How to survive and thrive in a multi-cluster world

Ivan Porta
Customer Engineer @ Buoyant

A little bit about Buoyant?

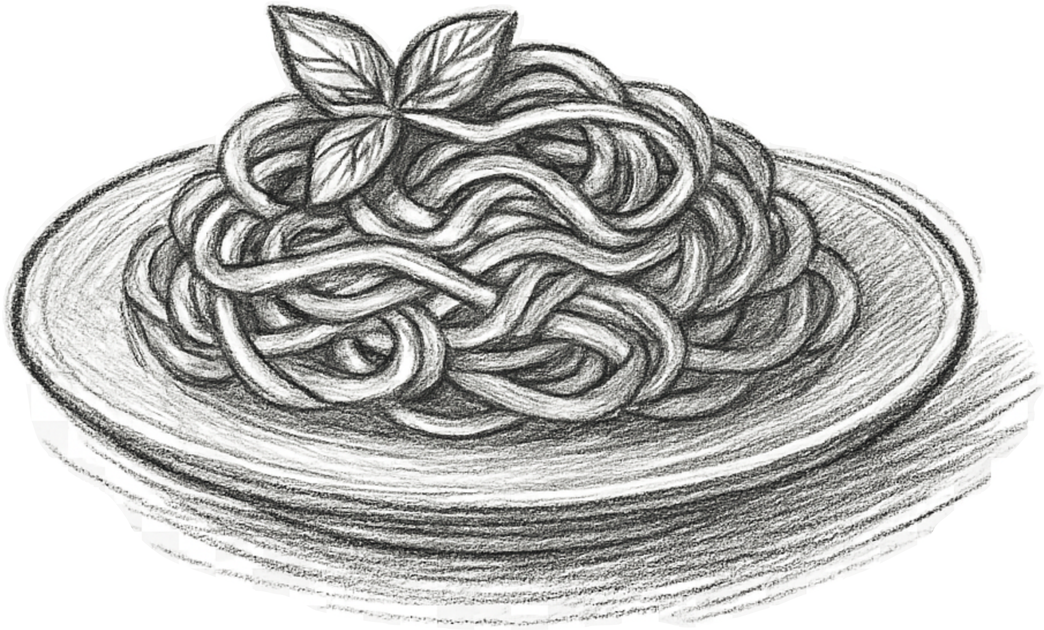
What is Buoyant Enterprise for Linkerd?

A tailored set of offerings for any size of enterprise

- **Buoyant Enterprise for Linkerd (BEL)** is the enterprise distribution of **Linkerd**, brought to you by Buoyant, the creators and maintainers of Linkerd
- BEL is a hardened distribution of Linkerd with additional tools, features, and testing designed for sustained production use
- Lifecycle automation
- Authorization policy generation
- High Availability Zonal Load Balancing (HAZL)
- FIPS-validated encryption
- Software Bills of Materials (SBOMs)
- External workload automation
- Buoyant Cloud
- BEL is **free** to use in non-production environments
- BEL is also **free** to use in production at companies with fewer than 50 employees
- Companies with 50 or more employees must purchase a license to run BEL in production

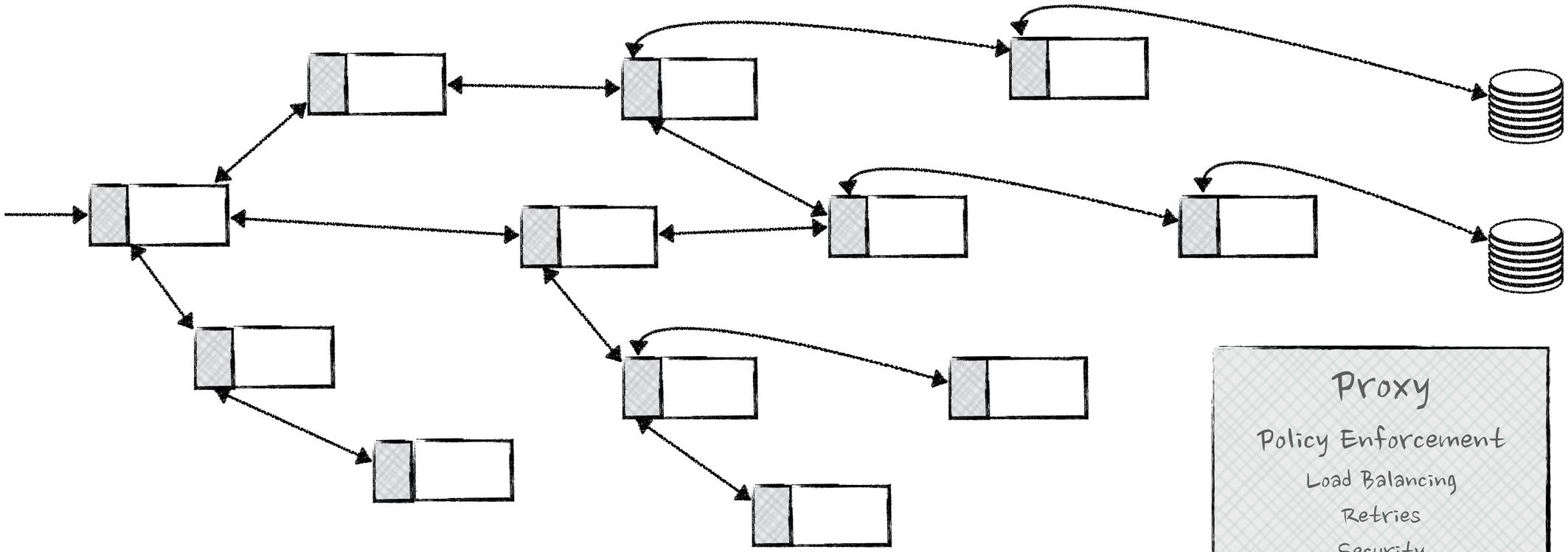
What Is a Service Mesh?

A service mesh is a dedicated **infrastructure layer** for handling service-to-service communication typically implemented as an array of lightweight network **proxies** that are deployed alongside application code, without the application needing to be aware.



What Is a Service Mesh?

Decouple Operations From Development



Proxy

Policy Enforcement

Load Balancing

Retries

Security

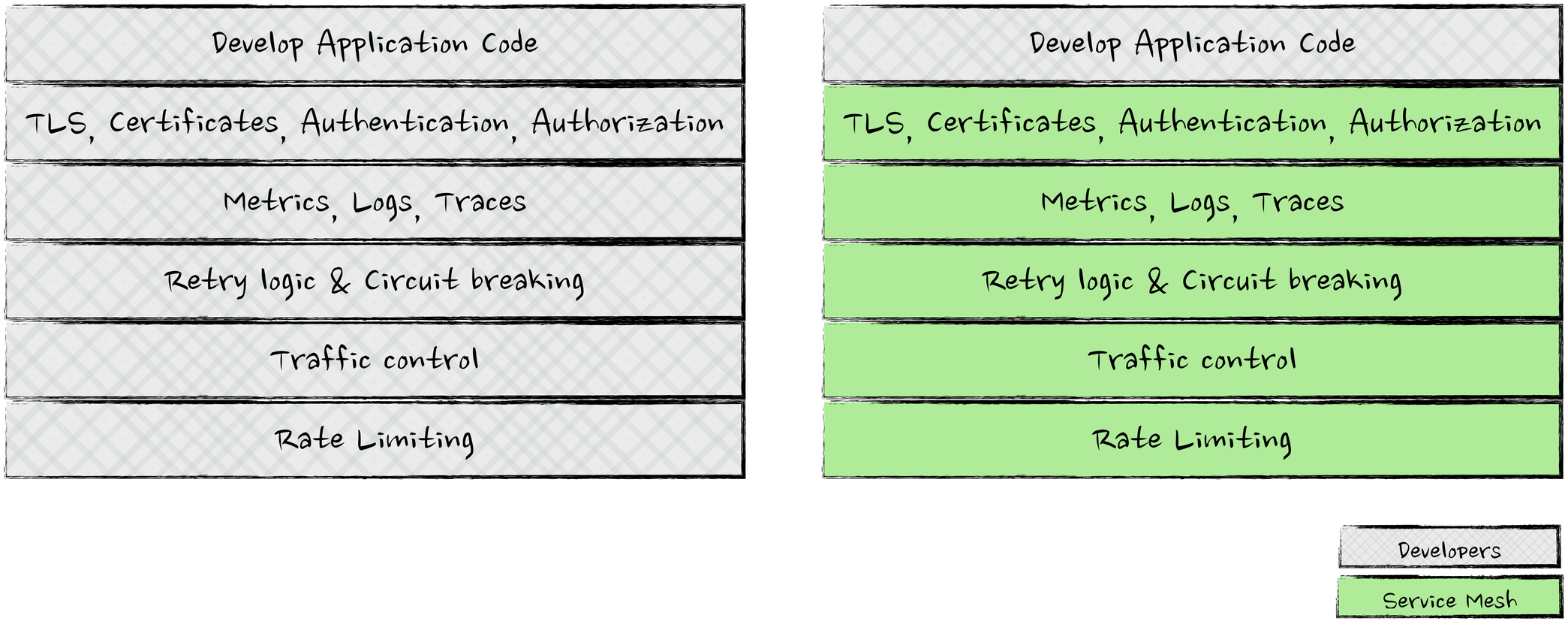
Logging

Tracing

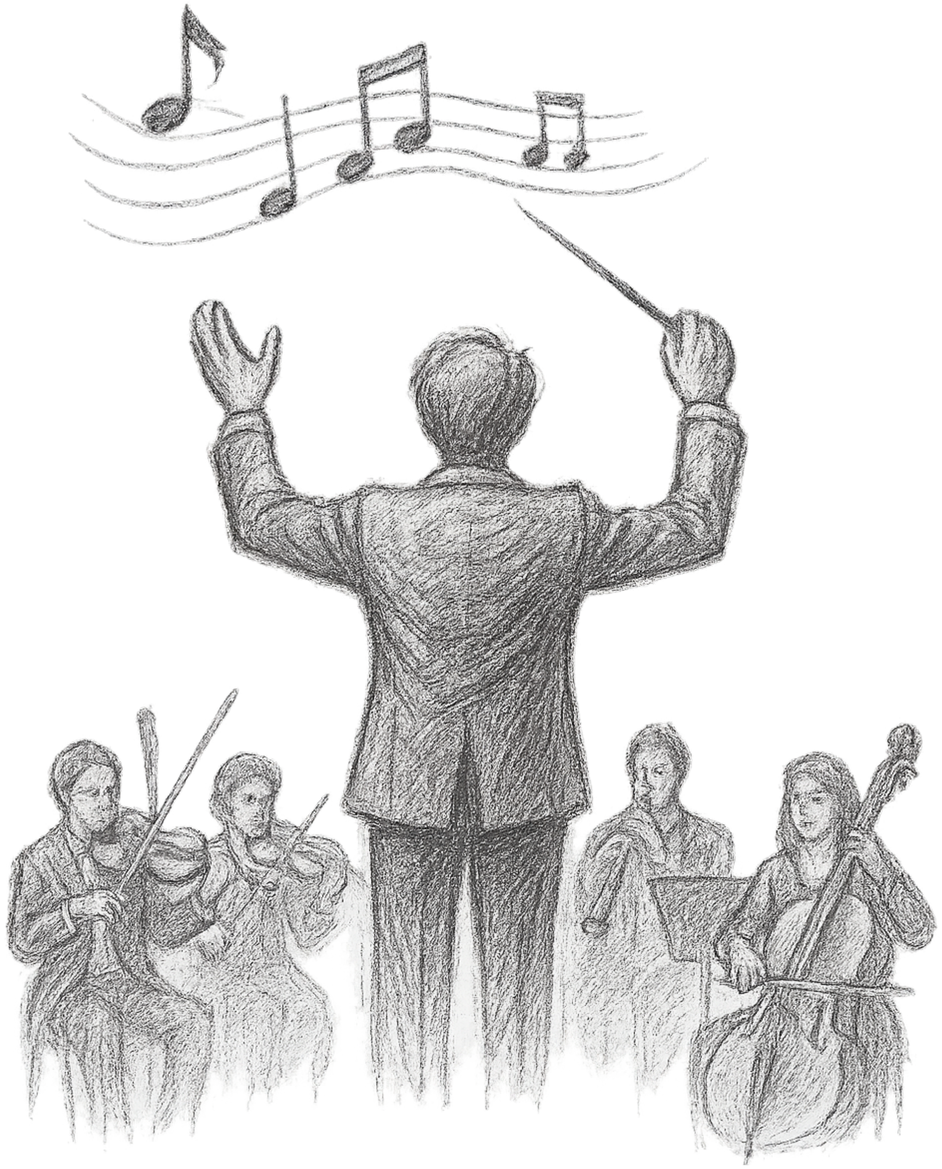
Routing

What Is a Service Mesh?

Reduce Complexity



How Does a Service Mesh Work?

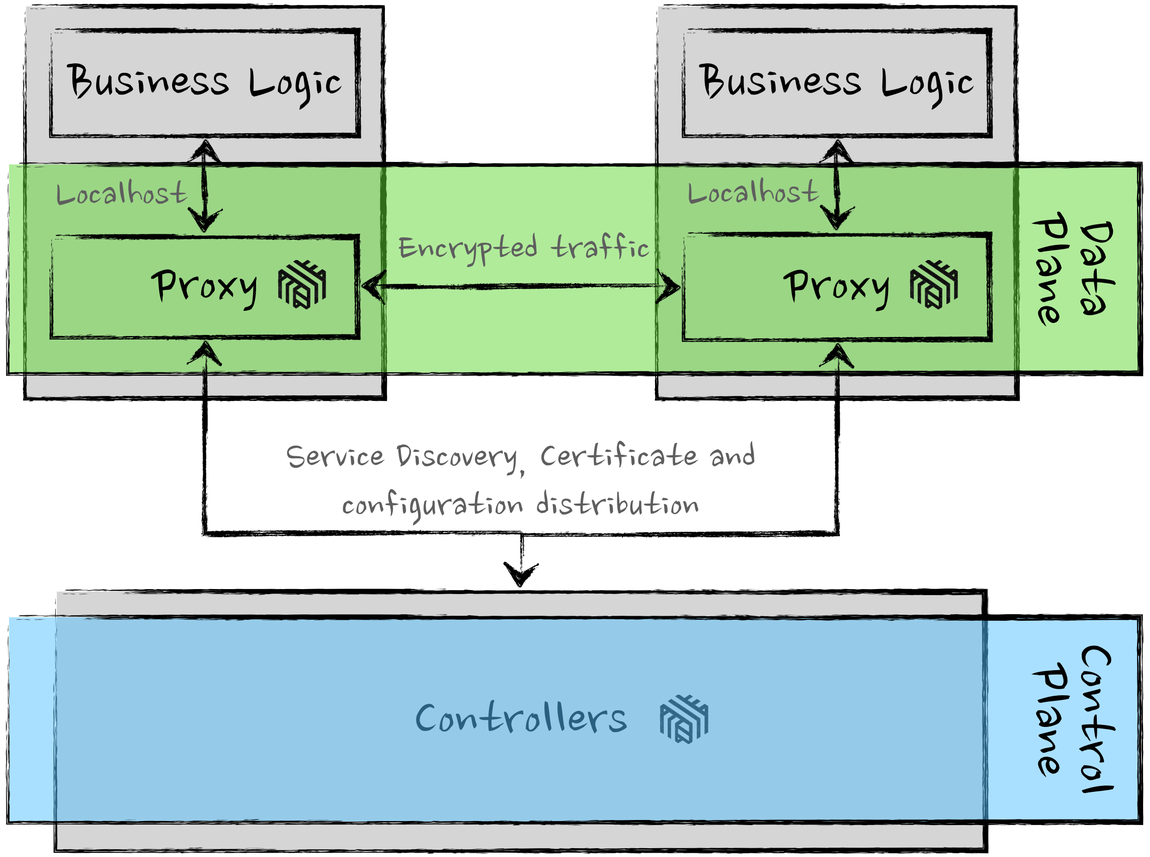


How Does a Service Mesh Work?

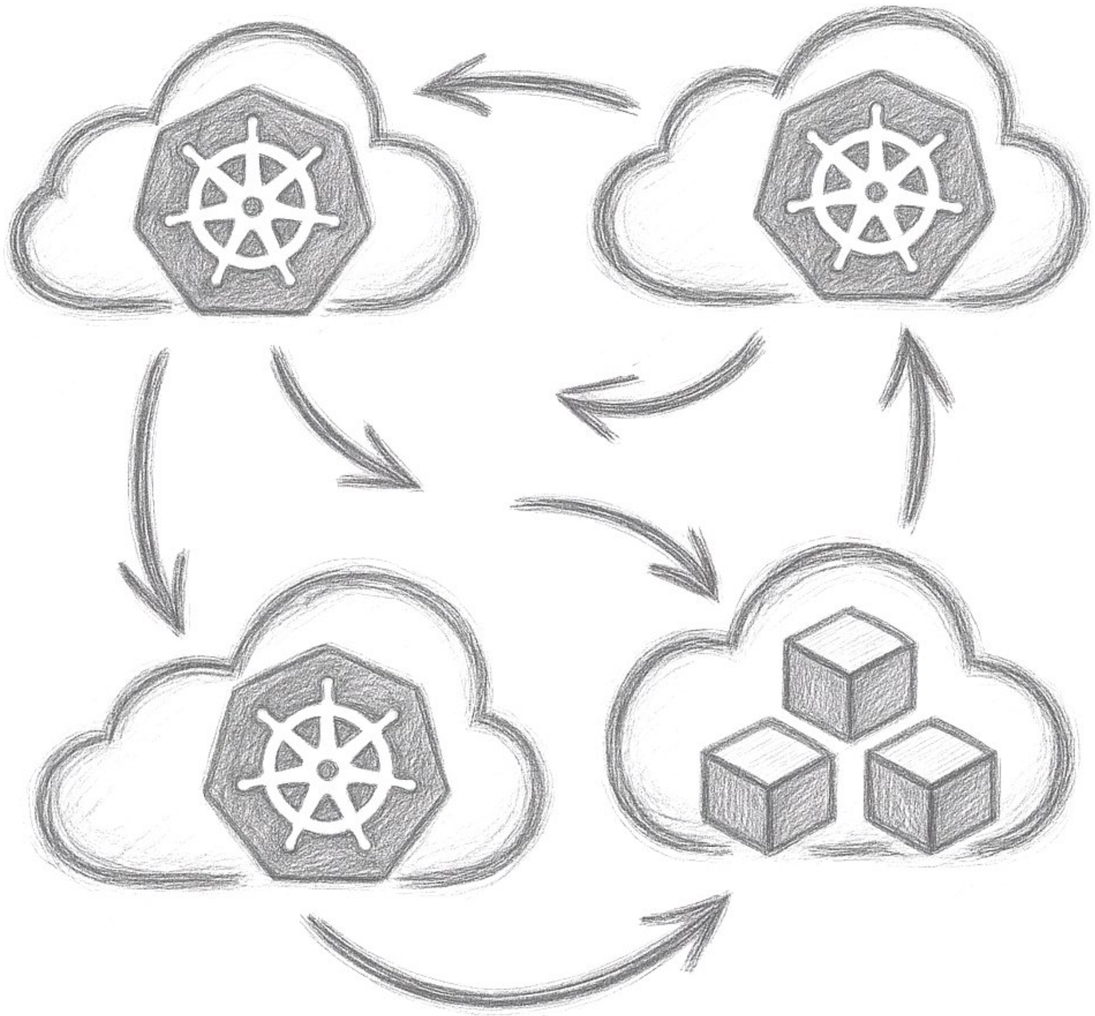
Two Layers: Data Plane and Control Plane

Data Plane: A set of sidecar proxies intercepts and controls every inbound and outbound request between microservices, enforcing policy and encrypting traffic.

