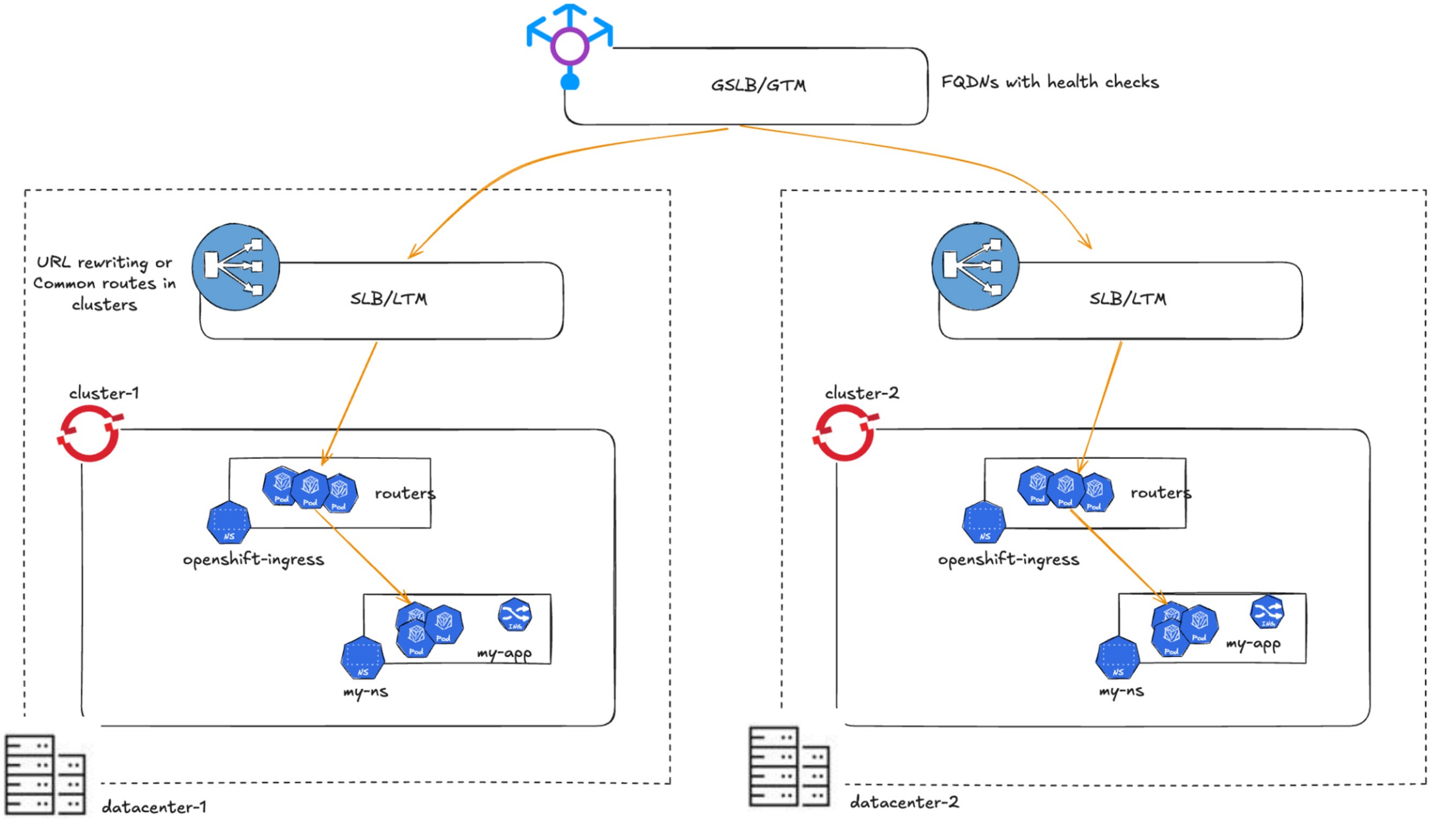
Red Hat Summit

Connect

Multi-Cluster Architecture with Service Mesh

Mervan Ileri
Senior Architect

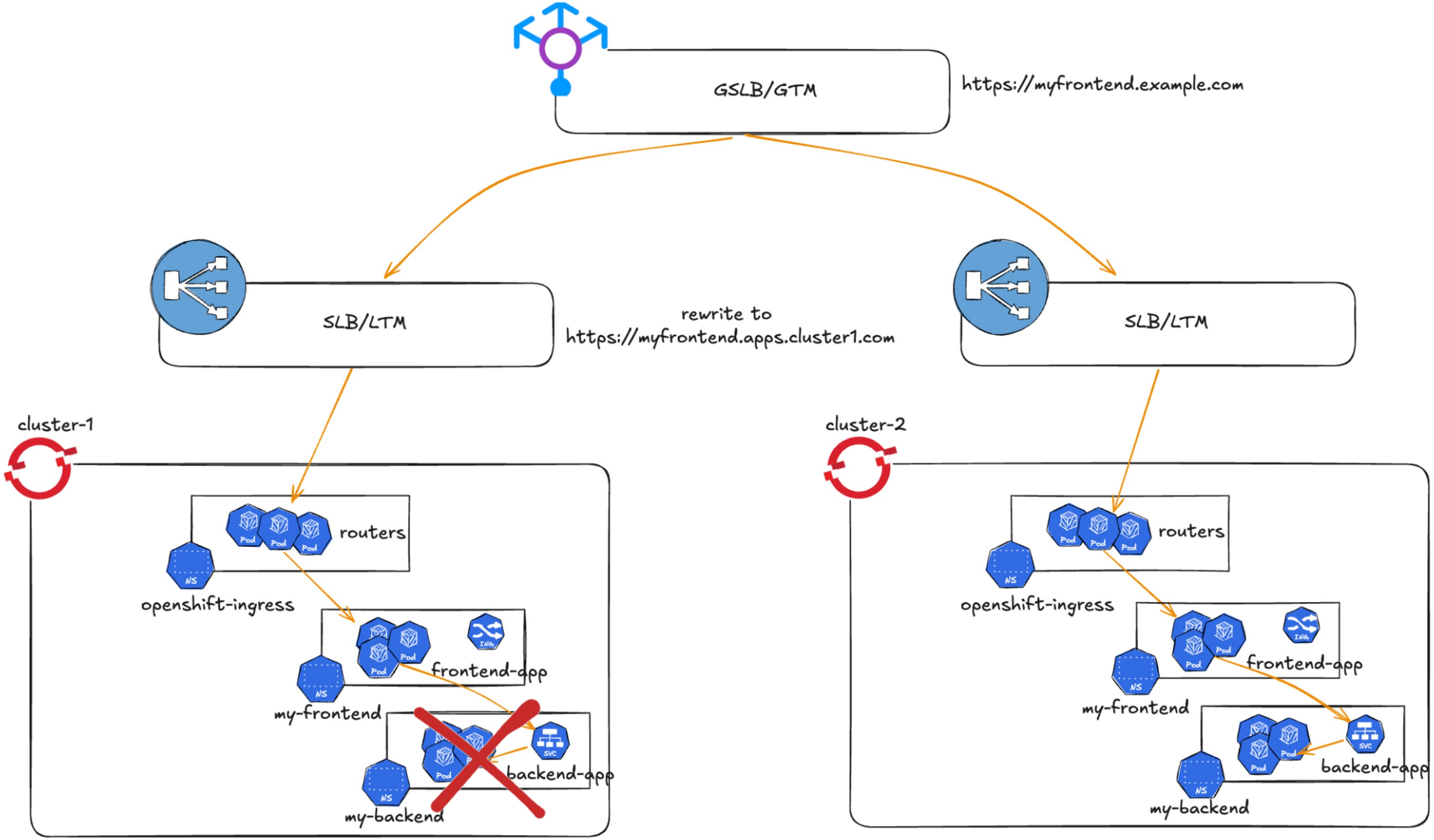
OpenShift Platform Redundancy



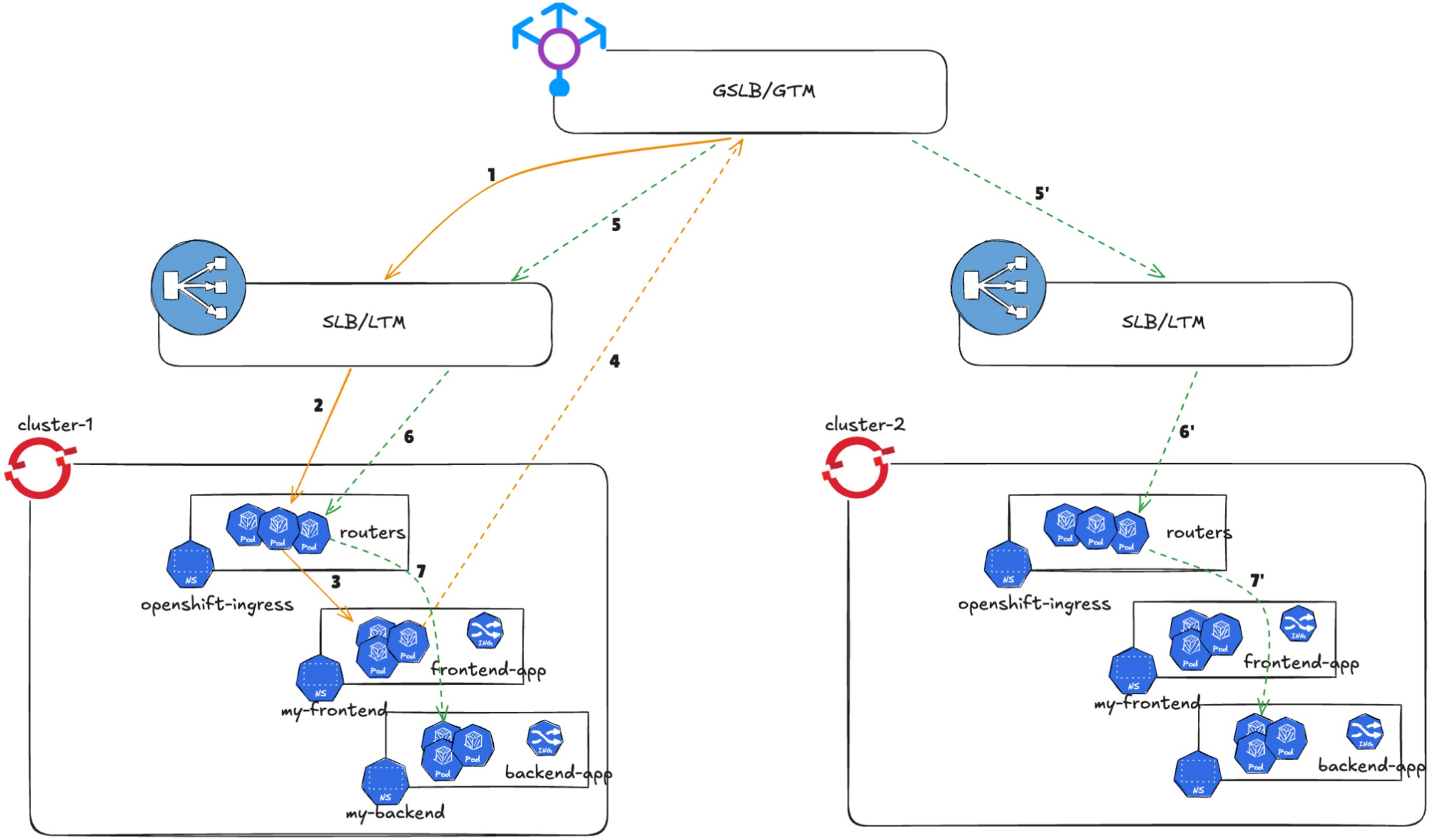
- GSLB acts as smart DNS & sends traffic to LTM.
- LTM can rewrite or send the traffic directly to the cluster.
- In case LTM sends directly, clusters will have a common Route/Ingress definition.
- Health checks for directing traffic to the healthy cluster.

https://www.redhat.com/en/blog/deploying-openshift-applications-multiple-datacenters

Example Scenario: Traffic Flow in an OpenShift Multi-Cluster Setup



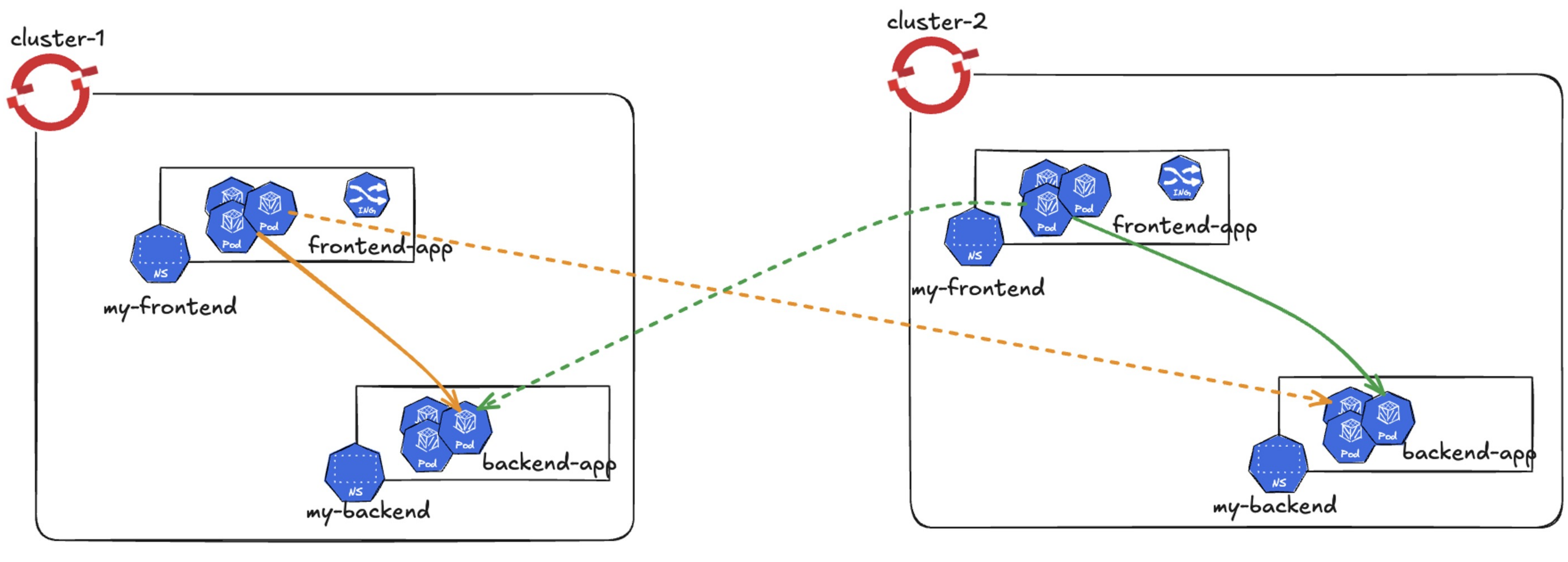
