TECHNISCHES WHITEPAPER

OpenKubes Anywhere

Einheitlicher Kubernetes-Betrieb über lokale, Bare-Metal-, Edge- und Cloud-Umgebungen

Version 1.0 · Juni 2026
Kubernauts GmbH · kubernauts.de

Zielgruppe: CTOs, Platform Engineers, Architekten, Investoren

Inhaltsverzeichnis

1. Zusammenfassung

OpenKubes Anywhere stellt ein einheitliches Kubernetes-Betriebsmodell für lokale Entwicklungsumgebungen, Bare-Metal-Rechenzentren, Edge-Standorte und Public Clouds bereit — gesteuert über eine einzige GitOps-basierte Control Plane. Während Unternehmen heute fragmentierte Toolchains für jede Umgebung betreiben, ersetzt OpenKubes Anywhere diese durch eine deklarative, kontinuierlich abgeglichene Plattform, die vollständig auf CNCF-Projekten aufbaut: ArgoCD, Cluster API und Crossplane.

Dieses Whitepaper beschreibt die aktuelle technische Implementierung, Architekturentscheidungen, unterstützte Umgebungen, das Sicherheitsmodell sowie die geplante Entwicklungs-Roadmap bis 2028. Es richtet sich an CTOs, die eine Plattformkonsolidierung evaluieren, Platform Engineers, die Multi-Environment-Kubernetes-Strategien entwickeln, sowie technische Architekten, die Cloud-native Infrastrukturframeworks bewerten.

Fähigkeit	Status	Technologie
GitOps Control Plane	GA	ArgoCD, Flux
Cluster-Lifecycle-Management	GA	Cluster API
Infrastructure as Code	GA	Crossplane
Bare-Metal-Provisionierung	GA	Metal3, iPXE
Edge-Fleet-Management	GA	K3s, ArgoCD
Public Cloud (EKS/AKS/GKE)	GA	Cluster API Providers
AI/MLOps-Plattform	GA	Kubeflow, KServe, Ray
Robotics / Industrial AI	Beta	ROS2, Open-RMF
Multi-Tenancy & Policy Engine	GA	Keycloak, Kyverno
Einheitliche Observability	GA	Prometheus, Grafana, OTel

Tabelle 1: Fähigkeitsmatrix von OpenKubes Anywhere

2. Problemstellung

Die Kubernetes-Einführung in Unternehmen ist ausgereift. Organisationen fragen nicht mehr, *ob* sie Kubernetes einsetzen sollen — sondern *wie* sie es konsistent über zunehmend heterogene Infrastrukturlandschaften betreiben können. Dies erzeugt mehrere sich gegenseitig verstärkende operative Herausforderungen:

Toolchain-Fragmentierung. Jede Umgebung (lokal, Bare Metal, Edge, Cloud) erfordert historisch separate Deployment-Tooling, CI/CD-Pipelines und Lifecycle-Prozesse. Platform Engineers verwalten mehrere Konfigurations-Repositories, Credential-Stores und Monitoring-Stacks.

Divergenz des Sicherheitsmodells. RBAC-Richtlinien, Netzwerkrichtlinien und Compliance-Kontrollen werden unabhängig pro Umgebung definiert. Dies schafft Audit-Lücken, erhöht den Compliance-Aufwand und erzeugt inkonsistente Sicherheitspositionen über die gesamte Flotte.

Cluster Drift. Ohne eine deklarative, kontinuierlich abgeglichene Control Plane divergieren Cluster-Konfigurationen über die Zeit. Manuelle Änderungen, Notfall-Patches und umgebungsspezifische Überschreibungen akkumulieren sich zu einem Zustand, der schwer zu reproduzieren oder zu prüfen ist.

Langsame Entwickler-Feedback-Schleifen. Entwickler benötigen Zugang zu Kubernetes-Umgebungen, die die Produktion genau widerspiegeln. Ohne ein konsistentes Local-to-Cloud-Betriebsmodell ist Umgebungsparität schwer zu erreichen und aufrechtzuerhalten.

Edge-Betriebskomplexität. Edge-Kubernetes-Deployments stehen vor einzigartigen Herausforderungen: intermittierende Konnektivität, Ressourceneinschränkungen, Remote-Fleet-Management und die Notwendigkeit offline-fähiger GitOps-Workflows.

OpenKubes Anywhere adressiert all diese Herausforderungen durch ein einziges, deklaratives, GitOps-natives Platform-Engineering-Framework, das vollständig auf abgeschlossenen und inkubierenden CNCF-Projekten aufbaut.