Control Plane: Central controllers manage configuration, service discovery, and certificate rotation, then push updates to the proxies.

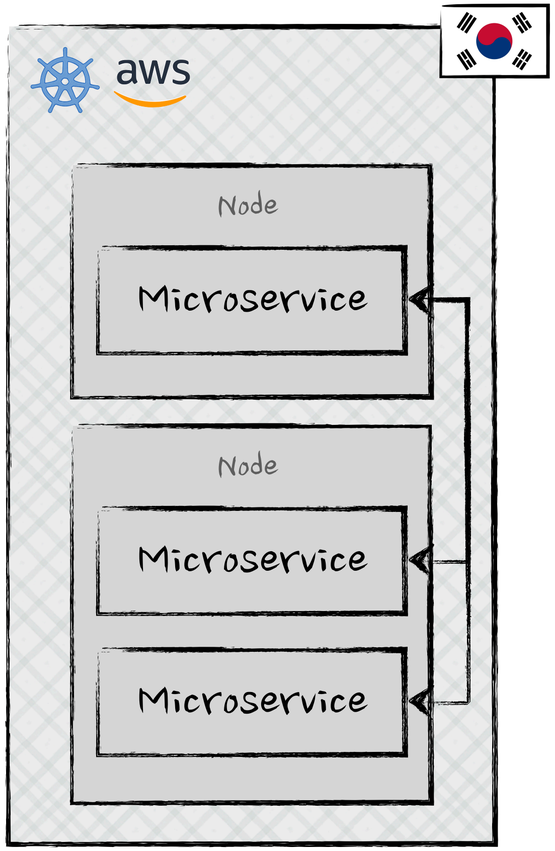


Multicluster



Cluster Architecture Evolution

Single But Fragile World

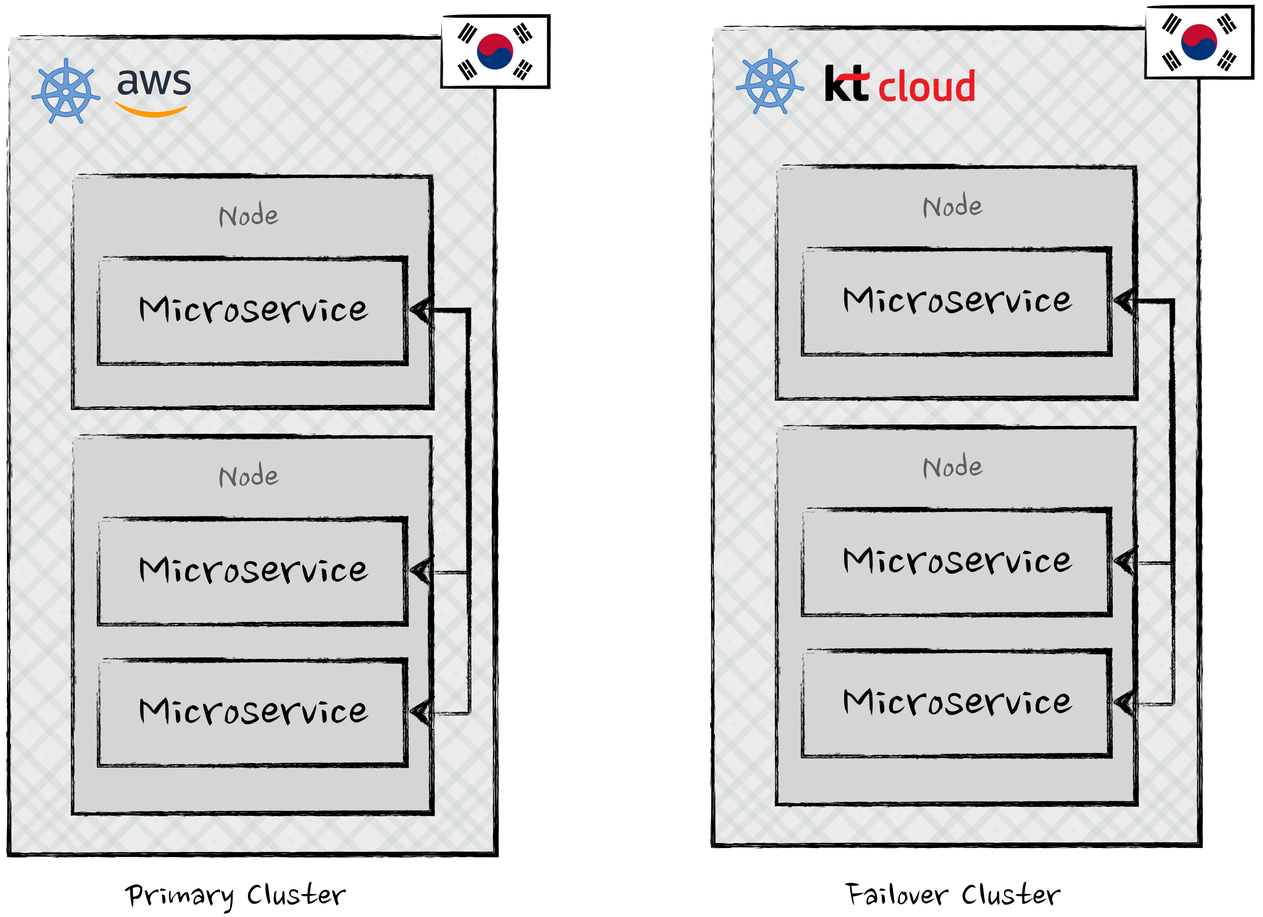


Challenges

- ✗ Improved Availability & Performance
- ✗ Establish Failure Boundaries
- ✗ Enhance Isolation
- ✗ Regulations & Compliance Requirements
- ✗ Cost-Efficiency
- ✗ Improved Scalability
- ✗ Eliminate Vendor-Lock

Cluster Architecture Evolution

An Archipelago of Islands - Failover Cluster

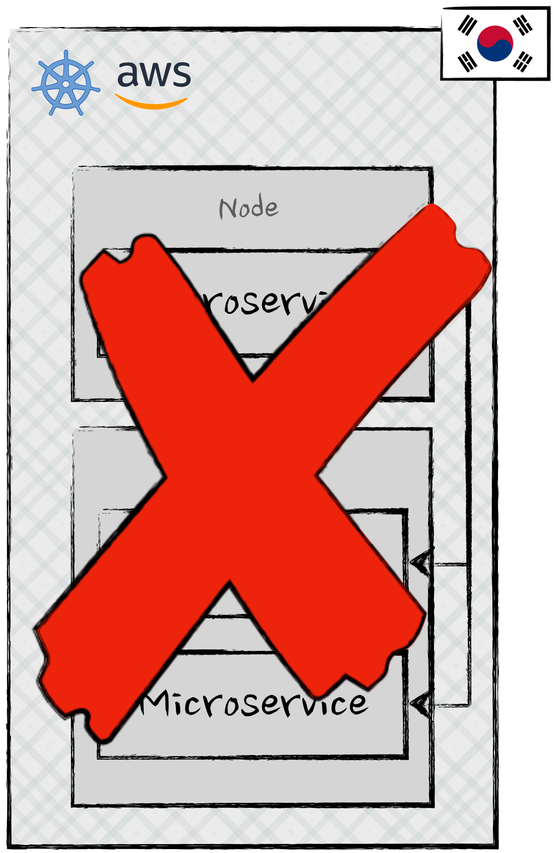


Challenges

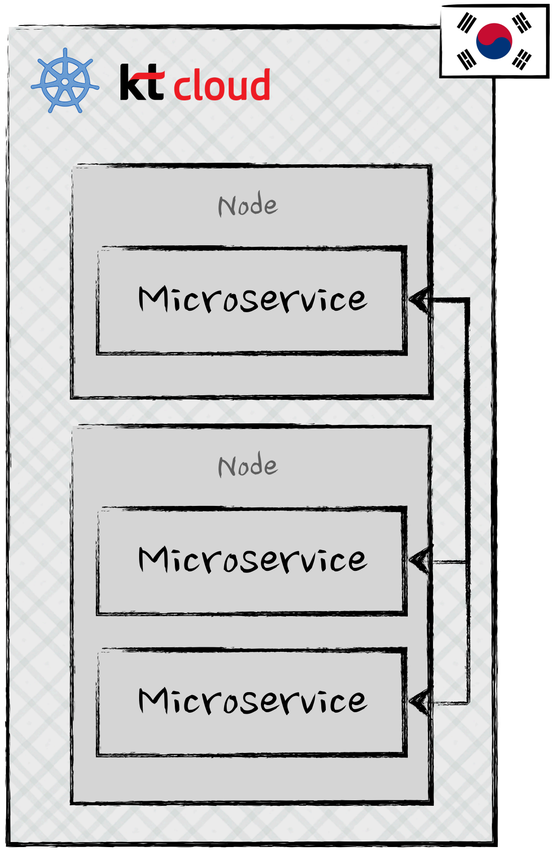
- ✓ Improved Availability & Performance
- ✓ Establish Failure Boundaries
- ✓ Enhance Isolation
- ✓ Regulations & Compliance Requirements
- ✗ Cost-Efficiency
- ✗ Improved Scalability
- ✓ Eliminate Vendor-Lock

Cluster Architecture Evolution

An Archipelago of Islands - Failover Cluster



Primary Cluster



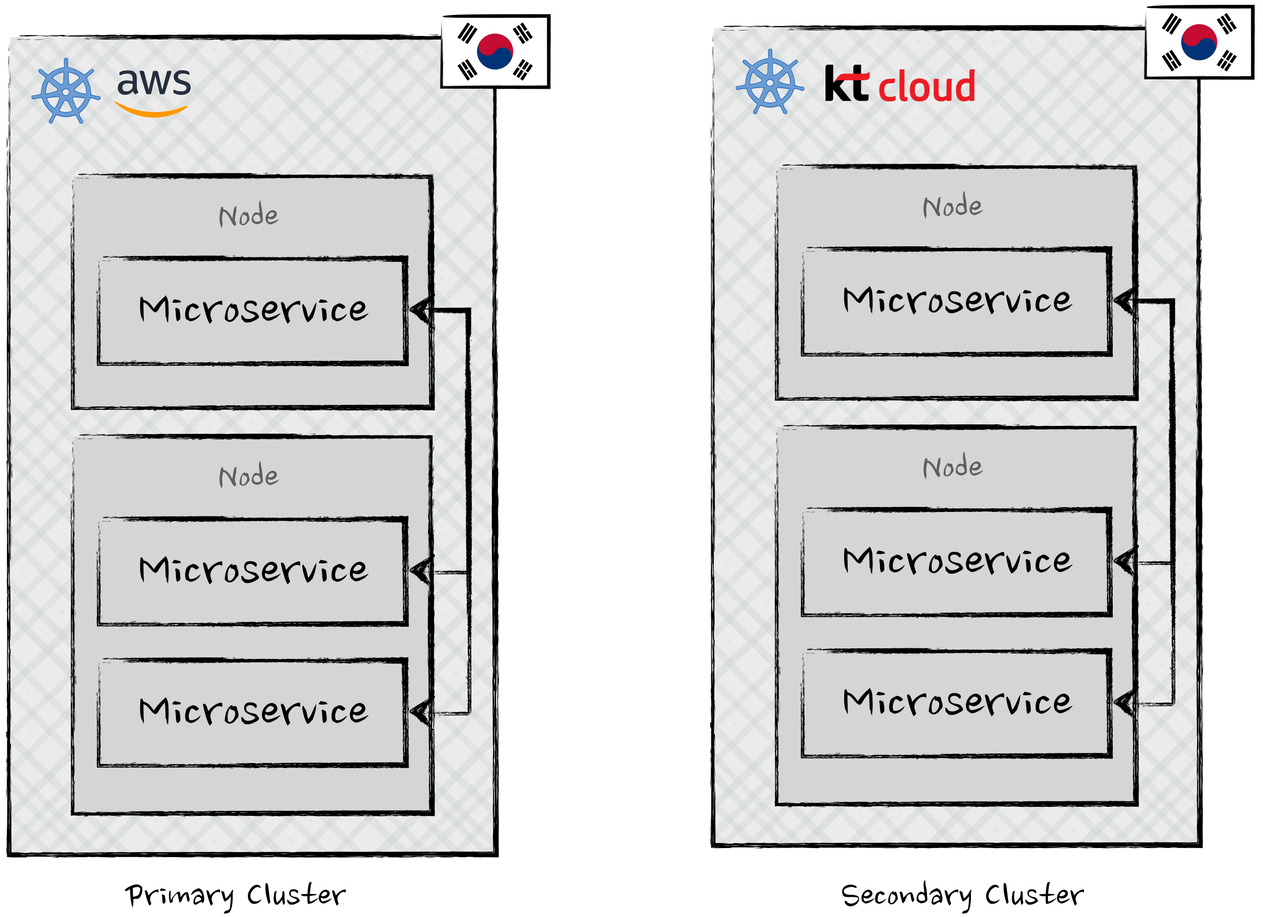
Failover Cluster

Challenges

- ✓ Improved Availability & Performance
- ✓ Establish Failure Boundaries
- ✓ Enhance Isolation
- ✓ Regulations & Compliance Requirements
- ✗ Cost-Efficiency
- ✗ Improved Scalability
- ✓ Eliminate Vendor-Lock

Cluster Architecture Evolution

An Archipelago of Islands - Primary/Secondary

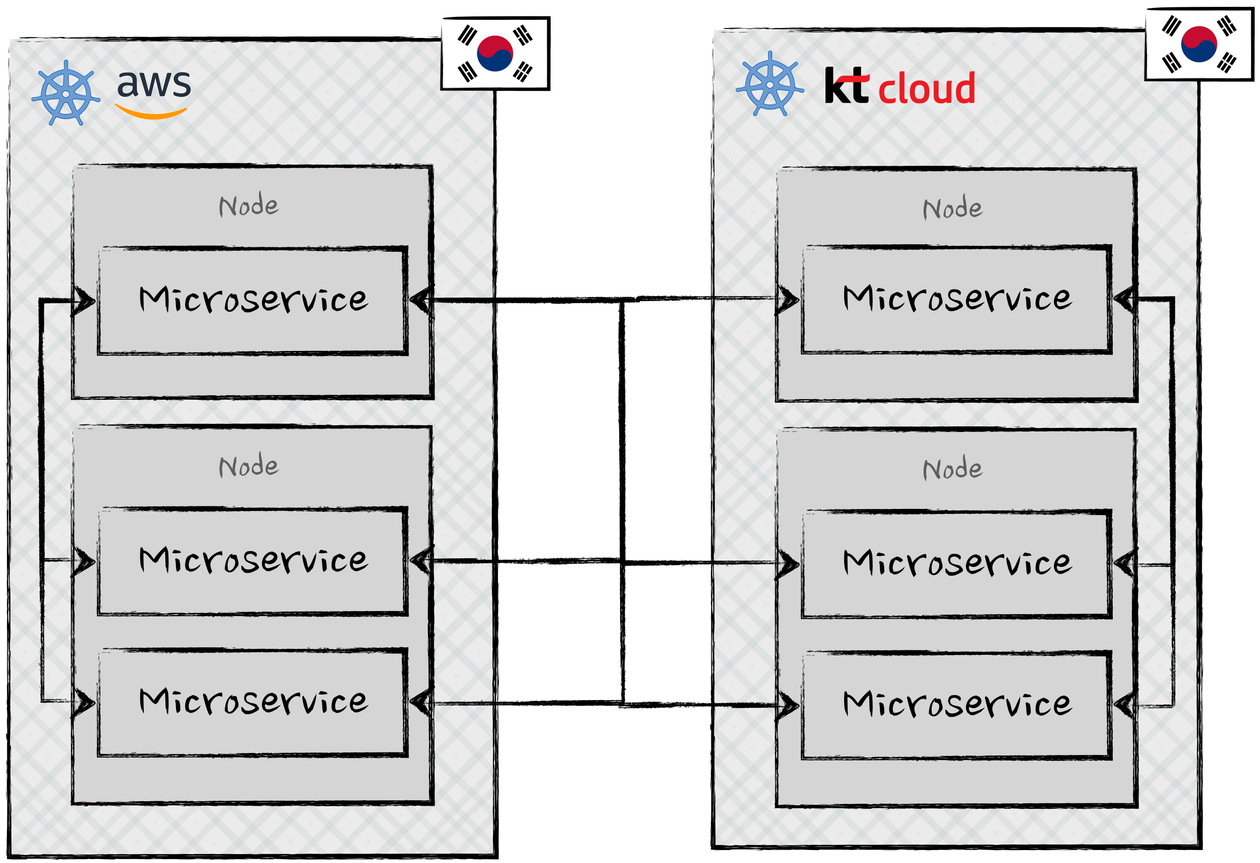


Challenges

- ✓ Improved Availability & Performance
- ✓ Establish Failure Boundaries
- ✓ Enhance Isolation
- ✓ Regulations & Compliance Requirements
- ✗ Cost-Efficiency
- ✗ Improved Scalability
- ✓ Eliminate Vendor-Lock

Cluster Architecture Evolution

Unity Makes Strength - Cooperative Clusters

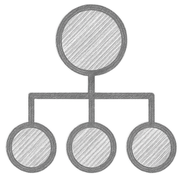


Challenges

- Improved Availability & Performance
- Establish Failure Boundaries
- Enhance Isolation
- Regulations & Compliance Requirements
- Cost-Efficiency
- Improved Scalability
- Eliminate Vendor-Lock

Why Embrace Multi-Cluster Deployments?

Common Challenges in Multi-Cluster Management



Cross-Cluster Networking

Observability

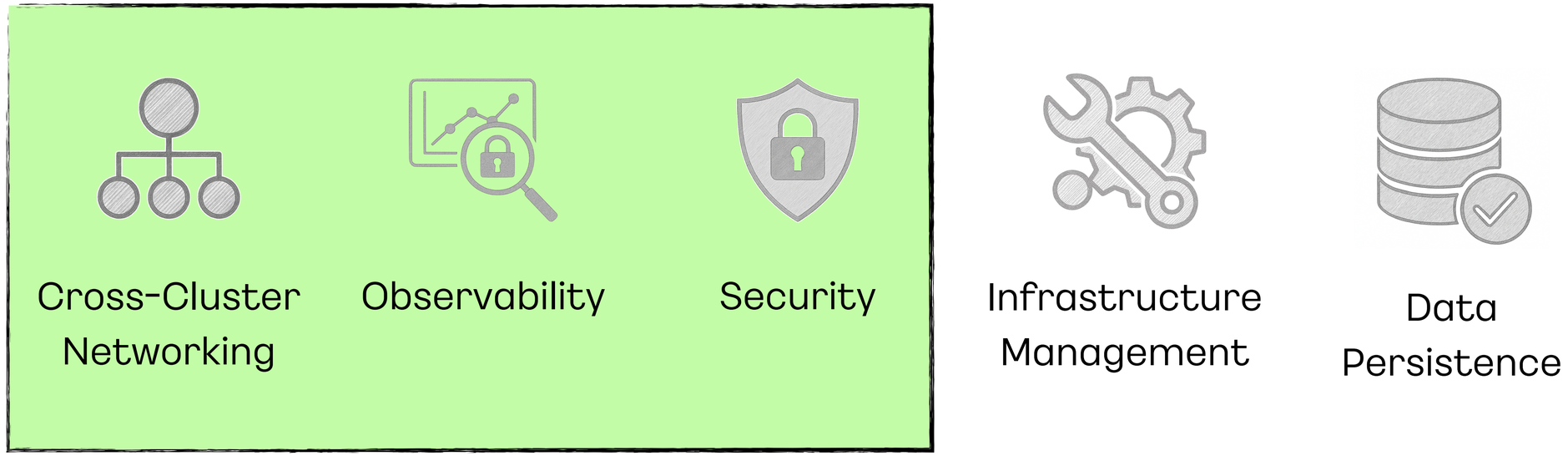
Security

Infrastructure Management

Data Persistence

Why Embrace Multi-Cluster Deployments?

