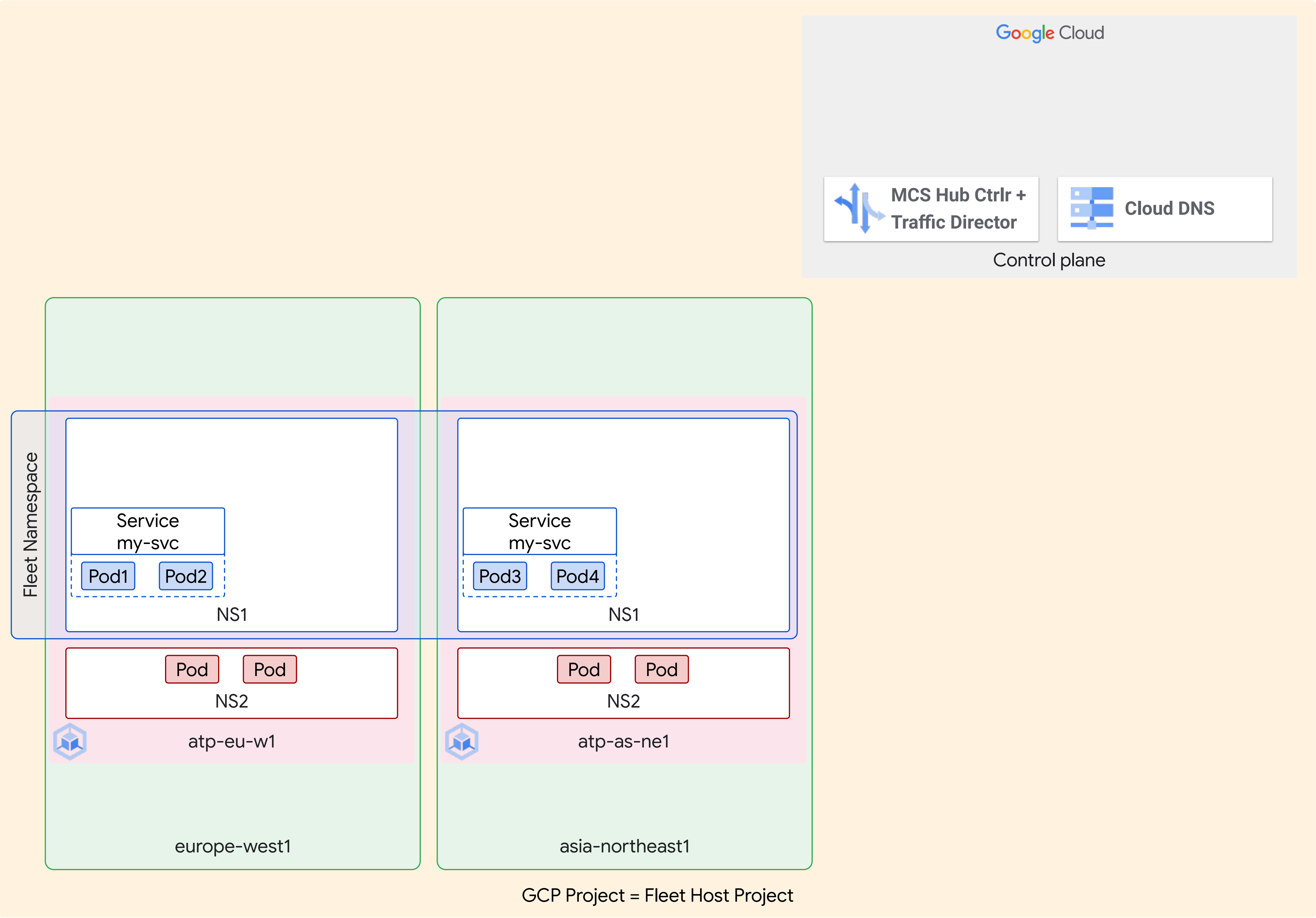
Closer to your clients/users

→ Increased east-west and north-south routing complexity (IP, DNS, load balancing)

No built-in cross-region service discovery

DNS-based failover strategy

Multi-Cluster Gateway Walk-through



3 Fleet and Multi-Cluster Service Enablement

✓ GKE Enterprise enablement at the project level

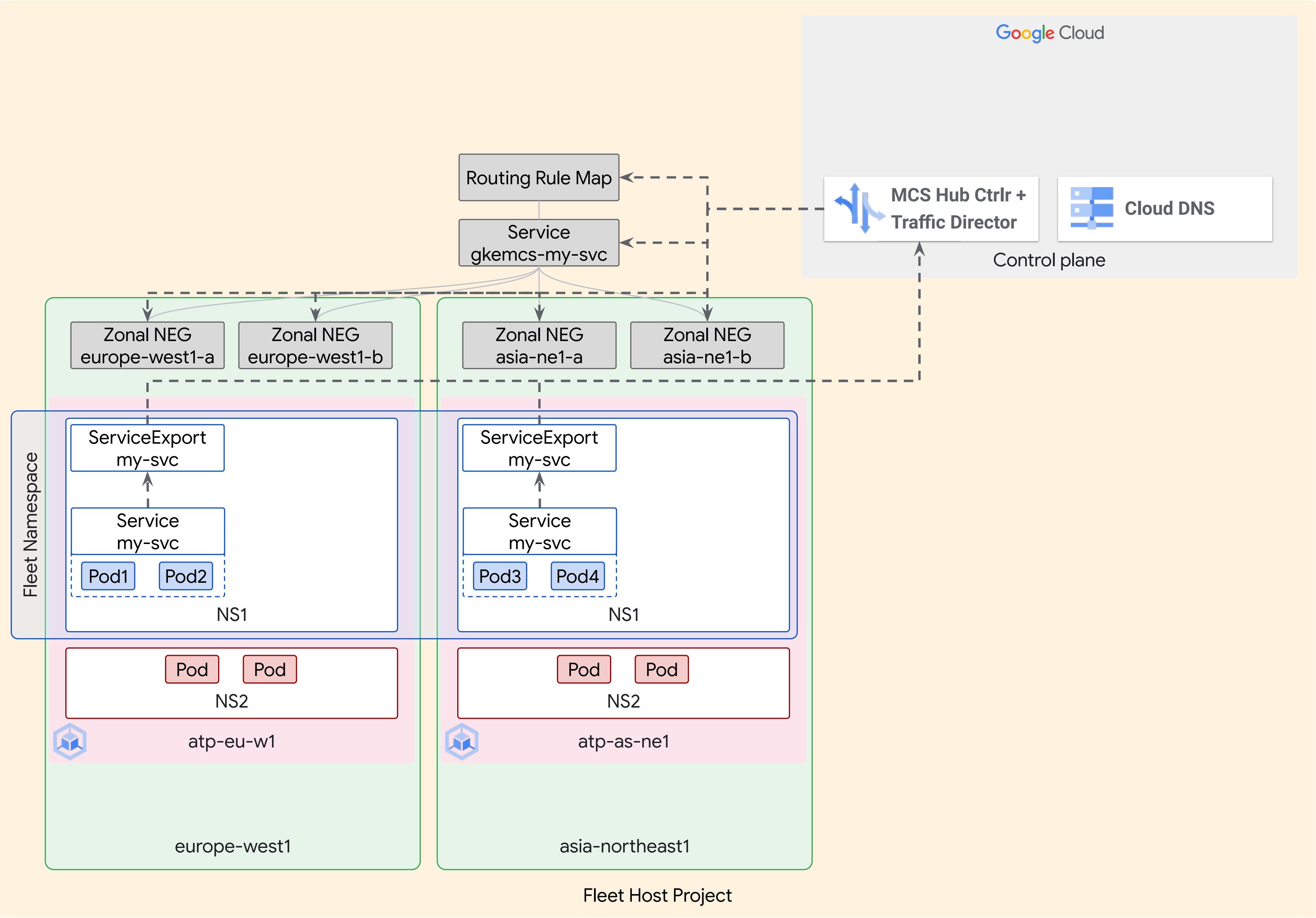
Networking APIs enablement at the project level (MCS, DNS, Traffic Director)