3. Warum jetzt?

Mehrere konvergierende Marktentwicklungen haben einen dringenden und konkreten Bedarf für eine einheitliche Kubernetes-Betriebsplattform geschaffen. Der Zeitpunkt für OpenKubes Anywhere ist kein Zufall — er spiegelt einen grundlegenden Wandel in den Enterprise-Infrastrukturanforderungen wider:

Broadcom / VMware-Disruption. Die Broadcom-Übernahme von VMware hat eine großangelegte Neubewertung von Enterprise-Virtualisierungsstrategien ausgelöst. Tausende Unternehmen migrieren aktiv von vSphere zu Kubernetes-nativen Alternativen. OpenKubes Anywhere — mit KubeVirt und Bare-Metal-Unterstützung — ist genau für diese Migration konzipiert und bietet einen Drop-in-Ersatz für VM-Workloads ohne Verlust operativer Disziplin.

AI-Infrastrukturbedarf. Generative KI- und LLM-Workloads erfordern GPU-beschleunigte, verteilte Infrastruktur, die Edge-, On-Premises- und Cloud-Umgebungen umspannt. Unternehmen können sich keine separaten Betriebsmodelle für KI- und Nicht-KI-Workloads leisten. OpenKubes AI integriert sich nativ in dieselbe Plattform, auf der auch Produktionsanwendungen laufen.

Edge-Computing-Wachstum. Industrielles IoT, autonome Systeme und Smart Manufacturing treiben das schnelle Wachstum von Edge-Kubernetes-Deployments voran. IDC prognostiziert über 55 Milliarden verbundene Edge-Geräte bis 2027. Das Management dieser Flotten in großem Maßstab erfordert genau das GitOps-basierte, offline-fähige Fleet-Management, das OpenKubes Edge bietet.

Sovereign-Cloud-Anforderungen. Regulatorische Rahmenbedingungen (NIS2, DORA, DSGVO, BSI C5) zwingen europäische und globale Unternehmen, Infrastruktursouveränität sicherzustellen — die Fähigkeit, Workloads auf vollständig kontrollierter Infrastruktur zu betreiben, einschließlich air-gapped und klassifizierter Umgebungen. OpenKubes Anywhere ist von Anfang an für Sovereign-Deployments konzipiert.

Platform-Engineering-Adoption. Die CNCF Platform Engineering Working Group und Gartner heben Platform Engineering als Top-Infrastrukturpriorität für 2025–2027 hervor. Unternehmen konsolidieren von maßgeschneiderten Kubernetes-Toolchains hin zu standardisierten Internal Developer Platforms (IDPs). OpenKubes Anywhere liefert die technische Grundlage für eine Enterprise-grade IDP.

Markttreiber	Auswirkung	OpenKubes-Antwort
VMware/Broadcom-Disruption	Massenhafte VM-zu-K8s-Migration	KubeVirt + Bare-Metal-Support
AI/LLM-Infrastrukturbedarf	GPU-Workloads in allen Umgebungen	OpenKubes AI Stack
Edge-Computing-Wachstum	55 Mrd.+ Edge-Geräte bis 2027	K3s + Offline-GitOps Fleet Mgmt
Sovereign-Cloud-Regularien	NIS2, DORA, BSI C5 Compliance	Air-Gap + RBAC + Crossplane
Platform-Engineering-Adoption	IDP-Konsolidierungstrend	Einheitliche Control Plane + CLI

Tabelle 2: Markttreiber und OpenKubes Anywhere-Antwort

4. Plattformarchitektur

OpenKubes Anywhere ist als geschichtetes Platform-Engineering-Framework strukturiert. Jede Schicht ist unabhängig austauschbar und kombinierbar, sodass Unternehmen die Plattform schrittweise einführen oder Komponenten in bestehende Toolchains integrieren können.

4.1 Control-Plane-Design

Die OpenKubes Control Plane ist ein Management-Kubernetes-Cluster, der alle Plattformkomponenten hostet. Er dient als Single Source of Truth für den Cluster-Fleet-Status, Anwendungs-Deployments und Infrastrukturressourcen. Der Management-Cluster betreibt:

- ArgoCD — Continuous Delivery und GitOps-Reconciliation-Engine
- Cluster API (CAPI) — deklaratives Cluster-Lifecycle-Management
- Crossplane — Infrastrukturressourcen-Provisionierung über Kubernetes-native API
- Keycloak — Identity Provider und OIDC-Broker
- Kyverno — Policy Engine für Governance und Compliance
- Prometheus + Grafana + OpenTelemetry Collector — Observability-Stack

4.2 GitOps-Pipeline

Alle Konfigurationen werden in Git gespeichert. Das Git-Repository ist die Single Source of Truth. ArgoCD gleicht kontinuierlich den Sollzustand (Git) mit dem Istzustand (Cluster) ab. Die Pipeline folgt einem strukturierten Promotion-Modell:

Phase	Beschreibung	Tooling
Source	Infrastruktur- und App-Manifeste in Git	Git (GitHub/GitLab/Gitea)
Validate	Policy-Prüfungen, Schema-Validierung, Security Scanning	Kyverno, Conftest, Trivy
Sync	ArgoCD Application Reconciliation pro Umgebung	ArgoCD, ApplicationSets
Verify	Health Checks, Smoke Tests, Status Gates	ArgoCD Health Checks, Argo Rollouts
Promote	Automatische oder manuelle Promotion über Umgebungen	ArgoCD Image Updater, Argo Workflows