The Limits of the Redundancy Model



- ➢ Needless Ingress/Route definition.
- ➢ Looser security.
- ➢ Traffic management is complicated.

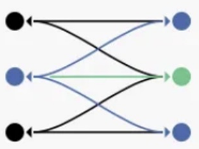
* Redundancy is at cluster level not at **application** level.

How to solve it?



Make backend application accessible to fronte nd in a secure manner and allowing for traffic management scenarios.

Istio Overview

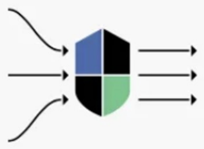


Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.

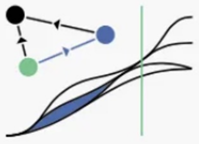
Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



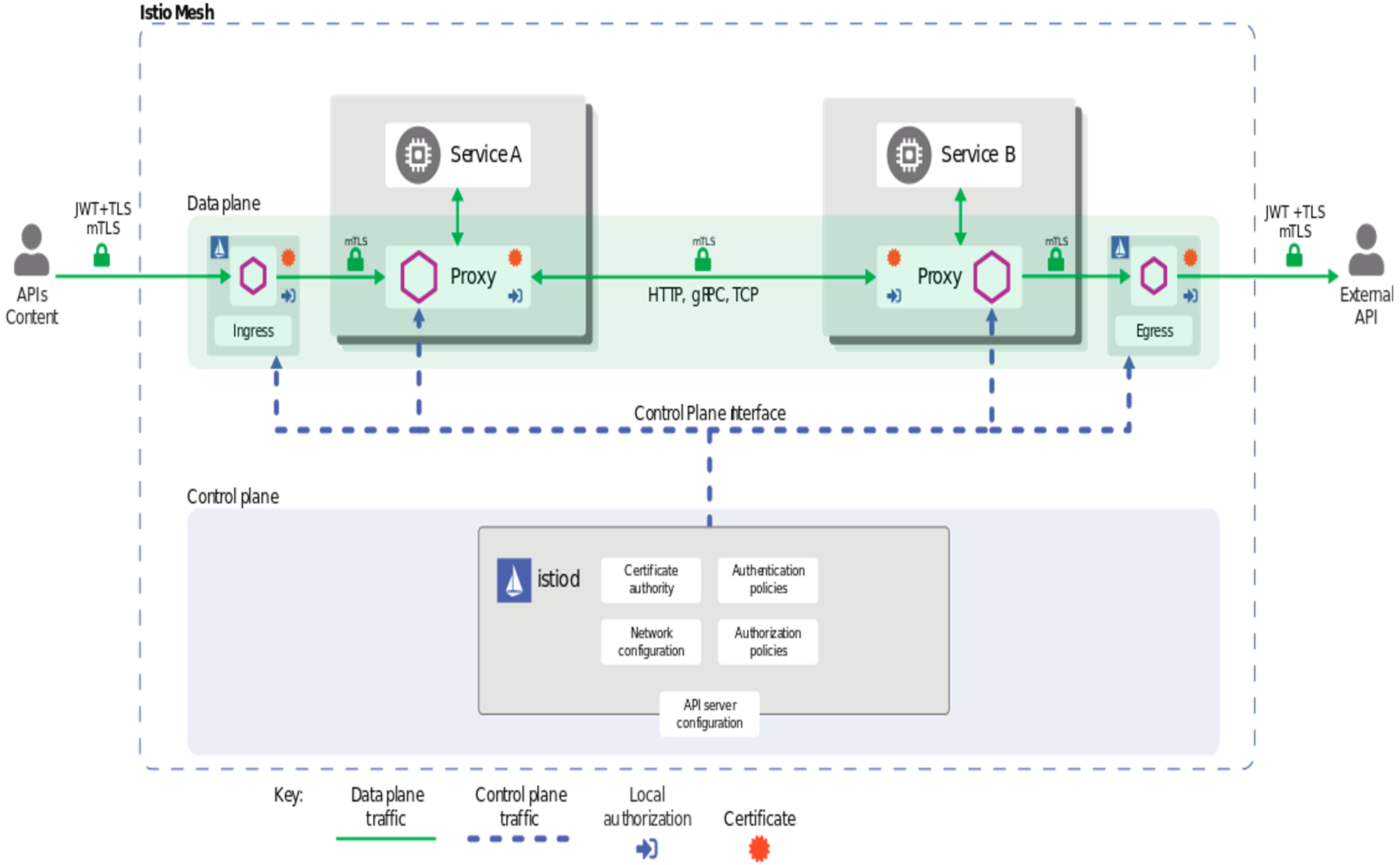
Control

Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.