Fleet feature enablement (Multi-cluster Service Discovery)

New controllers watching MCS APIs and connected to Traffic Director and Cloud DNS

Namespace-sameness for fleet consistency

Multi-Cluster Gateway Walk-through

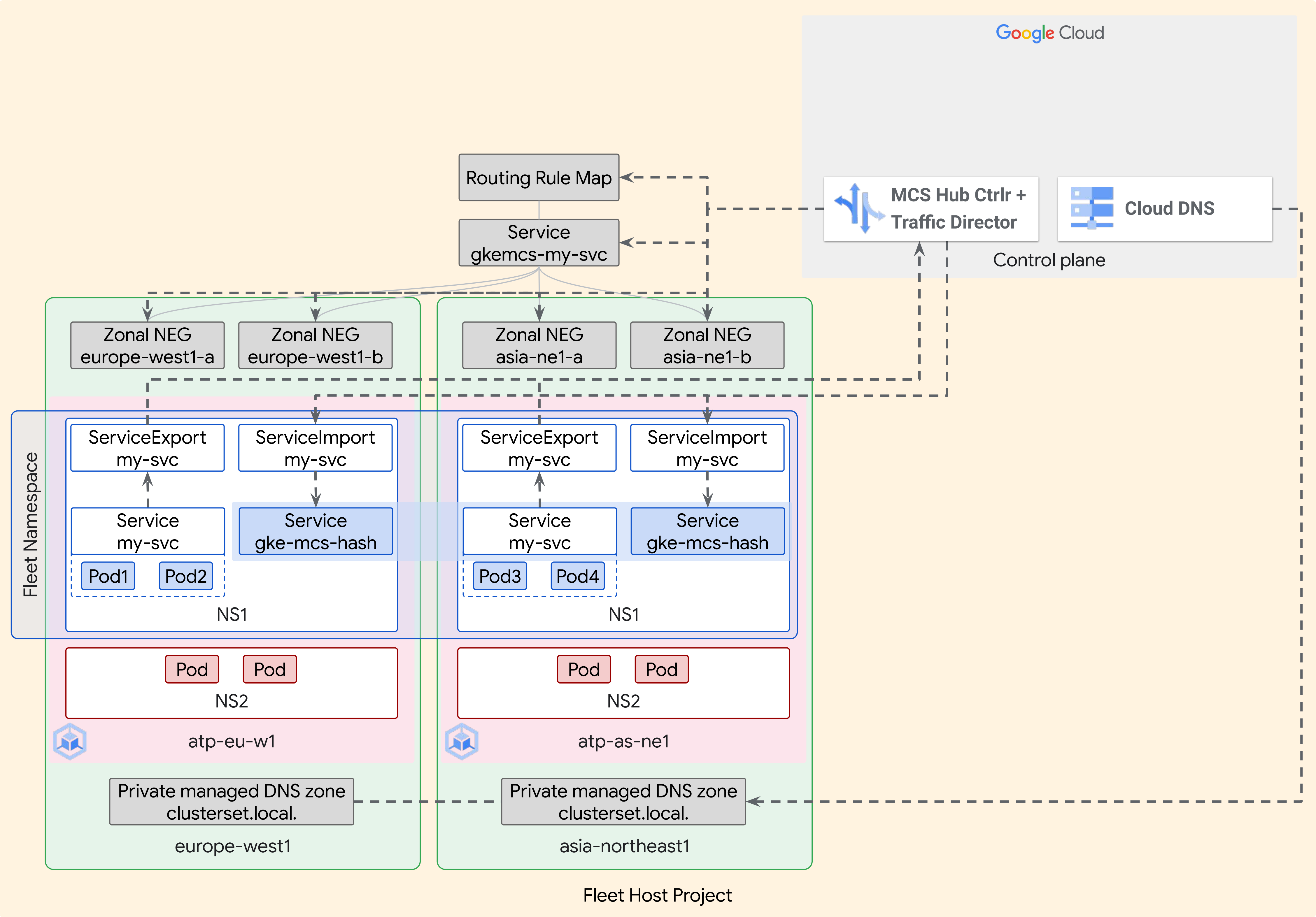


4 Exporting a Service to the fleet

✓ ServiceExport resources created on fleet members running the service

MCS Hub controller is updated with endpoints information and creates the appropriate resources (Routing rule map, Service, NEGs)