How a Service Mesh Simplifies Multi-Cluster

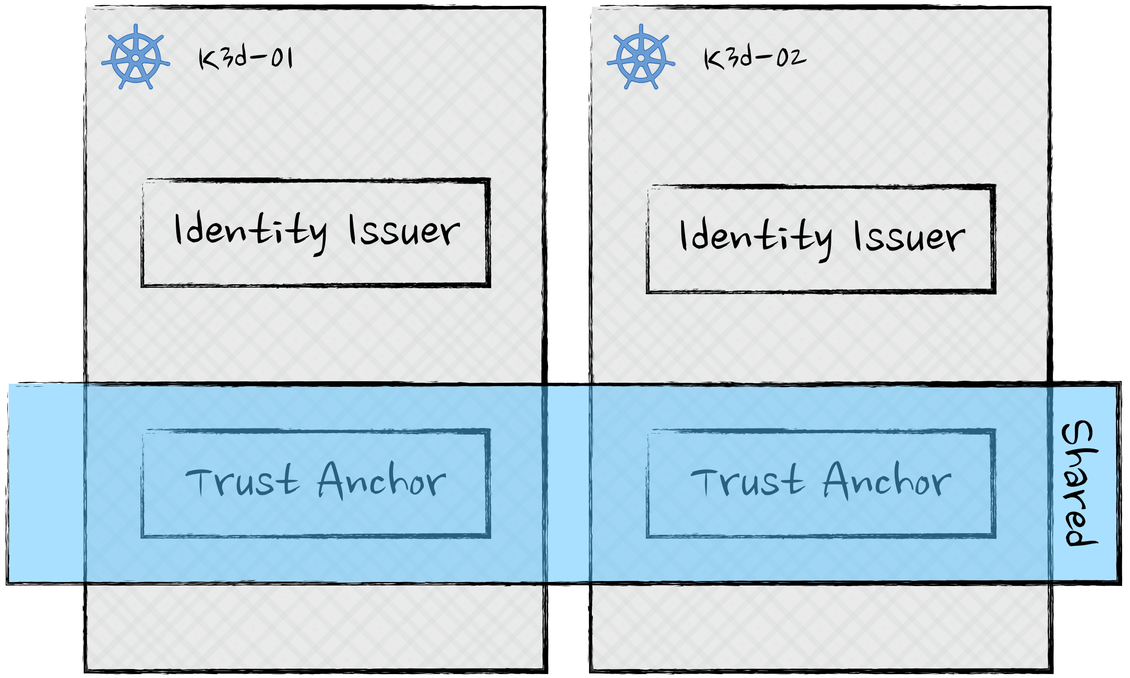


Linkerd Service Mesh

Multi-Cluster Requirements

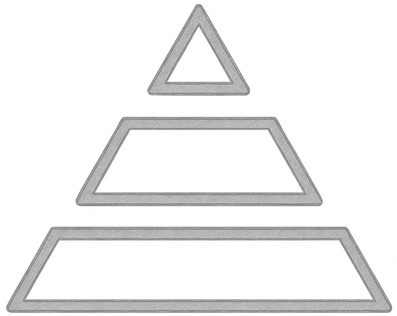
Root Trust Anchor: Single source of trust and shared trust anchor across all clusters to enables the control plane to encrypt inter-cluster requests and verify their identity, ensuring secure communication.

Intermediate Identity Issuer: Each cluster maintains its own intermediate identity issuer. It's used to sign the workload CSR.

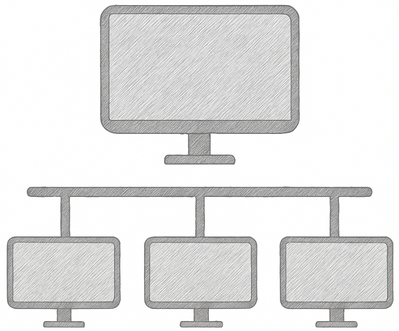


Linkerd Service Mesh

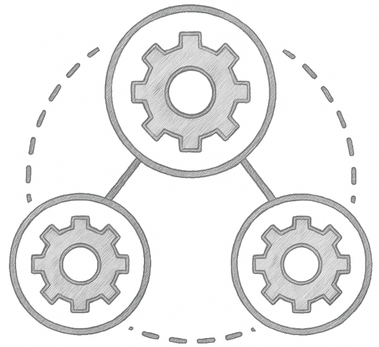
Multiple Multi-Cluster Solutions



Hierarchical



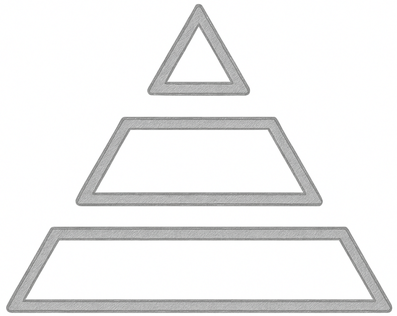
Flat



Federates

Linkerd Multi-Cluster Models

Hierarchical Network



Hierarchical

Pros:

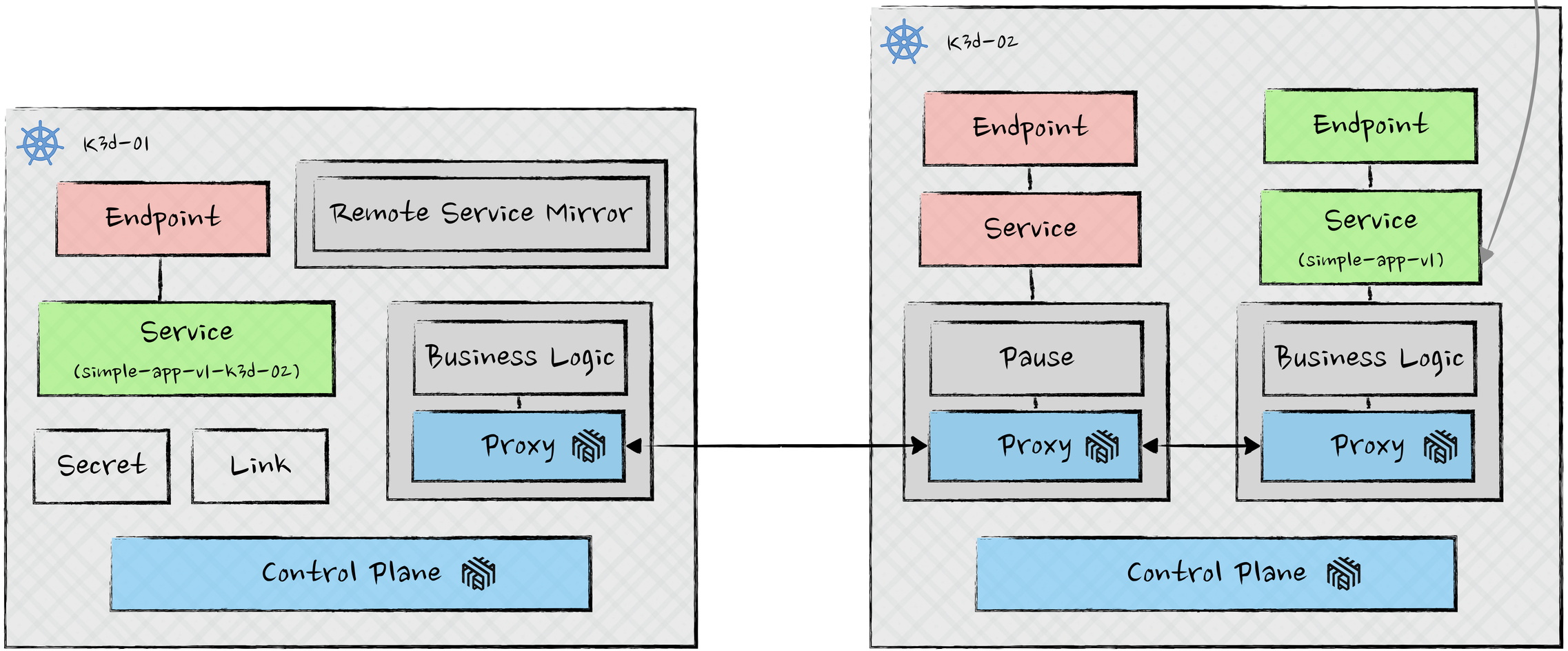
- No networking overhead as this configuration will tunnel traffic through the cluster's gateway.

Cons:

- The additional hop caused by the gateway will increase the latency.
- The connection reaching the server in the remote cluster will lose it's original identity, it will get the identity of the gateway.

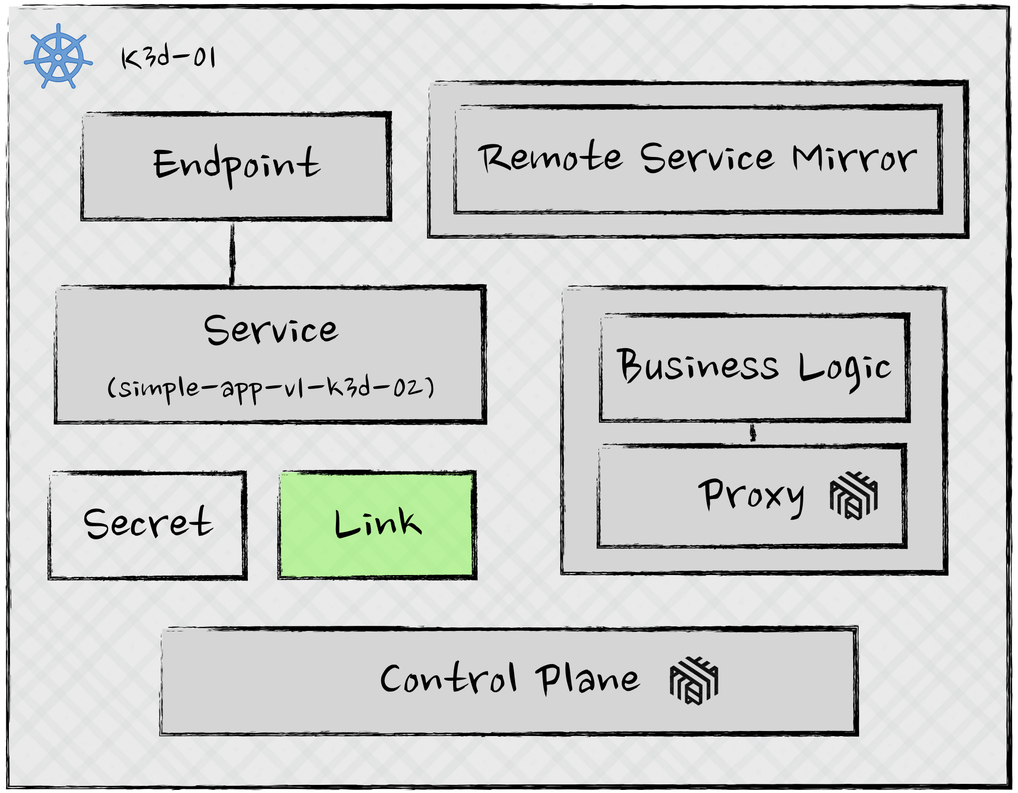
Linkerd Multi-Cluster Models

Hierarchical Network



Linkerd Multi-Cluster Models

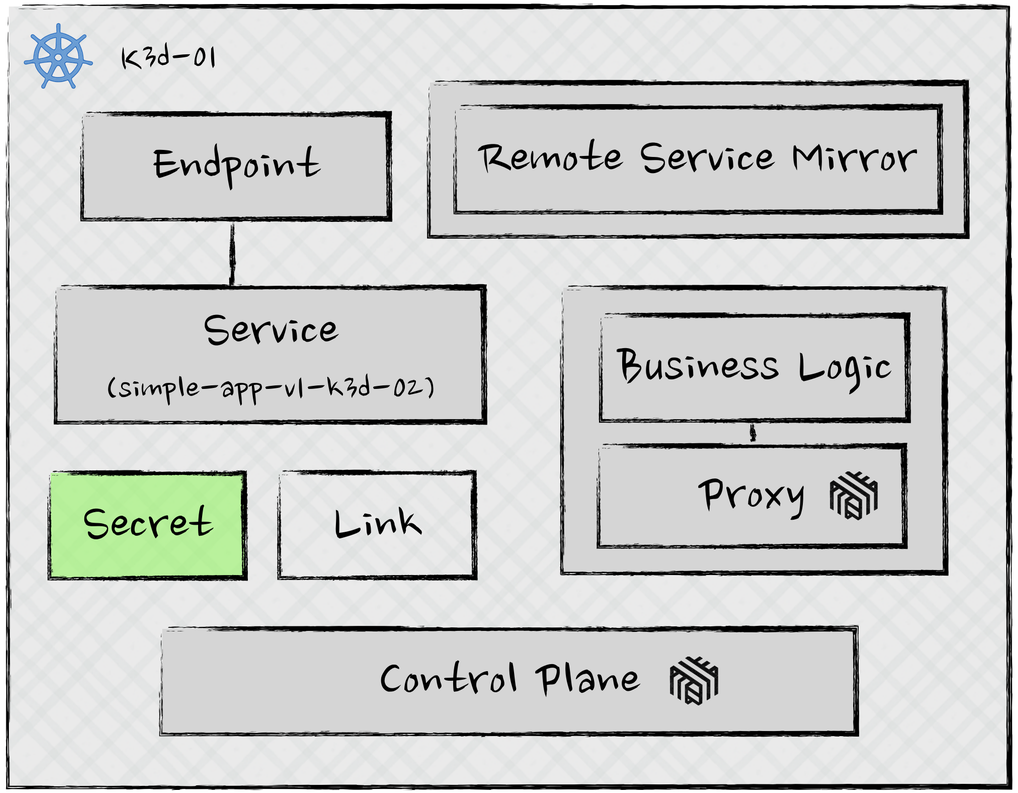
Hierarchical Network




```
Terminal — zsh
[k3d-01|default] gtrekter@MacBook-Pro-M4 playground-scripts % kubectl describe link -n linkerd-multicluster   k3d-02
Name:         k3d-02
Namespace:    linkerd-multicluster
Labels:       <none>
Annotations:  linkerd.io/created-by: linkerd/cli enterprise-2.18.0
API Version:  multicluster.linkerd.io/v1alpha3
Kind:         Link
Metadata:
  Creation Timestamp:  2025-05-26T12:52:38Z
  Generation:          1
  Resource Version:    20464
  UID:                 24a873c4-a0bf-4cc6-8e1a-cfcb8e48e4c2
Spec:
  Cluster Credentials Secret:  cluster-credentials-k3d-02
  Federated Service Selector:
    Match Labels:
      mirror.linkerd.io/federated:  member
  Gateway Address:                  172.20.0.15,172.20.0.16,172.20.0.17,172.20.0.18
  Gateway Identity:                 linkerd-gateway.linkerd-multicluster.serviceaccount.identity.linkerd.cluster.local
  Gateway Port:                     4143
  Probe Spec:
    Failure Threshold:  3
    Path:               /ready
    Period:             3s
    Port:               4191
    Timeout:            30s
  Remote Discovery Selector:
    Match Labels:
      mirror.linkerd.io/exported:  remote-discovery
  Selector:
    Match Labels:
      mirror.linkerd.io/exported:    true
  Target Cluster Domain:             cluster.local
  Target Cluster Linkerd Namespace:  linkerd
  Target Cluster Name:               k3d-02
Status:
  Mirror Services:
    Conditions:
      Last Transition Time:  2025-05-26T14:46:47Z
      Local Ref:
        Group:        core
        Kind:         Service
        Name:         simple-app-v1-k3d-02
        Namespace:    simple-app
      Message:
      Reason:         Mirrored
      Status:         True
      Type:           Mirrored
    Controller Name:  linkerd.io/service-mirror
    Remote Ref:
      Group:      core
      Kind:       Service
      Name:       simple-app-v1
      Namespace:  simple-app
```