Tabelle 3: GitOps-Pipeline-Phasen

4.3 Cluster-Lifecycle-Management

Cluster API (CAPI) stellt eine deklarative API für Cluster-Provisionierung, Skalierung, Upgrades und Löschung bereit. OpenKubes liefert getestete CAPI-Provider-Konfigurationen für:

- CAPM3 (Metal3) — Bare-Metal-Cluster-Provisionierung via BMC/IPMI/Redfish
- CAPA — Amazon EKS- und EC2-basierte Cluster

- CAPZ — Azure AKS und verwaltete Node Pools
- CAPG — Google GKE-Cluster
- CAPD — lokale Docker-basierte Cluster für Entwicklung und CI

ClusterClass-Templates bieten meinungsstarke, wiederverwendbare Cluster-Topologien, die organisatorische Standards für Node-Sizing, Netzwerk, Sicherheit und Add-ons kodieren.

4.4 Infrastructure as Code mit Crossplane

Crossplane erweitert Kubernetes mit Composite Resource Definitions (XRDs) und Compositions, die Cloud-Provider-APIs in organisationsspezifische Infrastruktur-Primitive abstrahieren. Platform Teams definieren wiederverwendbare Abstraktionen (z.B. 'DatabaseCluster', 'ObjectStoreBucket'), die Entwickler über Standard-Kubernetes-Manifeste nutzen — ohne cloud-provider-spezifisches Wissen.

5. Umgebungsabdeckung

5.1 Lokale Entwicklung — Derselbe GitOps-Workflow vom Laptop bis zur Produktion

k3s · Multipass · KubeVirt · bonsai · GitOps

OpenKubes Local ist eine der markantesten Fähigkeiten der Plattform. Entwickler betreiben eine vollständig GitOps-verwaltete Kubernetes-Umgebung auf ihrer Workstation — keine Simulation, keine abgespeckte Annäherung, sondern dieselbe Cluster-Topologie, dieselben ArgoCD ApplicationSets und dieselben Kyverno-Policies wie in der Produktion. Wenn ein Entwickler nach Git pusht, läuft dieselbe Reconciliation-Schleife wie im Rechenzentrum.

Aufgebaut auf k3s (leichtgewichtiges Kubernetes), Multipass (VM-Isolation auf macOS, Linux und Windows) und KubeVirt (VM-Workloads neben Container-Workloads) ermöglicht OpenKubes Local Full-Stack-Entwicklung einschließlich legacy-VM-basierter Anwendungen. bonsai bietet opinioniertes lokales Cluster-Bootstrapping mit vorkonfiguriertem GitOps, Observability und Security-Tooling — eine produktionsreife Entwicklungsumgebung in Minuten. Das ist ein einzigartig starkes Wertversprechen: *Was lokal läuft, läuft überall.*

5.2 Bare Metal & Sovereign Cloud

Cluster API · Metal3 · ArgoCD · iPXE

OpenKubes Bare Metal bietet deklaratives Cluster-Lifecycle-Management für On-Premises-Rechenzentren und souveräne Infrastruktur. Metal3 (Cluster API Provider für Metal3) verwaltet Bare-Metal-Host-Erkennung, -Inspektion, -Provisionierung und -Deprovisionierung über BMC-Schnittstellen (IPMI, Redfish).

Nodes werden über iPXE-Netzwerkboot provisioniert, wodurch die manuelle OS-Installation entfällt. Ironic verwaltet das Hardware-Inventar. Nach der Provisionierung werden Cluster identisch zu Cloud-Clustern über CAPI und ArgoCD verwaltet. Dieses Modell ist vollständig air-gap-fähig für souveräne und klassifizierte Umgebungen.

5.3 Edge Computing

K3s · GitOps · Offline · Fleet Management

OpenKubes Edge adressiert die operativen Herausforderungen des Kubernetes-Betriebs in Fabriken, Lagerhallen, Einzelhandelsstandorten und verteilten Industriestandorten. K3s bietet eine produktionsreife, ressourceneffiziente Kubernetes-Distribution für ressourcenbeschränkte Hardware.

Der ArgoCD-Agent-Modus ermöglicht Pull-basierte GitOps-Reconciliation, die intermittierende WAN-Konnektivität toleriert. Fleet-Management über ApplicationSets ermöglicht konsistentes, skalierbares Deployment über tausende Edge-Standorte von einem einzigen Management-Cluster aus.

5.4 Public Cloud

EKS · AKS · GKE · Crossplane

OpenKubes Cloud bietet einheitliches Lifecycle-Management für verwaltete Kubernetes-Services über AWS, Azure und Google Cloud — ohne Vendor Lock-in. Cluster API Provider (CAPA, CAPZ, CAPG) verwalten Cluster-Erstellung, Node-Pool-Skalierung und Kubernetes-Version-Upgrades über dieselbe deklarative API wie für Bare Metal.

