MayaData

Technical White Paper

# OpenEBS 1.3

## Data Agility for the Enterprise

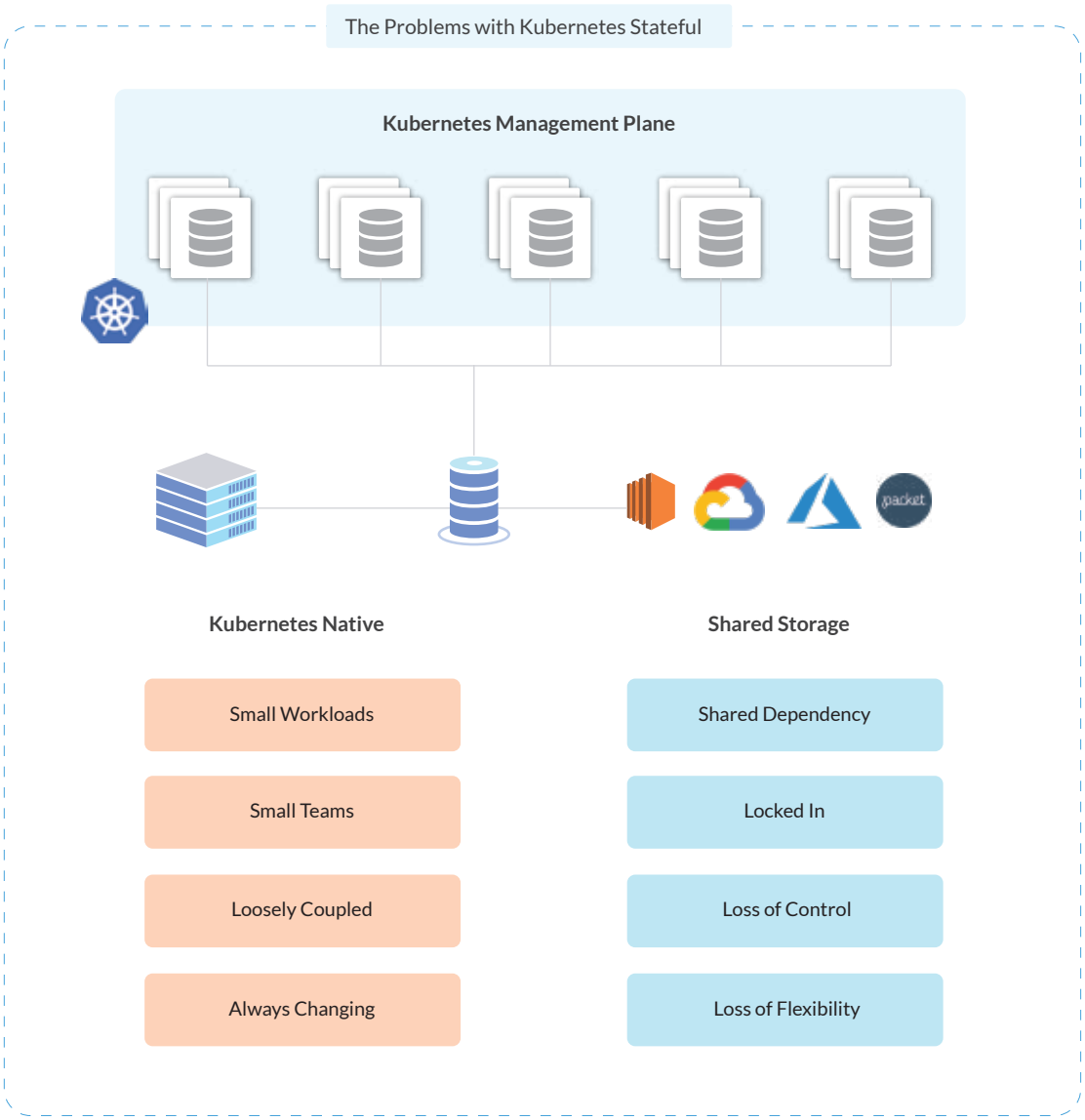
OpenEBS: Built in Kubernetes, Architected for Kubernetes

