



KubeCon



CloudNativeCon

Europe 2026

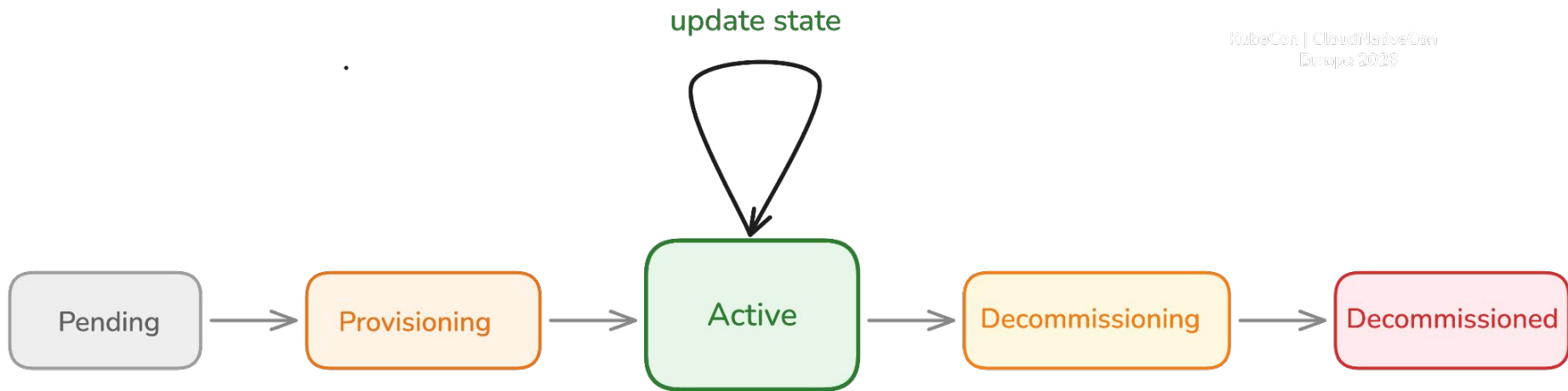
#KubeCon #CloudNativeCon

Scaling the Kubernetes Ecosystem: Uber's Approach to Cluster Lifecycle Management

Silvio Simunic & Vadim Plakhtinskii, Uber



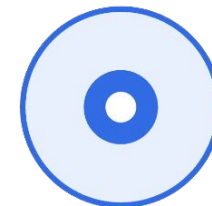
What is cluster lifecycle?





Manual Steps to Create 1 Cluster

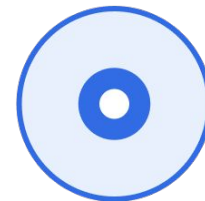
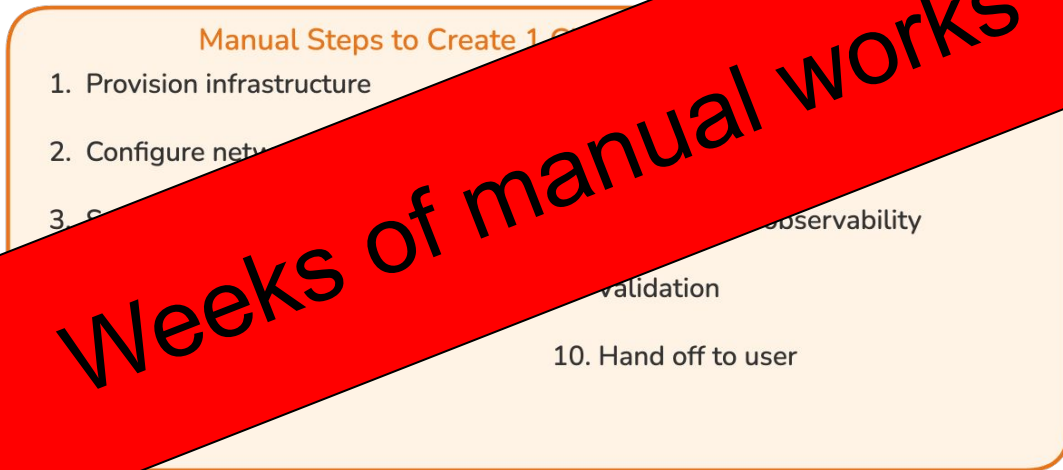
1. Provision infrastructure
2. Configure networking
3. Set up etcd cluster
4. Deploy control plane
5. Custom configs per team
6. Install custom operators
7. Security & RBAC setup
8. Monitoring & observability
9. Validation
10. Hand off to user