Linkerd Multi-Cluster Models

Hierarchical Network

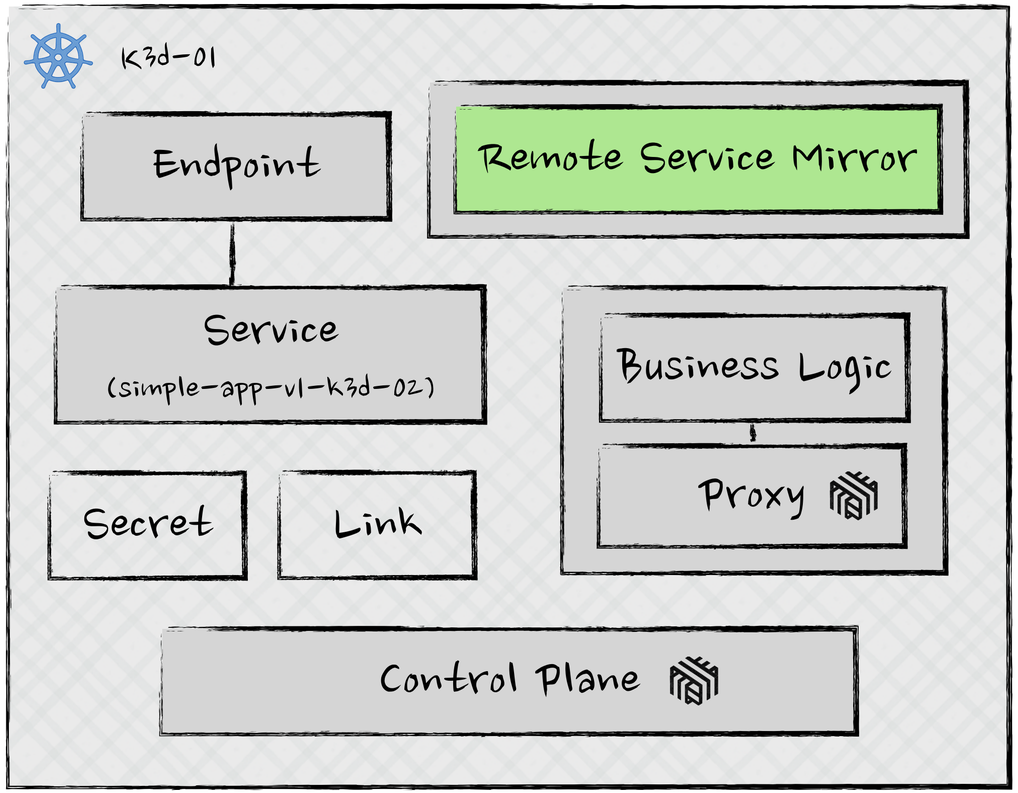


Terminal — zsh

```
NAMESPACE             NAME                                          TYPE                                  DATA   AGE
kube-system           k3d-01-agent-0.node-password.k3s              Opaque                                1      152m
kube-system           k3d-01-agent-1.node-password.k3s              Opaque                                1      152m
kube-system           k3d-01-agent-2.node-password.k3s              Opaque                                1      152m
kube-system           k3d-01-server-0.node-password.k3s             Opaque                                1      152m
kube-system           k3s-serving                                   kubernetes.io/tls                     2      153m
linkerd-multicluster  cluster-credentials-k3d-02                    mirror.linkerd.io/remote-kubeconfig   1      147m
linkerd-multicluster  linkerd-service-mirror-remote-access-default-token  kubernetes.io/service-account-token   3      148m
linkerd               buoyant-license                               Opaque                                1      151m
linkerd               cluster-credentials-k3d-02                    mirror.linkerd.io/remote-kubeconfig   1      147m
linkerd               linkerd-identity-issuer                       Opaque                                2      151m
linkerd               linkerd-policy-validator-k8s-tls              kubernetes.io/tls                     2      151m
linkerd               linkerd-proxy-injector-k8s-tls                kubernetes.io/tls                     2      151m
linkerd               linkerd-sp-validator-k8s-tls                  kubernetes.io/tls                     2      151m
linkerd               sh.helm.release.v1.linkerd-control-plane.v1   helm.sh/release.v1                    1      151m
linkerd               sh.helm.release.v1.linkerd-preview-crds.v1    helm.sh/release.v1                    1      151m
[k3d-01|default] gtrekter@MacBook-Pro-M4 playground-scripts % kubectl get secrets -n linkerd-multicluster   cluster-credentials-k3d-02 -o jso
npath='{.data.kubeconfig}' | base64 -d
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUJkakNDQVIyZ0F3SUJBZ01CQURBS0JnZ3Foa2pPUFFRREFqQWpNU0V3SHdZRFZRUUREQ
mhyTTNNdGMyVnkKZG1WeUxXTmhRREUzTkRneU5qTTNNVGd3SGhjTk1qVXdOVEkyTVRJME9ETTRXaGNOTXpVd05USTBNVEkwT0RNNApXakFqTVNFd0h3WURWUVFEREJock0zTXRjMlZ5ZG
1WeUxXTmhRREUzTkRneU5qTTNNVGd3V1RBVEJnY3Foa2pPClBRSUJCZ2dxaGtqT1BRTUJCd05DQUFTTGVScVd3SEw2UDhidnEvcXVTa2M2ZXdxNEFlc1Z4NGQ3SSsyZEtoRloKR0ZMTUJ
uREl2QytFMFY5d3Nycjdmb1VlMVE5aWE4NkNMaFJBeXNob2JSZ3VvMEl3UURBT0JnTlZIUThCQWY4RQpCQU1DQXFRd0R3WURWUjBUQVFIL0JBVXdBd0VCL3pBZEJnTlZIUTRFRmdRVTNw
YmxWSnFjVllvUmR3bmkramxzCjRHY3V3OTB3Q2dZSUtvWkl6ajBFQXdJRFJ3QXdSQUlnTVUrYitGSzhqRHpvVDQ1S2w2S2JmQ21XR2s0NHNXSUEKc0tQS3dib0ZVWU1DSUZ1UThFRE1KZ
UFRaWNSSjBvK1JHUVhDdy9mV1Q1NlZZWC9rWEczSXJLdHEKLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
    server: https://172.20.0.15:6443
  name: k3d-02
contexts:
- context:
    cluster: k3d-02
    user: linkerd-service-mirror-remote-access-default
  name: k3d-02
current-context: k3d-02
kind: Config
preferences: {}
users:
- name: linkerd-service-mirror-remote-access-default
  user:
    token: eyJhbGciOiJSUzI1NiIsImtpZCI6ImlBZjNJSWZtZUQ4MnZBMWttYkNDalNpUjBiMjZPclg0OW1UNnU5c3huaW8ifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2N
vdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJsaW5rZXJkLW11bHRpY2x1c3RlciIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0
Lm5hbWUiOiJsaW5rZXJkLXNlcnZpY2UtbWlycm9yLXJlbW90ZS1hY2Nlc3MtZGVmYXVsdC10b2tlbiIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50Lm
m5hbWUiOiJsaW5rZXJkLXNlcnZpY2UtbWlycm9yLXJlbW90ZS1hY2Nlc3MtZGVmYXVsdCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50LnVpZCI6Ij
Q4ZDQzMTg1LTY4MTAtNDcxMy04ZDFkLTZjYjBiZDJkMjdlMiIsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDpsaW5rZXJkLW11bHRpY2x1c3RlcjpsaW5rZXJkLXNlcnZpY2UtbWl
ycm9yLXJlbW90ZS1hY2Nlc3MtZGVmYXVsdCJ9.C2haOSAwE7hMrwdg0fRR2o3cOvnEu7EPW6wEyzv_I5q1pMwPYtbPt3H9YLaVun57MW_DGFBAlM0Ghue8DleWgb8ImXxzC_o2v4ykqNr
rJPSRaQ1_0pb5P2V4BZ47MiE3IFW7N2lBGwjfN95Rp-qCjJx7h6HfSqDpBYFGVuoTzSPA6XsqKPTo97S2p3ezFWbyeIbi3yRVTaIYOqwf91iLVI6HtnSTy6f52lFBPBDJ4A2dBTwj0DDZ
zNz04av_k2ODVtpiN0VJGMLkP_TSRO89Bg-MumejYH9lSdhzmcz7QUJdTIt0bgeoXYnafxX9Ds6y4LZiXTpGWApUlTRcPwALWA
[k3d-01|default] gtrekter@MacBook-Pro-M4 playground-scripts %
```

Linkerd Multi-Cluster Models

Hierarchical Network

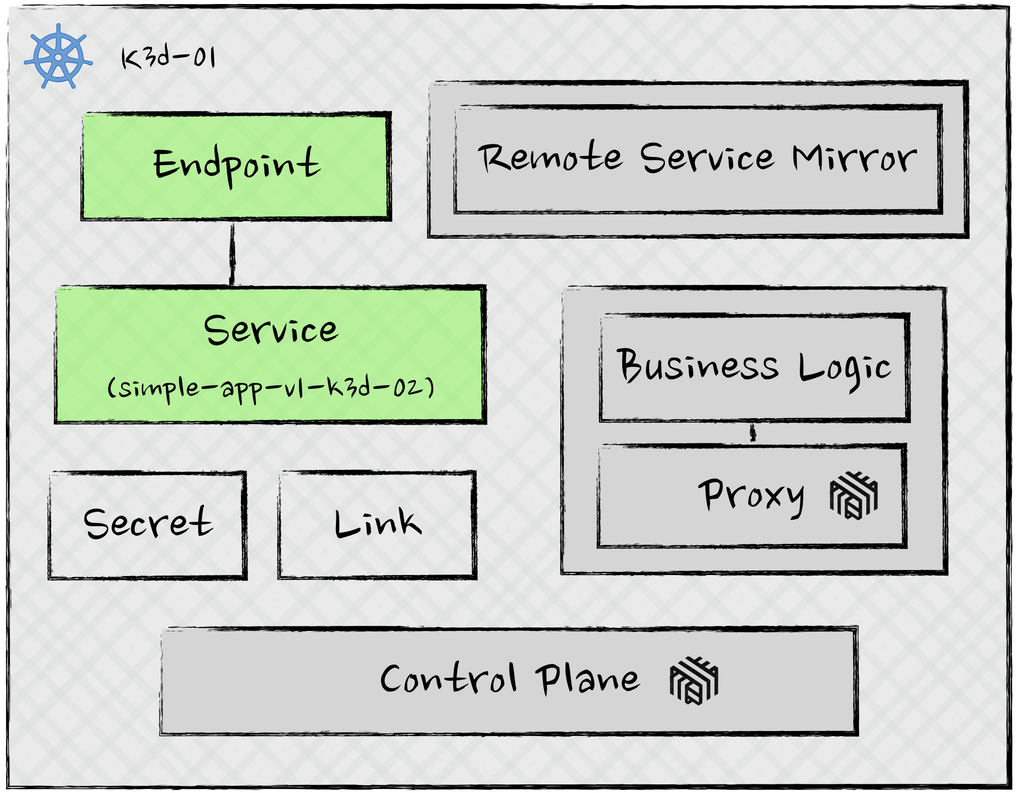


Terminal — zsh

```
], labels [[]]}}" cluster=k3d-02
time="2025-05-26T14:46:42Z" level=info msg="Received: RemoteServiceUnexported: {name: simple-app-v1, namespace: simple-app }" cluster=k3d-02
time="2025-05-26T14:46:42Z" level=info msg="GET https://10.247.0.1:443/apis/multicluster.linkerd.io/v1alpha3/namespaces/linkerd-multicluster/links
/k3d-02 200 OK in 1 milliseconds"
time="2025-05-26T14:46:42Z" level=info msg="patching link status linkerd-multicluster/k3d-02" cluster=k3d-02
time="2025-05-26T14:46:42Z" level=debug msg="skipped processing endpoints object simple-app/simple-app-v1: missing mirror.linkerd.io/exported labe
l" cluster=k3d-02
time="2025-05-26T14:46:42Z" level=info msg="PATCH https://10.247.0.1:443/apis/multicluster.linkerd.io/v1alpha3/namespaces/linkerd-multicluster/lin
ks/k3d-02/status 200 OK in 2 milliseconds"
time="2025-05-26T14:46:42Z" level=info msg="Deleting mirrored service simple-app/simple-app-v1-k3d-02" cluster=k3d-02
time="2025-05-26T14:46:42Z" level=info msg="DELETE https://10.247.0.1:443/api/v1/namespaces/simple-app/services/simple-app-v1-k3d-02 200 OK in 8 m
illiseconds"
time="2025-05-26T14:46:42Z" level=info msg="Successfully deleted service: simple-app/simple-app-v1-k3d-02" cluster=k3d-02
time="2025-05-26T14:46:43Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-
mirror-write-k3d-02 200 OK in 4 milliseconds"
time="2025-05-26T14:46:45Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-
mirror-write-k3d-02 200 OK in 2 milliseconds"
time="2025-05-26T14:46:45Z" level=debug msg="Gateway is healthy" probe-key=k3d-02
time="2025-05-26T14:46:47Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-
mirror-write-k3d-02 200 OK in 3 milliseconds"
time="2025-05-26T14:46:47Z" level=info msg="Received: OnUpdateCalled: {svc: Service: {name: simple-app-v1, namespace: simple-app, annotations: [[]
], labels [[]]}}" cluster=k3d-02
time="2025-05-26T14:46:47Z" level=info msg="Received: RemoteServiceExported: {service: Service: {name: simple-app-v1, namespace: simple-app, annot
ations: [[]], labels [[]]}}" cluster=k3d-02
time="2025-05-26T14:46:47Z" level=info msg="GET https://10.247.0.1:443/api/v1/namespaces/simple-app 200 OK in 2 milliseconds"
time="2025-05-26T14:46:47Z" level=info msg="Creating a new service mirror for simple-app/simple-app-v1" cluster=k3d-02
time="2025-05-26T14:46:47Z" level=info msg="POST https://10.247.0.1:443/api/v1/namespaces/simple-app/services 201 Created in 6 milliseconds"
time="2025-05-26T14:46:47Z" level=info msg="Resolved gateway [[{172.20.0.15  <nil> nil} {172.20.0.16  <nil> nil} {172.20.0.17  <nil> nil} {172.20.
0.18  <nil> nil}]:4143] for simple-app/simple-app-v1" cluster=k3d-02
time="2025-05-26T14:46:47Z" level=info msg="Creating a new endpoints for simple-app/simple-app-v1" cluster=k3d-02
time="2025-05-26T14:46:47Z" level=info msg="POST https://10.247.0.1:443/api/v1/namespaces/simple-app/endpoints 201 Created in 2 milliseconds"
time="2025-05-26T14:46:47Z" level=info msg="v1 Endpoints is deprecated in v1.33+; use discovery.k8s.io/v1 EndpointSlice"
time="2025-05-26T14:46:47Z" level=info msg="GET https://10.247.0.1:443/apis/multicluster.linkerd.io/v1alpha3/namespaces/linkerd-multicluster/links
/k3d-02 200 OK in 2 milliseconds"
time="2025-05-26T14:46:47Z" level=info msg="patching link status linkerd-multicluster/k3d-02" cluster=k3d-02
time="2025-05-26T14:46:47Z" level=info msg="PATCH https://10.247.0.1:443/apis/multicluster.linkerd.io/v1alpha3/namespaces/linkerd-multicluster/lin
ks/k3d-02/status 200 OK in 3 milliseconds"
time="2025-05-26T14:46:47Z" level=info msg="Received: &{%!s(*v1.Endpoints=&{{ } {simple-app-v1  simple-app  0306822c-6130-40c3-8f3b-a00f3ccb111e 1
2310 0 {{0 63883860648 0x39f9de0}} <nil> <nil> map[endpoints.kubernetes.io/managed-by:endpoint-controller mirror.linkerd.io/exported:true] map[] [
] [] [{k3s Update v1 0x4000d865d0 FieldsV1 0x4000d865e8 }]} [{[{10.24.0.10  0x4000a7ef10 0x4000425490}] [] [{ 5678 TCP <nil>}]}]}})" cluster=k3d-0
2
time="2025-05-26T14:46:47Z" level=debug msg="skipped processing endpoints object simple-app/simple-app-v1: missing service.kubernetes.io/headless
label" cluster=k3d-02
time="2025-05-26T14:46:47Z" level=debug msg="Link update ignored (only status changed): k3d-02"
time="2025-05-26T14:46:48Z" level=debug msg="Gateway is healthy" probe-key=k3d-02
time="2025-05-26T14:46:49Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-
mirror-write-k3d-02 200 OK in 3 milliseconds"
time="2025-05-26T14:46:51Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-
mirror-write-k3d-02 200 OK in 2 milliseconds"
time="2025-05-26T14:46:51Z" level=debug msg="Gateway is healthy" probe-key=k3d-02
time="2025-05-26T14:46:53Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-
mirror-write-k3d-02 200 OK in 5 milliseconds"
time="2025-05-26T14:46:54Z" level=debug msg="Gateway is healthy" probe-key=k3d-02
time="2025-05-26T14:46:55Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-
mirror-write-k3d-02 200 OK in 1 milliseconds"
time="2025-05-26T14:46:57Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-
```

Linkerd Multi-Cluster Models

Hierarchical Network

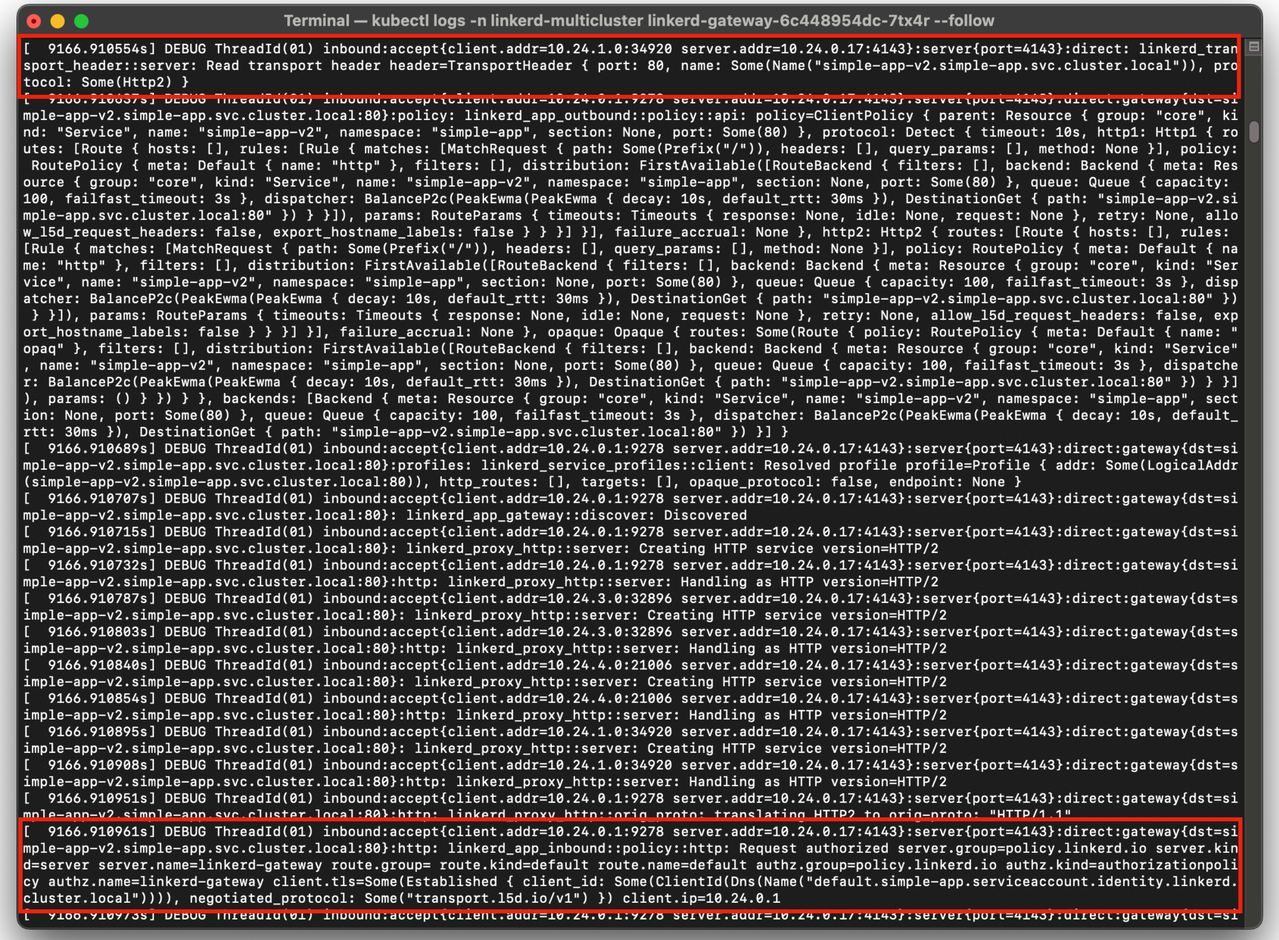


Terminal — zsh

```
[k3d-01|default] gtrekter@MacBook-Pro-M4 playground-scripts % kubectl get svc -n simple-app simple-app-v1-k3d-02 -o yaml
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"name":"simple-app-v1","namespace":"simple-app"},"spec":{"ports":[{"port":80,"
targetPort":5678}],"selector":{"app":"simple-app-v1","version":"v1"}}}
    mirror.linkerd.io/remote-resource-version: "1332"
    mirror.linkerd.io/remote-svc-fq-name: simple-app-v1.simple-app.svc.cluster.local
  creationTimestamp: "2025-05-27T06:06:20Z"
  labels:
    mirror.linkerd.io/cluster-name: k3d-02
    mirror.linkerd.io/mirrored-service: "true"
  name: simple-app-v1-k3d-02
  namespace: simple-app
  resourceVersion: "1609"
  uid: 7b52bc35-87bc-4b55-bf0b-6e01a908cffc
spec:
  clusterIP: 10.247.130.193
  clusterIPs:
  - 10.247.130.193
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - port: 80
    protocol: TCP
    targetPort: 5678
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
[k3d-01|default] gtrekter@MacBook-Pro-M4 playground-scripts % kubectl get endpoints -n simple-app simple-app-v1-k3d-02 -o yaml
Warning: v1 Endpoints is deprecated in v1.33+; use discovery.k8s.io/v1 EndpointSlice
apiVersion: v1
kind: Endpoints
metadata:
  annotations:
    mirror.linkerd.io/remote-gateway-identity: linkerd-gateway.linkerd-multicluster.serviceaccount.identity.linkerd.cluster.local
    mirror.linkerd.io/remote-svc-fq-name: simple-app-v1.simple-app.svc.cluster.local
  creationTimestamp: "2025-05-27T06:06:20Z"
  labels:
    mirror.linkerd.io/cluster-name: k3d-02
    mirror.linkerd.io/mirrored-service: "true"
  name: simple-app-v1-k3d-02
  namespace: simple-app
  resourceVersion: "1611"
  uid: a1ae45c0-d31a-4291-945a-738eba2ddc2a
subsets:
- addresses:
  - ip: 172.20.0.15
  - ip: 172.20.0.16
  - ip: 172.20.0.17
  - ip: 172.20.0.18
  ports:
  - port: 4143
```

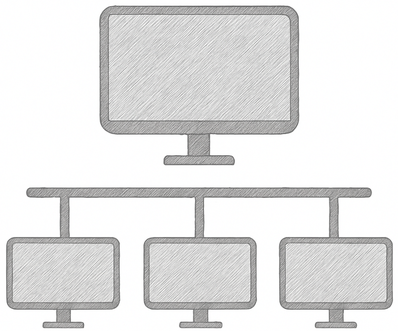
Linkerd Multi-Cluster Models

Hierarchical Network



Linkerd Multi-Cluster Models

Flat Network



Flat

Pros:

- Because there is no gateway in the middle, the identities are preserved.
- The lack of additional hop caused by the proxy, reduces the overall latency.