## Table of Contents

## Software-Defined Data Management: Container Attached Storage

Developers face several challenges when it comes to data management for stateful Kubernetes applications:

- Storage APIs & behaviors differ across
    - Cloud environments lock users into the cloud API, limiting workload mobility
    - Moving from dev in one cloud to production on another cloud or on-prem requires rewiring storage
- Centralized storage models do not align well with Kubernetes decentralized, loosely coupled architecture where data disaggregates at a low level of granularity & control
- Many non-CAS storage solutions require PVCs be tightly coupled to the Kernel modules of worker nodes and cannot be safely rescheduled by Kubernetes
- Data is not replicated with localPVs so node and disk failures mean data loss
- Cloud challenges:
    - Slow I/O performance
    - Fixed volume size (only EBS volumes can be resized)
    - Detach and attach is not instantaneous which means very slow failover
    - Cross zone availability is difficult, e.g. EBS instances are limited to single zones
- On-premise challenges:
    - Storage on-prem often requires manual administrator allocation
    - Centralized architectures draw workloads towards storage nodes, limiting workload mobility, performance scalability and overall throughout
    - I/O “blending” hampers predictable, scalable performance
    - NVMe introduces additional storage hierarchy design considerations




The effects of these constraints are pervasive:

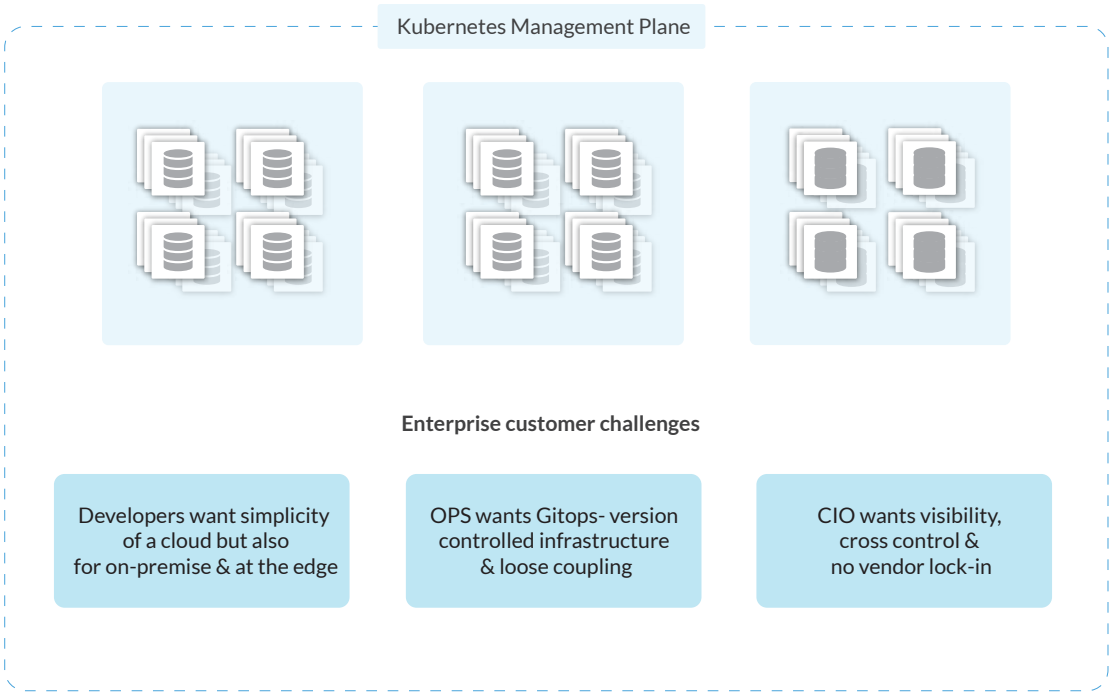
- Developers struggle to force-fit disaggregated cloud-native datasets into monolithic, shared dependency storage systems.
- Developers get pulled into discussions and meetings about shared data infrastructure.
- Developers must explicitly manage complex IT landscapes.
- Overall storage costs seemingly rise forever.
- Stateful workloads that worked well on Kubernetes environment A don't work on environment B especially when dev is in the cloud and production on-premises or vice

Storage system lock-in decreases developer agility and increases spending.