1 Cluster



1 Engineer



1 Cluster

Weeks of manual works

Uber's Kubernetes Footprint

- **> 100** of clusters in production
- **15+** zones across regions
- **> 10** cluster types serving different workloads
- **On-prem + cloud** - hybrid infrastructure
- **1000+** engineers who want to customise everything

Layers



KubeCon



CloudNativeCon

Europe 2026

1. **Users** - just want to run their operators
2. **Custom controllers** - manage specific workloads
3. **Configuration** - manages permission, custom objects
4. **Native control plane** - manages k8s-apiserver, k8s-scheduler, etc.
5. **Foundation** - we need more hosts (cloud, on-prem, etc.)

Where We Started

- X** Manual cluster creation - several weeks per cluster
- X** 15+ manual provisioning/decommission steps
- X** New cluster type - require a lot of engineering work
- X** Configuration drift - clusters diverging with no continuous reconciliation
- X** No single source of truth - config scattered across multiple systems

Goal state



KubeCon



CloudNativeCon

Europe 2026

- Flexible **enough (but not too much)** cluster configuration
- Automated cluster provisioning
- Reconciliation of clusters
- Automated cluster decommissioning



KubeCon



CloudNativeCon

Europe 2026

Scaling Configuration for 100+ Clusters



Cluster Dimensions: Type, Topology & Lifecycle

Dimension 1: TYPE

Stateless

Stateful

Batch

Benchmarking

Federator

Dimension 2: TOPOLOGY

Zonal

Regional

Global

Dimension 3: LIFECYCLE

Pending

Provisioning

Active

Decommissioned

Decommissioning

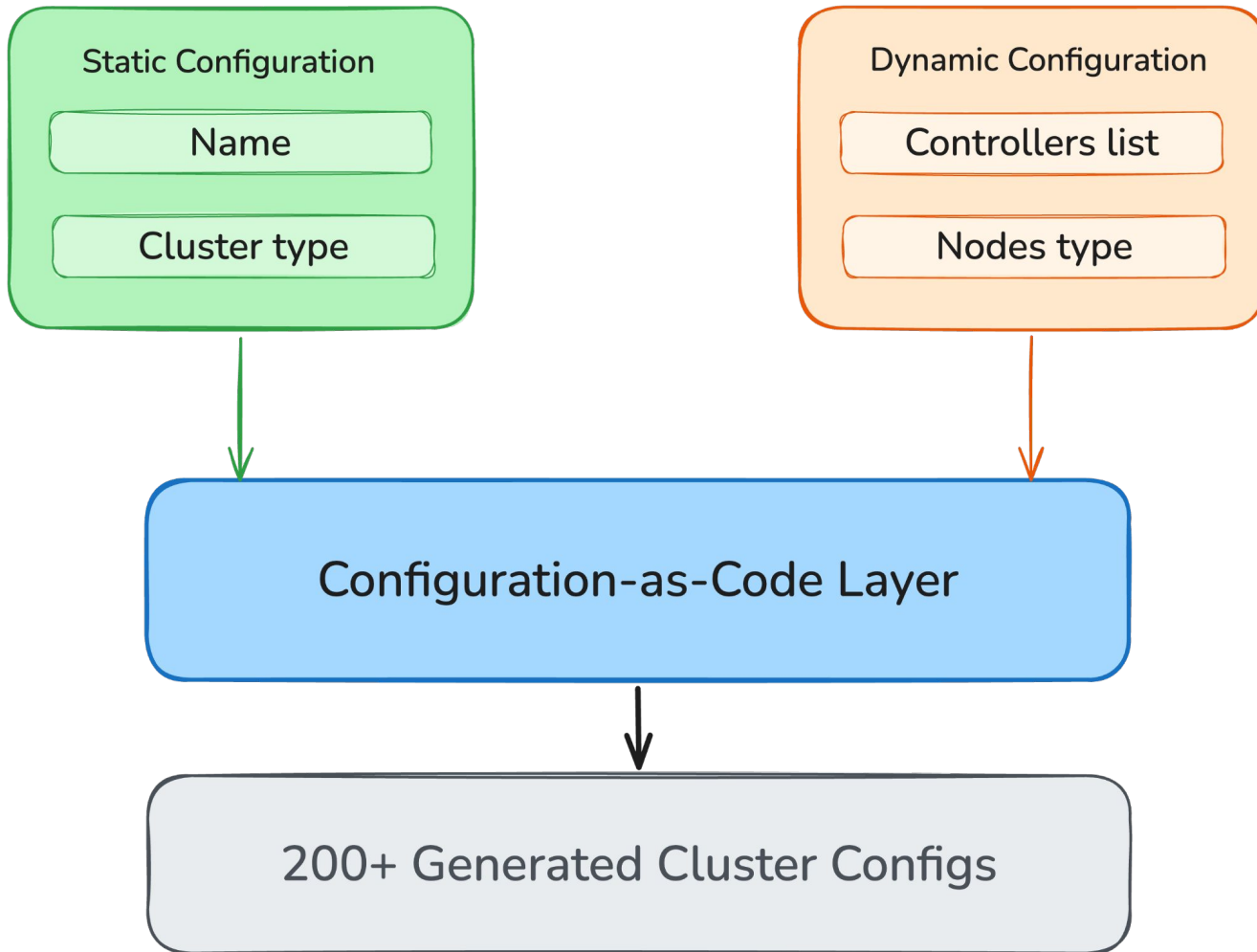


KubeCon



CloudNativeCon

Europe 2026





KubeCon