Crossplane Compositions abstrahieren cloud-provider-verwaltete Services (RDS, Azure Database, Cloud SQL) in portable Resource Definitions. Anwendungen nutzen Infrastrukturressourcen über Standard-Kubernetes-CRDs unabhängig vom zugrundeliegenden Cloud-Provider.

6. Sicherheitsframework

OpenKubes implementiert ein Zero-Trust-Sicherheitsmodell über den gesamten Plattform-Stack. Sicherheit ist kein Add-on — sie ist in die Plattformarchitektur auf jeder Ebene integriert.

Schicht	Komponente	Fähigkeit
Identität	Keycloak	OIDC/OAuth2-Identitäts-Brokering, SSO, MFA, LDAP-Integration
Zugriffskontrolle	Kubernetes RBAC + Kyverno	Feingranulares RBAC, Policy-as-Code, Admission Control
Netzwerk	Cilium / Calico	Netzwerkrichtlinien, mTLS, Zero-Trust East-West-Traffic
Supply Chain	Cosign + Trivy	Image-Signing, SBOM-Generierung, Vulnerability-Scanning
Secrets	Vault / ESO	Secret-Management, dynamische Secrets, External Secret Operator
Audit	Falco + OpenSearch	Runtime-Bedrohungserkennung, Audit-Log-Aggregation, Alerting
Compliance	Kyverno + OPA	CIS-Benchmarks, PCI-DSS, SOC2, Policy-Enforcement

Tabelle 4: Sicherheitsframework-Komponenten

7. Observability-Stack

Vollständige Observability ist ab Werk enthalten und wird konsistent über alle Umgebungen bereitgestellt. Der Observability-Stack basiert auf offenen Standards (OpenTelemetry, Prometheus Remote Write, OTLP) und gewährleistet so Portabilität über Umgebungen und Cloud-Provider.

Metriken — Prometheus. Cluster-, Node-, Workload- und benutzerdefinierte Anwendungsmetriken. Thanos für Langzeitspeicherung und clusterübergreifende Abfragen.

Dashboards — Grafana. Vorgefertigte Dashboards für Cluster-Health, Workload-Performance, GitOps-Sync-Status und Sicherheitsposture.

Logs — OpenSearch + Fluentd. Zentralisierte Log-Aggregation mit Volltextsuche, Audit-Log-Retention und compliance-konformes Log-Management.

Traces — Tempo + OpenTelemetry. Distributed Tracing für Anwendungs- und Infrastrukturkomponenten. OTLP-native Sammlung und Speicherung.

Alerting — Alertmanager + PagerDuty. Multi-Channel-Alerting mit Routing, Silencing und Eskalationsrichtlinien.

KI-Einblicke — OpenKubes AI Ops (Beta). Anomalie-Erkennung und prädiktives Alerting mit ML-Modellen, die auf Cluster-Telemetrie trainiert wurden.

8. AI & MLOps-Integration

OpenKubes AI stellt einen vollständig integrierten MLOps- und AI-Platform-Engineering-Stack bereit, der über Edge-, Rechenzentrum- und Cloud-Umgebungen durch dieselben GitOps-Workflows wie alle anderen Plattformkomponenten eingesetzt werden kann.

Komponente	Technologie	Zweck
ML-Pipeline-Orchestrierung	Kubeflow Pipelines	Wiederverwendbare, portable ML-Pipelines mit Versionskontro
Experiment-Tracking	MLflow	Experiment-Metadaten, Model Registry, Artifact-Speicherung
Verteiltes Training	Ray / Kubeflow Training	Großskaliges Modelltraining über GPU-Node-Pools
Model Serving (Standard)	KServe	Serverless Model Inference mit Autoscaling
Model Serving (High-Perf)	Triton Inference Server	NVIDIA GPU-optimiertes Multi-Modell-Serving
LLM-Inferenz	vLLM / Ollama	Effiziente LLM-Inferenz für Open-Weight-Modelle
Vektordatenbank	Qdrant / Milvus	Embedding-Speicherung und semantische Suche
Feature Store	Feast	Feature Engineering und Serving für ML-Pipelines
GPU-Scheduling	NVIDIA GPU Operator	GPU-Ressourcenmanagement über heterogene Nodes

Tabelle 5: AI/MLOps-Stack-Komponenten

9. Robotics & Industrial AI

OpenKubes Robotics überträgt dasselbe Betriebsmodell, das für Cloud-native Anwendungen genutzt wird, auf das Management von Flotten autonomer Industriesysteme. Eine Fabrikhalle wird zur weiteren Umgebung in der GitOps-Pipeline — provisioniert, abgesichert, überwacht und lifecycle-gesteuert über dieselbe Control Plane wie jeder Cloud-Cluster. Für CTOs in Fertigung, Logistik und industrieller Automatisierung bedeutet dies: Ein Platform-Team betreibt sowohl IT- als auch OT-Workloads ohne separate Toolchains oder Kompetenzsilos.