Multi-Cluster Gateway Walk-through



5 [MC]Service Import on fleet members

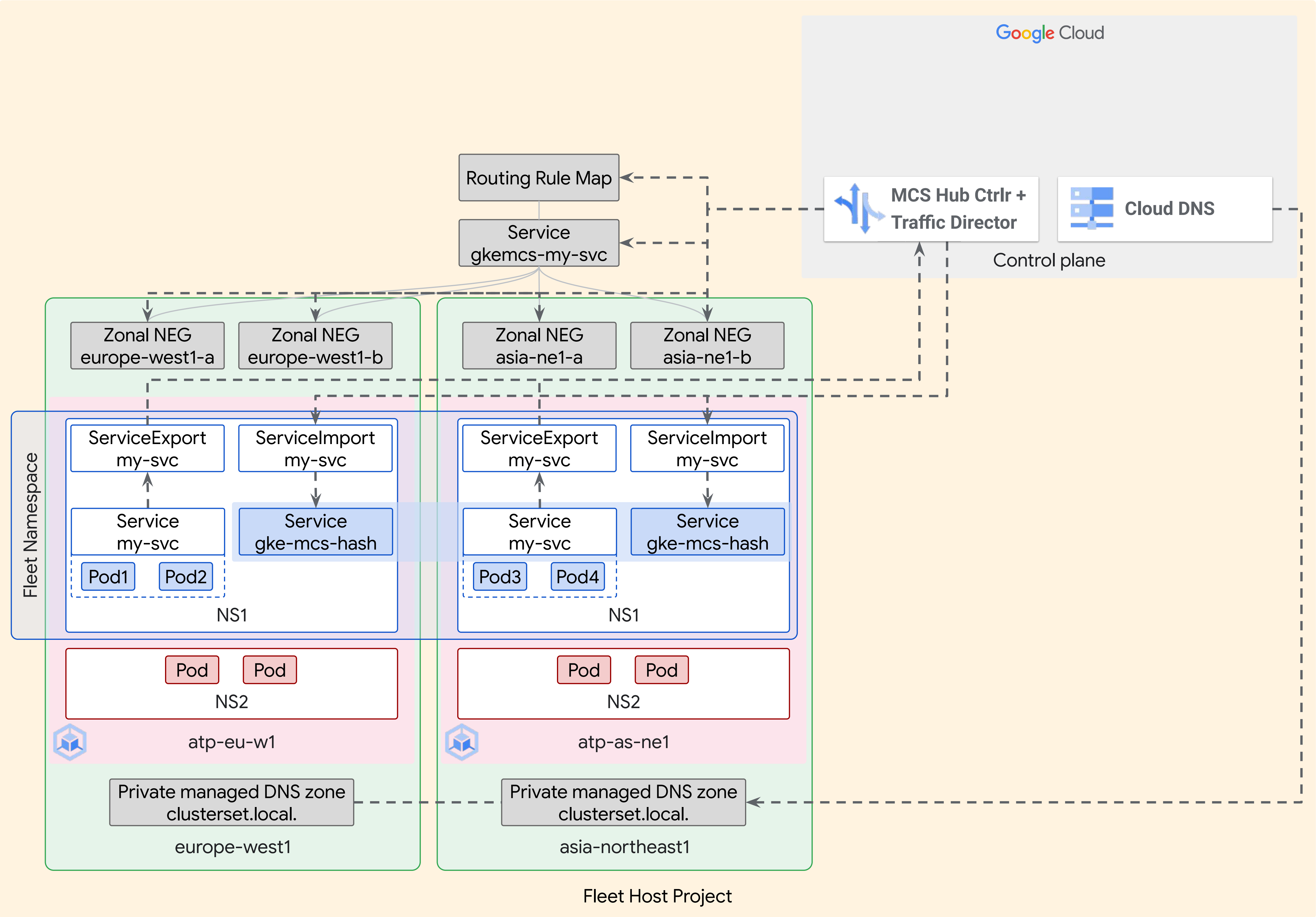
✓ ServiceImport resources dynamically created on other members with the fleet namespace created (ClusterSetIP)

Creation of a [Multi-Cluster] Service (ClusterIP) on each fleet member populated with all the endpoint IP addresses

Traffic Director pushes remote endpoints information to all clusters

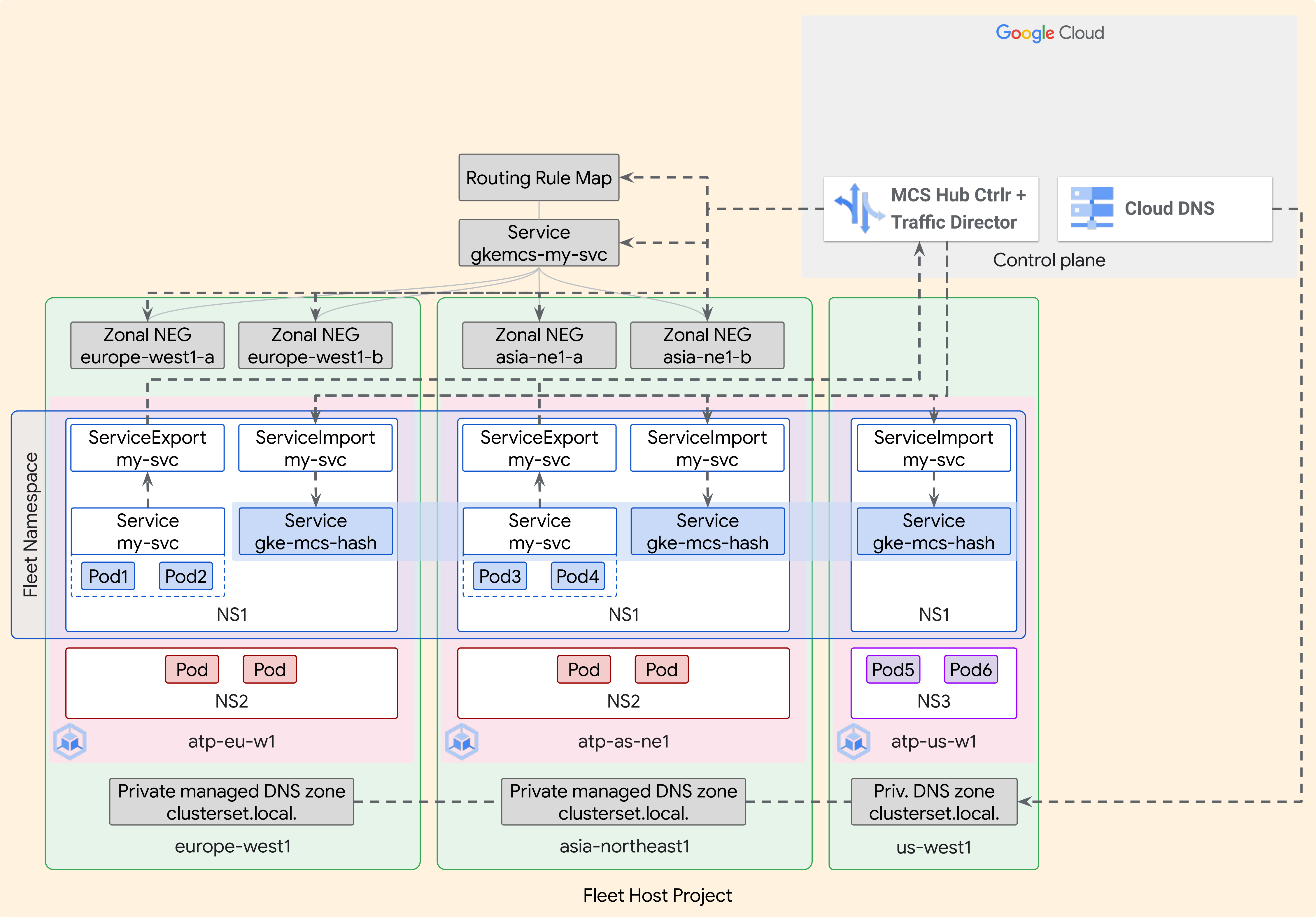
Service name registered with private managed DNS zone (Cloud DNS) with clusterset.local. domain