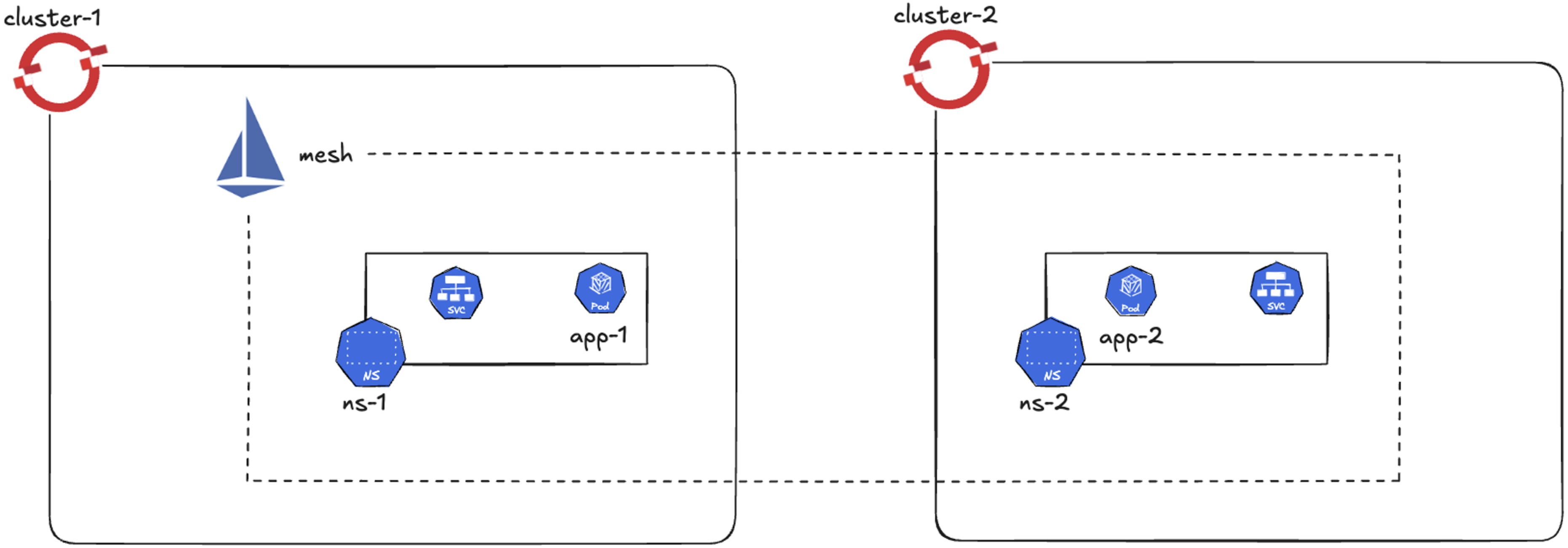
Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.

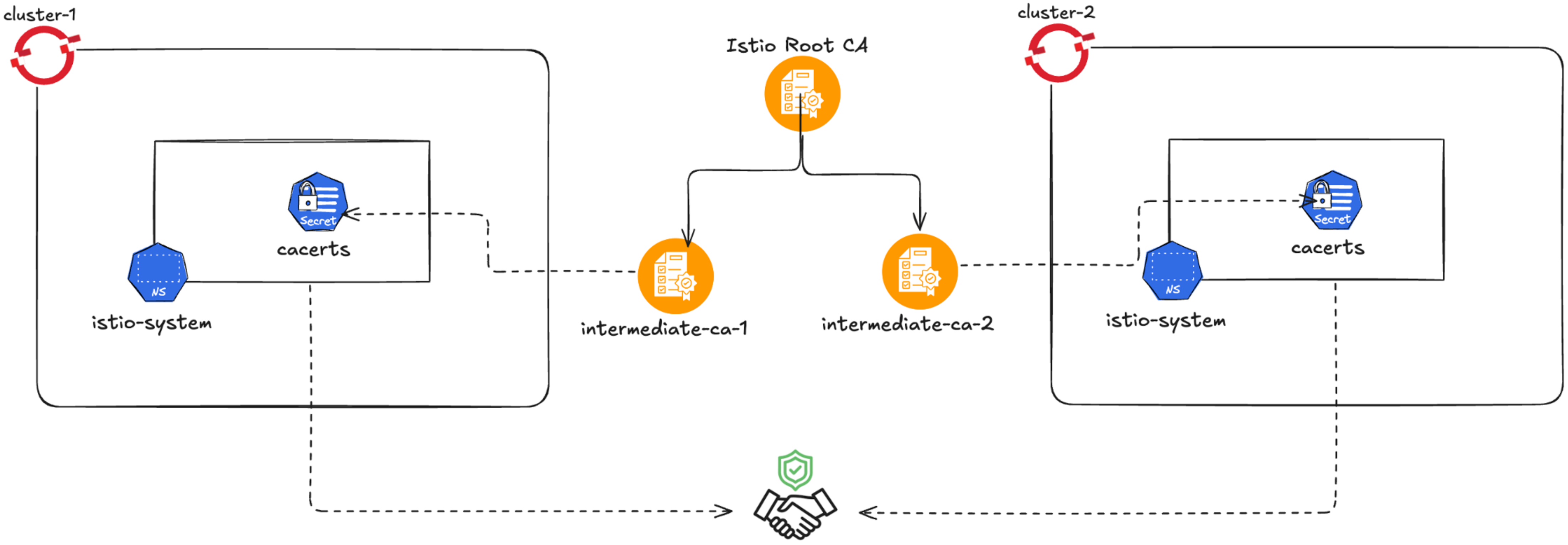


Istio Multicluster Deployment Models

- ➢ Multi-primary deployment model.
- ➢ Primary/remote deployment model.
- ➢ External control plane model.

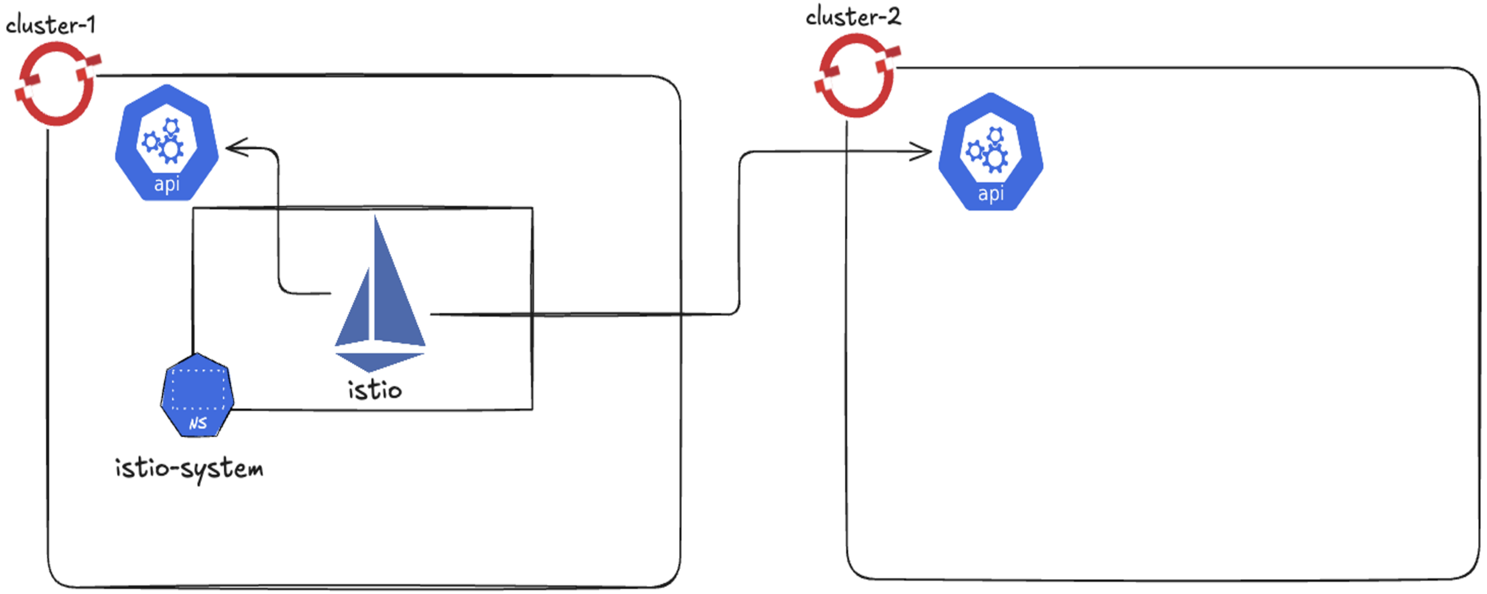


Requirement: Common Trust



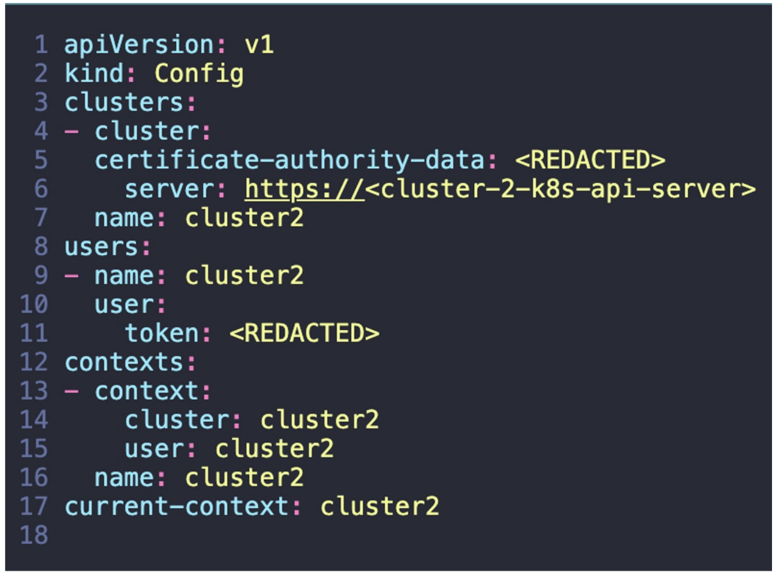
```
kubectl create secret generic cacerts -n istio-system \
  --from-file=west/ca-cert.pem \
  --from-file=west/ca-key.pem \
  --from-file=west/root-cert.pem \
  --from-file=west/cert-chain.pem
```