## Kubernetes Data Agility Requirements

A next generation architecture is needed to satisfy three key Kubernetes user types:

- **Developers** want the control & simplicity of cloud but also for on-premise & at the edge
- **Ops** wants distributed GitOps - version controlled infrastructure with loose coupling
- **CxOs** wants visibility, cost control & no vendor lock-in

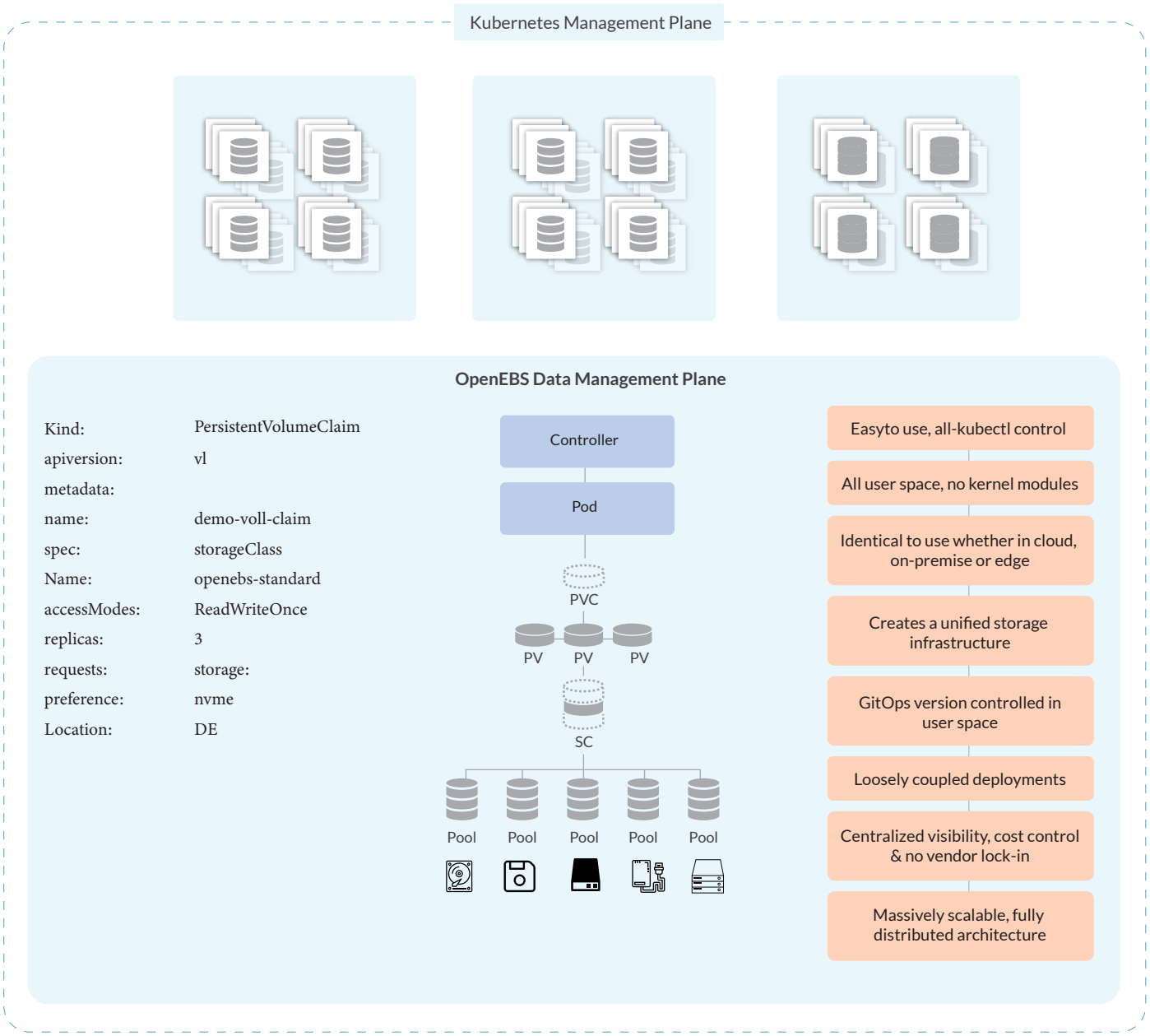



## OpenEBS Kubernetes-Native Data Management

MayaData's OpenEBS Container Attached Storage (CAS) is a next generation architecture that meets all three of these user requirements by deploying a Data Management plane that architecturally mirrors Kubernetes' application management plane.