Multi-Cluster Gateway Walk-through



6 Expand the fleet with new members

Multi-Cluster Gateway Walk-through



6 Expand the fleet with new members

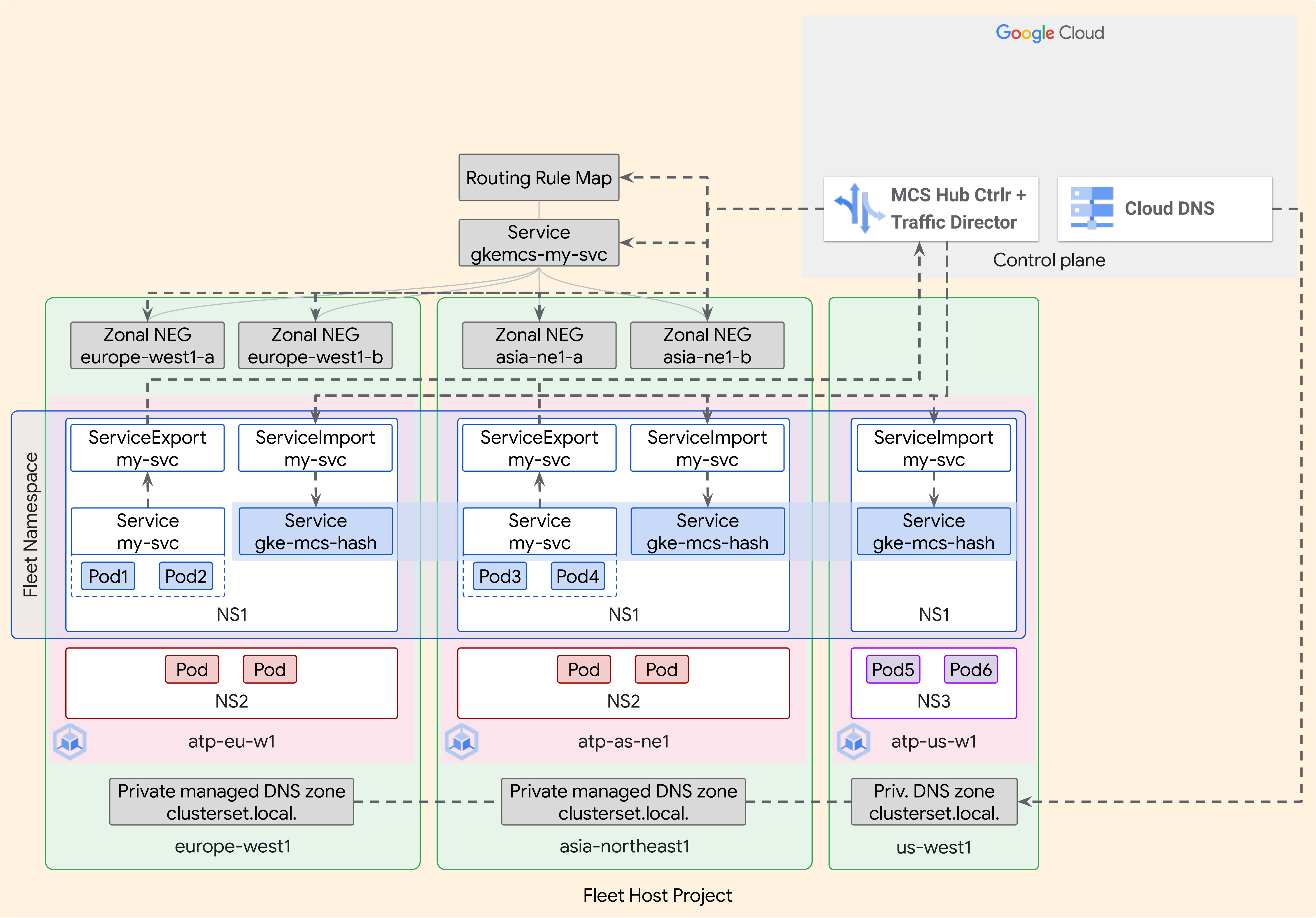
✓ New cluster added to the fleet

Fleet namespace added to the new cluster

Multi-cluster Service imported to the new cluster in the fleet namespace

Connectivity to all pods backing a multi-cluster service is automated