Requirement: Discovery

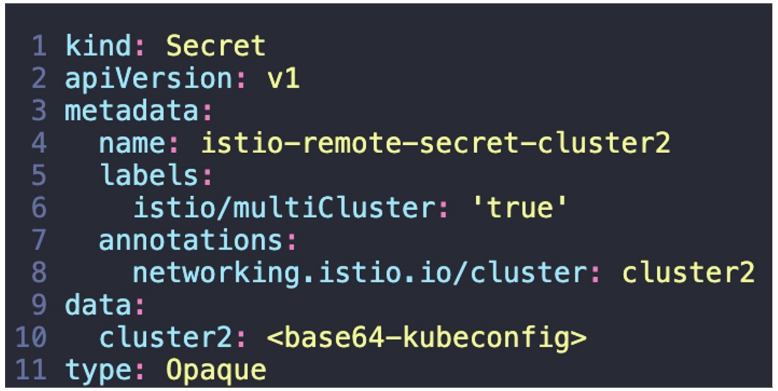


```
istioctl create-remote-secret \
  --context="${CLUSTER2}" --name=cluster-2 | \
  kubectl apply -f - --context="${CLUSTER1}"
```

Through `istio-reader-service-account`.

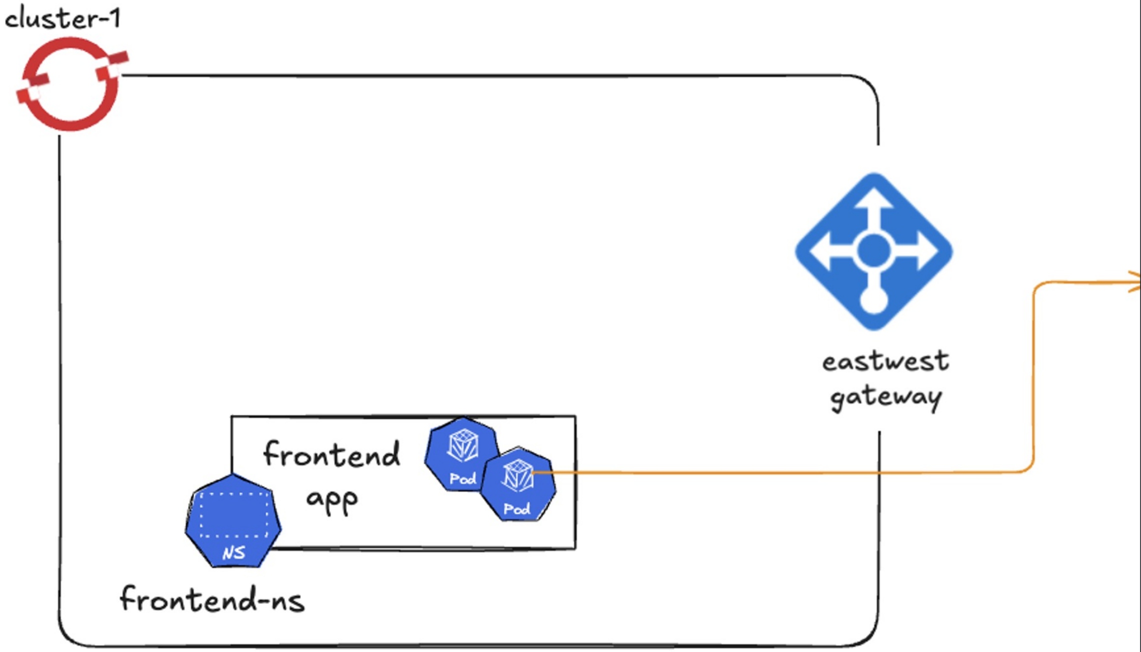


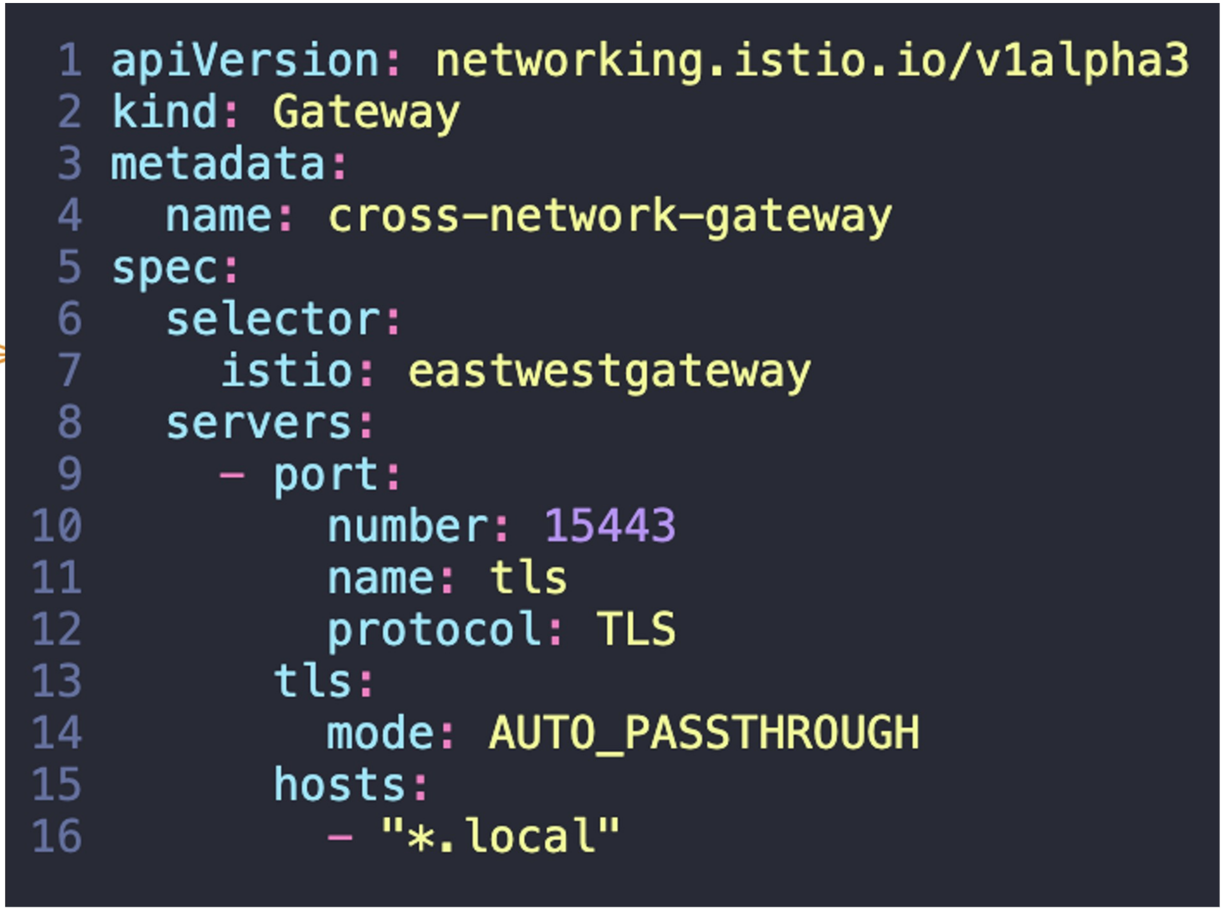
```
apiVersion: v1
kind: Config
clusters:
- cluster:
  certificate-authority-data: <REDACTED>
    server: https://<cluster-2-k8s-api-server>
  name: cluster2
users:
- name: cluster2
  user:
    token: <REDACTED>
contexts:
- context:
    cluster: cluster2
    user: cluster2
  name: cluster2
current-context: cluster2

```



```
kind: Secret
apiVersion: v1
metadata:
  name: istio-remote-secret-cluster2
  labels:
    istio/multiCluster: 'true'
  annotations:
    networking.istio.io/cluster: cluster2
data:
  cluster2: <base64-kubeconfig>
type: Opaque
```