ROS2. Robot Operating System 2 — containerisierte ROS2-Nodes, verwaltet über Kubernetes Deployments und DaemonSets. Unterstützt Multi-Roboter-Koordination und Lifecycle-Management.

Open-RMF. Open Robotics Middleware Framework — Fleet-Management für heterogene Roboterflotten. Auf Kubernetes für Hochverfügbarkeit und Skalierbarkeit deployed.

DDS-Netzwerk. Data Distribution Service für Echtzeit-, deterministisches Publish-Subscribe-Kommunikation zwischen ROS2-Nodes über Kubernetes-Pods.

Multus CNI. Multi-homed Networking, das Pods die gleichzeitige Verbindung zu mehreren Netzwerken ermöglicht — erforderlich für industrielle Feldbusse, OT-Netzwerk- und IT-Netzwerk-Isolation.

SR-IOV. Single Root I/O Virtualization für hardware-beschleunigtes, latenzarmes Networking. Ermöglicht Near-Bare-Metal-Netzwerk-Performance für zeitkritische Workloads.

GPU-Scheduling. NVIDIA GPU Operator und Device Plugins für das Scheduling von KI-Inferenz- und Computer-Vision-Workloads auf GPU-bestückten industriellen Edge-Nodes.

10. Entwicklungs-Roadmap

Die OpenKubes Anywhere Roadmap ist in vier Horizonte unterteilt, die den aktuellen allgemeinen Verfügbarkeitsstatus, kurzfristige Lieferung, mittelfristige Entwicklung und die langfristige Plattformvision widerspiegeln.

10.1 Aktueller Stand — Allgemein Verfügbar ■ GA

- ✓ GitOps-gesteuerter Cluster-Lifecycle über ArgoCD + Cluster API
- ✓ Bare-Metal-Provisionierung über Metal3/Ironic mit iPXE-Boot
- ✓ Edge-Kubernetes-Fleet-Management mit Offline-GitOps-Unterstützung
- ✓ Public-Cloud-Unterstützung: EKS, AKS, GKE über CAPI-Provider
- ✓ Infrastructure as Code über Crossplane mit AWS-, Azure-, GCP-Providern
- ✓ Zero-Trust-Sicherheit: Keycloak, Kyverno, Cilium, Falco
- ✓ Vollständige Observability: Prometheus, Grafana, Tempo, OpenSearch, OTel
- ✓ AI/MLOps-Stack: Kubeflow, MLflow, KServe, Triton, Ray
- ✓ Lokale Entwicklung: k3s + Multipass + KubeVirt

10.2 Kurzfristig — H2 2026 ■ Geplant

OpenKubes UI (Alpha). Einheitliches Web-Dashboard für Cluster-Fleet-Management, GitOps-Sync-Status und Observability über alle Umgebungen. Aufgebaut auf React mit Kubernetes-nativem API-Backend.

Cluster API v1.10-Unterstützung. Übernahme von CAPI v1.10-Features einschließlich ClusterClass-Topology-Mutation-Hooks und verbesserter MachineHealthCheck-Semantik.

AI Ops Integration. Integration von Anomalie-Erkennungsmodellen in den Observability-Stack für prädiktives Alerting bei Cluster-Health-Metriken und Workload-Performance-Degradation.

vCluster Multi-Tenancy. Virtual-Cluster-Unterstützung über vCluster für Self-Service-Namespace-Isolation mit voller Kubernetes-API-Kompatibilität.

SOPS + External Secrets Operator GA. Einheitliches Secret-Management über alle Umgebungen mit SOPS-verschlüsselten Git-Secrets und ESO für Cloud-Provider-Secret-Store-Integration.

OpenKubes CLI (okctl). Einheitliche CLI für Cluster-Bootstrapping, Umgebungs-Onboarding, GitOps-Status-Inspektion und Plattform-Upgrade-Management.

10.3 Mittelfristig — 2027 ■ Experimentell

Autonome Cluster-Remediation. ML-gesteuertes Remediation-Engine, das Cluster-Health-Anomalien erkennt und automatisierte korrigierende GitOps-Workflows ohne menschliche Intervention ausführt.

Multi-Cluster Service Mesh. Istio/Cilium-basiertes Multi-Cluster-Service-Mesh für transparente, mTLS-verschlüsselte Service-zu-Service-Kommunikation über alle Umgebungstypen.

OpenKubes Marketplace. Self-Service-Katalog von vorvalidierten Plattform-Add-ons, Anwendungsvorlagen und Infrastruktur-Compositions. GitOps-natives One-Click-Deployment.

Cost-Management-Integration. Kubecost-Integration mit umgebungsübergreifender Kostenzuordnung, Chargeback-Reporting und Rightsizing-Empfehlungen.

Confidential Computing. Intel TDX / AMD SEV-Integration für vertrauliche VM-Workloads auf Bare-Metal- und Cloud-Umgebungen.