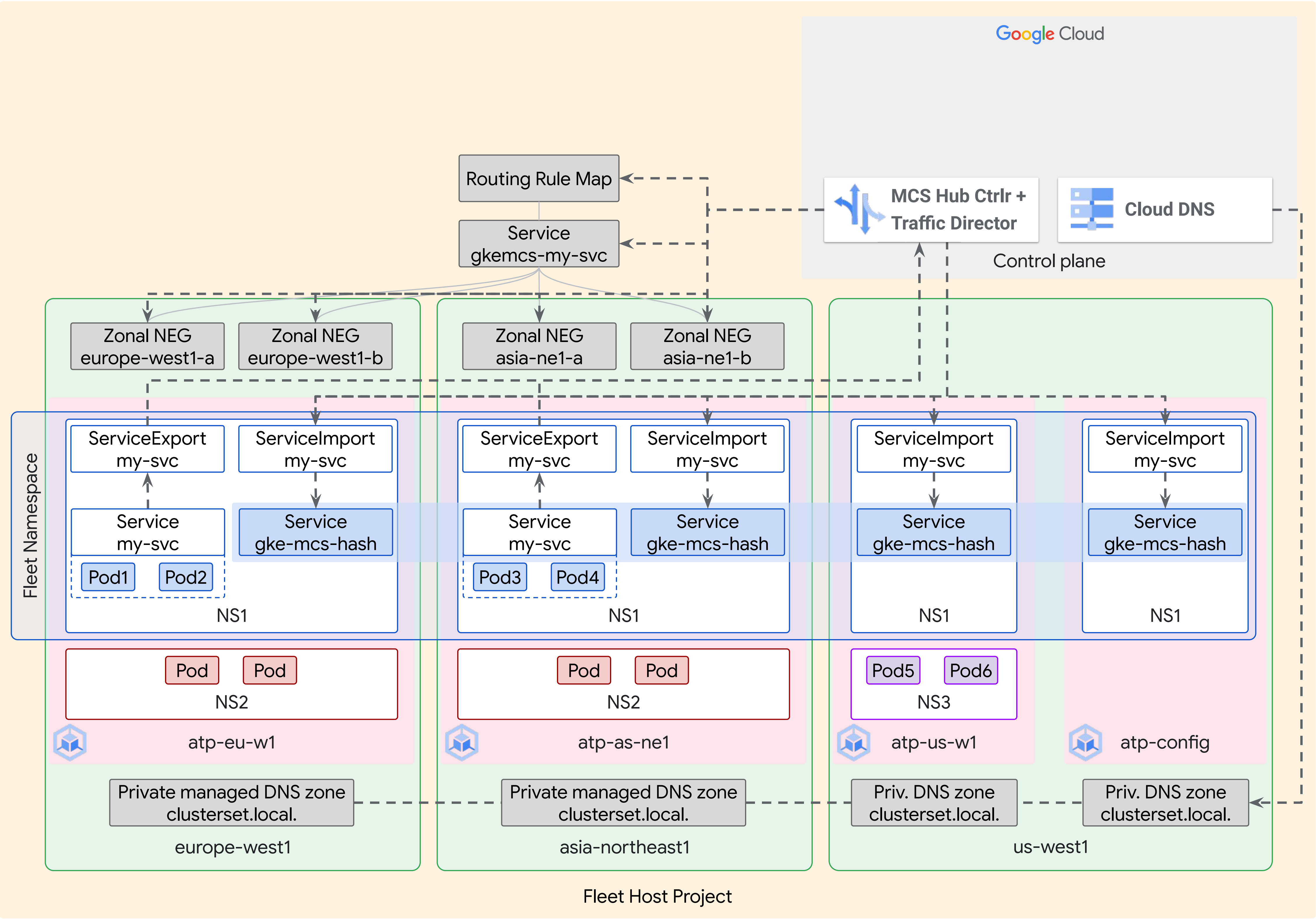
Multi-Cluster Gateway Walk-through



7

Adding a config cluster to manage your (multi-cluster) Gateway

Multi-Cluster Gateway Walk-through



7 Adding a config cluster to manage your (multi-cluster) Gateway

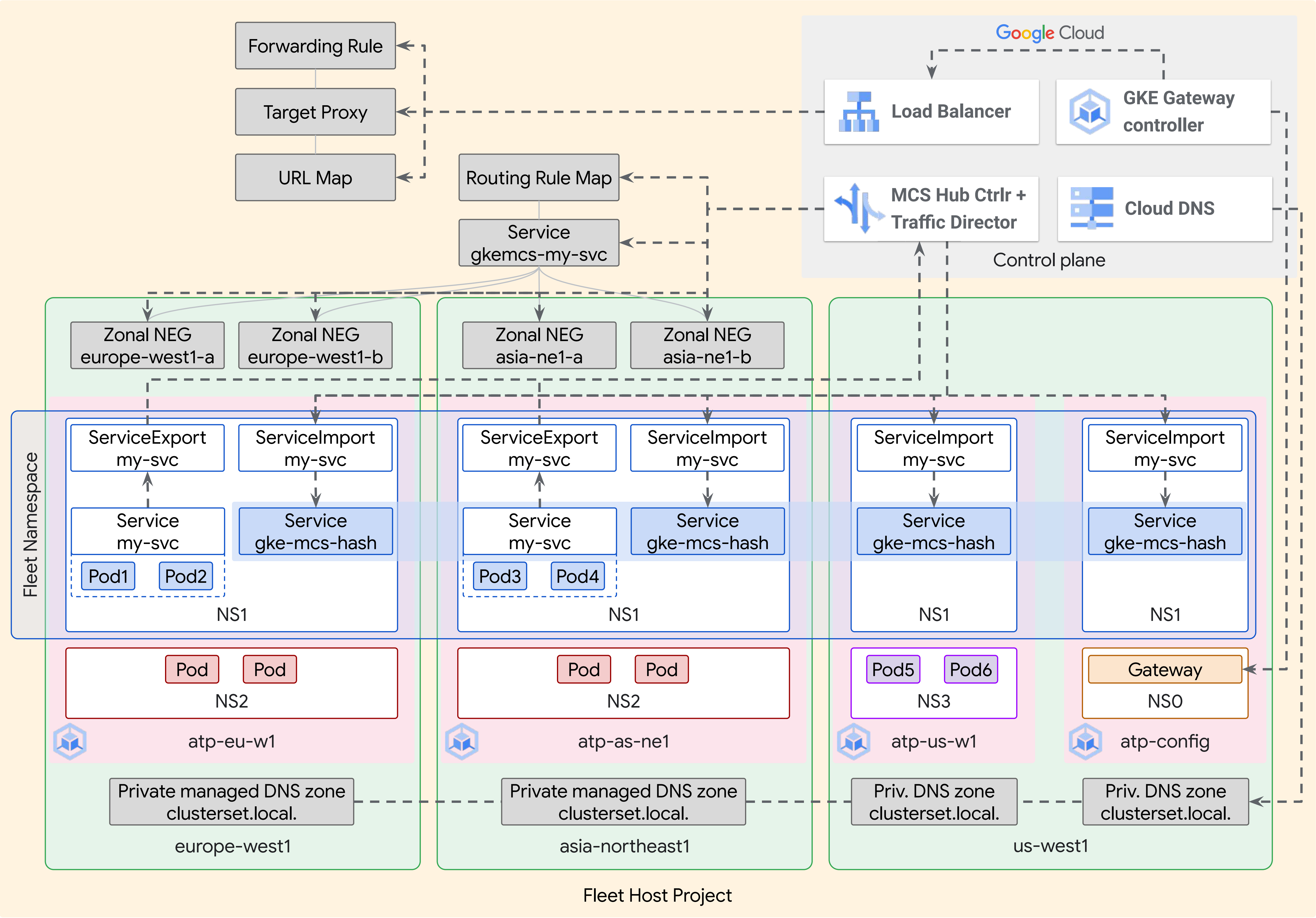
✓ New cluster (config-cluster) added to the fleet

Fleet namespace extended to the config cluster

Multi-cluster Service imported to the new cluster in the fleet namespace

Connectivity to all pods backing a multi-cluster service is automated

Multi-Cluster Gateway Walk-through



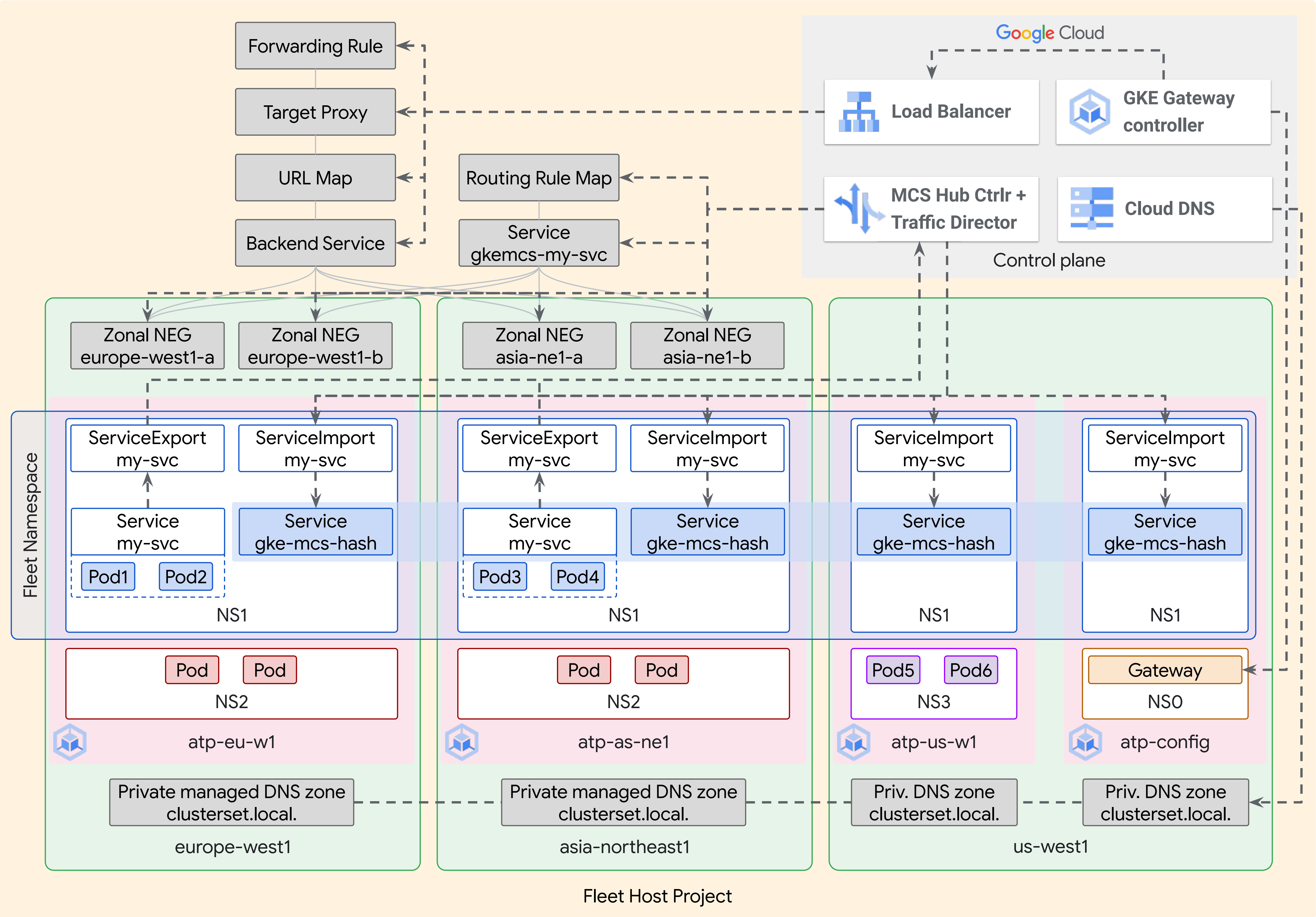
8 Multi-cluster Gateway creation

✓ Gateway resource added to the config-cluster

Google-hosted GKE Gateway controller watches Gateway APIs on the config-cluster

GKE Gateway controller creates Cloud Load Balancing resources in the fleet host project

Multi-Cluster Gateway Walk-through



9 Targeting Multi-Cluster Services with a Gateway

✓ HTTPRoute resource added to the config-cluster

Corresponding backend service created in the fleet host project

Existing Zonal NEGs attached to the backend service to enable connectivity with Google Front End network

Users can connect to the distributed service in the fleet

Ingress Traffic to GKE Fleets

With GKE Enterprise

Multi-Cluster Gateways GKEE Deployment Patterns

Shared Global Gateway

A platform-managed Gateway with global connectivity for different teams

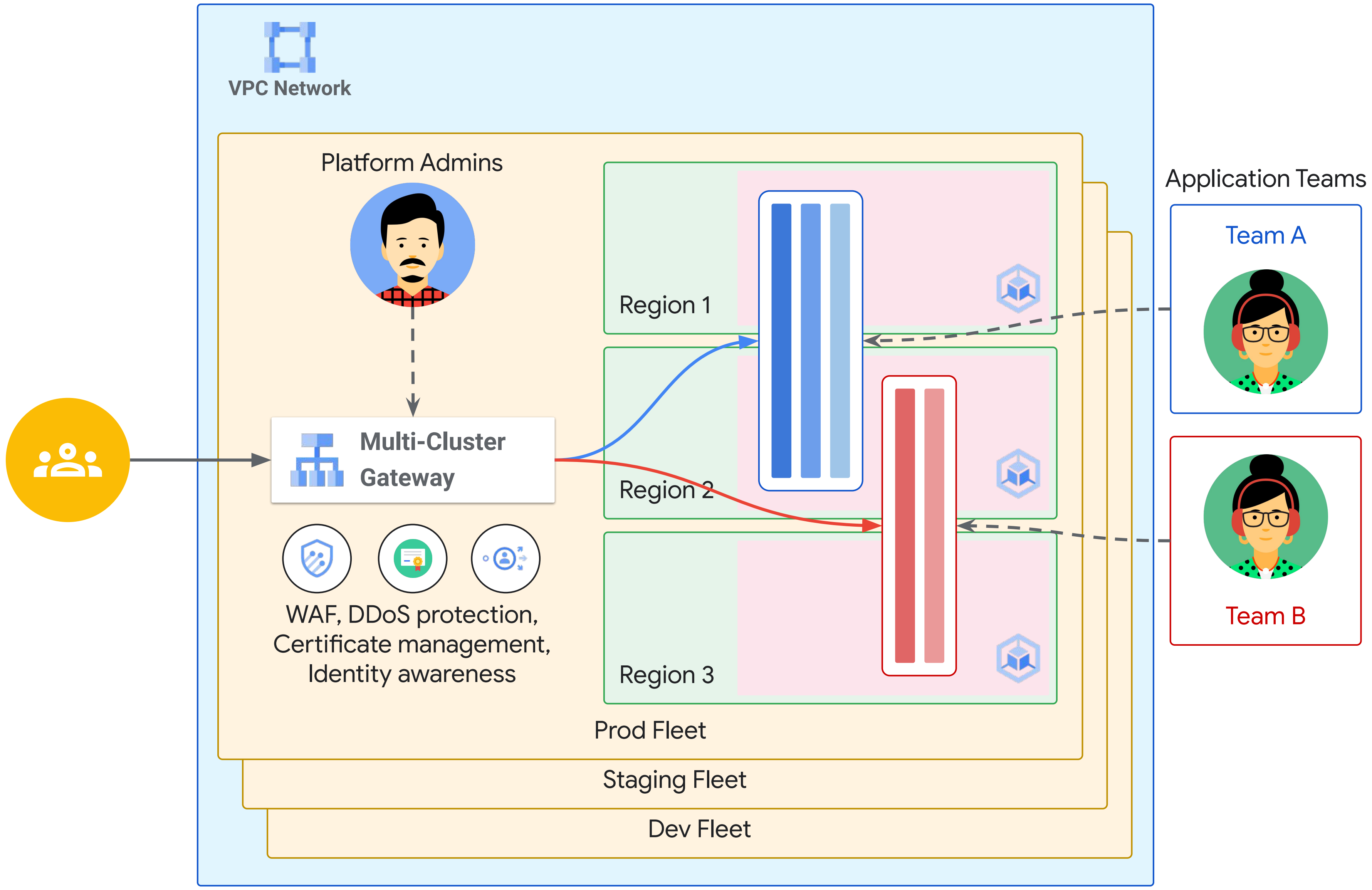
Shared Internal Regional Gateways

A set of platform-managed Gateways with regional connectivity only and DNS Failover routing policies

Dedicated Gateway per Team

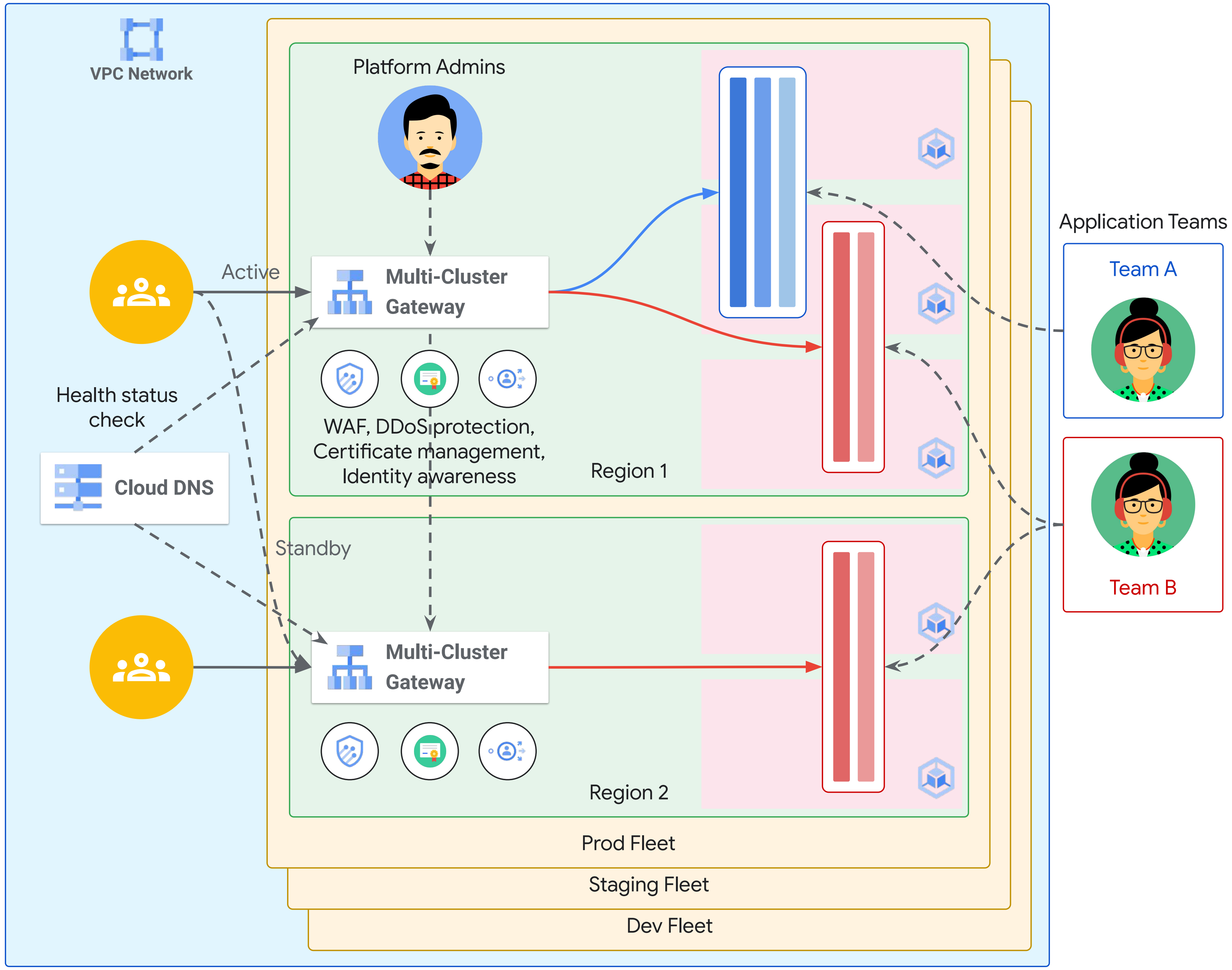
A platform-managed Gateway dedicated to a Team for more flexibility and control