Requirement: Connectivity

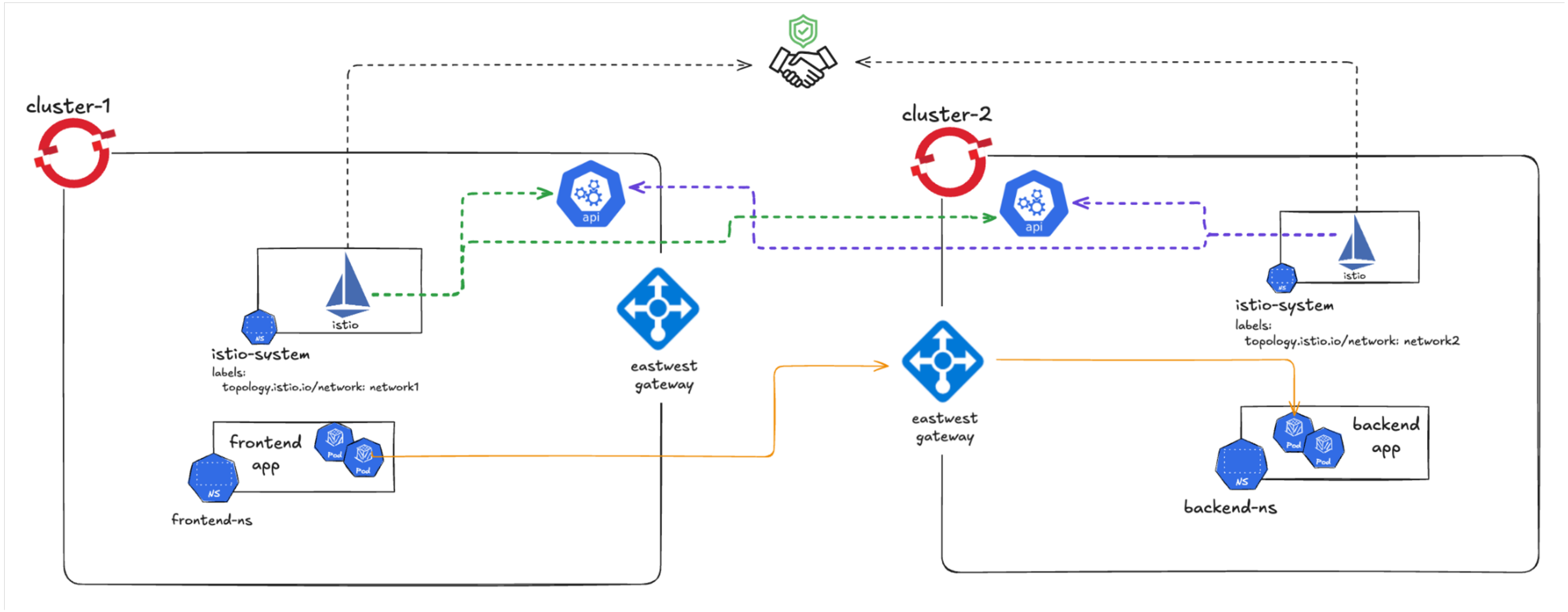




```yaml
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: cross-network-gateway
spec:
  selector:
    istio: eastwestgateway
  servers:
    - port:
        number: 15443
        name: tls
        protocol: TLS
      tls:
        mode: AUTO_PASSTHROUGH
      hosts:
        - "*.local"
```

- ➢ East-West gateway stores SNI clusters.
- ➢ For a service: direction, port, subset, FQDN encoded into SNI.
  - `outbound|8080|v1|backend-svc.backend-ns.svc.cluster.local` turns into:
    `outbound_.8080_.v1_.backend-svc.backend-ns.svc.cluster.local.`
- ➢ Expose services in the cluster.

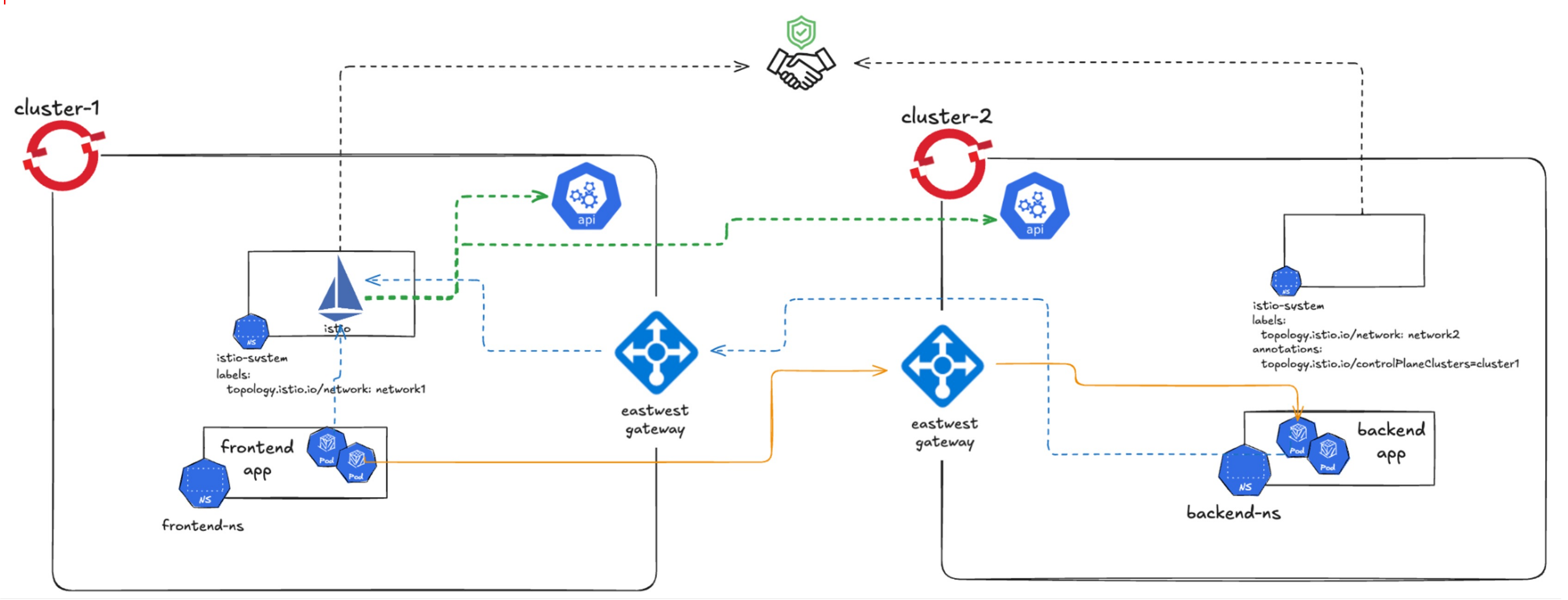
Multicluster: Multi-primary Model



- ➢ Separate Istio installations in both clusters.
- ➢ Multiple points of control plane redundancy.

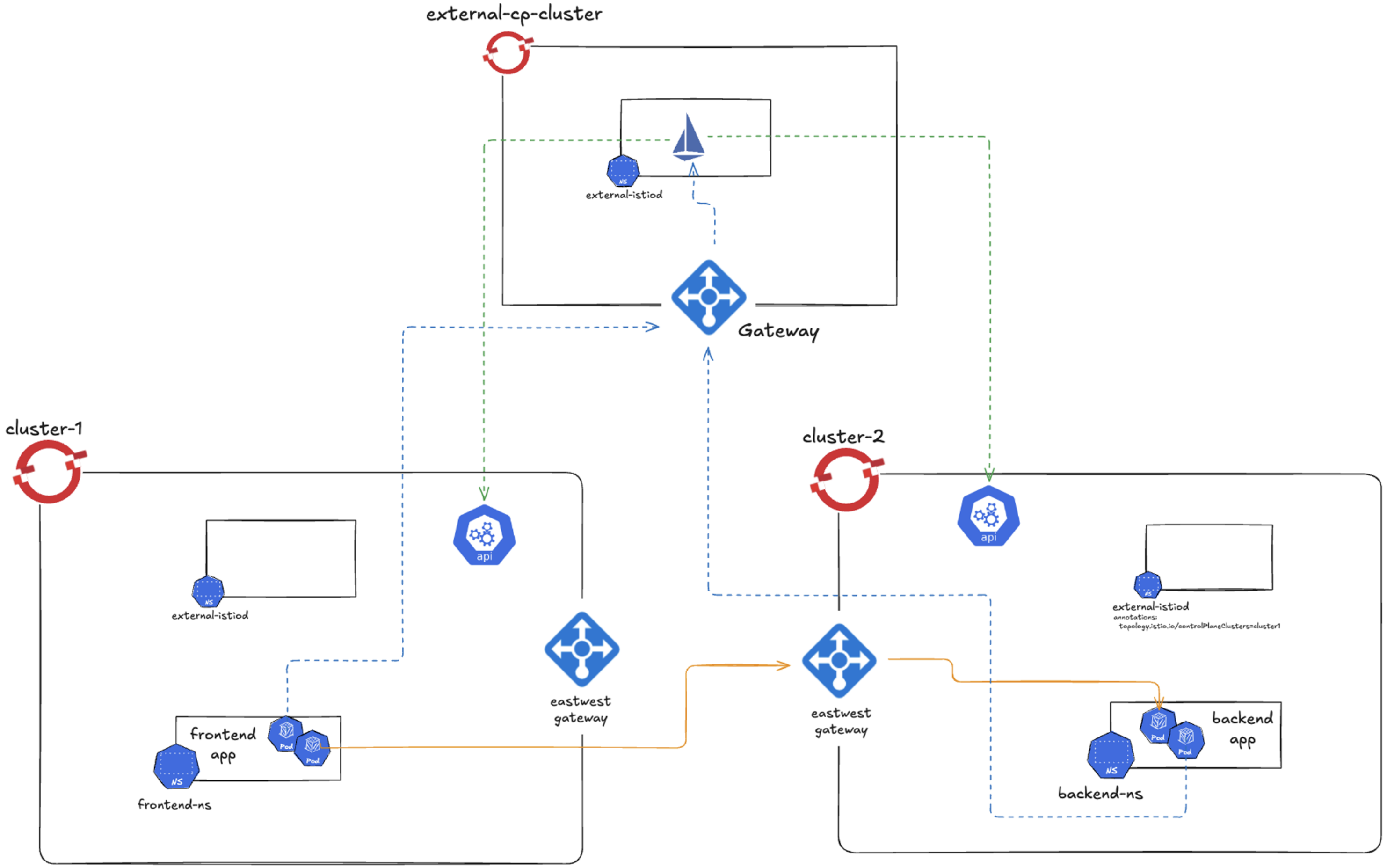
Red Hat

Multicluster: Primary/Remote Model



- One control plane on cluster1.
- No control plane on cluster2.