OpenKubes Robotics GA. Vollständige allgemeine Verfügbarkeit der Robotics-Plattform mit zertifizierten ROS2-Distributionen und Industrial-AI-Workload-Templates.

10.4 Langfristig — 2028+ ■ Vision

Die langfristige Plattformvision konzentriert sich auf autonome, selbstheilende Infrastruktur, die den operativen Aufwand minimiert und gleichzeitig Entwicklerproduktivität und Plattformzuverlässigkeit maximiert:

- Intent-driven Infrastructure — Geschäftsergebnisse deklarieren, Plattform löst Infrastruktur-Topologie
- KI-natives Cluster-Scheduling — Workload-Placement-Optimierung mit ML-Modellen
- Umgebungsübergreifende Workload-Mobilität — Live-Migration zustandsbehafteter Workloads
- Quantensichere Kryptographie — Post-Quantum-TLS und Key-Management über die Plattform
- Carbon-aware Scheduling — Workload-Scheduling optimiert für Verfügbarkeit erneuerbarer Energie
- OpenKubes SaaS-Angebot — vollständig verwaltete Control Plane mit Bring-Your-Own-Infrastructure-Modell
- OpenKubes IMP (Forschung) — Infrastructure Memory Platform: Git-natives Infrastrukturgedächtnis und Recovery-Framework für autonomen Plattformbetrieb.

11. Referenzarchitektur

Das nachfolgende Diagramm veranschaulicht die OpenKubes Anywhere Plattformschichten — vom Git-Repository als Single Source of Truth an der Spitze, über die einheitliche Management-Control-Plane, bis zu den vier Zielumgebungstypen. Alle Umgebungen werden kontinuierlich über dieselbe GitOps-Pipeline abgeglichen.

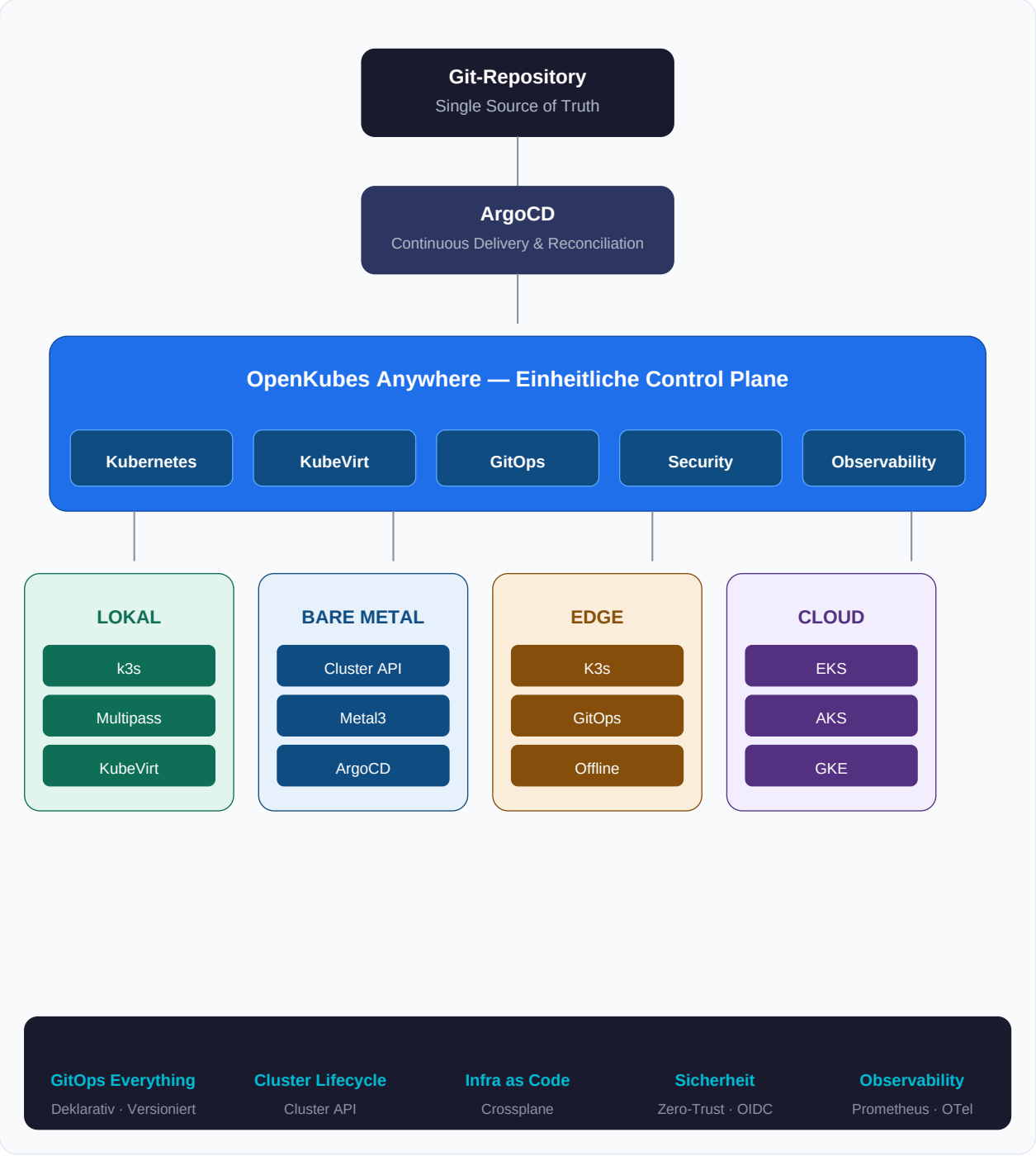

Abbildung 1: OpenKubes Anywhere Plattformarchitektur — vom Git-Repository zum Workload-Cluster über alle Umgebungen

Die nachfolgende Tabelle ordnet jede Architekturschicht ihrem Deployment-Modell zu:

Schicht	Komponente	Deployment-Modell
Git Source of Truth	GitHub / GitLab / Gitea	SaaS oder Self-hosted, HA-Konfiguration
Management-Cluster	Kubernetes (RKE2 / K3s)	3-Node HA Control Plane, Bare Metal oder Cloud-VM
GitOps-Engine	ArgoCD	HA-Deployment auf Management-Cluster, SSO über Keycloak
Cluster-Lifecycle	Cluster API + Provider	Auf Management-Cluster, Provider je Umgebungstyp
Infra-Provisionierung	Crossplane + Provider	Auf Management-Cluster, XRDs je Organisation
Identität	Keycloak	HA-Deployment, externer PostgreSQL-Backend
Observability	Prometheus + Thanos	Föderation über Workload-Cluster, Langzeitspeicher in O
Secrets	Vault + ESO	HA-Vault-Cluster, ESO auf Workload-Clustern
Workload-Cluster	CAPI-verwaltete Cluster	Je Umgebung, bootstrapped über ArgoCD ApplicationSets

Tabelle 6: Referenzarchitektur-Komponenten-Zuordnung

Erwartete Geschäftsergebnisse

Die folgenden Ergebnisse spiegeln den messbaren Geschäftswert der Einführung eines einheitlichen Kubernetes-Betriebsmodells wider:

Betriebsmodell	Geschäftsergebnis	Auswirkung
Eine GitOps-Pipeline	Reduzierte Betriebskomplexitä	Weniger Tools, weniger Kontextwechsel, geringerer Schulungsaufw
Ein Sicherheitsmodell	Geringerer Compliance-Aufwand	Einheitlicher Audit-Scope, konsistente Richtlinien in allen Umgebur
Ein Observability-Stack	Schnellere Incident-Reaktion	Einheitliche Dashboards, korrelierte Alerts, keine Datensilos
Ein Lifecycle-Modell	Schnellere Plattformbereitstellung	Standardisierte Cluster-Provisionierung von Tagen auf Minuten
GitOps-natives IaC	Reduzierter Konfigurationsdrift	Deklarativer Zustand eliminiert manuelle Änderungsakkumula
Sovereign Deployment	Regulatorische Compliance	NIS2, DORA, BSI C5, DSGVO ohne Cloud-Provider-Abhängi
Open-Source-Basis	Kein Vendor Lock-in	Vollständige Portabilität über Cloud-Provider und

Tabelle 7: Erwartete Geschäftsergebnisse durch OpenKubes Anywhere

12. Warum OpenKubes Anywhere?

Enterprise-Kubernetes-Plattformen sind keine neue Kategorie. OpenShift, Rancher, Anthos und Azure Arc sind alle ausgereifte, gut unterstützte Produkte. Die Frage, die ein CTO, Architekt oder Investor stellen wird, ist nicht *was* OpenKubes Anywhere tut — sondern *warum OpenKubes Anywhere statt der Alternativen.* Die Antwort liegt in vier grundlegenden Designentscheidungen, die es von jeder kommerziellen Plattform auf dem Markt unterscheiden.

GitOps-first durch Design, nicht durch Integration

Die meisten Plattformen bieten GitOps als Add-on — eine optionale Integrationsschicht über einer Control Plane, die für imperativen Betrieb konzipiert wurde. OpenKubes Anywhere ist von Grund auf auf das GitOps-Betriebsmodell aufgebaut. Git ist kein Deployment-Trigger; es ist das System of Record. Jeder Cluster, jede Infrastrukturressource, jede Sicherheitsrichtlinie und jede Anwendung wird deklarativ in Git definiert und kontinuierlich über ArgoCD abgeglichen.

CNCF-nativ, nicht CNCF-kompatibel

Kommerzielle Plattformen hüllen CNCF-Projekte typischerweise in proprietäre Abstraktionen ein, die einen weichen Lock-in erzeugen: benutzerdefinierte Ressourcen-Schemas, proprietäre APIs und anbieterspezifisches Tooling. OpenKubes Anywhere verwendet CNCF-Projekte direkt — ArgoCD, Cluster API, Crossplane, Prometheus, OpenTelemetry — ohne proprietäre Wrapper. Platform Teams arbeiten mit denselben Tools und APIs wie in einer selbst gebauten Plattform, aber vorintegriert, vorgetestet und produktionsvalidiert.