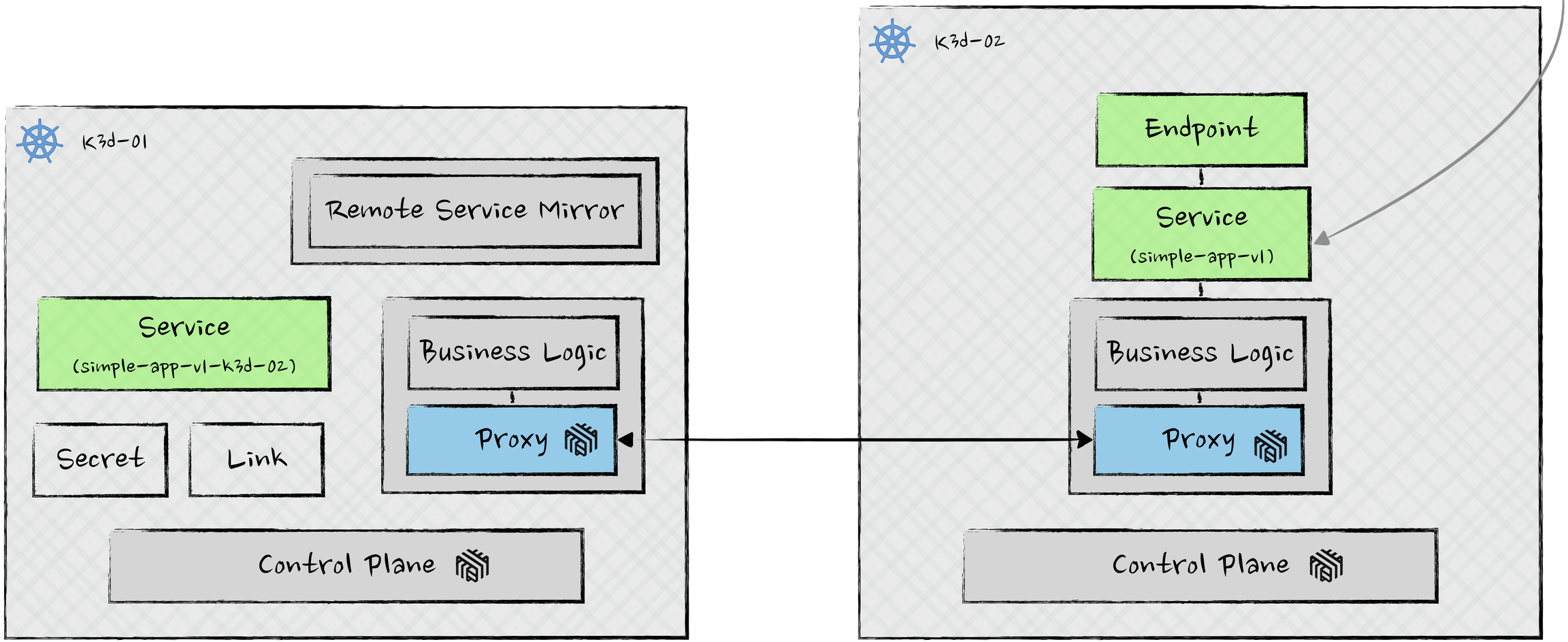
Cons:

- The clusters needs to use non overlapping CIDR ranges.
- The pods need to have connectivity with the pods in the remote cluster.

Linkerd Multi-Cluster Models

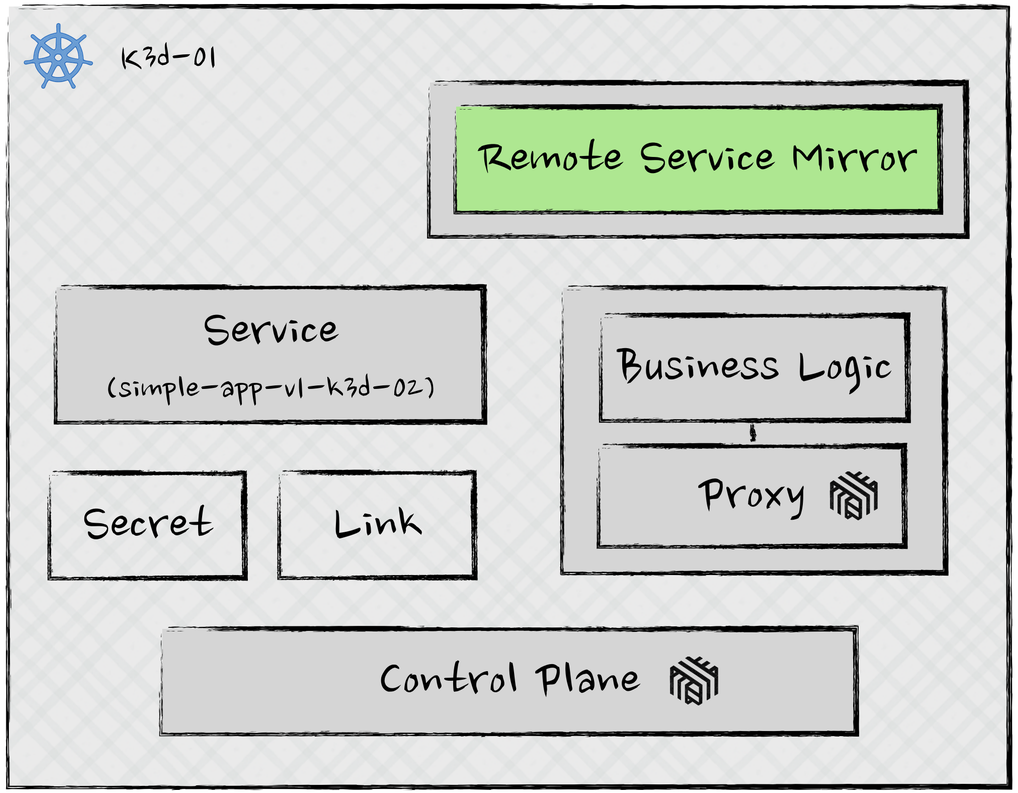
Flat Network

The Remote Service Mirror connect to the cluster using the credentials stored in the secret and watches for Services with:
label:
mirror.linkerd.io/exported=remote-discovery



Linkerd Multi-Cluster Models

Flat Network

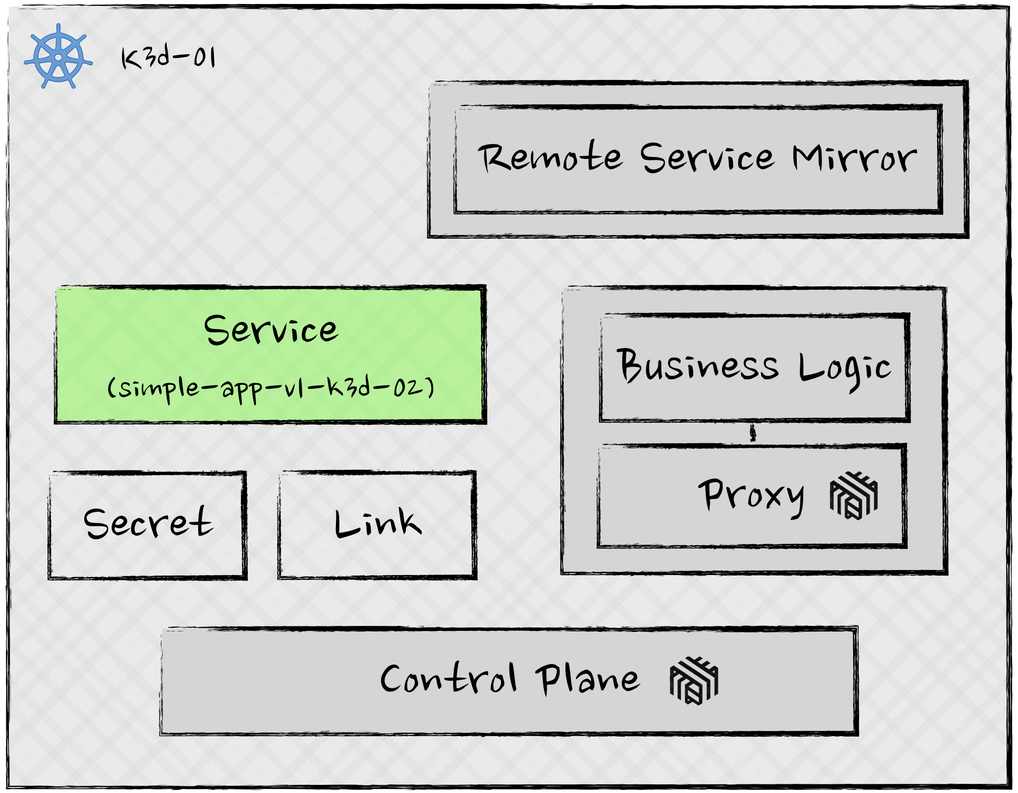


Terminal — zsh

```
time="2025-05-27T05:35:38Z" level=info msg="Received: OnUpdateCalled: {svc: Service: {name: simple-app-v1, namespace: simple-app, annotations: [[]], labels [[]]}}" cluster=k3d-02
time="2025-05-27T05:35:38Z" level=info msg="Received: RemoteServiceUnexported: {name: simple-app-v1, namespace: simple-app }" cluster=k3d-02
time="2025-05-27T05:35:38Z" level=info msg="GET https://10.247.0.1:443/apis/multicluster.linkerd.io/v1alpha3/namespaces/linkerd-multicluster/links/k3d-02 200 OK in 1 milliseconds"
time="2025-05-27T05:35:38Z" level=info msg="patching link status linkerd-multicluster/k3d-02" cluster=k3d-02
time="2025-05-27T05:35:38Z" level=debug msg="skipped processing endpoints object simple-app/simple-app-v1: missing mirror.linkerd.io/exported label" cluster=k3d-02
time="2025-05-27T05:35:38Z" level=info msg="PATCH https://10.247.0.1:443/apis/multicluster.linkerd.io/v1alpha3/namespaces/linkerd-multicluster/links/k3d-02/status 200 OK in 3 milliseconds"
time="2025-05-27T05:35:38Z" level=info msg="Deleting mirrored service simple-app/simple-app-v1-k3d-02" cluster=k3d-02
time="2025-05-27T05:35:38Z" level=info msg="DELETE https://10.247.0.1:443/api/v1/namespaces/simple-app/services/simple-app-v1-k3d-02 200 OK in 6 milliseconds"
time="2025-05-27T05:35:38Z" level=info msg="Successfully deleted service: simple-app/simple-app-v1-k3d-02" cluster=k3d-02
time="2025-05-27T05:35:40Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 5 milliseconds"
time="2025-05-27T05:35:42Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 2 milliseconds"
time="2025-05-27T05:35:42Z" level=info msg="Received: OnUpdateCalled: {svc: Service: {name: simple-app-v1, namespace: simple-app, annotations: [[]], labels [[]]}}" cluster=k3d-02
time="2025-05-27T05:35:42Z" level=info msg="Received: RemoteServiceExported: {service: Service: {name: simple-app-v1, namespace: simple-app, annotations: [[]], labels [[]]}}" cluster=k3d-02
time="2025-05-27T05:35:42Z" level=info msg="GET https://10.247.0.1:443/api/v1/namespaces/simple-app 200 OK in 1 milliseconds"
time="2025-05-27T05:35:42Z" level=info msg="Creating a new service mirror for simple-app/simple-app-v1" cluster=k3d-02
time="2025-05-27T05:35:42Z" level=debug msg="skipped processing endpoints object simple-app/simple-app-v1: missing mirror.linkerd.io/exported label" cluster=k3d-02
time="2025-05-27T05:35:42Z" level=info msg="POST https://10.247.0.1:443/api/v1/namespaces/simple-app/services 201 Created in 8 milliseconds"
time="2025-05-27T05:35:42Z" level=info msg="GET https://10.247.0.1:443/apis/multicluster.linkerd.io/v1alpha3/namespaces/linkerd-multicluster/links/k3d-02 200 OK in 1 milliseconds"
time="2025-05-27T05:35:42Z" level=info msg="patching link status linkerd-multicluster/k3d-02" cluster=k3d-02
time="2025-05-27T05:35:42Z" level=info msg="PATCH https://10.247.0.1:443/apis/multicluster.linkerd.io/v1alpha3/namespaces/linkerd-multicluster/links/k3d-02/status 200 OK in 3 milliseconds"
time="2025-05-27T05:35:42Z" level=debug msg="Link update ignored (only status changed): k3d-02"
time="2025-05-27T05:35:44Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 4 milliseconds"
time="2025-05-27T05:35:46Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 6 milliseconds"
time="2025-05-27T05:35:48Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 1 milliseconds"
time="2025-05-27T05:35:50Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 6 milliseconds"
time="2025-05-27T05:35:52Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 3 milliseconds"
time="2025-05-27T05:35:54Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 3 milliseconds"
time="2025-05-27T05:35:56Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 6 milliseconds"
time="2025-05-27T05:35:58Z" level=info msg="PUT https://10.247.0.1:443/apis/coordination.k8s.io/v1/namespaces/linkerd-multicluster/leases/service-mirror-write-k3d-02 200 OK in 2 milliseconds"
```

Linkerd Multi-Cluster Models

Flat Network

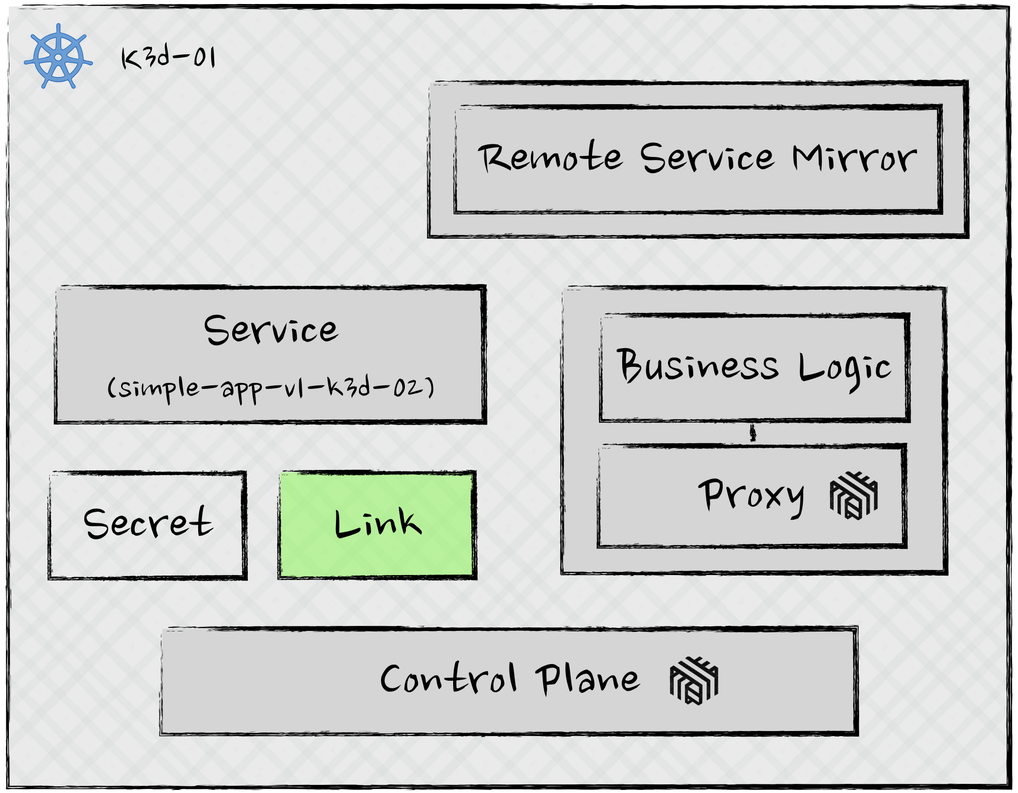


Terminal — zsh

```
[k3d-01|default] gtrekter@MacBook-Pro-M4 playground-scripts % kubectl get svc -n simple-app simple-app-v1-k3d-02 -o yaml
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"name":"simple-app-v1","namespace":"simple-app"},"spec":{"ports":
[{"port":80,"targetPort":5678}],"selector":{"app":"simple-app-v1","version":"v1"}}}
    mirror.linkerd.io/remote-resource-version: "2066"
    mirror.linkerd.io/remote-svc-fq-name: simple-app-v1.simple-app.svc.cluster.local
  creationTimestamp: "2025-05-27T05:35:42Z"
  labels:
    mirror.linkerd.io/cluster-name: k3d-02
    mirror.linkerd.io/mirrored-service: "true"
    multicluster.linkerd.io/remote-discovery: k3d-02
    multicluster.linkerd.io/remote-service: simple-app-v1
  name: simple-app-v1-k3d-02
  namespace: simple-app
  resourceVersion: "2373"
  uid: d51cf5e6-66a0-4bf0-b1e7-9f46a9149903
spec:
  clusterIP: 10.247.79.82
  clusterIPs:
  - 10.247.79.82
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - port: 80
    protocol: TCP
    targetPort: 5678
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
[k3d-01|default] gtrekter@MacBook-Pro-M4 playground-scripts %
```