Multicluster: External Control Plane Model

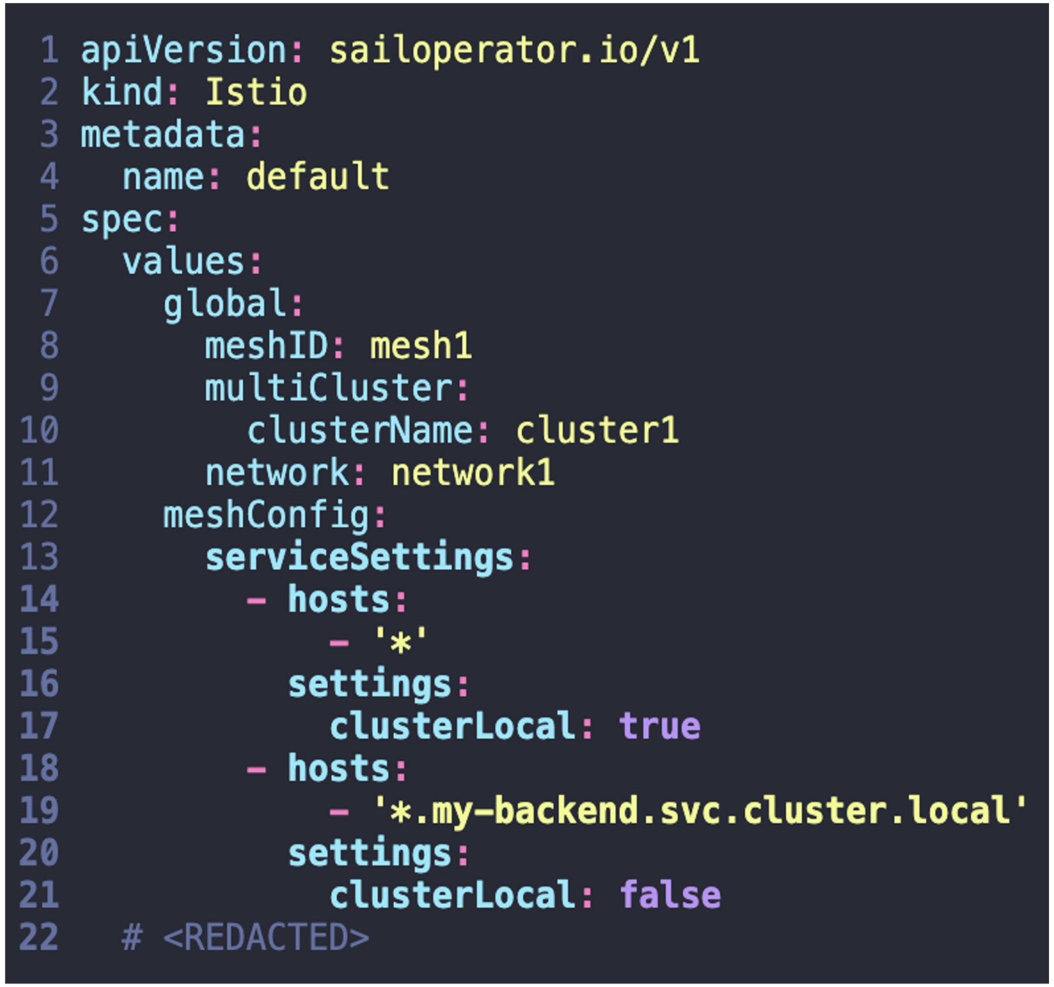


- ➢ Dedicated cluster for istio installations.
- ➢ Cluster can serve for more than one istio installations.

Istio Multi-Cluster Architecture Comparison

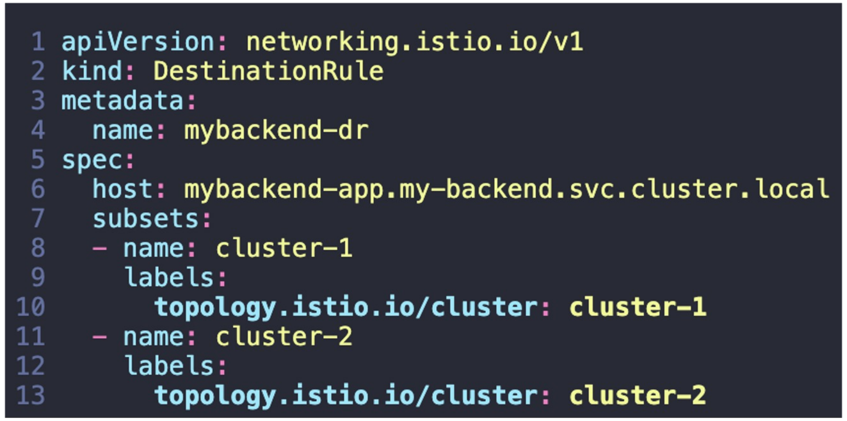
Architecture	Core Goal & Use Case	Pros	Cons
Primary / Remote	Simplicity & Efficiency. Best for single-region, low-latency networks.	✓ Low operational cost ✓ Simplest setup ✓ Minimal resource consumption	✗ Single point of failure (CP) ✗ Sensitive to network latency ✗ Not for unstable networks
Multi-primary	High Availability & Fault Isolation. Best for multi-region, geo-distributed.	✓ High Availability (No SPOF) ✓ Excellent fault isolation	✗ Highest resource cost. ✗ Highest operational complexity
External CP	Operational Separation. 'Istio-as-a-Service' for platform teams.	✓ Separation of concerns ✓ Isolated CP resources ✓ Centralized policy enforcement	✗ Requires dedicated CP cluster ✗ Complex initial setup ✗ High cost for small deployments

Multi-Cluster Traffic Management 1/2

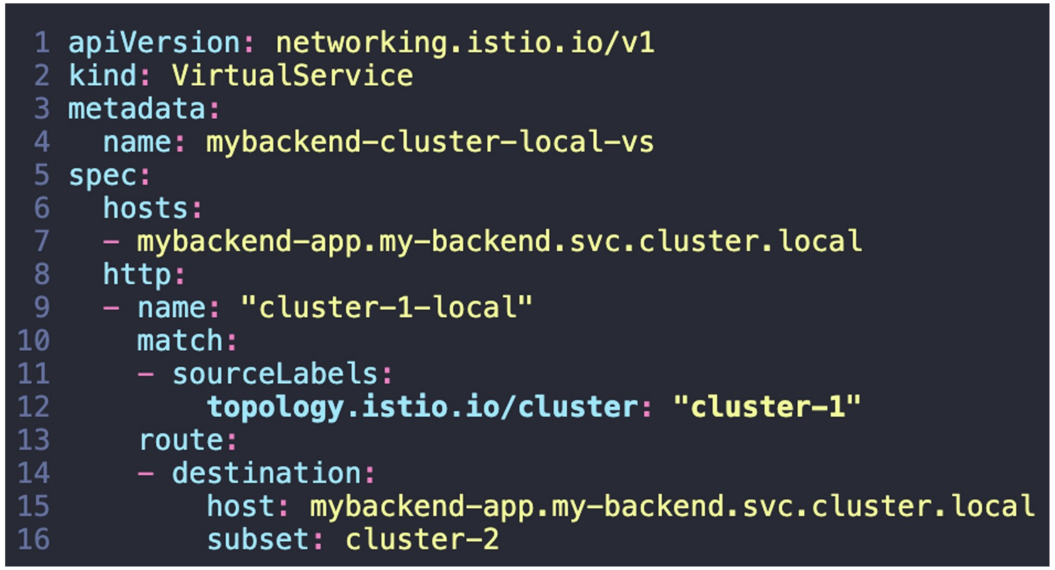


```yaml
apiVersion: sailoperator.io/v1
kind: Istio
metadata:
  name: default
spec:
  values:
    global:
      meshID: mesh1
      multiCluster:
        clusterName: cluster1
      network: network1
    meshConfig:
      serviceSettings:
        - hosts:
            - '*'
          settings:
            clusterLocal: true
        - hosts:
            - '*.my-backend.svc.cluster.local'
          settings:
            clusterLocal: false
  # <REDACTED>
```