OpenEBS unifies disaggregated storage into a natively-managed data agility component in the Kubernetes application layer. For Kubernetes applications, OpenEBS creates a unified software-defined storage infrastructure on top of the variegated hardware and software offerings typically deployed in an enterprise WAN. This simplifies developer's lives, gives control to DevOps and provides complete usage visibility to CxOs.

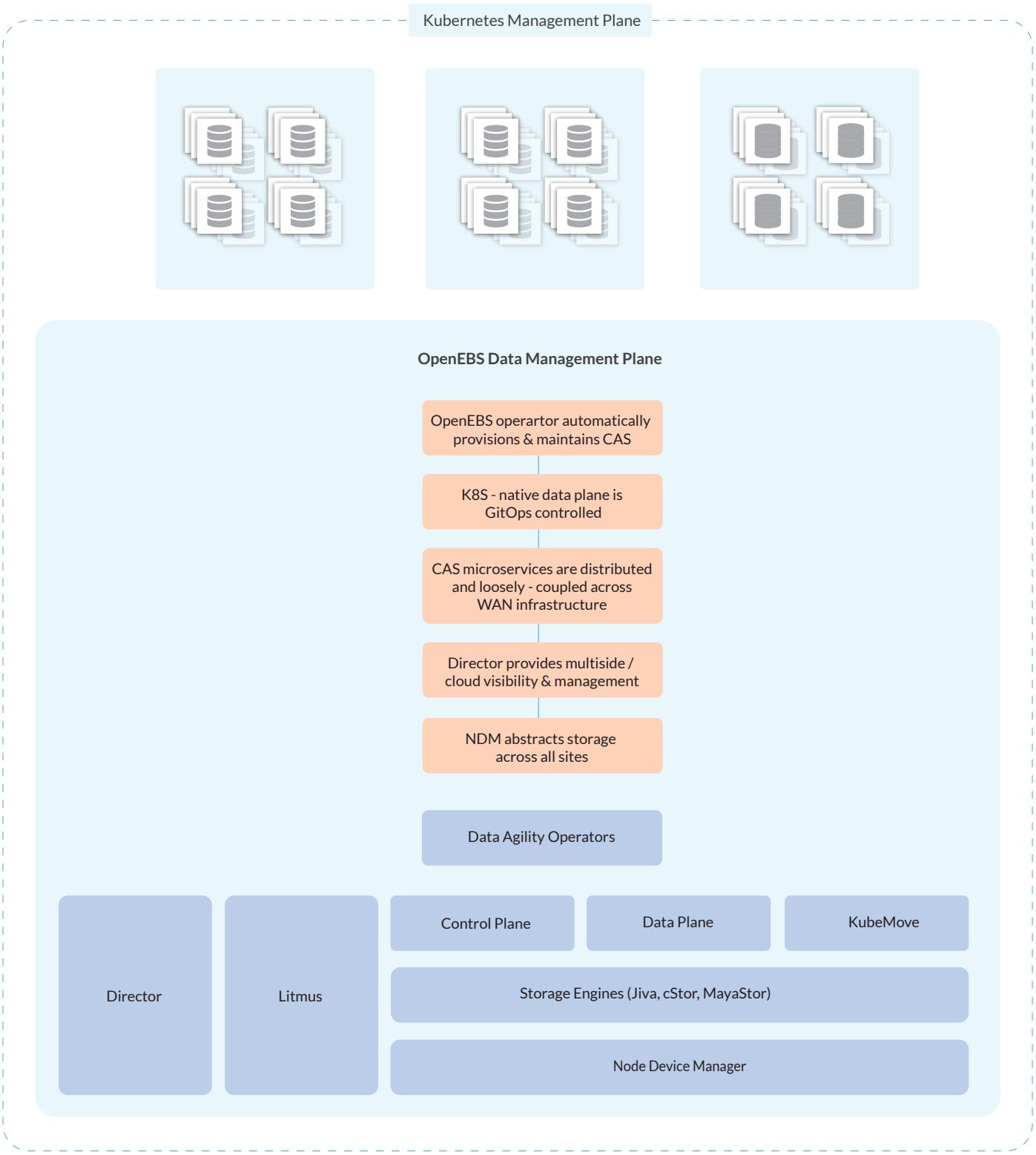
OpenEBS makes the management of stateful applications across an enterprise WAN easy, predictable and resilient. What Kubernetes does for distributed application management, OpenEBS does for distributed data management.




