KubeCon | CloudNativeCon

North America 2023

Patterns of Multi-Cluster Kubernetes

Dan Mckean
Sr Product Manager, MongoDB

George Hantzaras
Engineering Director, MongoDB

Mircea Cosbuc
Sr Engineer, MongoDB

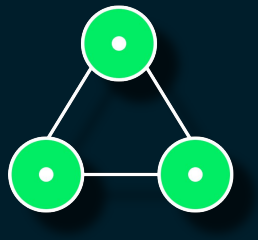
Agenda

- Motivations for Multi-Cluster
- Architectures
- Network Centric Patterns
- Cluster Inventory
- Workload Distribution
- Networking
- Multi-Cluster Operators and Controllers
- Future of Multi-cluster

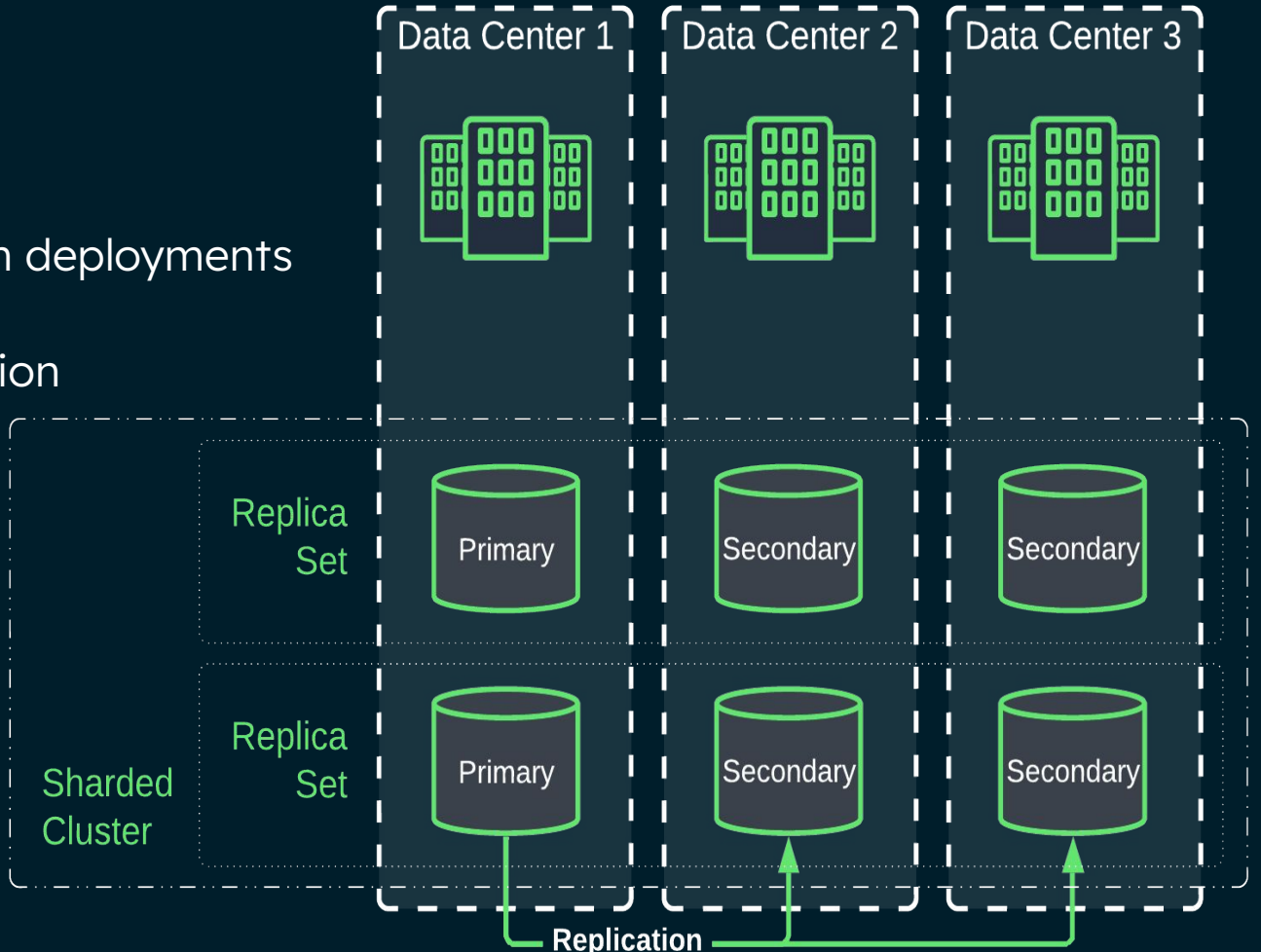
Motivations for Multi-Cluster

Motivations for Multi-Cluster Deployments

- Improved Performance & Reduced Latency
- Compliance with Regulations
- High Availability, Disaster Recovery, and Data Redundancy