Linkerd Multi-Cluster Models

Flat Network

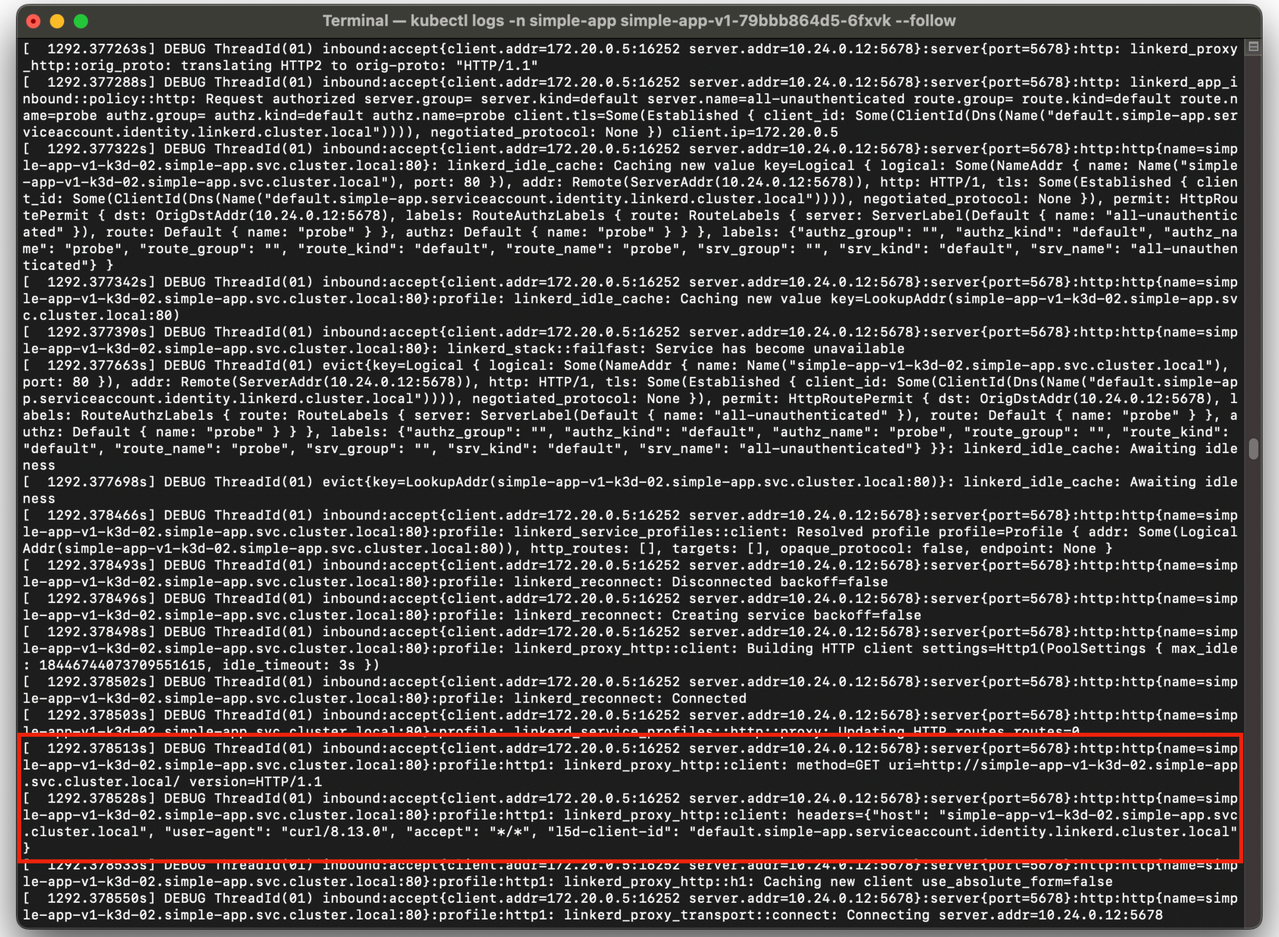


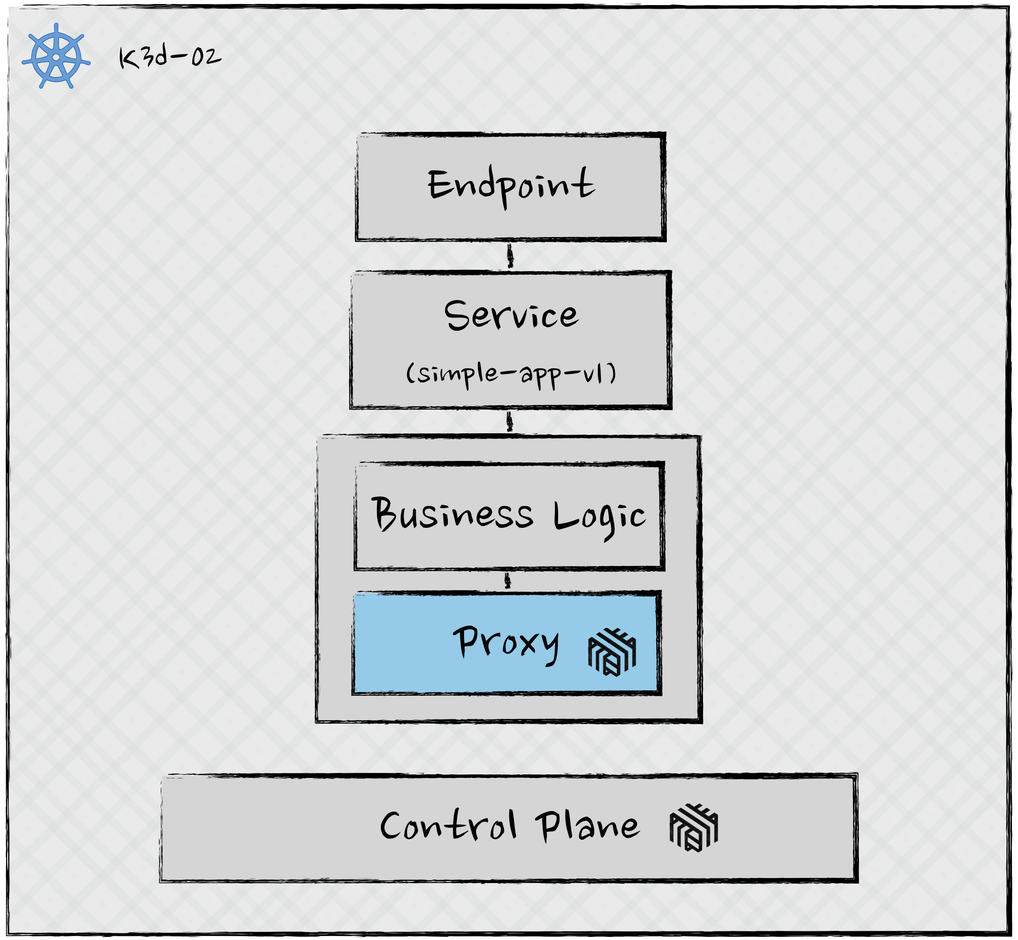
Terminal — zsh

```
[k3d-01|default] gtrekter@MacBook-Pro-M4 playground-scripts % kubectl describe link -n linkerd-multicluster k3d-02
Name:         k3d-02
Namespace:    linkerd-multicluster
Labels:       <none>
Annotations:  linkerd.io/created-by: linkerd/cli enterprise-2.18.0
API Version:  multicluster.linkerd.io/v1alpha3
Kind:         Link
Metadata:
  Creation Timestamp:  2025-05-27T05:32:10Z
  Generation:          1
  Resource Version:    3583
  UID:                 7eb2e5a3-de51-4ef5-ab38-820fee7b0f76
Spec:
  Cluster Credentials Secret:  cluster-credentials-k3d-02
  Federated Service Selector:
    Match Labels:
      mirror.linkerd.io/federated:  member
  Probe Spec:
    Failure Threshold:  3
    Timeout:            30s
  Remote Discovery Selector:
    Match Labels:
      mirror.linkerd.io/exported:    remote-discovery
  Target Cluster Domain:             cluster.local
  Target Cluster Linkerd Namespace:  linkerd
  Target Cluster Name:               k3d-02
Status:
  Mirror Services:
    Conditions:
      Last Transition Time:  2025-05-27T05:35:42Z
      Local Ref:
        Group:        core
        Kind:         Service
        Name:         simple-app-v1-k3d-02
        Namespace:    simple-app
      Message:
      Reason:         Mirrored
      Status:         True
      Type:           Mirrored
    Controller Name:  linkerd.io/service-mirror
    Remote Ref:
      Group:      core
      Kind:       Service
      Name:       simple-app-v1
      Namespace:  simple-app
    Conditions:
      Last Transition Time:  2025-05-27T05:44:34Z
      Local Ref:
        Group:        core
        Kind:         Service
        Name:         simple-app-v2-k3d-02
```

Linkerd Multi-Cluster Models

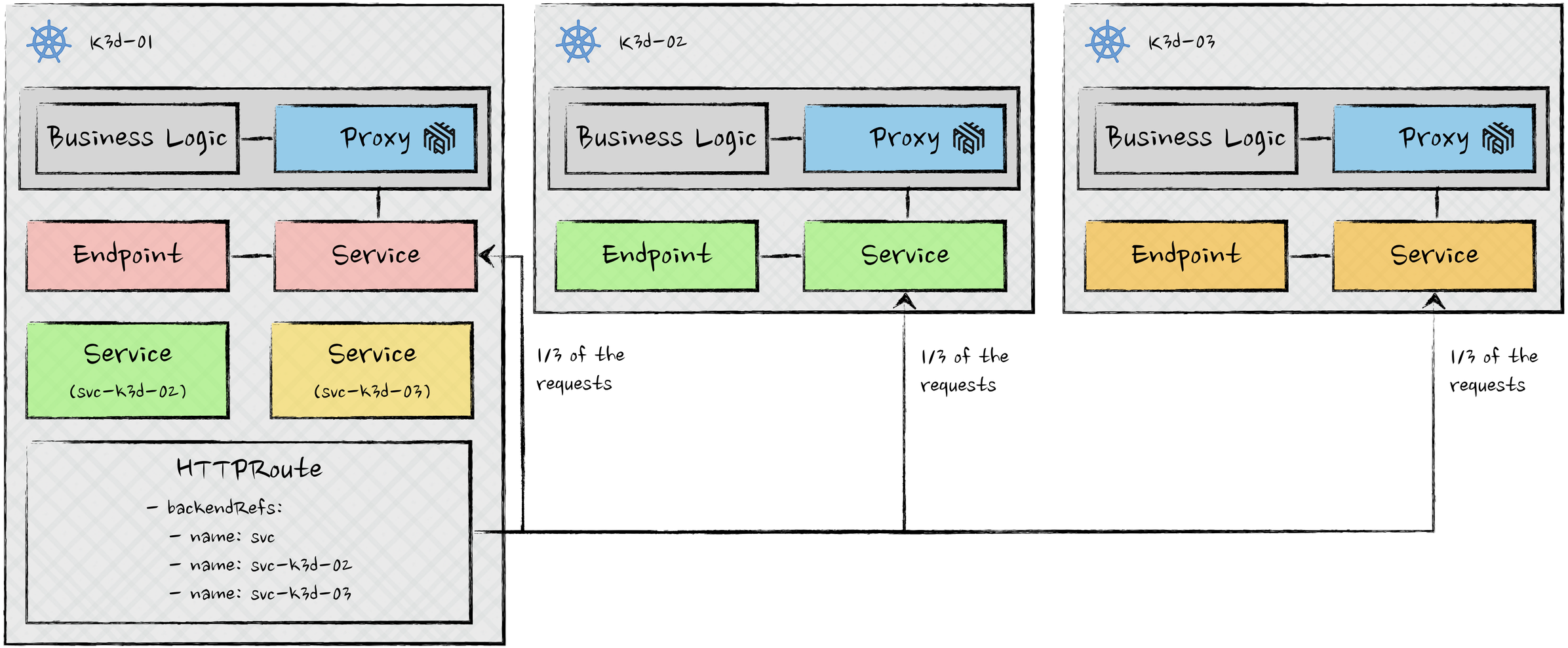
Flat Network





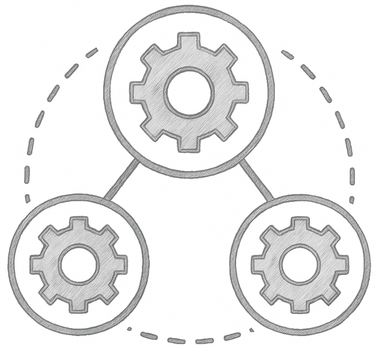
Linkerd Multi-Cluster Models

Flat Network



Linkerd Multi-Cluster Models

Federated Services



Federates

Pros:

- All the members of the federated service will have a single service independently from the cluster.
- Use Linkerd Proxy Exponentially Weighted Moving Average (EWMA) to route the requests.

Cons:

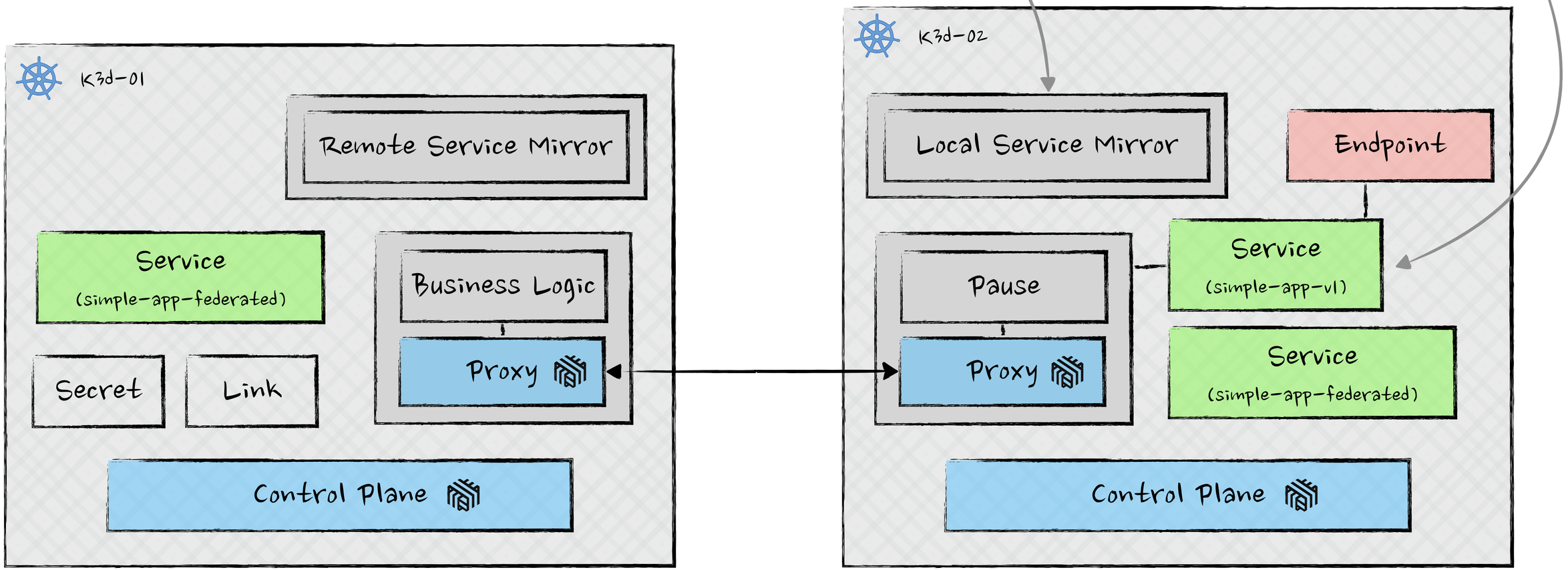
- You will still need to face the same challenges of the flat network.

Linkerd Multi-Cluster Models

Federated Services

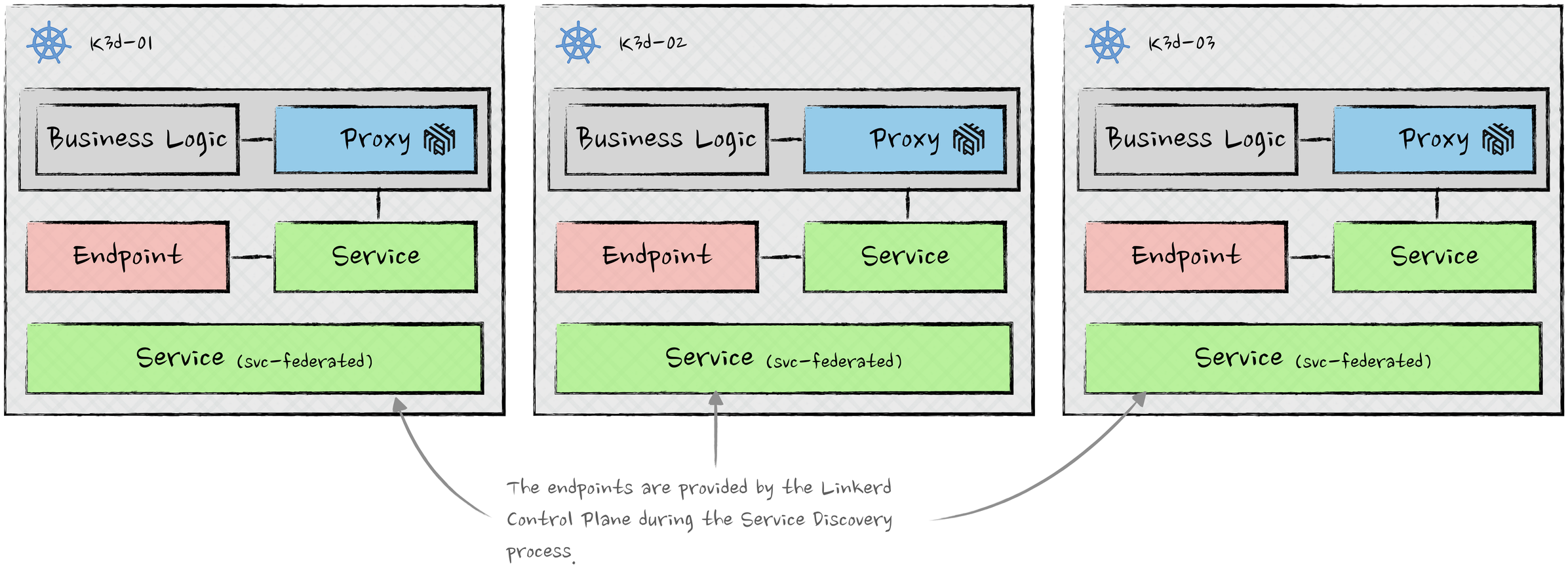
The Remote Service Mirror connect to the cluster using the credentials stored in the secret and watches for Services with:
label:
mirror.linkerd.io/federated=member

The Local Service Mirror watches services in the local cluster with the federated label.