OpenEBS deploys a loosely-coupled, decentralized but unified storage infrastructure entirely in user space, no kernel modifications required. Developers specify data management requirements and replication rules using standard YAML configuration files which allows the data layer itself to be tested, deployed and version-managed right alongside its Kubernetes application, regardless of the infrastructure it is deployed on.

## OpenEBS Kubernetes-Native Architecture

The OpenEBS architecture gives developers fine-grained, self-service control of disaggregated storage across data centers, multiple clouds and all the way out to ARM infrastructure at the 5G edge.



**OpenEBS Node Device Manager** abstracts all device types so that storage resource management is consistent regardless of where the cluster is deployed. Device support includes:

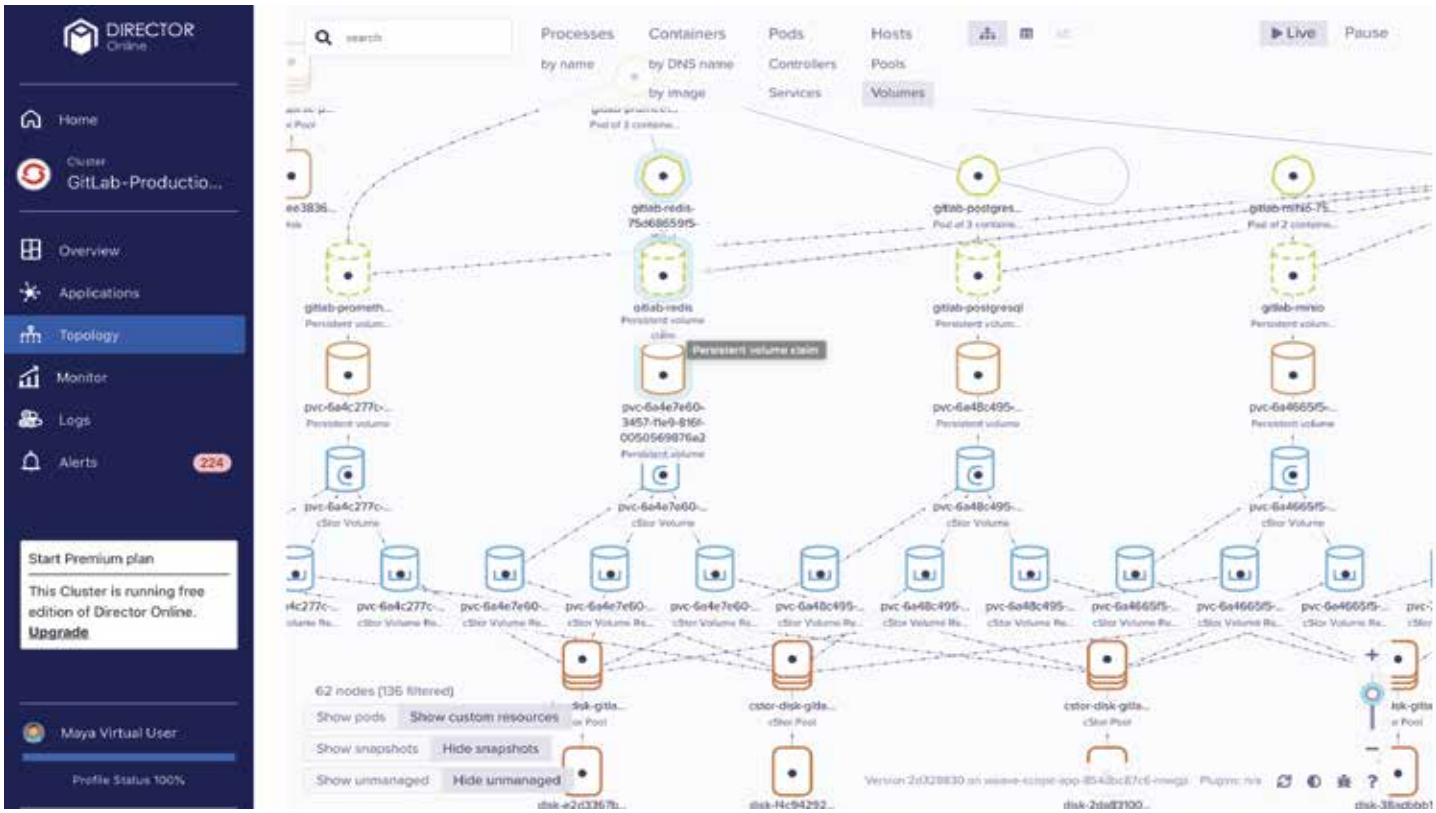
- Cloud Volumes (EBS, S3, GCS, ABS)
- Physical Block Devices (SSD, NVMe)
- Block Storage (LVM, ZFS)
- Virtual Devices (VMDK, VSAN)
- Scale Out Storage (Ceph, Gluster)

OpenEBS creates storage pools on top of these physical resources. The pluggable **OpenEBS Storage Engine** provisions Kubernetes PV resources on these pools for different application use cases:

- **Local PV Provisioner** - Dynamic Local PV for self replicating workloads
- **Jiva** - Lightweight deployment for Edge applications
- **cStor** - Space efficient deployment for Cloud & Data Center applications
- **MayaStor** - Low latency deployment for converged and segregated storage leveraging, among others, NVMe/NVMe over Fabrics (NVMe-oF)

Architected for distributed, K8s-class scalability, **OpenEBS Container Attached Storage** provides a dedicated software-defined storage controller for each container application, leveraging the fully distributed Kubernetes architecture to deliver bare metal performance for each.
Data layer operational state is accessible via standard Kubernetes metrics exported to collectors like Prometheus and FluentD, as well as via an included visual analytics and operational management tool called **MayaData Director** that can be hosted or run on-prem.