Keeping traffic local with exceptions



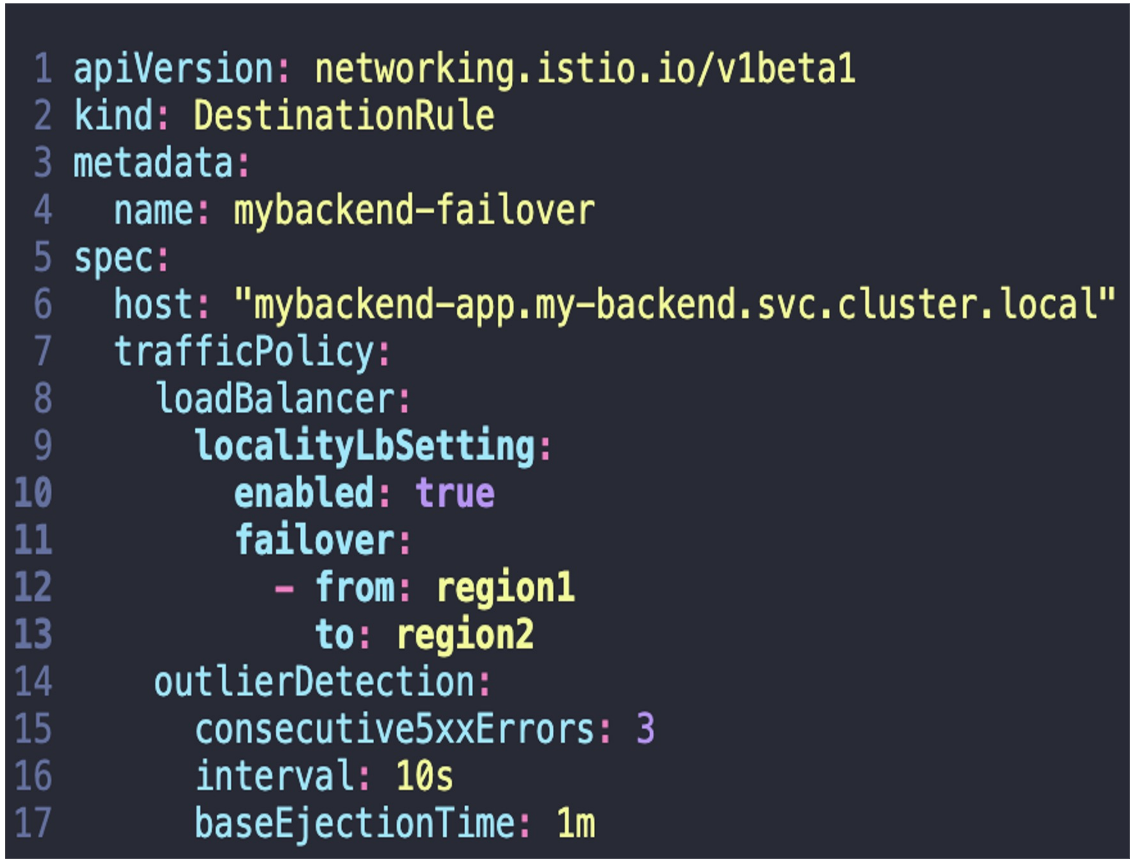
```yaml
apiVersion: networking.istio.io/v1
kind: DestinationRule
metadata:
  name: mybackend-dr
spec:
  host: mybackend-app.my-backend.svc.cluster.local
  subsets:
  - name: cluster-1
    labels:
      topology.istio.io/cluster: cluster-1
  - name: cluster-2
    labels:
      topology.istio.io/cluster: cluster-2
```



```yaml
apiVersion: networking.istio.io/v1
kind: VirtualService
metadata:
  name: mybackend-cluster-local-vs
spec:
  hosts:
  - mybackend-app.my-backend.svc.cluster.local
  http:
  - name: "cluster-1-local"
    match:
    - sourceLabels:
        topology.istio.io/cluster: "cluster-1"
    route:
    - destination:
        host: mybackend-app.my-backend.svc.cluster.local
        subset: cluster-2
```

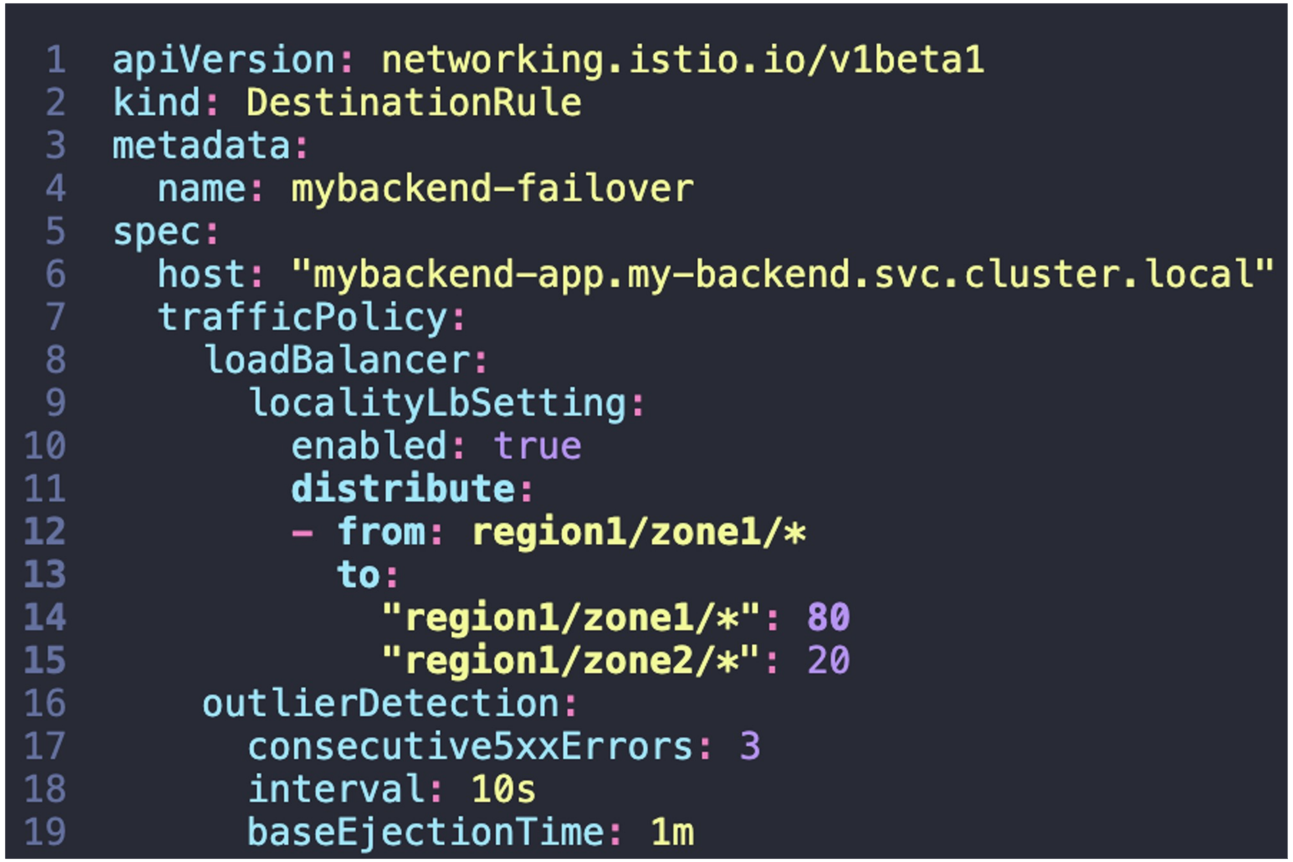
Shift traffic via service partitioning

Multi-Cluster Traffic Management 2/2



```yaml
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: mybackend-failover
spec:
  host: "mybackend-app.my-backend.svc.cluster.local"
  trafficPolicy:
    loadBalancer:
      localityLbSetting:
        enabled: true
        failover:
          - from: region1
            to: region2
    outlierDetection:
      consecutive5xxErrors: 3
      interval: 10s
      baseEjectionTime: 1m
```