Shared Global Gateway per Environment



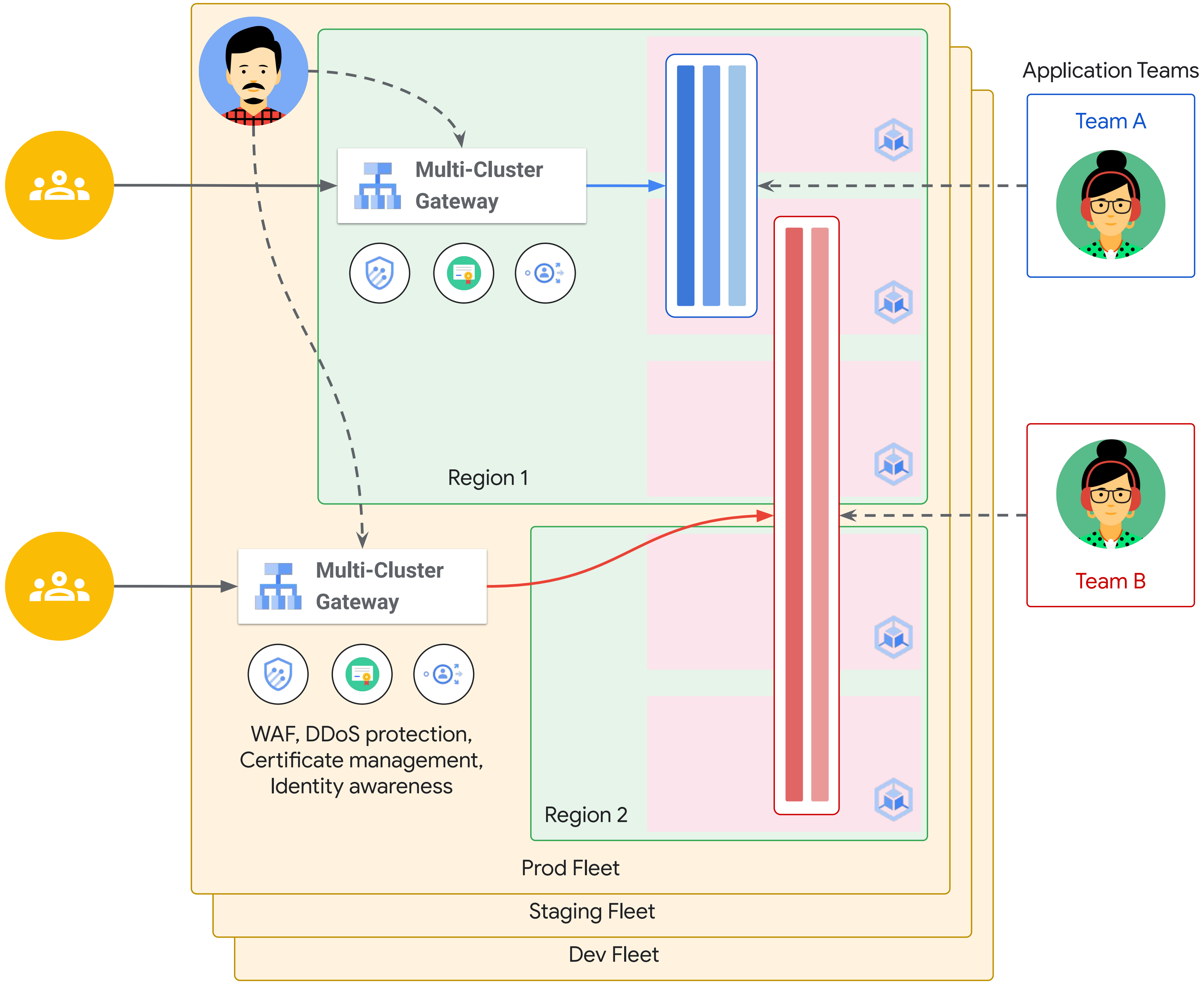
- Global Multi-Cluster Gateway managed by the Platform Team
- Routes and Services managed by the Application Teams
- Proximity routing with anycast IP and consistent security policy across the organization and environments
- One Multi-Cluster Gateway per fleet project. (Future - One Gateway for multiple cluster projects)

Shared Internal Regional Gateways per Environment



- Regional Multi-Cluster Gateways managed by the Platform Team
- Routes and Services managed by the Application Teams
- Local routing with a Gateway regional IP address and differentiated security policies per region
- Failover across regions with Global access and DNS failover routing policies, with flexible traffic shifting strategies
- Future - Cross-region backends to unlock Global Internal Gateways for GKE

Dedicated Gateway per Team



- Gateways managed by the Platform Team or the Application teams
- Routes and Services managed by the Application Teams
- Flexible design that allows some Applications to be globally distributed and some to remain local to a region

Key Takeaways

GKE Enterprise Multi-Cluster Networking is essential to help you build regionally and globally distributed secure applications

- GKE Enterprise with fleet-based multi-cluster management simplifies the operation model for a multi-tenant platform

- Multi-Cluster Services provides cross-geo redundancy for your critical services

- Multi-Cluster Gateway, a fleet-managed load balancing solution with advanced traffic management and security capabilities

- Flexible designs and deployment patterns, at the edge of the fleet and beyond, to help your platform & application teams deliver faster together

We are interested in your feedback!

Connect with a GKE/Serverless PM or UX researcher.

kubernetes turns 10!

#k8sturns10

Ready to build what's next?

Tap into **special offers** designed to help you **implement what you learned** at Google Cloud Next.

Scan the code to receive personalized guidance from one of our experts.

Or visit **g.co/next/24offers**

Thank you