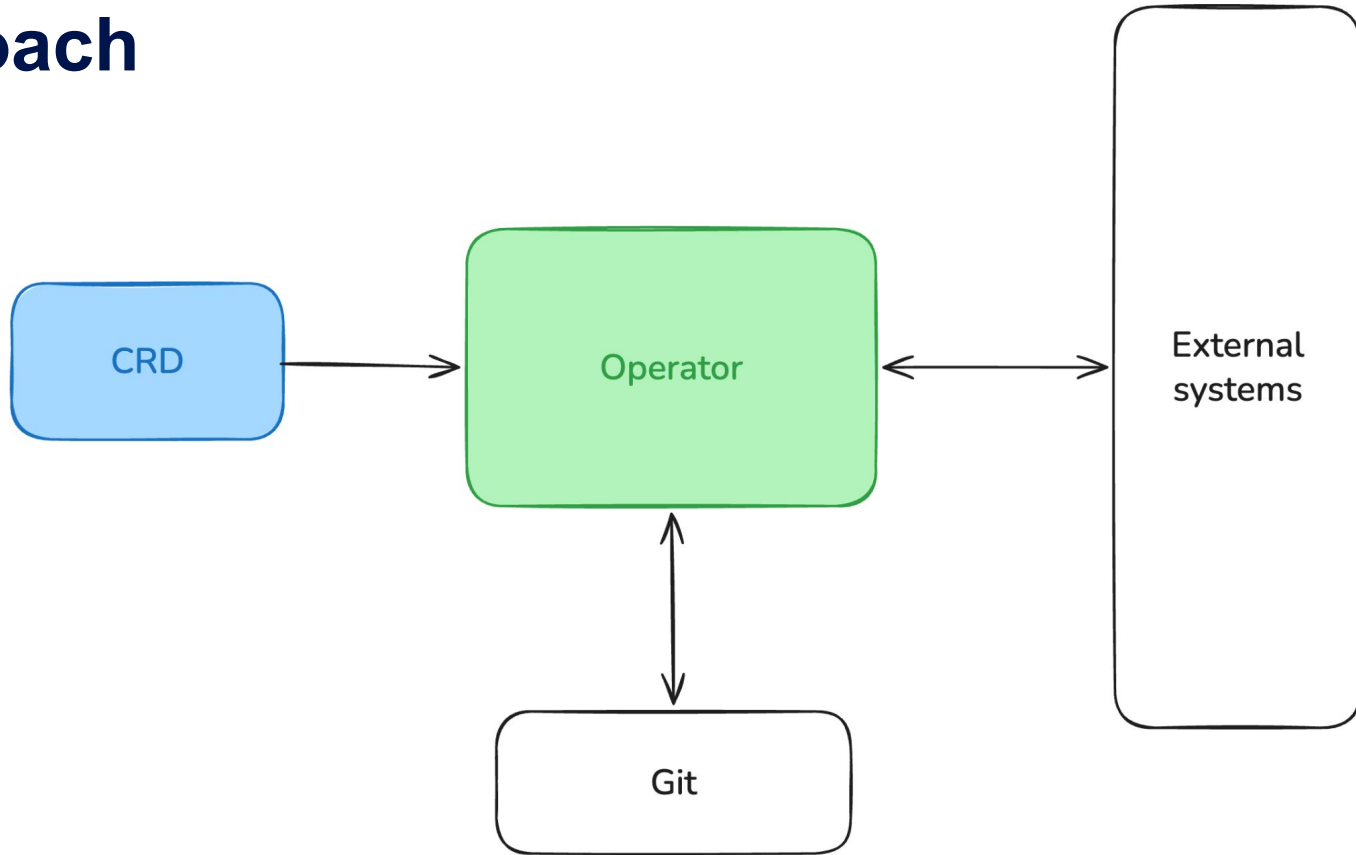
CloudNativeCon

Europe 2026

Automating Cluster Provisioning at Scale

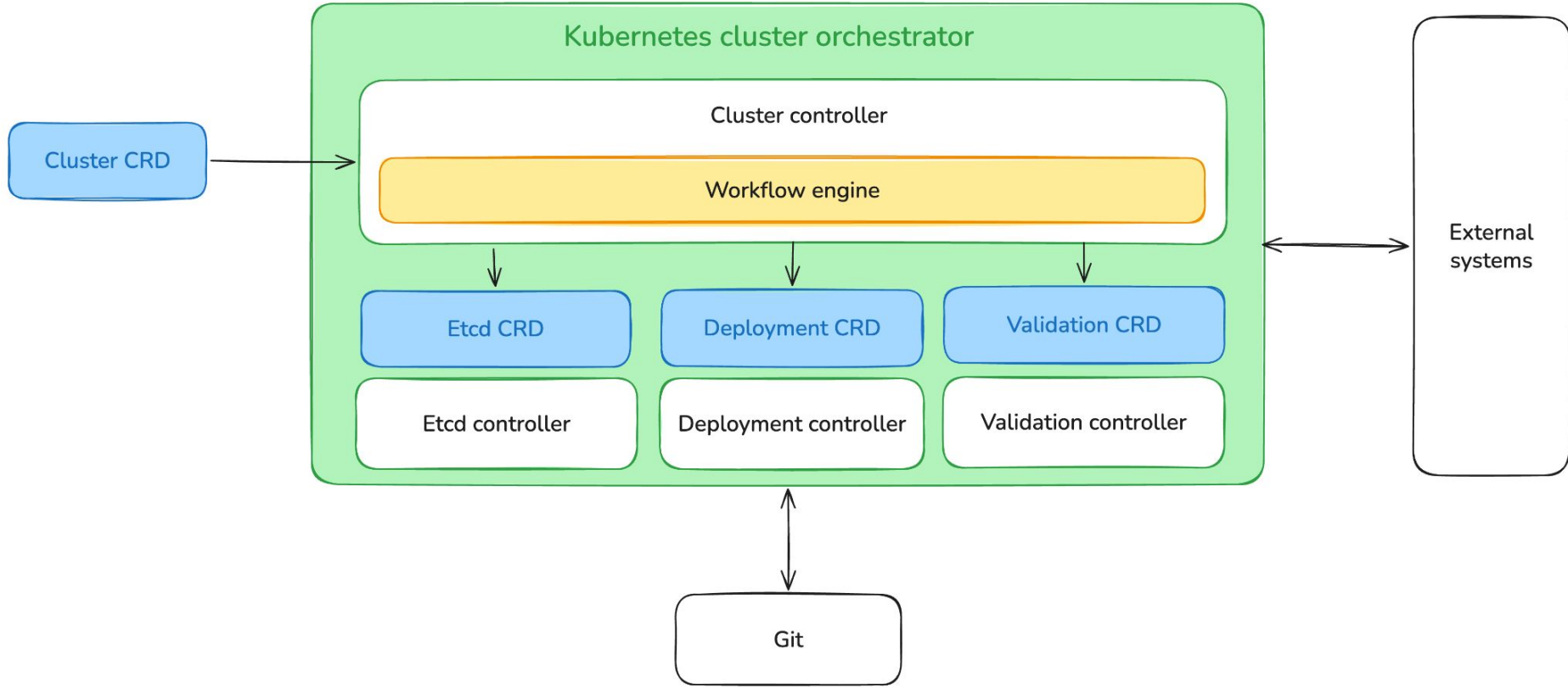


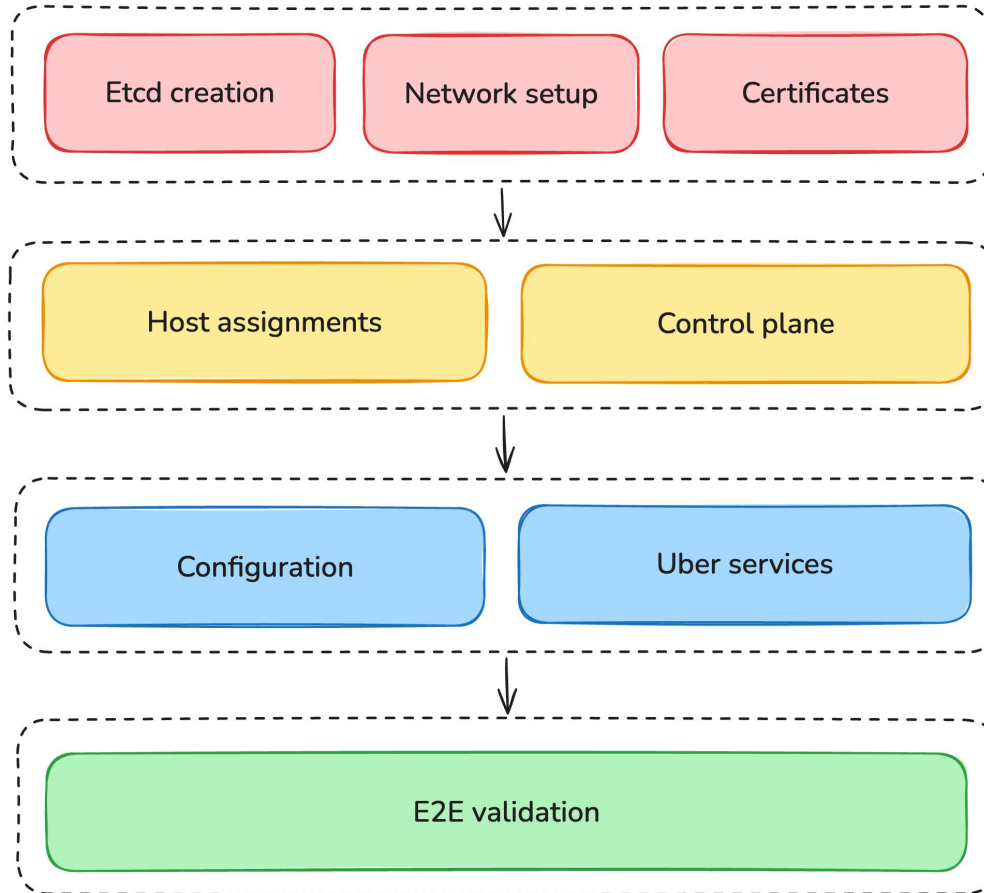
Approach



```
apiVersion: v1beta1
kind: Cluster
metadata:
  name: phx1-stateless
spec:
  region: phx
  zones: [phx1]
  type: stateless
  environment: production
  kubeletSpecs:
    - hostGroup: k8s-kubelet
      serverTypes: serverT1
      count: 5000
status:
  conditions:
    - type: EtcSetup
      status: True
  phase: Provisioning
```







External Dependencies



KubeCon



CloudNativeCon

Europe 2026

- Deployment engines, capacity, certificates, provisioning APIs
- Patterns for resilient integration: idempotent calls, rate limits, retries, cooldowns



KubeCon

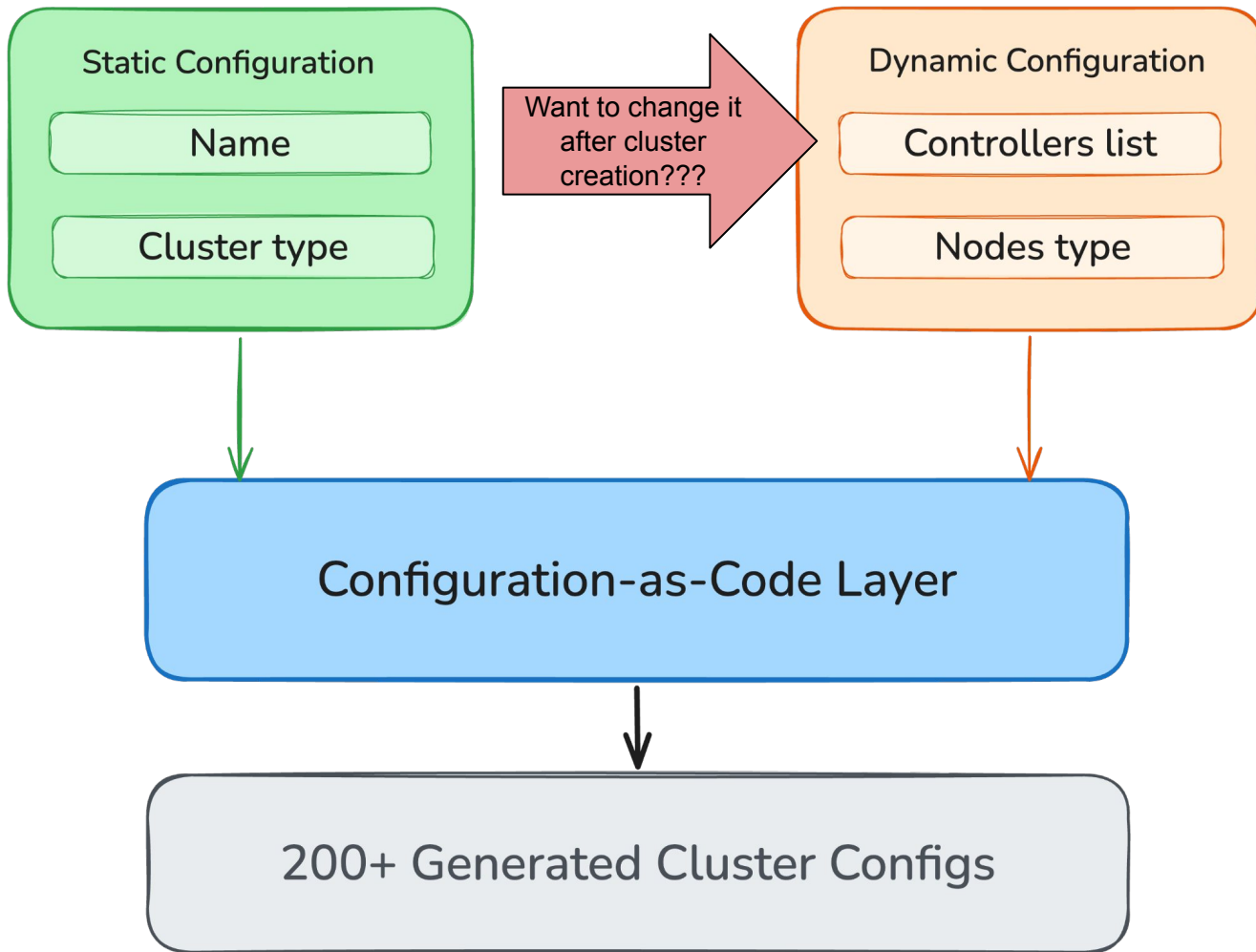


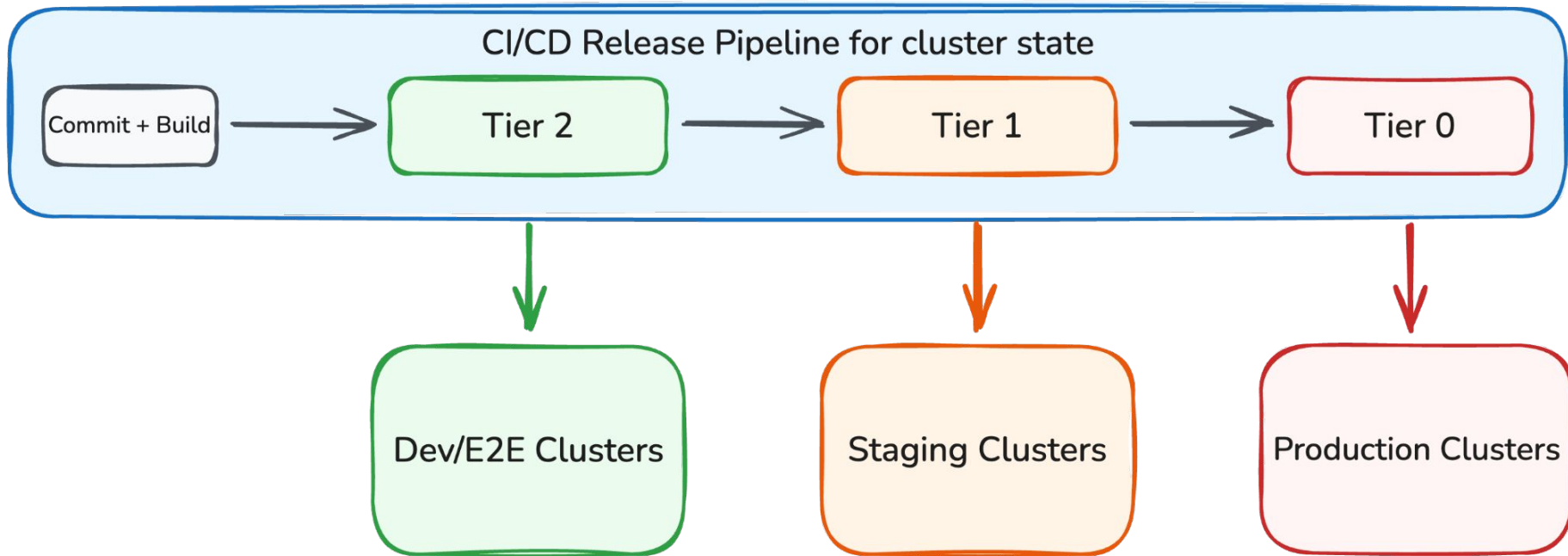
CloudNativeCon

Europe 2026

Safely Changing Configuration for 100+ Clusters in Production









KubeCon



CloudNativeCon

Europe 2026

Safely Decommissioning Clusters



Decommissioning Workflow



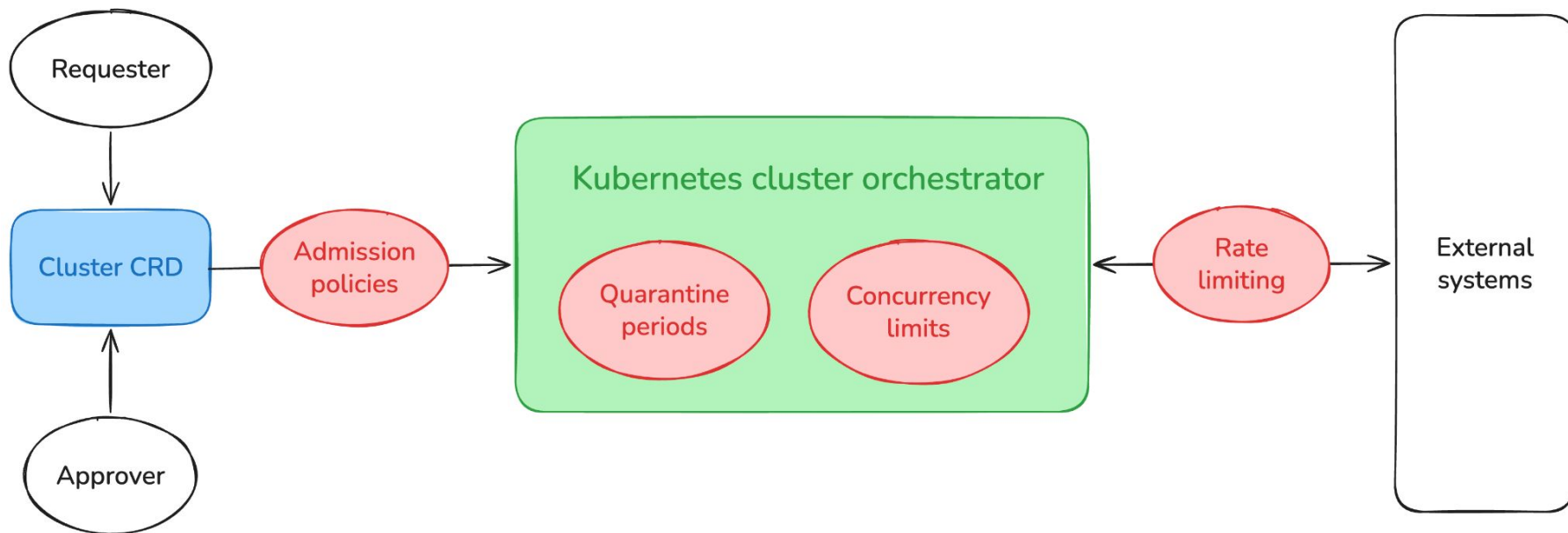
KubeCon



CloudNativeCon

Europe 2026

- Same model as provisioning (controllers, CRDs, steps)
- Finalizers ensure cleanup of child resources
- History in status field for auditing





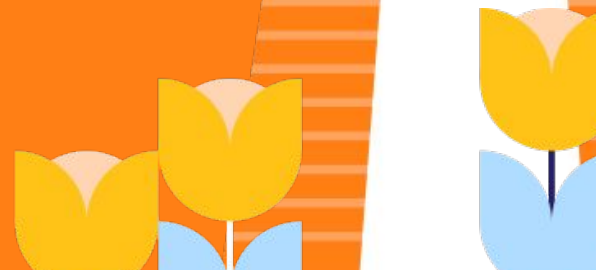
KubeCon



CloudNativeCon

Europe 2026

Summary and Key Takeaways



- Configuration is centralized in a single source of truth
- Provisioning and decommissioning are fully automated
- Cluster setup time is reduced from weeks to hours
- Cluster lifecycle is integrated with the Uber ecosystem

Key Lessons Learned



KubeCon



CloudNativeCon

Europe 2026

- Integration complexity is the main cost
- Keep declarative configuration in Git as final state
- Start with small-scoped automation that can be expanded



KubeCon



CloudNativeCon

Europe 2026

Thank You!

Questions?