Sovereign und Air-Gap-ready von Anfang an

Cloud-gehostete Control Planes (Anthos, Azure Arc) führen eine Abhängigkeit vom Anbieter für das Cluster-Management ein. Für Unternehmen, die unter NIS2, DORA, BSI C5 oder klassifizierten Anforderungen operieren, ist das ein Ausschlusskriterium. OpenKubes Anywhere läuft vollständig auf Infrastruktur, die der Kunde kontrolliert — einschließlich vollständig air-gapped-Umgebungen ohne ausgehende Konnektivität. Dies ist nicht ein Workaround; es ist das Standard-Deployment-Modell.

AI und Robotics als erstklassige Plattformbürger

Die meisten Kubernetes-Plattformen behandeln AI/ML als separate Aufgabe. OpenKubes AI und OpenKubes Robotics sind keine separaten Produkte, die nachträglich angebaut wurden. Sie laufen auf denselben Clustern, verwaltet durch dieselbe GitOps-Pipeline, gesichert durch dasselbe Zero-Trust-Framework, überwacht durch denselben einheitlichen Observability-Stack. Ein Platform Team, das OpenKubes Anywhere betreibt, verwaltet Cloud-native Anwendungen, LLM-Inferenz-Workloads und Fabrikhallen-Robotik-Systeme unter einem einheitlichen Betriebsmodell.

Fähigkeit	OpenKubes	OpenShift	Rancher	Anthos	Azure Arc
GitOps-first (nativ)	■	Teilweise	Teilweise	Teilweise	Teilweise
CNCF-nativ (keine Wrapper)	■	■	Teilweise	■	■
Air-Gap / Sovereign Cloud	■	■	■	■	■
Bare Metal (Metal3/CAPI)	■	■	Teilweise	■	■
Edge-Fleet-Management	■	Teilweise	■	Teilweise	■
AI/MLOps integriert	■	Teilweise	■	Teilweise	Teilweise
Robotics / Industrial AI	■	■	■	■	■
Open Source (kein Lock-in)	■	■	Teilweise	■	■
Multi-Cloud ohne Lock-in	■	Teilweise	■	Teilweise	■

Tabelle 8: Wettbewerbs-Feature-Vergleich. ■ = native/volle Unterstützung, Teilweise = mit Einschränkungen, ■ = nicht unterstützt. Bewertungen basieren auf der Standard-Plattformarchitektur ohne Drittanbieter-Erweiterungen. Alle genannten Produkte sind Marken der jeweiligen Eigentümer.

OpenKubes Anywhere konkurriert nicht allein auf Features. Es konkurriert auf Philosophie:

Open by default. Declarative by design. Sovereign by architecture. Extensible without vendor permission.

Für Unternehmen, die über cloud-provider-verwaltetes Kubernetes hinausgewachsen sind und eine Plattform benötigen, die vom Laptop des Entwicklers über die Fabrikhalle bis zur Multi-Cloud-Produktionsumgebung gleich funktioniert — OpenKubes Anywhere ist die einzige Plattform, die von Grund auf für dieses Betriebsmodell entwickelt wurde.

13. Fazit

OpenKubes Anywhere repräsentiert einen bedeutenden Schritt vorwärts im Enterprise Platform Engineering. Durch die Vereinheitlichung des Kubernetes-Betriebs über alle Umgebungen über eine einzige GitOps-gesteuerte Control Plane können Unternehmen operative Fragmentierung eliminieren, Plattformkomplexität reduzieren und die Entwicklerproduktivität steigern.

Vollständig auf CNCF-abgeschlossenen und inkubierenden Projekten aufgebaut — ArgoCD, Cluster API, Crossplane, Prometheus, OpenTelemetry — vermeidet OpenKubes Anywhere Vendor Lock-in durch Design. Platform Teams behalten die volle Kontrolle über ihre Infrastruktur und profitieren gleichzeitig von der Geschwindigkeit eines vorintegrierten, produktionsvalidierten Platform-Engineering-Frameworks.

Die Roadmap spiegelt eine klare Entwicklung wider: von der heutigen operativen Vereinheitlichung hin zur autonomen, KI-getriebenen Infrastruktur von morgen. Das heute aufgebaute Fundament — deklarativ, versioniert, auditierbar — ist die Voraussetzung für alles, was folgt.

Kubernetes hat Anwendungsportabilität gelöst. OpenKubes Anywhere löst operative Portabilität. Die Zukunft ist nicht nur Cloud. Die Zukunft ist Anywhere.

Kubernauts GmbH Im Mediapark 4C, 50670 Köln

Web https://kubernauts.de/de/openkubes/openkubes-anywhere/

Telefon +49 221 379 90 680