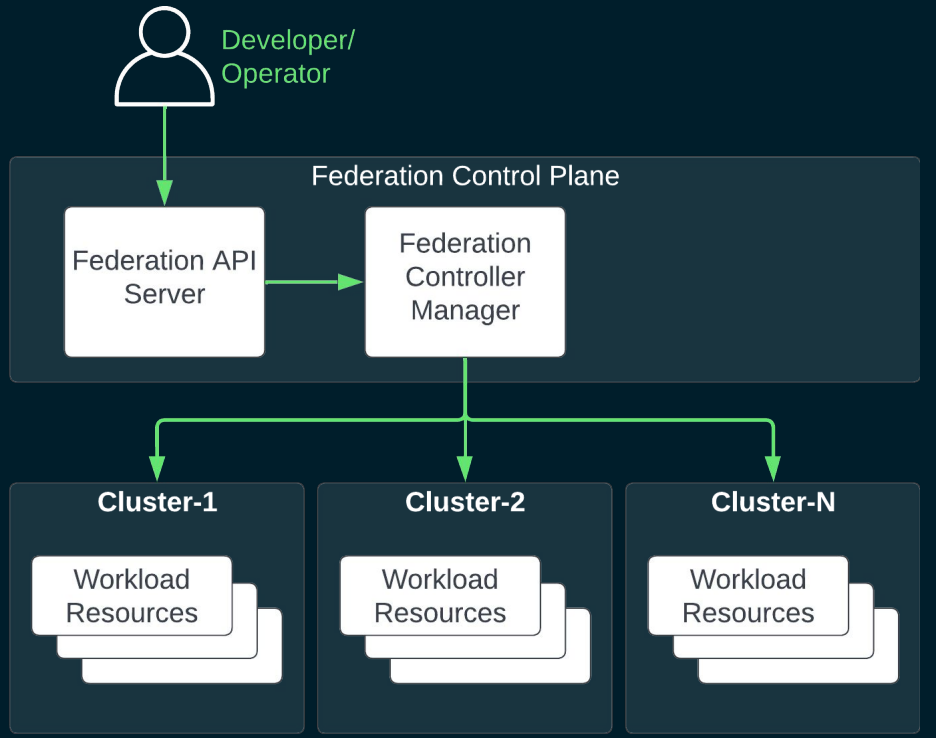


- Multi AZ, multi region deployments
- Cross region replication
- Segmenting data by location with multi-region shards
- Read-only

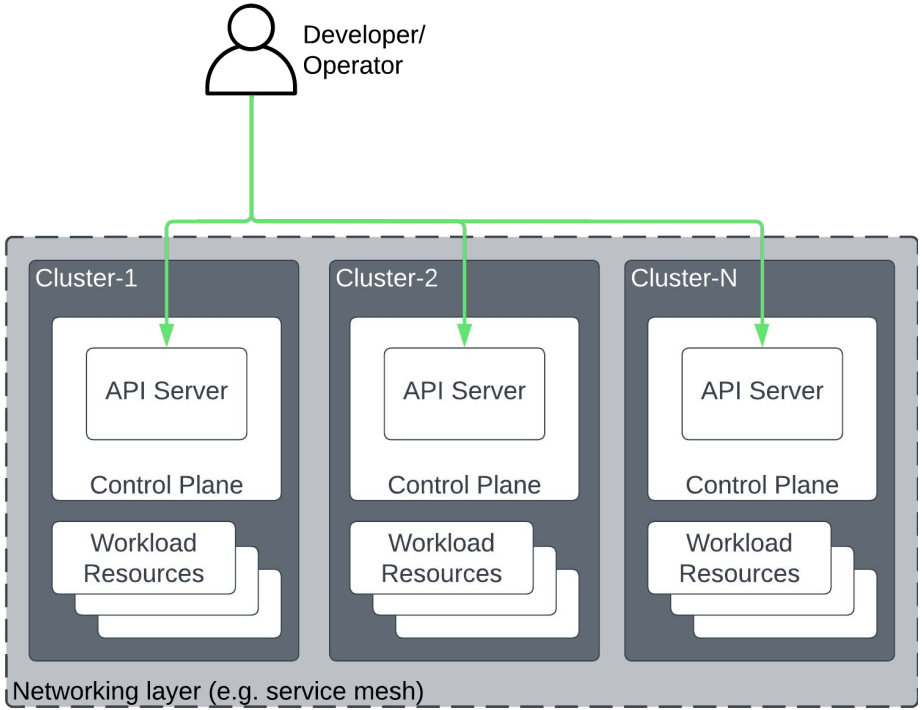


Architectures

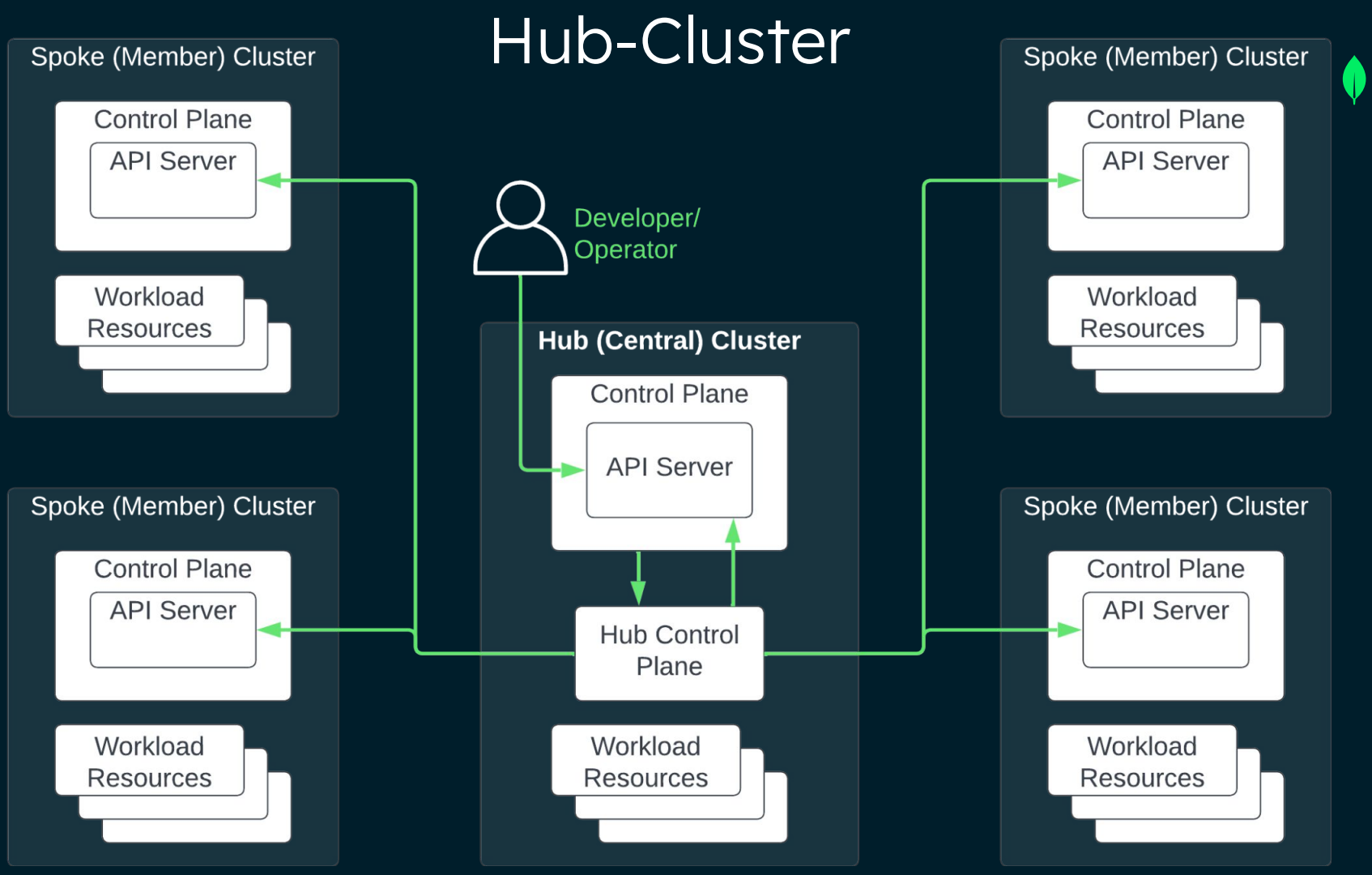
Kubernetes-Centric
(Cluster Federation)



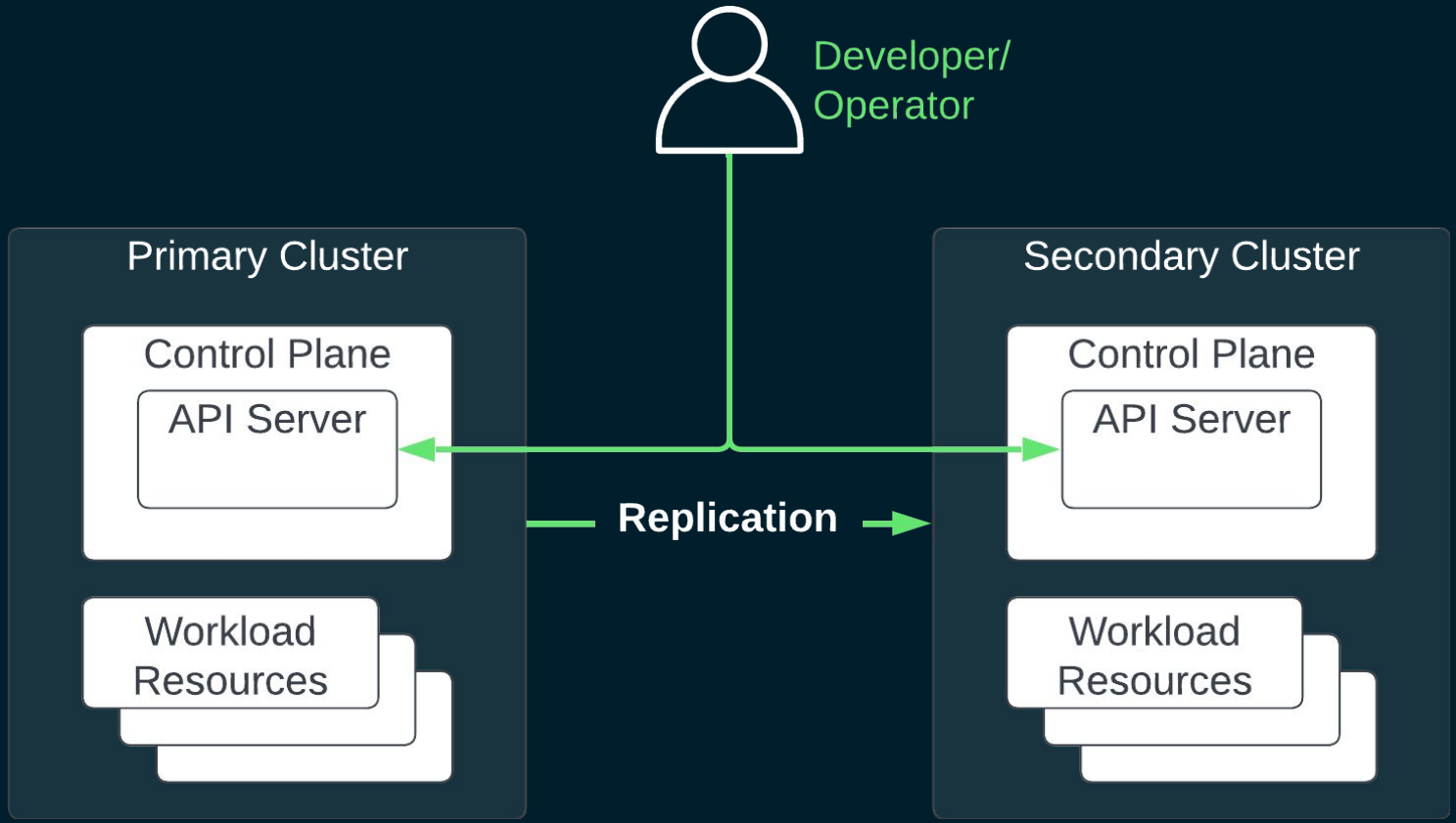
Network-Centric
(Workload distribution)



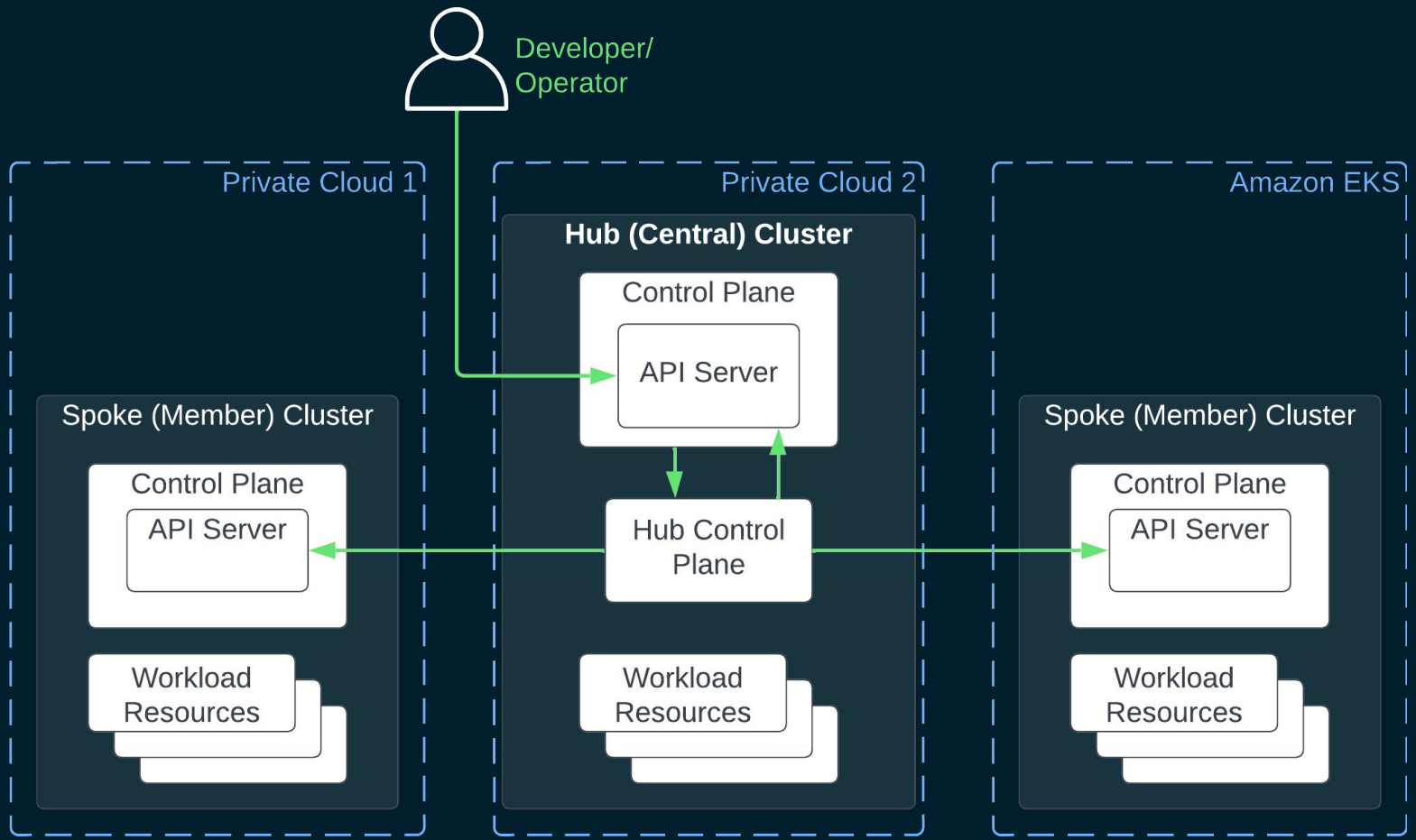
Network-Centric Patterns


Hub-Cluster
Spoke (Member) Cluster
Control Plane
API Server
Workload Resources
Spoke (Member) Cluster
Control Plane
API Server
Workload Resources
Developer/ Operator
Hub (Central) Cluster
Control Plane
API Server
Hub Control Plane
Workload Resources
Spoke (Member) Cluster
Control Plane
API Server
Workload Resources
Spoke (Member) Cluster
Control Plane
API Server
Workload Resources

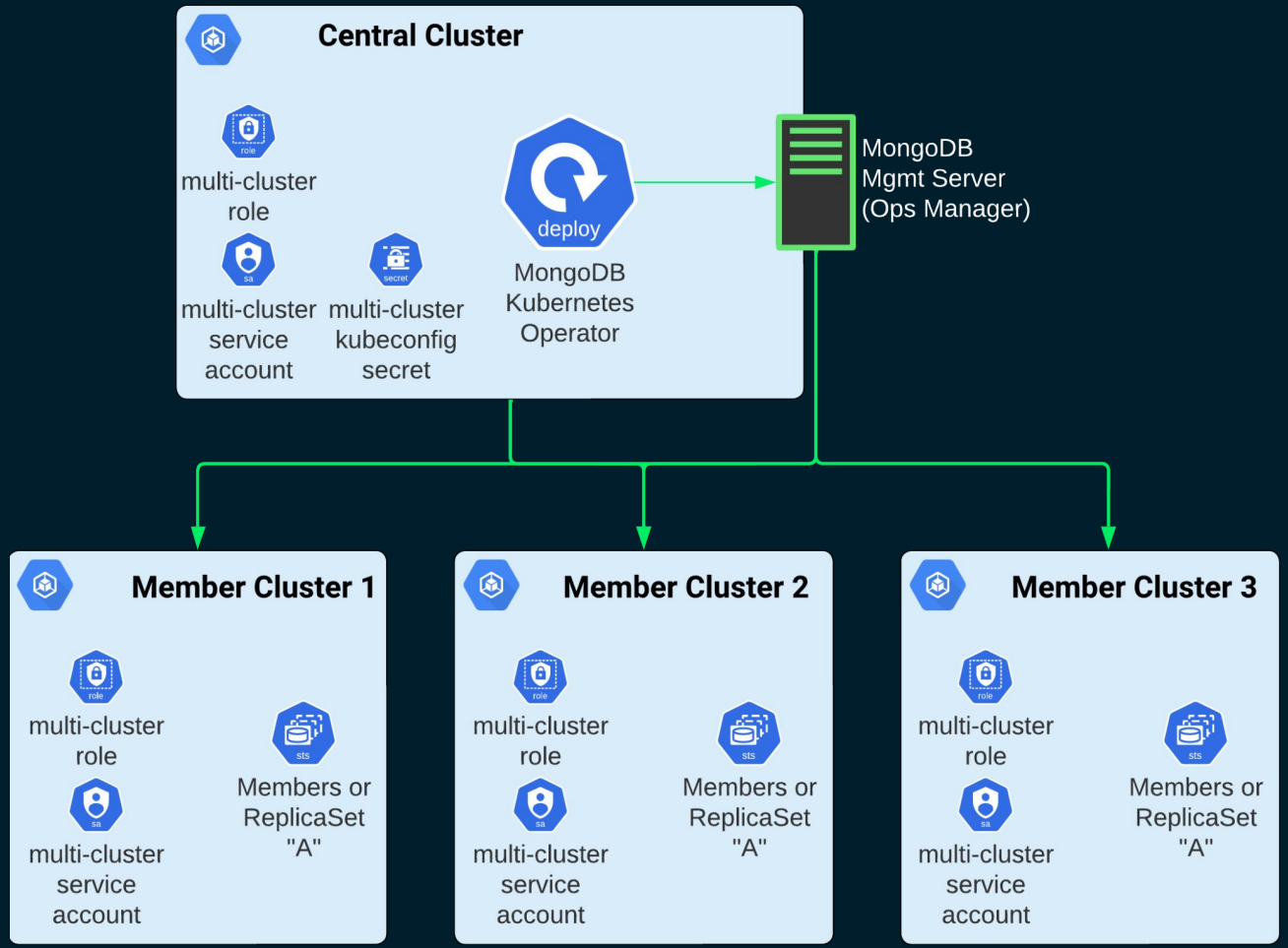
Replicated Application Architecture



Multi-cloud


Developer/
Operator
Private Cloud 1
Private Cloud 2
Amazon EKS
Hub (Central) Cluster
Control Plane
API Server
Hub Control
Plane
Spoke (Member) Cluster
Control Plane
API Server
Workload
Resources
Spoke (Member) Cluster
Control Plane
API Server
Workload
Resources
Workload
Resources

MongoDB Architecture



Cluster Inventory

Cluster Inventory

Requirements

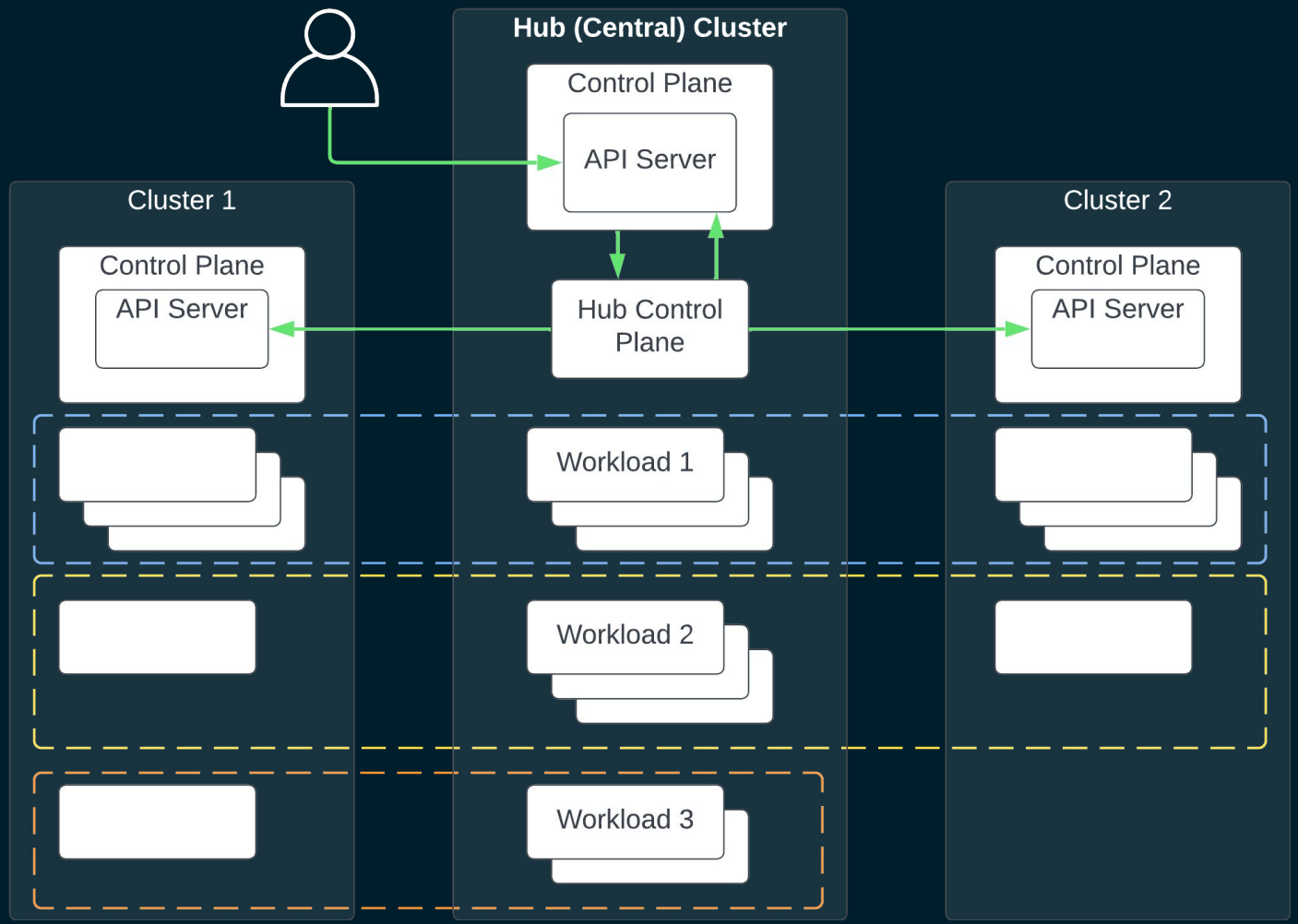
- Cluster credentials
- Cluster health checks
- Cluster resource tracking

MongoDB Cluster Inventory

- Creation of RBAC on member clusters
- Store the kubeconfig of member clusters

```
kubeconfig:
  apiVersion: v1
  clusters:
  - cluster:
      certificate-authority-data: <cluster-1-ca.crt>
      server: https://127.0.0.1:41727
    name: cluster-1
  contexts:
  - context:
      cluster: cluster-1
      namespace: mongodb
      user: cluster-user-1
    name: cluster-1
  kind: Config
  users:
  - name: cluster-user-1
    user:
      token: <cluster-1-token>
```

Workload Distribution



Considerations:

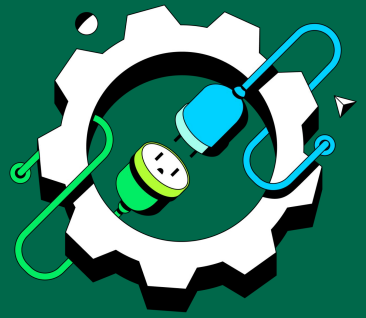
- Redundancy
- Geographical
- Resources

Workload Distribution

- Declarative approach using *spec.clusterSpecList*
- Users request distribution of the database nodes
- In case of Kubernetes cluster failure, we can re-distribute statefulsets to healthy clusters

```yaml
apiVersion: mongodb.com/v1
kind: MongoDBMultiCluster
metadata:
  name: multi-replica-set
spec:
  version: 6.0.6-ent
  type: ReplicaSet
  persistent: false
  duplicateServiceObjects: false
  credentials: my-credentials
  opsManager:
    configMapRef:
      name: my-project
  clusterSpecList:
    - clusterName: cluster1.mongokubernetes.com
      members: 2
      statefulSet:
        spec:
          # to override the default storage class for the pv
          volumeClaimTemplates:
          - metadata:
              name: data
            spec:
              accessModes: [ "ReadWriteOnce" ]
              storageClassName: "gp2"
    - clusterName: cluster2.mongokubernetes.com
      members: 1
      statefulSet:
        spec:
          volumeClaimTemplates:
          - metadata:
              name: data
            spec:
              accessModes: [ "ReadWriteOnce" ]
              storageClassName: "gp2"
    - clusterName: cluster3.mongokubernetes.com
      members: 1
```

Networking

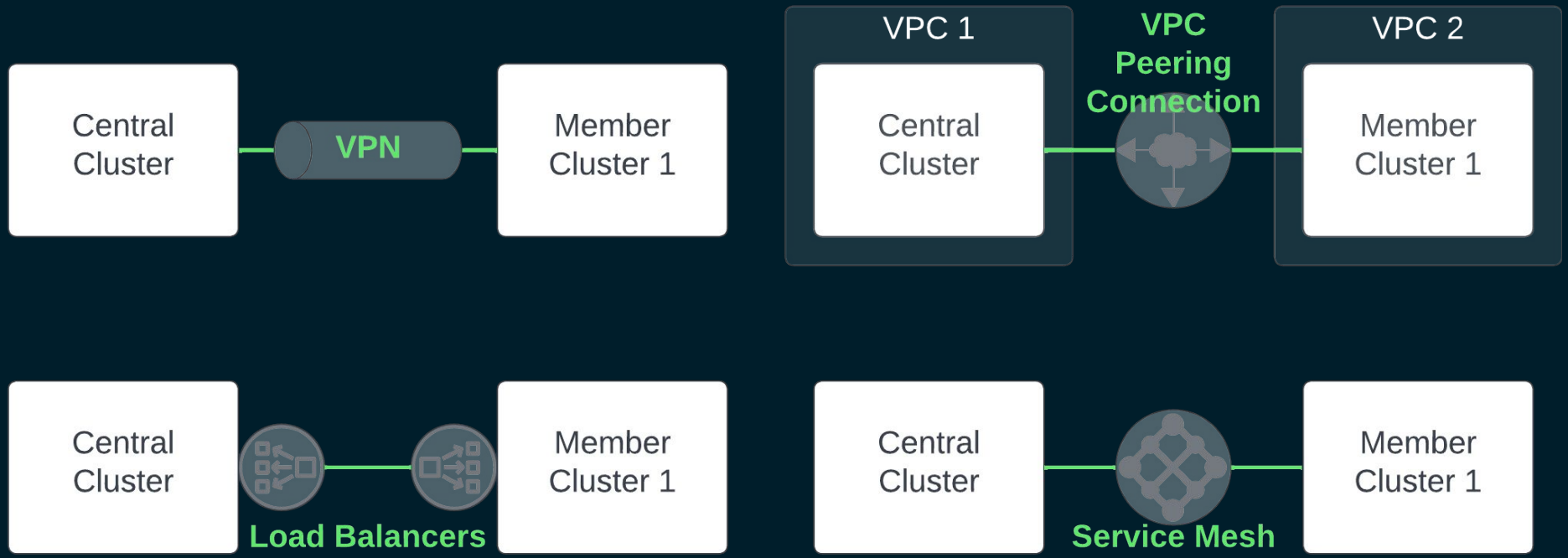


Networking

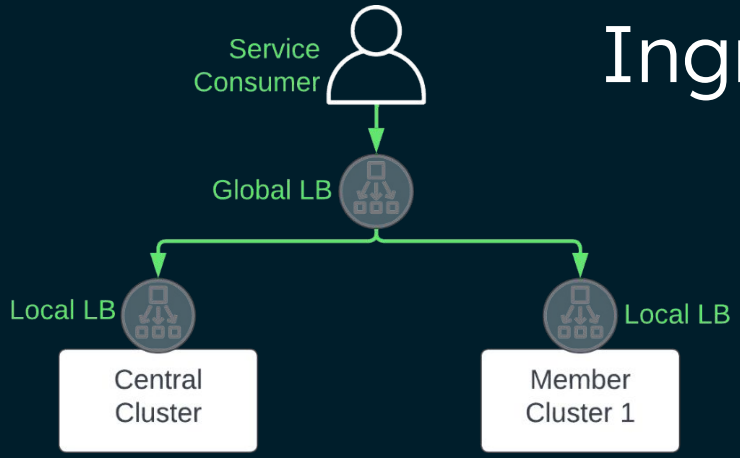
Considerations

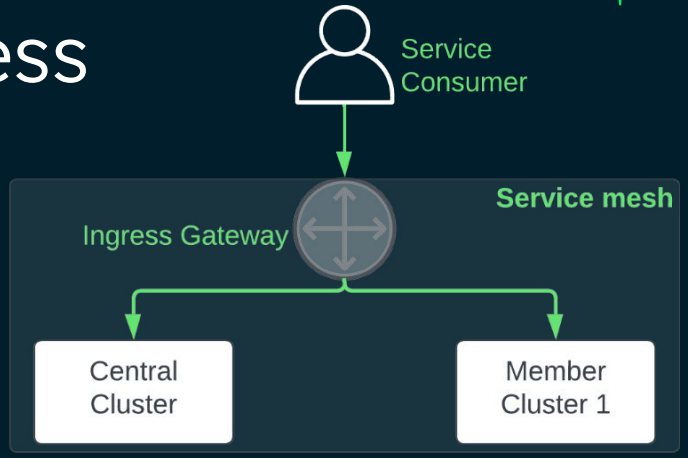
- Network topology
- Security
- Performance
- Reliability & resilience