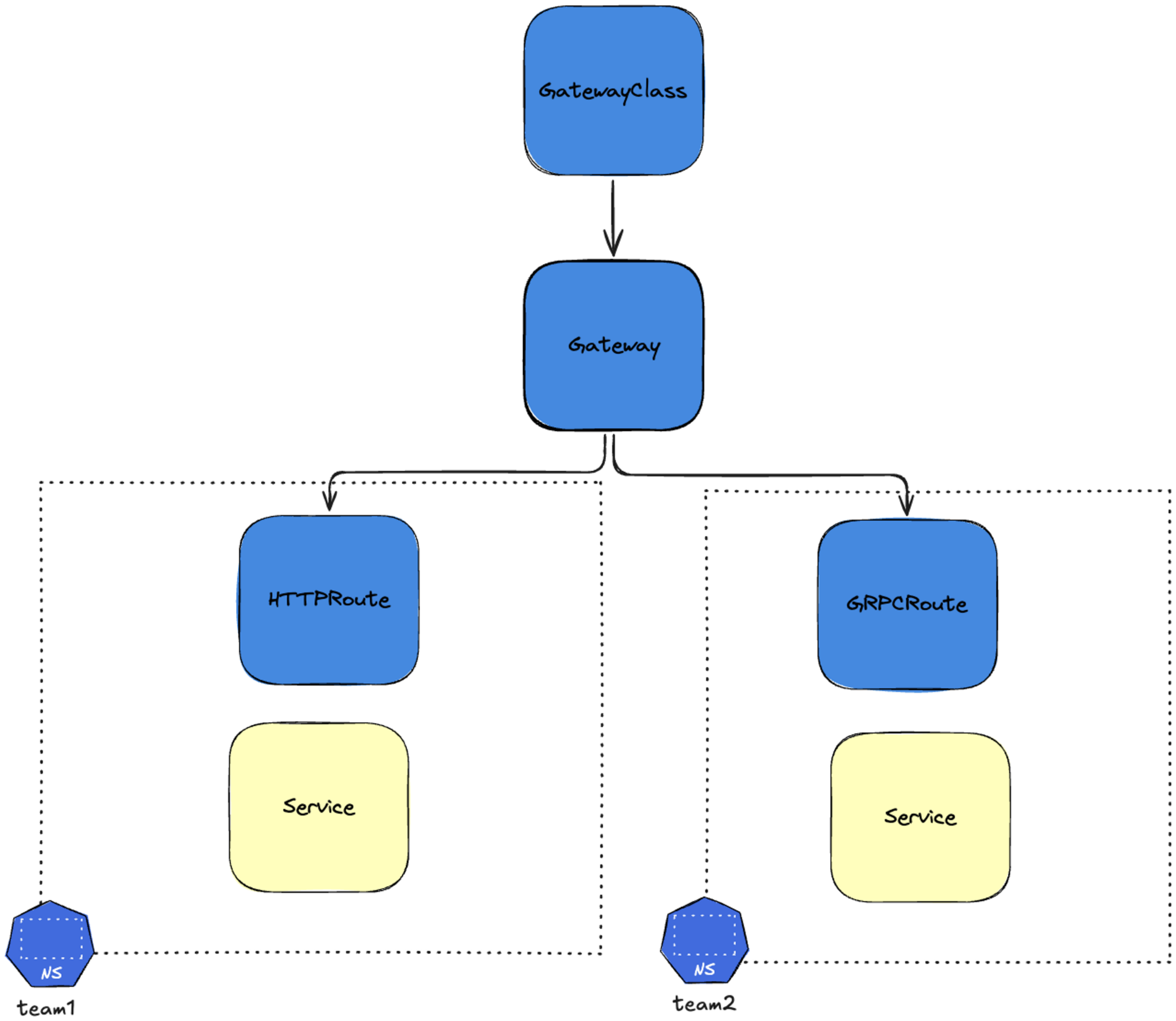
Locality with failover



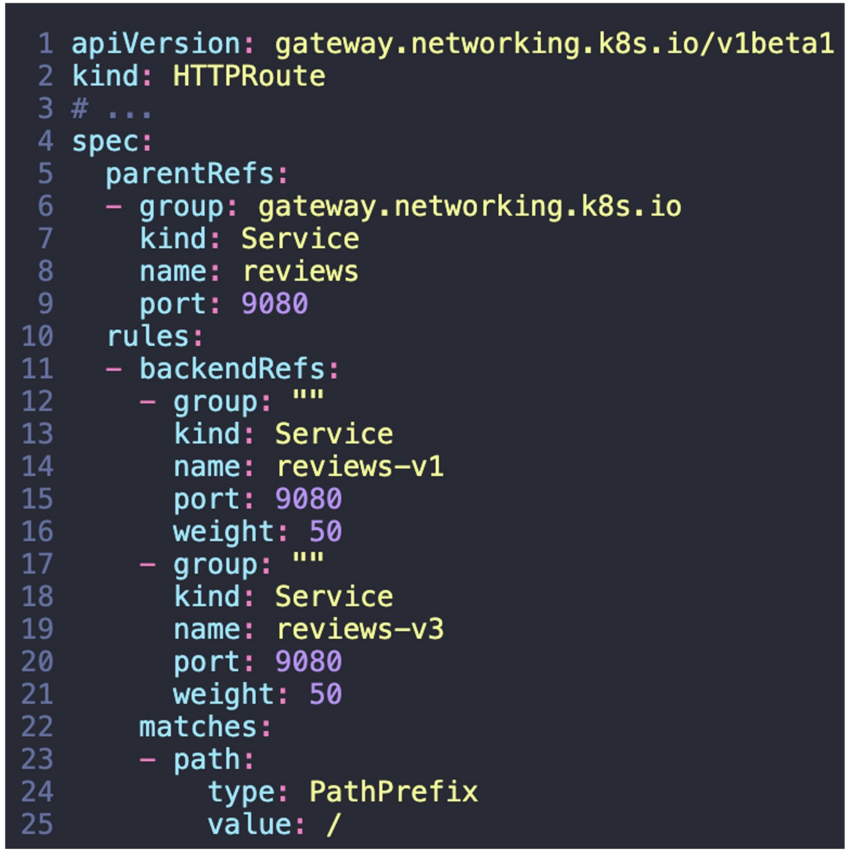
```yaml
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: mybackend-failover
spec:
  host: "mybackend-app.my-backend.svc.cluster.local"
  trafficPolicy:
    loadBalancer:
      localityLbSetting:
        enabled: true
        distribute:
        - from: region1/zone1/*
          to:
            "region1/zone1/*": 80
            "region1/zone2/*": 20
    outlierDetection:
      consecutive5xxErrors: 3
      interval: 10s
      baseEjectionTime: 1m
```

Locality with distribution

New Feature: K8S Gateway API

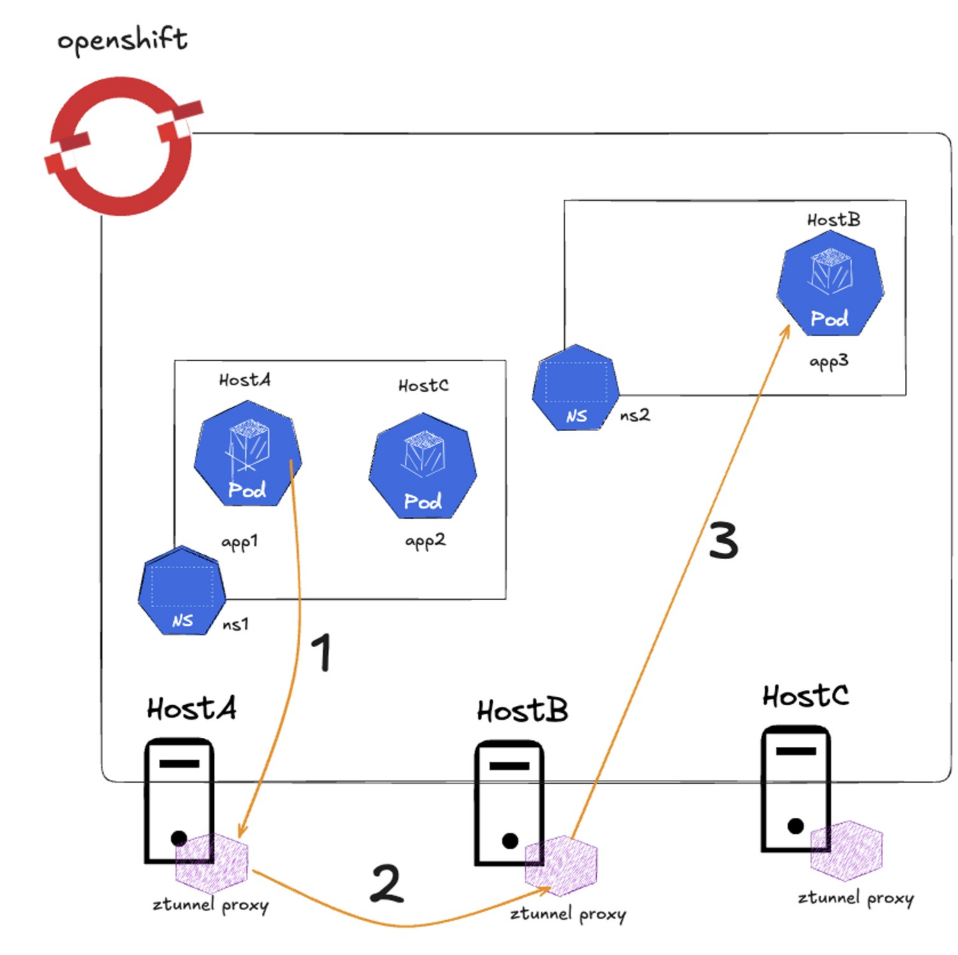


Gateway API Version 1.2.1

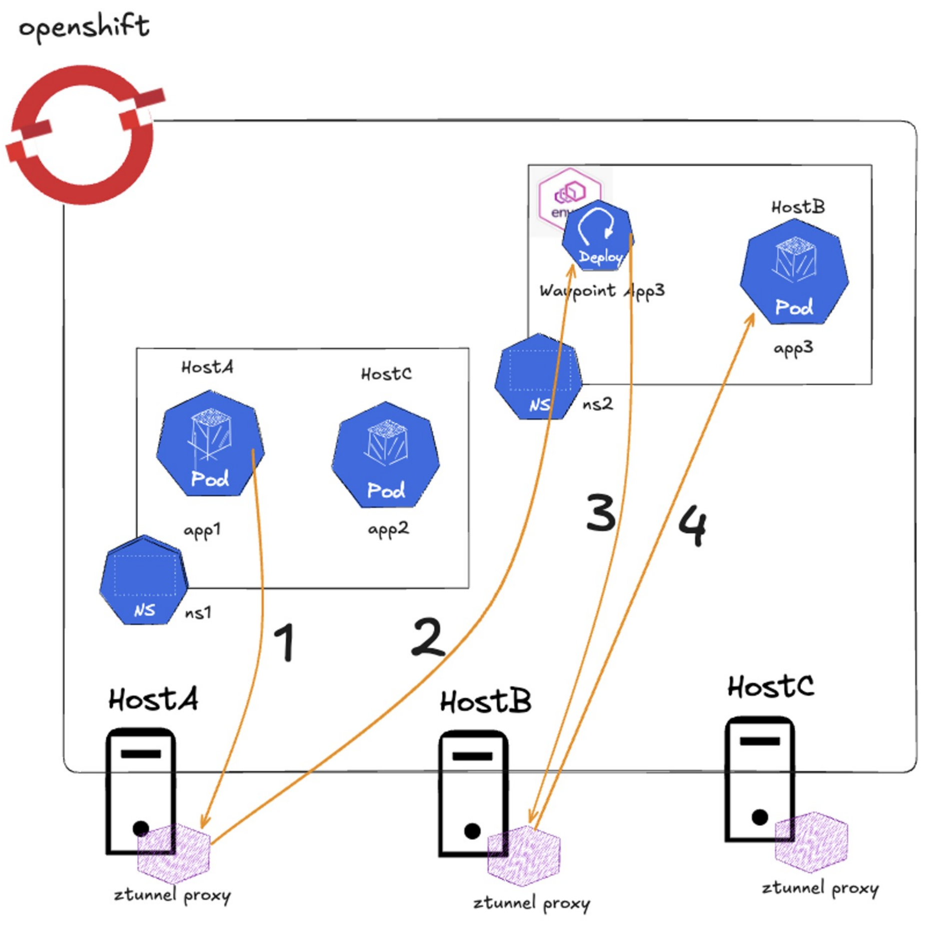


```yaml
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
# ...
spec:
  parentRefs:
  - group: gateway.networking.k8s.io
    kind: Service
    name: reviews
    port: 9080
  rules:
  - backendRefs:
    - group: ""
      kind: Service
      name: reviews-v1
      port: 9080
      weight: 50
    - group: ""
      kind: Service
      name: reviews-v3
      port: 9080
      weight: 50
    matches:
    - path:
        type: PathPrefix
        value: /
```

New Feature: Istio Ambient Mesh



For Layer 4 features, only ZTunnel proxies are used.



For Layer 7 features, use waypoint proxies.

Red Hat Summit

Connect

Thank you

in linkedin.com/company/red-hat

f facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat