Kubernetes in the Cloud-Native World

Landscape • DevOps Role • Future • Alternatives & Complements

- **Audience:** Advanced / Intermediate (beginner-accessible)
- **Focus:** Theory, architecture, decision frameworks

Agenda

1. **Kubernetes & Cloud-Native Landscape (CNCF projects)**
2. **The Role of Kubernetes in DevOps**
3. **The Future of Kubernetes (WASM, Edge, etc.)**
4. **Alternatives & Complements (Nomad, Docker Swarm, OpenShift)**

1. Kubernetes & Cloud-Native Landscape

Positioning inside the CNCF

Kubernetes (orchestration core): Scheduling, service discovery, scaling, self-healing

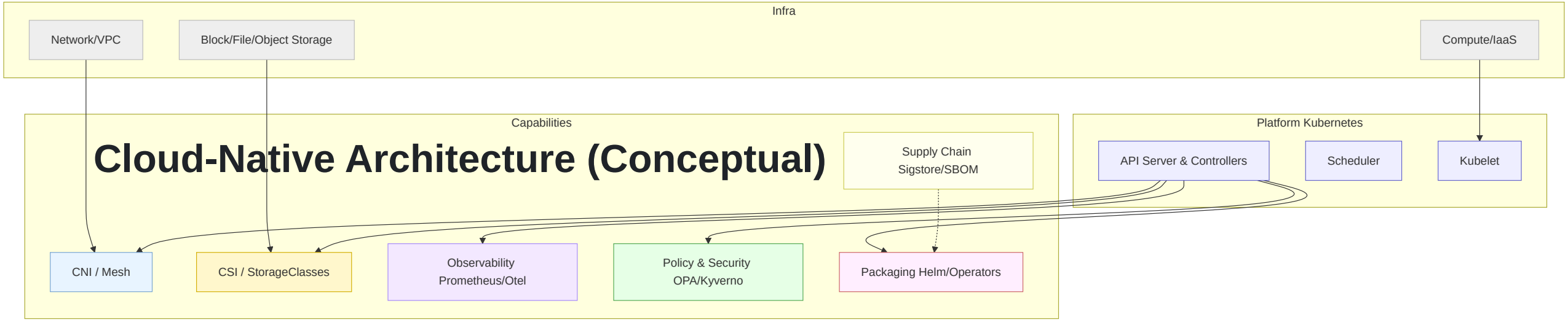
Surrounding pillars:

- **Provisioning:** Cluster lifecycle & IaC (kubeadm, Cluster API, Terraform)
- **Networking:** CNI plugins (Calico, Cilium), Ingress/Gateway API, service mesh (Istio/Linkerd)
- **Storage:** CSI drivers, dynamic provisioning, snapshots

- **Observability:** Prometheus, OpenTelemetry, Loki, Tempo
- **Security:** OPA/Gatekeeper, Kyverno, Sigstore, SPIFFE/SPIRE, Falco
- **App packaging:** Helm, Kustomize, Operators/OLM
- **Supply chain:** SLSA, in-toto, SBOM (SPDX/CycloneDX)

> Kubernetes is the control center, but value emerges from the ecosystem: networking, storage, observability, security, packaging, and supply chain.

Cloud-Native Architecture (Conceptual)



Layering: Infra → Kubernetes control/data plane → cloud-native capabilities

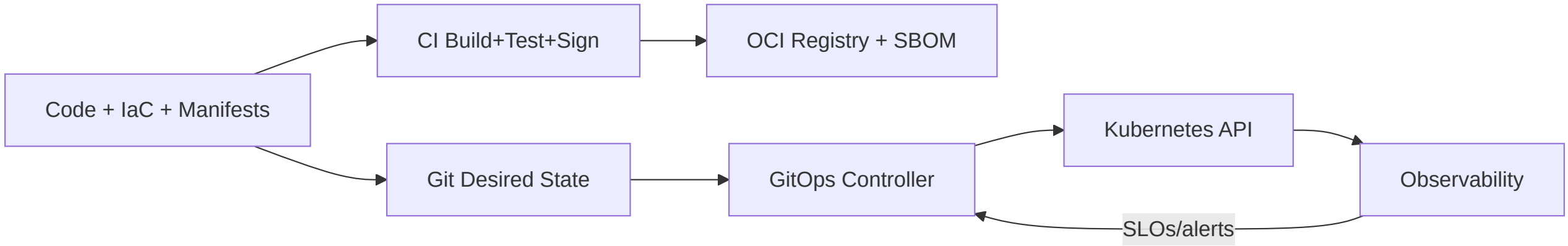
CNCF Maturity Lens

- **Graduated projects:** Production-proven (Kubernetes, Prometheus, Envoy, ...)
- **Incubating/Sandbox:** Emerging capabilities—evaluate carefully for risk & support
- **Selection principle:** Prefer graduated for core controls; incubating for edge cases

2. The Role of Kubernetes in DevOps

From Commit to Cluster (Theory)

- **Git as source of truth:** GitOps (desired state) with controllers (Argo CD/Flux)
- **Pipelines:** CI builds artifacts + attestations; CD reconciles desired → actual
- **Environment parity:** Overlays per env, policy gates at admission
- **Feedback loops:** Metrics, logs, traces and SLO-driven rollouts


CI Build+Test+Sign
OCI Registry + SBOM
Code + IaC + Manifests
Git Desired State
GitOps Controller
Kubernetes API
Observability
SLOs/alerts

Outcome: Faster, safer delivery via declarative reconciliation and policy

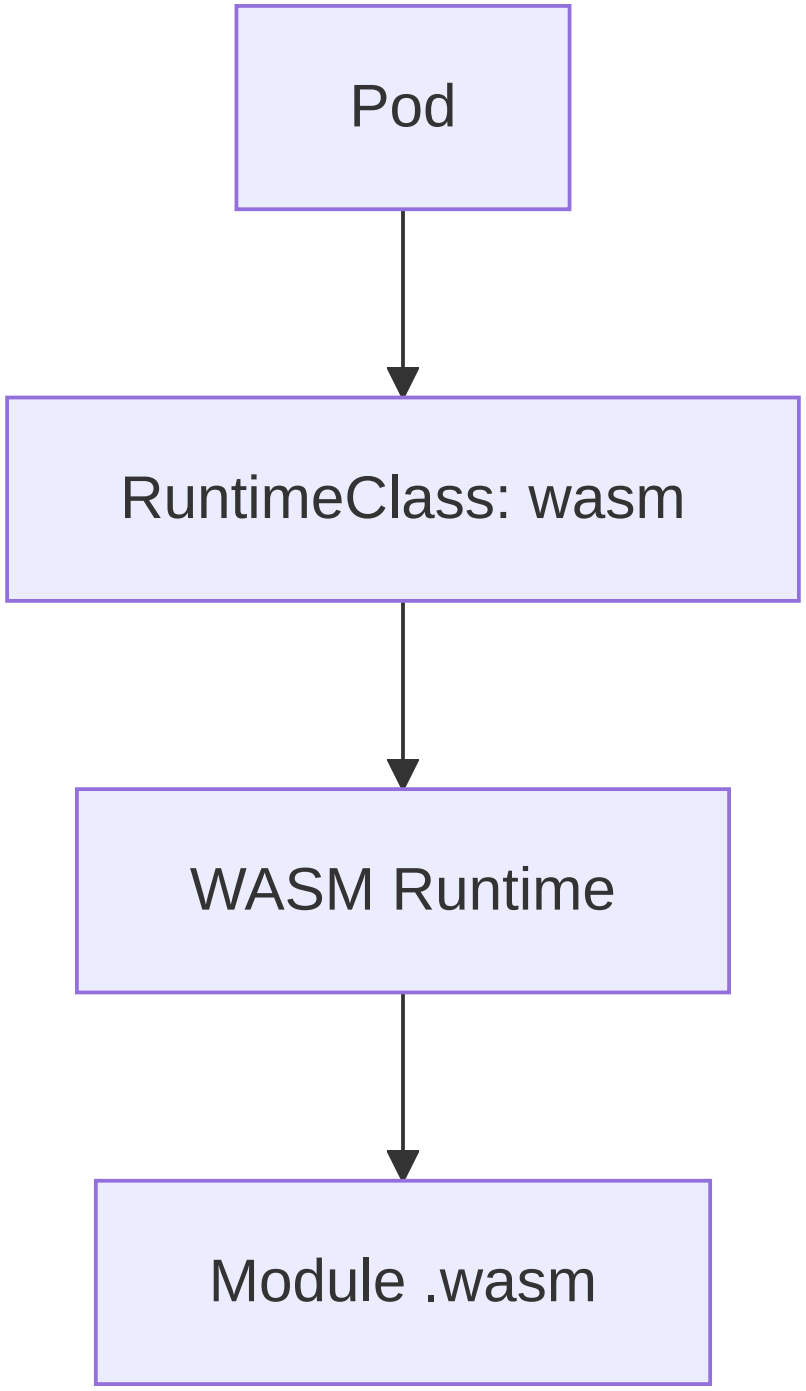
DevOps Guardrails on K8s

- **Security gates:** Signed images, SBOM required, validating admission policies
- **Progressive delivery:** Canary/Blue-Green via Service/Ingress/Service Mesh
- **Release safety:** PDBs, readiness gates, surge/unavailable thresholds
- **Ops observability:** Golden signals (latency, traffic, errors, saturation)

3. The Future of Kubernetes

WASM Workloads (WebAssembly)

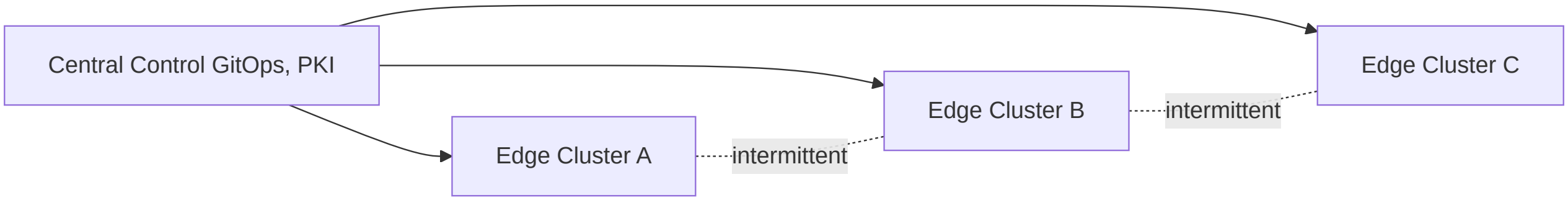
- **Why WASM:** Fast start, small footprint, strong sandboxing, polyglot toolchains
- **K8s Integration:**
  - **Runtimes:** wasmtime, wasmEdge, spin via CRI-shim or runtime-class
  - **Workload types:** Sidecar filters (Proxy-WASM), batch, edge functions
  - **Trade-offs:** Syscall model limits; I/O via host or capability APIs; ecosystem still maturing


Pod
RuntimeClass: wasm
WASM Runtime
Module .wasm

Pattern: Mix containers for heavy I/O with WASM modules for fast, safe logic

Edge & Fleet Kubernetes

- **Drivers:** Low latency, data locality, intermittent connectivity, cost of egress
- **Approaches:** k3s/microk8s at the edge; central control with GitOps & remote ops; partial mesh
- **Challenges:** Topology awareness, constrained nodes, offline upgrades, security of remote sites


Central Control GitOps, PKI
Edge Cluster A
Edge Cluster B
Edge Cluster C
intermittent
intermittent

Data: Prefer stateless at edge; replicate only necessary state; use RWX carefully

Platform as a Product

- **Internal Developer Platforms (IDP):** K8s as substrate + golden paths + templates
- **Abstractions:** Backstage/Portal, higher-level APIs (CRDs), “platform engineering” teams
- **Outcome:** Self-service with curated defaults; pave the road without blocking flexibility

4. Alternatives & Complements

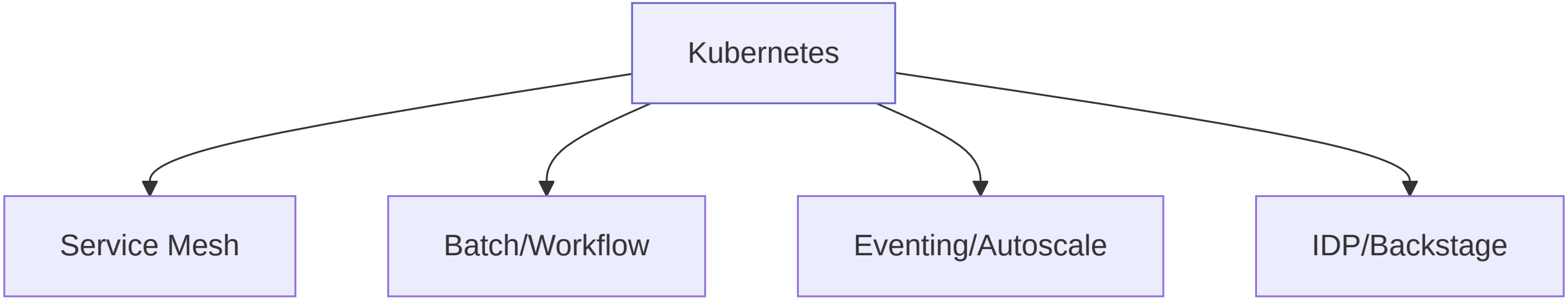
Nomad, Docker Swarm, OpenShift (and more)

Solution	Scope	Strengths	Trade-offs	When it fits
Nomad (HashiCorp)	Scheduler for containers, VMs	Simple, single binary; multi-workload; ties well with Consul/Vault	Smaller ecosystem vs. K8s; fewer built-ins	Heterogeneous workloads, HashiCorp stack shops
Docker Swarm	Container orchestrator	Easy to learn; integrated with Docker	Limited features; declining mindshare	Small teams, simple deployments

Solution	Scope	Strengths	Trade-offs	When it fits
OpenShift	Enterprise K8s distro	Opinionated operators, secure defaults, route/registry, pipelines	Vendor coupling; added complexity	Enterprises needing opinionated K8s + support
Serverless (Knative)	Event-driven autoscale to zero	Simple developer UX, scale-to-zero	Cold start, pattern-fit required	Event/API workloads with bursty traffic

Complementary Patterns with Kubernetes

- **Service Mesh:** Identity, mTLS, traffic policy (Istio/Linkerd)
- **Batch at Scale:** Argo Workflows, Volcano for HPC/ML batch scheduling
- **Data Plane Variants:** DPDK/eBPF acceleration; WASM as in-mesh filters
- **Eventing:** Knative Eventing, CloudEvents; queue-backed autoscaling (KEDA)


Kubernetes
Service Mesh
Batch/Workflow
Eventing/Autoscale
IDP/Backstage

Guideline: Use K8s as a platform kernel, compose capabilities as needs grow

Decision Framework (Summary)

- **Team skills:** Ops maturity for networking, security, observability
- **Workload fit:** Stateless vs. stateful, batch vs. services, edge vs. centralized
- **Ecosystem needs:** Mesh, policy, supply chain, data services
- **Governance:** Compliance, tenancy, cost controls
- **Buy vs. build:** Vanilla K8s, managed services, or opinionated distros

Key Takeaways

- Kubernetes is the core orchestrator in a rich CNCF ecosystem
- In DevOps, K8s enables declarative delivery and strong policy gates
- Future: WASM and Edge expand the runtime & topology space
- Alternatives (Nomad/Swarm) and complements (OpenShift/Knative) fit specific contexts
- Treat K8s as a platform kernel; compose capabilities deliberately

References & Further Reading

- CNCF Landscape & project docs
- GitOps: Argo CD, Flux
- Service Mesh: Istio, Linkerd; Gateway API
- Observability: Prometheus, OpenTelemetry
- Security: Sigstore, SPIFFE/SPIRE, OPA/Kyverno
- Serverless & Eventing: Knative, KEDA