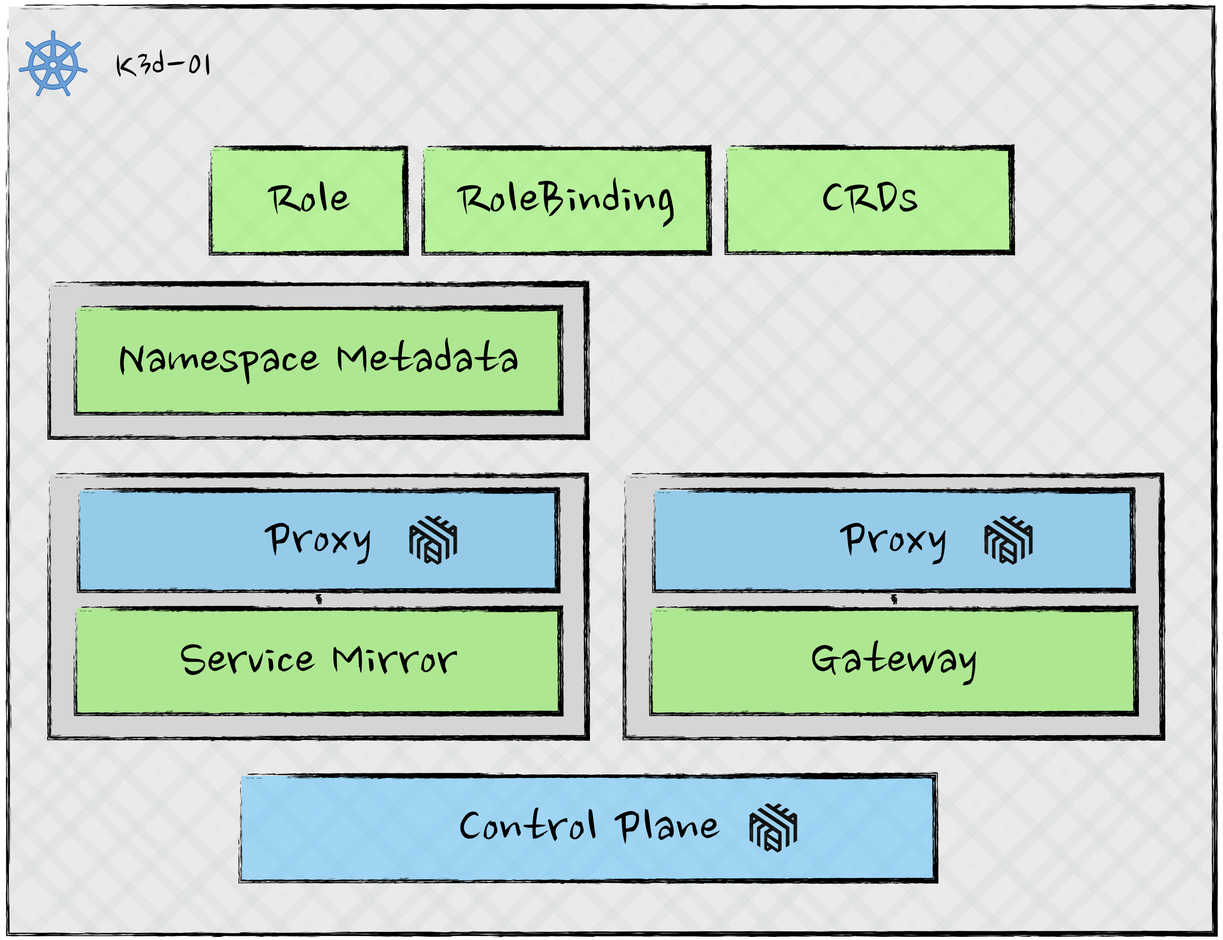
Linkerd Multi-Cluster Models

Federated Services



Phases to Link Clusters

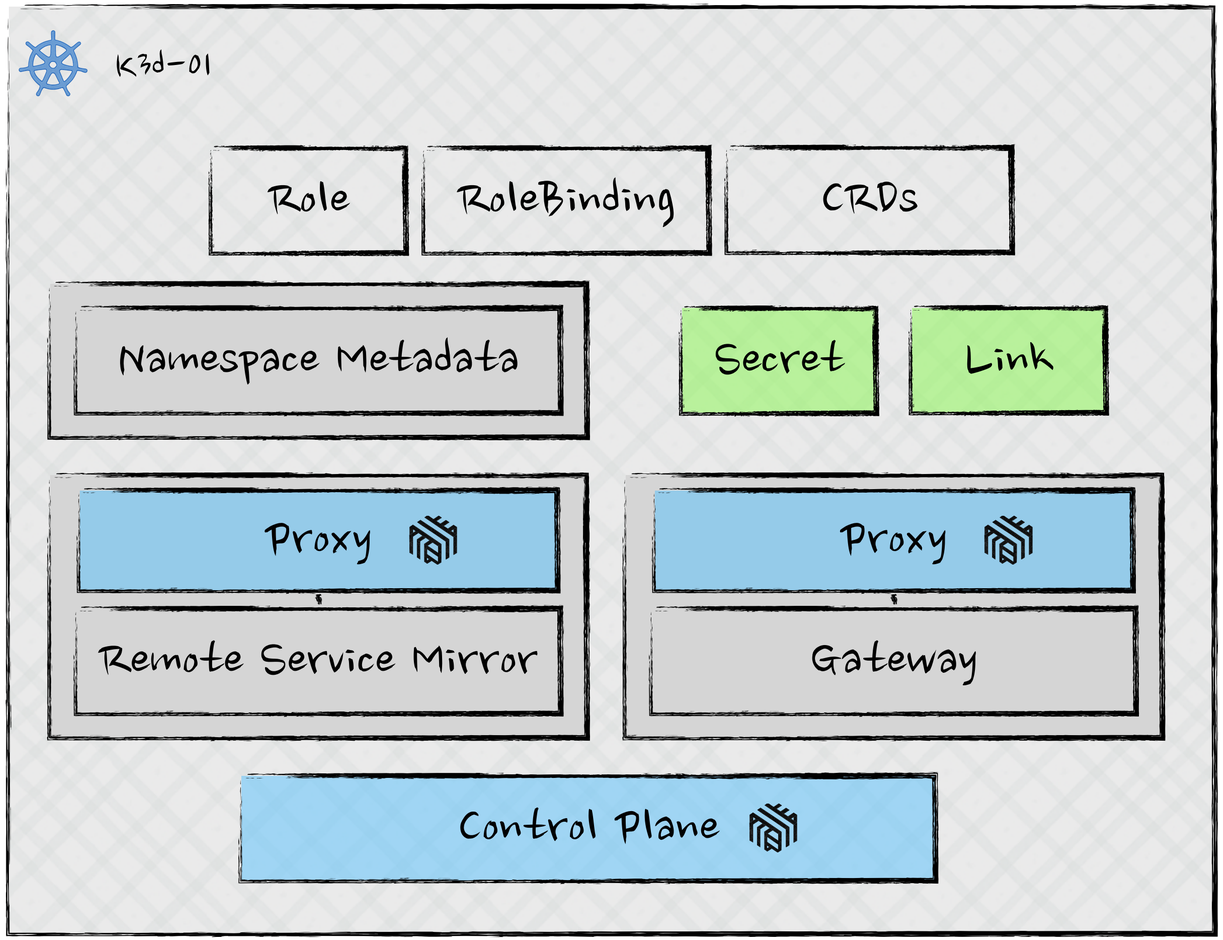
Deploy all the core resources and get ready to accept Links



- By Installing the Linkerd Multicluster Helm chart in the cluster, you will deploy the Service Mirror controller, the RBAC, LInk CRD, and, optionally, the Gateway.
- The Service Mirror will immediatly start watching for Link resources and Secrets of type mirror.linkerd.io/remote-kubeconfig to connect remote clusters.

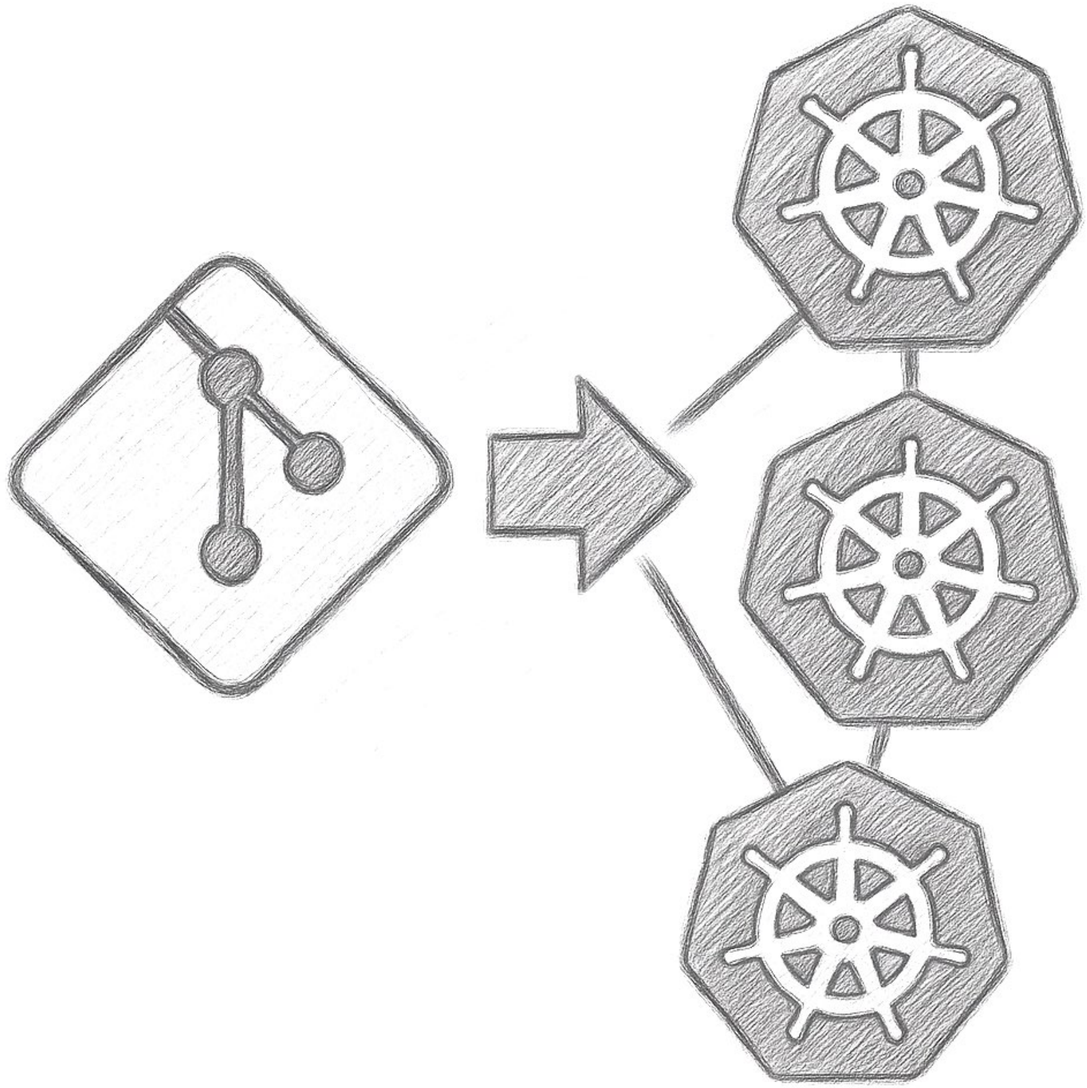
Phases to Link Clusters

Generate and apply the cluster Link



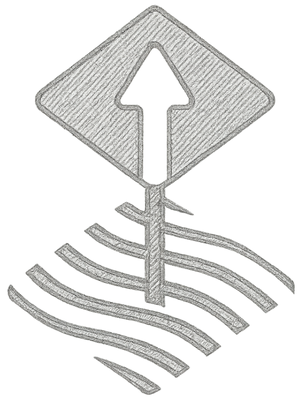
- Run linkerd CLI link-gen to get the service-account token and credentials from the remote cluster, and a Link resource with references to the remote Kubernetes API endpoint and (optionally) the Gateway address/port.
- Once applied to the local cluster, the Service Mirror detects it and begins mirroring the remote services.

GitOps

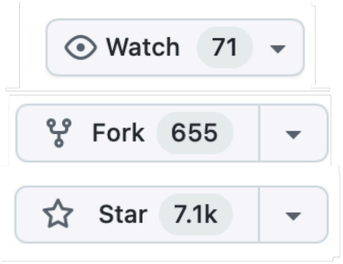


Scaling Cluster Management with GitOps

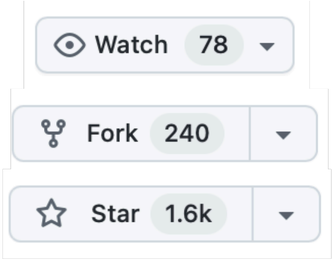
GitOps Makes It Easy



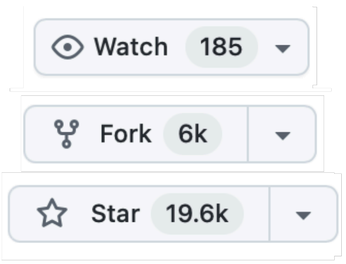
Flux



Rancher Fleet

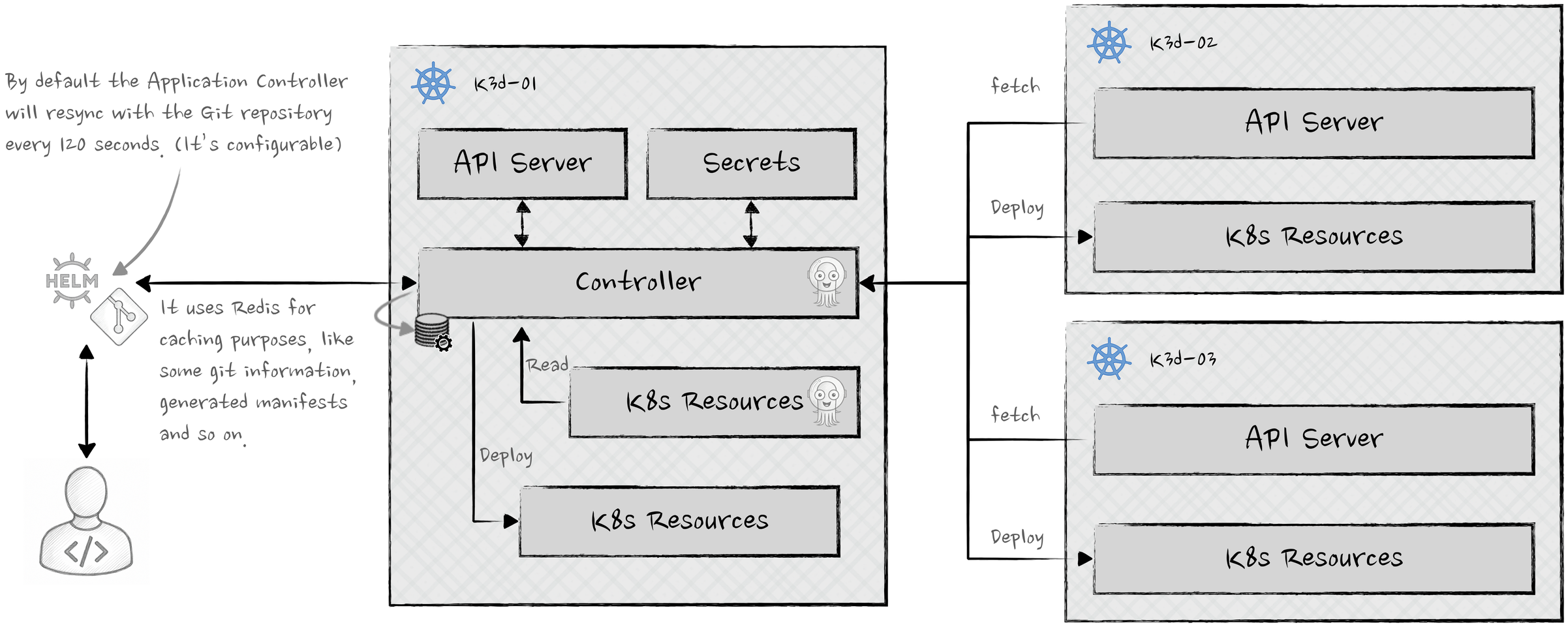


Argo



Scaling Cluster Management with GitOps

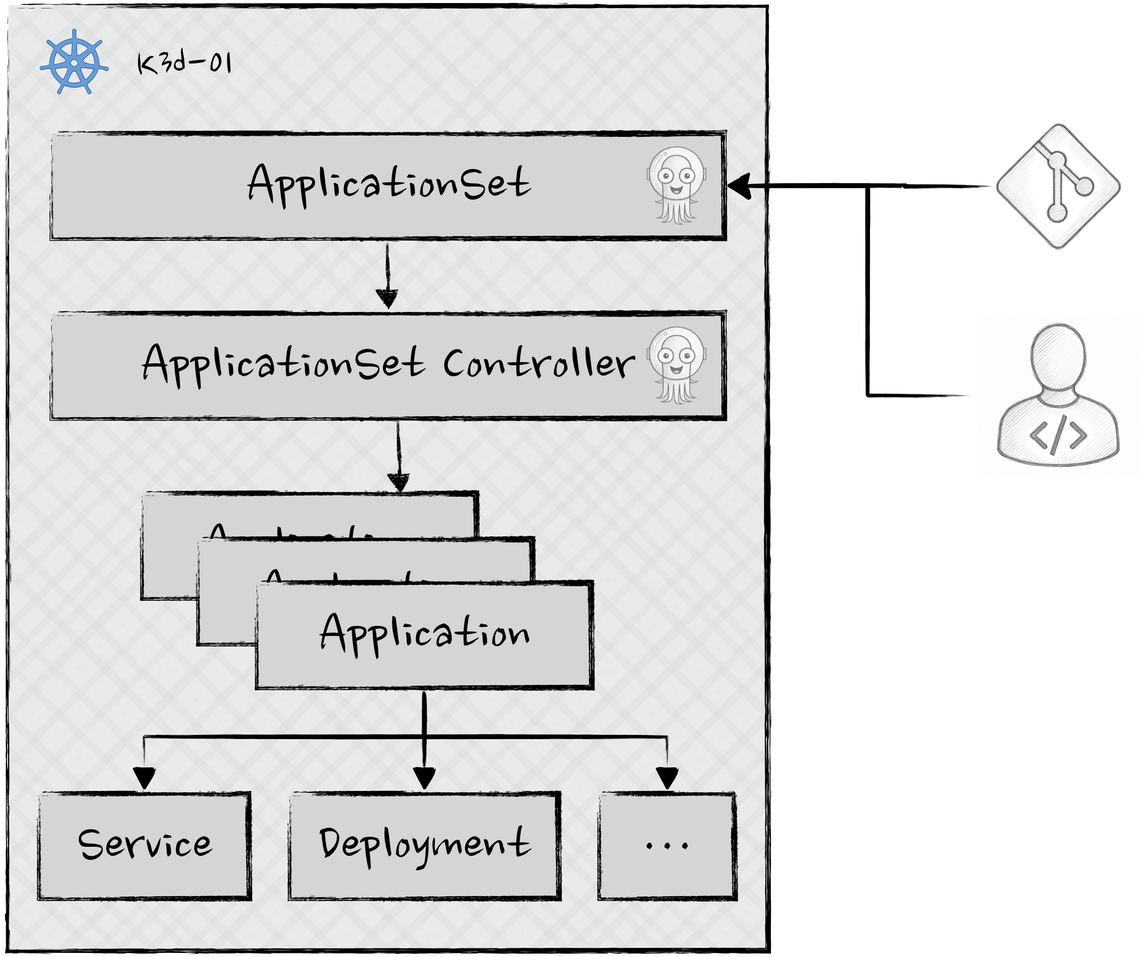
GitOps Workflows with Argo CD



Scaling Cluster Management with GitOps

Automating at Scale: ApplicationSets

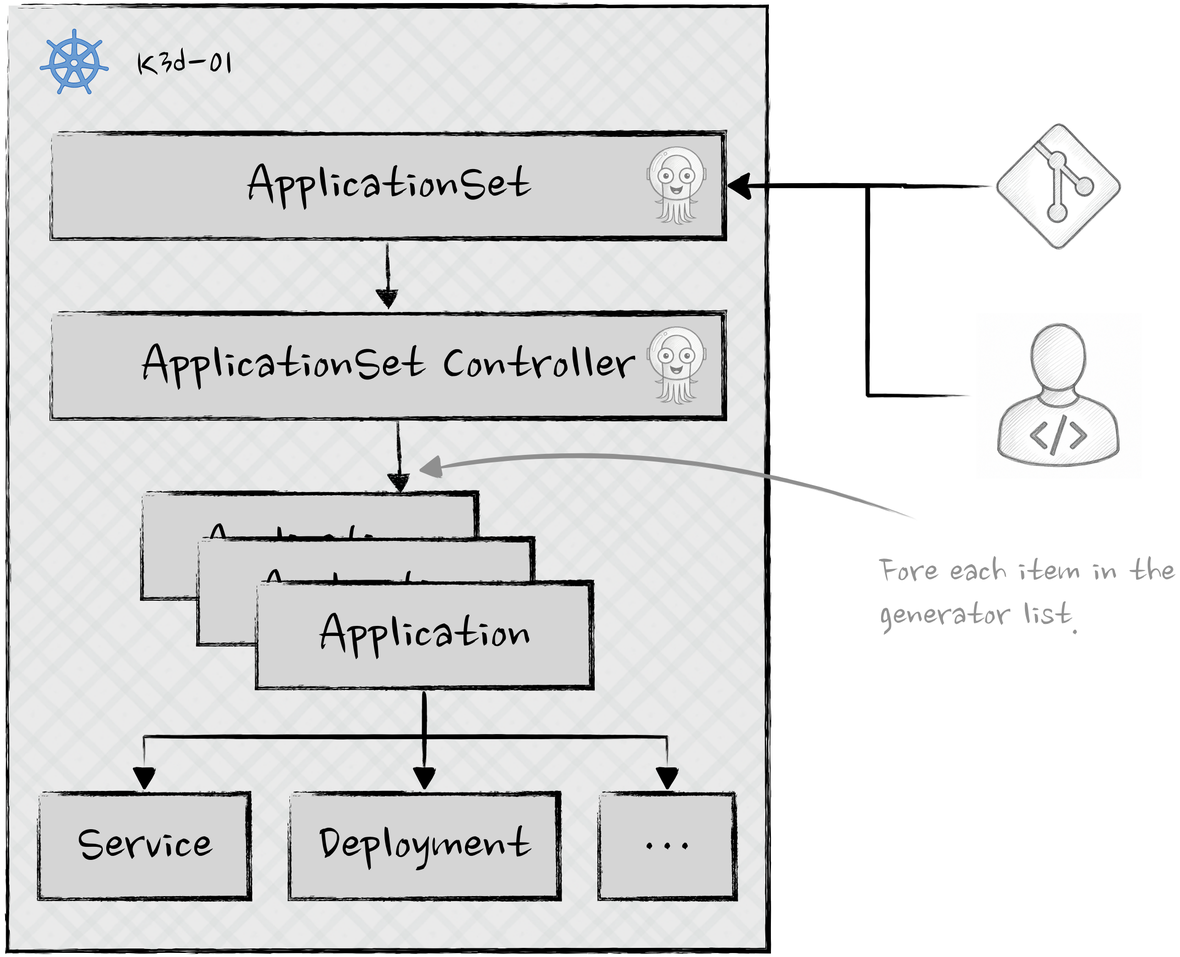
```
Terminal — -zsh
gtrekter@MacBook-Pro-M4 Repositories % cat ./linkerd-operator-argocd/sample/linkerd.yaml
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: linkerd-enterprise
  namespace: argocd
spec:
  syncPolicy:
    applicationsSync: create-update
    preserveResourcesOnDeletion: true
  template:
    metadata:
      name: "linkerd-enterprise-in-cluster"
    spec:
      project: default
      destination:
        server: "https://kubernetes.default.svc"
        namespace: linkerd
      sources:
      - repoURL: "https://helm.buoyant.cloud"
        chart: linkerd-enterprise-crds
        targetRevision: "2.17.1"
        helm:
          releaseName: linkerd-enterprise-crds
          valuesObject:
            fullnameOverride: linkerd-enterprise-crds
            enableHttpRoutes: false
      - repoURL: "https://helm.buoyant.cloud"
        chart: linkerd-enterprise-control-plane
        targetRevision: "2.17.1"
        helm:
          parameters:
            - name: license
              value: **********************
            - name: identity.issuer.scheme
              value: kubernetes.io/tls
          valuesObject:
            proxy:
              resources:
                cpu:
                  request: 50m
                memory:
                  request: 64Mi
gtrekter@MacBook-Pro-M4 Repositories %
```