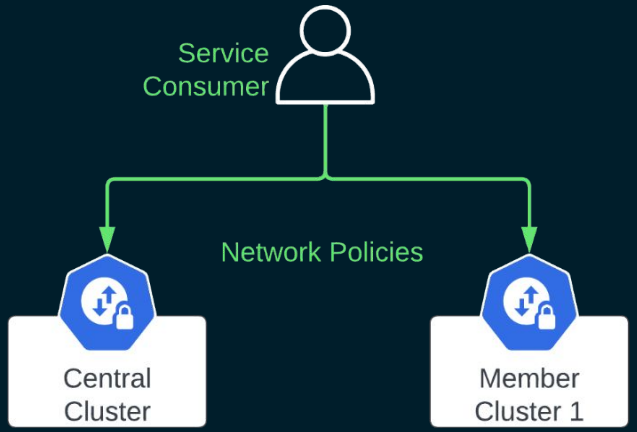
Inter-Cluster Networking

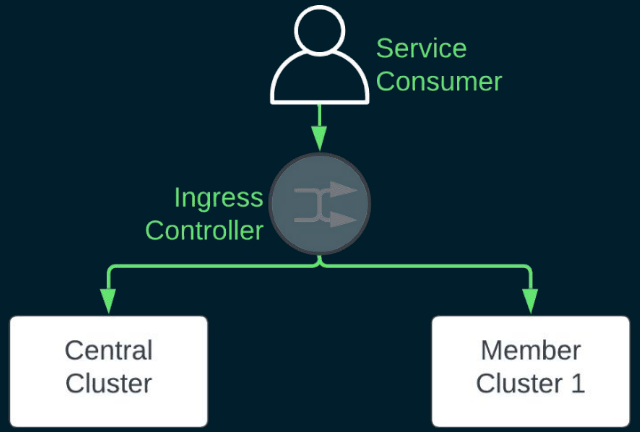


Ingress & Egress

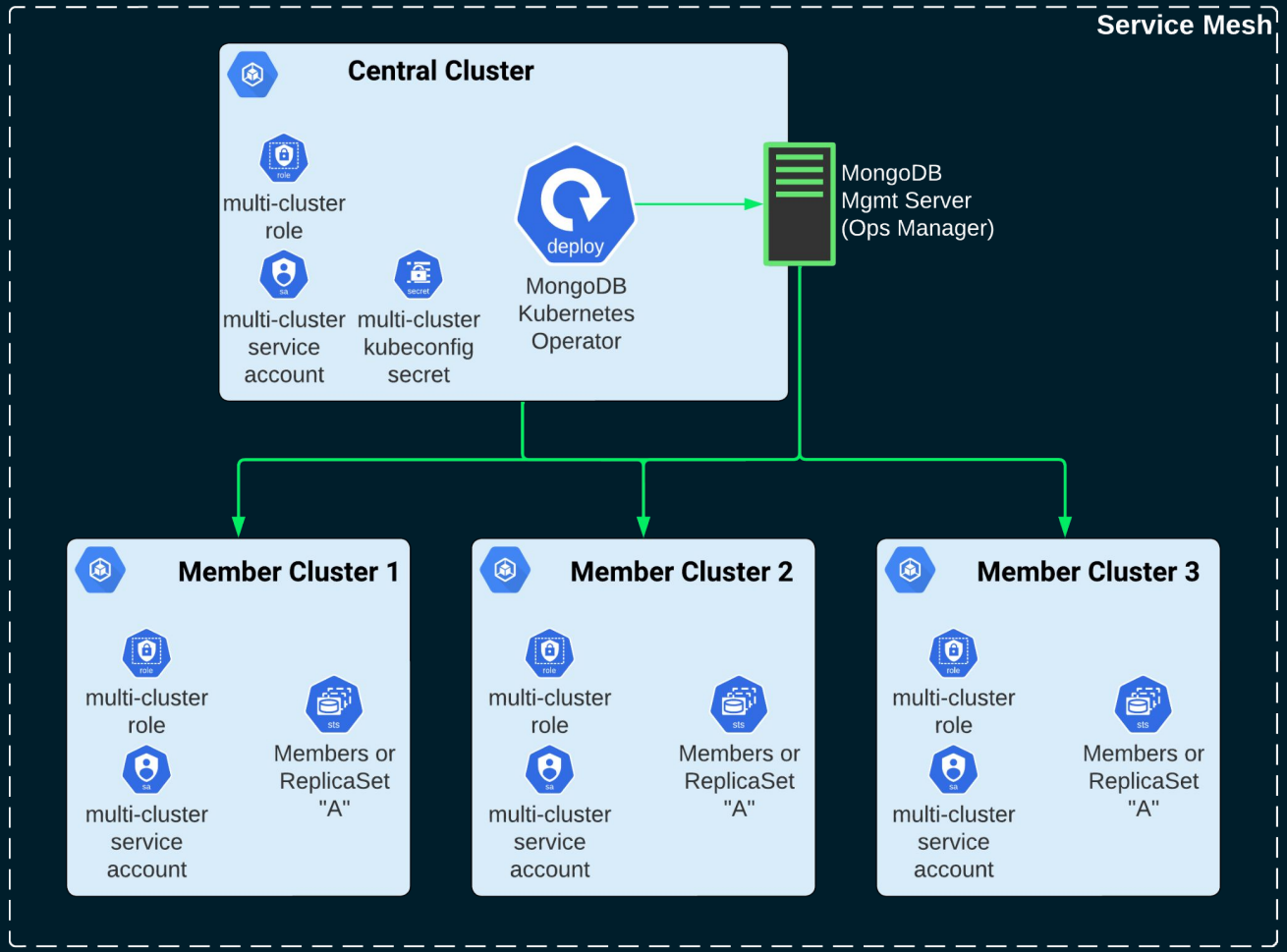




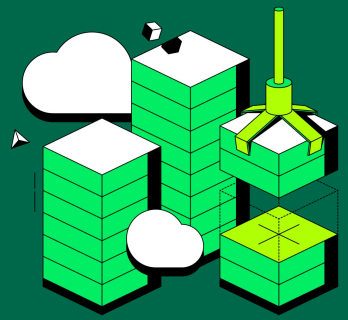




Using a service mesh



Multi-Cluster Controllers and Operators



Controller & Operator

Design Considerations

- Consistent state across clusters
- Resiliency - specifically to cluster failure
- Scalability
- Security

Functional Needs

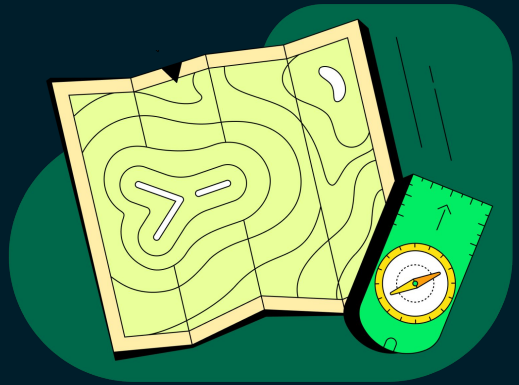
- Cluster Inventory & Workload Distribution
- Failure handling
- Monitoring and Observability

Existing Solutions

Karmada	Open Cluster Management
Considers cluster availability and actual resources	Considers cluster availability and actual resources
Automated or manual cross-cluster failover	Automated or manual cross-cluster failover
Cross-cluster service discovery	Cross-cluster service discovery
Uses Submariner to establish inter-cluster connectivity	Uses Cluster Proxy to establish L4 connectivity, or DNS integration
More declarative	More prescriptive
Optional karmada-agent on each member/spoke clusters	Klusterlet agent on each member/spoke clusters
Can retrofit a single cluster workload to multi-cluster	

MongoDB's Journey

- Workload distribution: Prescriptive → Declarative
- Resource Awareness
- Automated failover → More control
- Operator High Availability



Future of Multi-Cluster

Future of Multi-Cluster

- More powerful & user-friendly multi-cluster management tools
- Increased use of AI and ML to automate tasks
- Better network support for multi-cluster Kubernetes
- Better support for hybrid and multi-cloud deployments

Q&A

Dan Mckean
Senior Product Manager, Kubernetes

George Peter Hantzaras
Director of Engineering, Kubernetes

linktr.ee/kubecon_multi_cluster_patterns