Scaling Cluster Management with GitOps

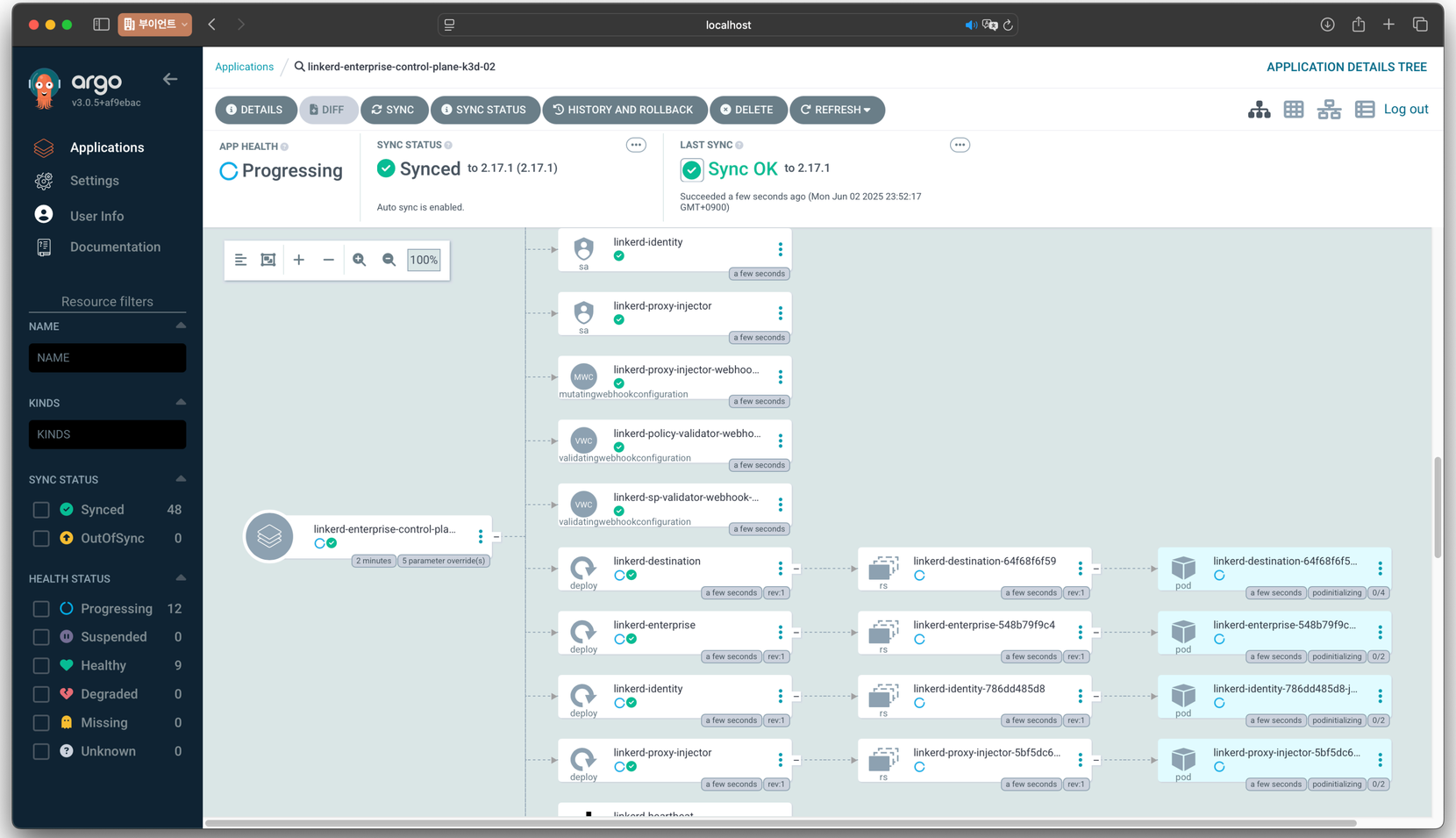
Automating at Scale: Generators

```
Terminal — -zsh
gtrekter@MacBook-Pro-M4 Repositories % cat ./linkerd-operator-argocd/sample/linkerd.yaml
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: linkerd-enterprise
  namespace: argocd
spec:
  generators:
    - list:
        elements:
          - cluster: k3d-01
            url: "https://172.18.0.13:6443"
            values:
              env: laboratory
          - cluster: in-cluster
            url: "https://kubernetes.default.svc"
            values:
              env: development
  syncPolicy:
    applicationsSync: create-update
    preserveResourcesOnDeletion: true
  template:
    metadata:
      name: "linkerd-enterprise-{{ .cluster }}"
      labels:
        environment: "{{ .values.env }}"
    spec:
      project: default
      destination:
        server: "{{ .url }}"
        namespace: linkerd
      sources:
        - repoURL: "https://helm.buoyant.cloud"
          chart: linkerd-enterprise-crds
          targetRevision: "2.17.1"
          helm:
            releaseName: linkerd-enterprise-crds
            valuesObject:
              fullnameOverride: linkerd-enterprise-crds
              enableHttpRoutes: false
        - repoURL: "https://helm.buoyant.cloud"
          chart: linkerd-enterprise-control-plane
          targetRevision: "2.17.1"
          helm:
            parameters:
              - name: license
                value: **********
              - name: identity.issuer.scheme
                value: kubernetes.io/tls
            valuesObject:
              proxy:
                resources:
                  cpu:
                    request: 50m
                  memory:
                    request: 64Mi
gtrekter@MacBook-Pro-M4 Repositories %
```



Scaling Cluster Management with GitOps

Unified Multi-Cluster Dashboard



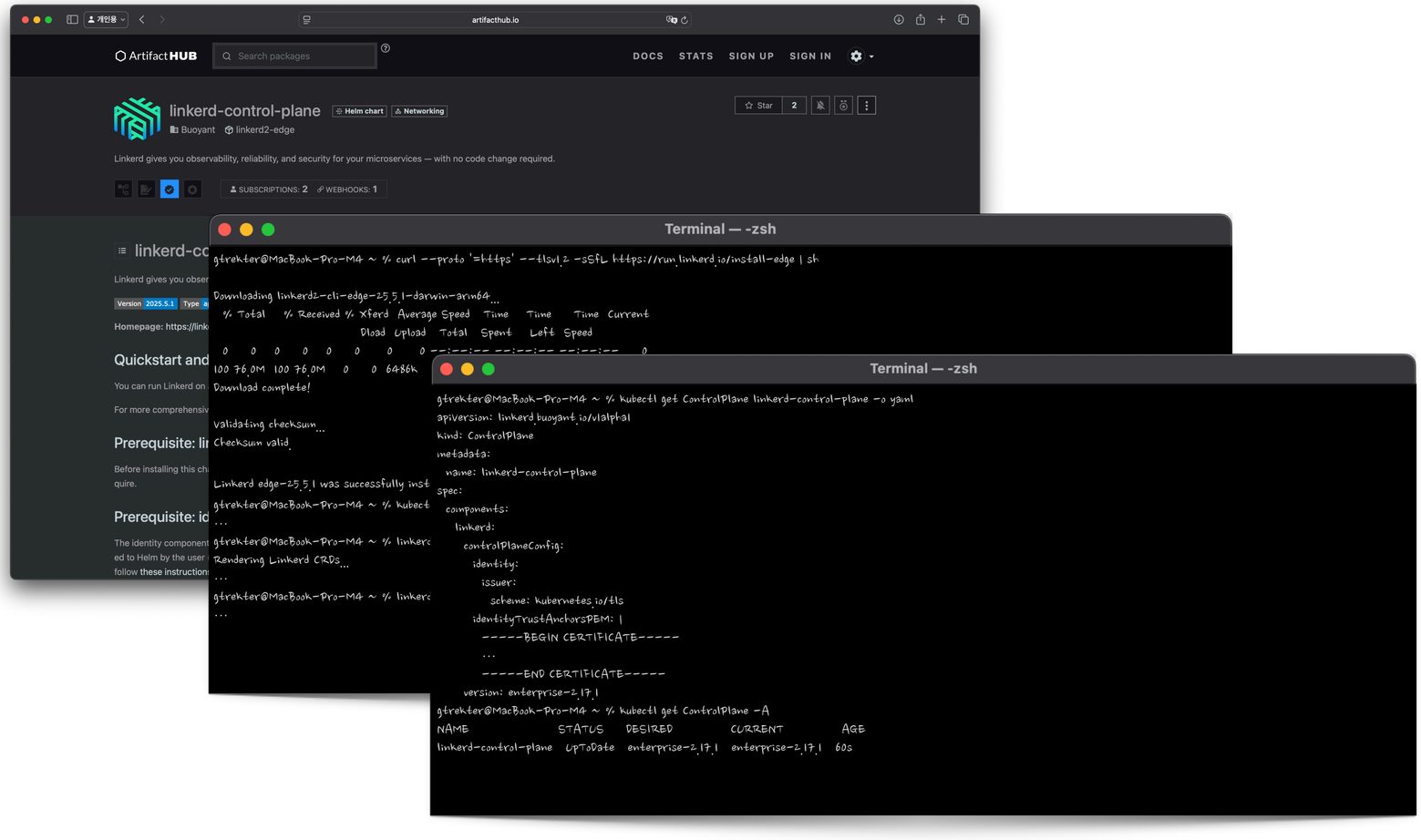
Getting Started

Mesh in 3 Commands

Install Linkerd via CLI, Helm, or Operator

linkerd.io

buoyant.io

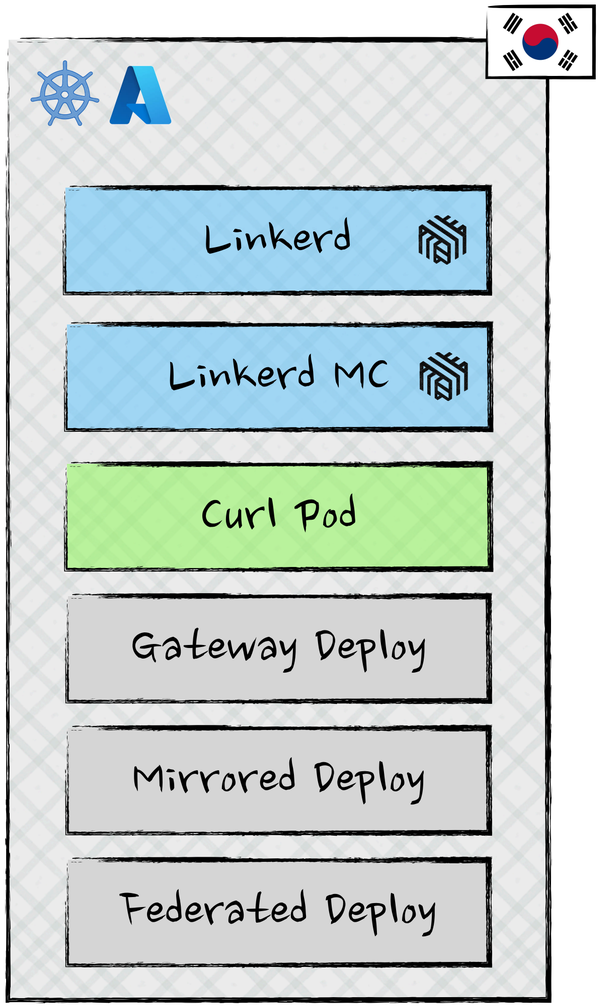


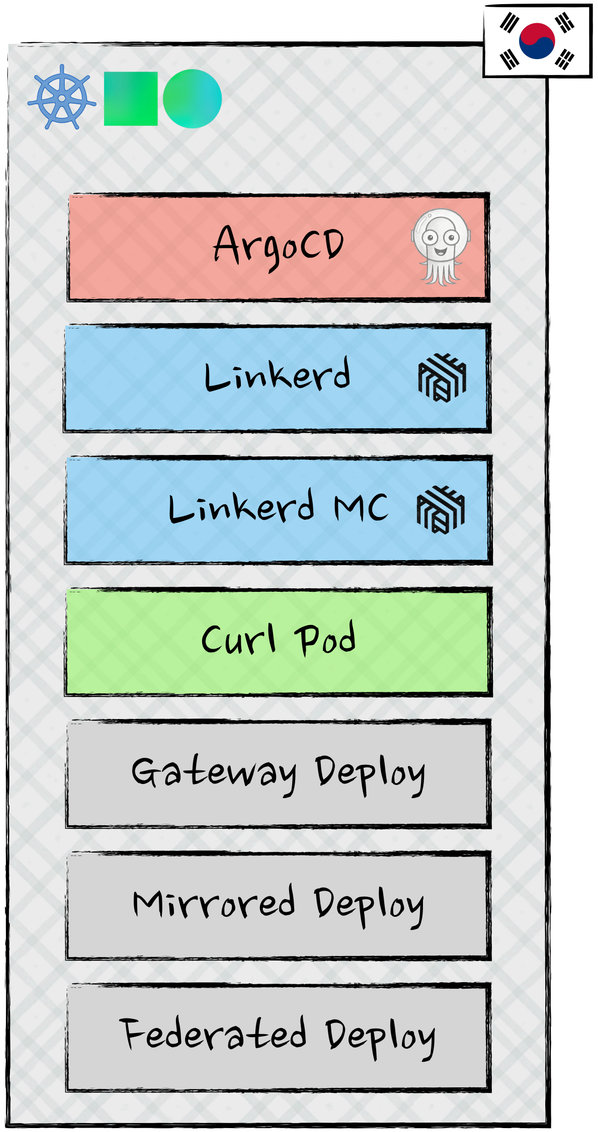
Demo

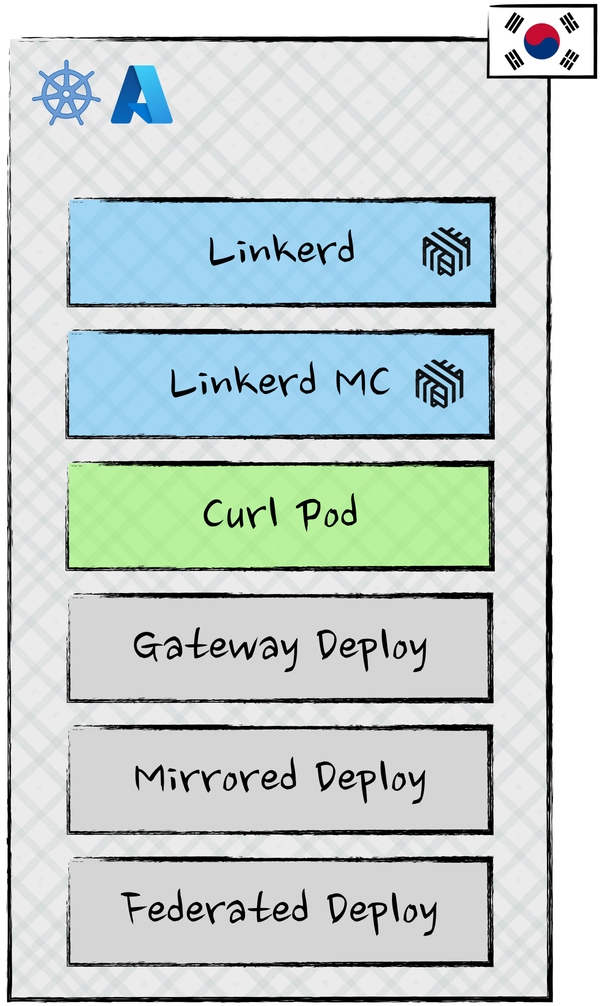
Scan to try it & view the code:
https://github.com/GTRekter/DevKorea-2026

Architecture

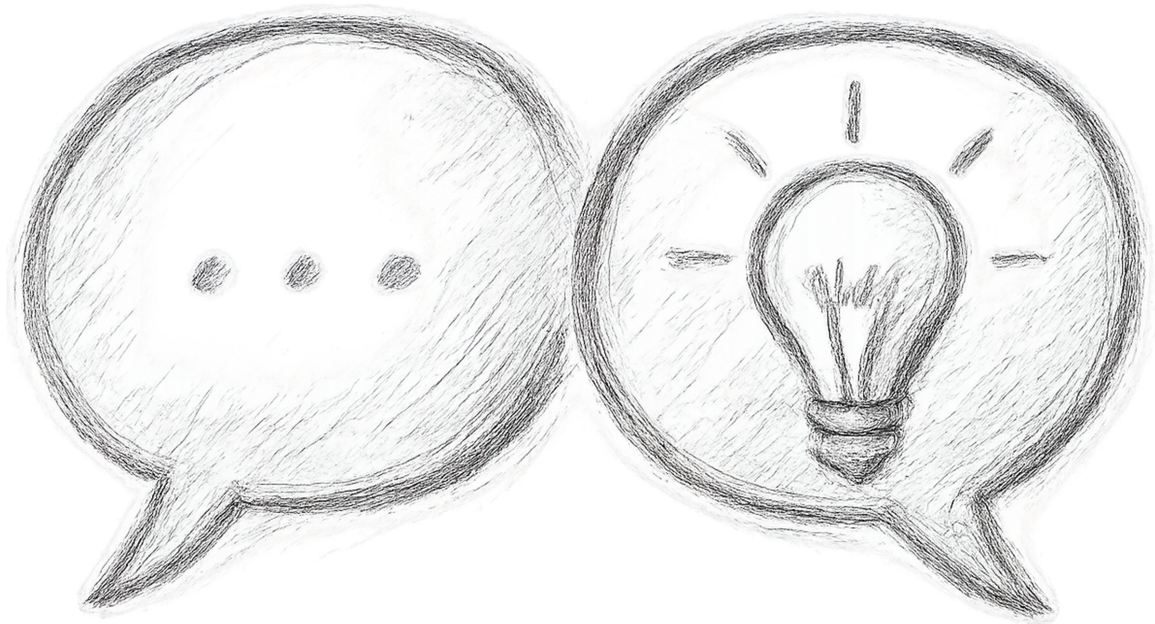
Components







Q&A




Thank you 谢谢 Gracias धन्यवाद شكراً Obrigado Danke ありがとうございます Merci Cảm ơn 감사합니다 ขอบคุณ Dziękuję Дякую Salamat Ευχαριστώ