## MayaData Director Topology View



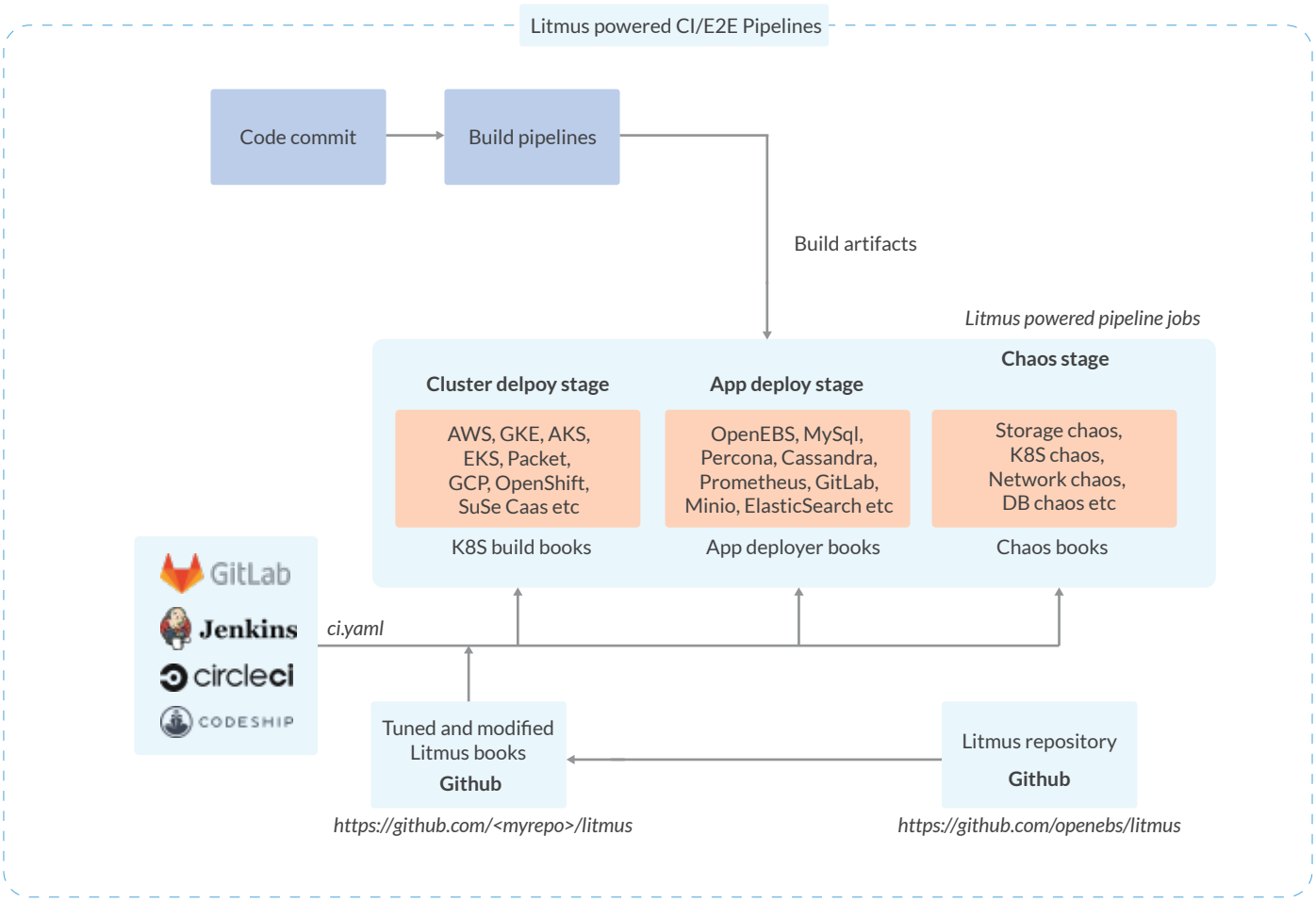
## MayaData Director Disk Management View


## OpenEBS Kubernetes-Native CI/CD

Like Kubernetes' declarative management of application state via git-version-controlled YAML files, OpenEBS uses a state-based, declarative operator to manage both control plane and data plane of the data layer across all infrastructures and media types, no kernel mods required. And since OpenEBS deployments are managed with simple kubectl commands, the entire data layer is readily integrated into devops version-managed CI/CD workflows alongside their applications.

```
Kind:          PersistentVolumeClaim
apiversion:    vl
metadata:
name:          demo-voll-claim
spec:          storageClass
Name:          openebs-standard
accessModes:   ReadWriteOnce
replicas:      3
requests:      storage:
preference:    nvme
Location:      DE
```

Because of its declarative architecture, OpenEBS system reliability can be tested and measured alongside applications with a MayaData-sponsored open source system testing tool called **Litmus**.

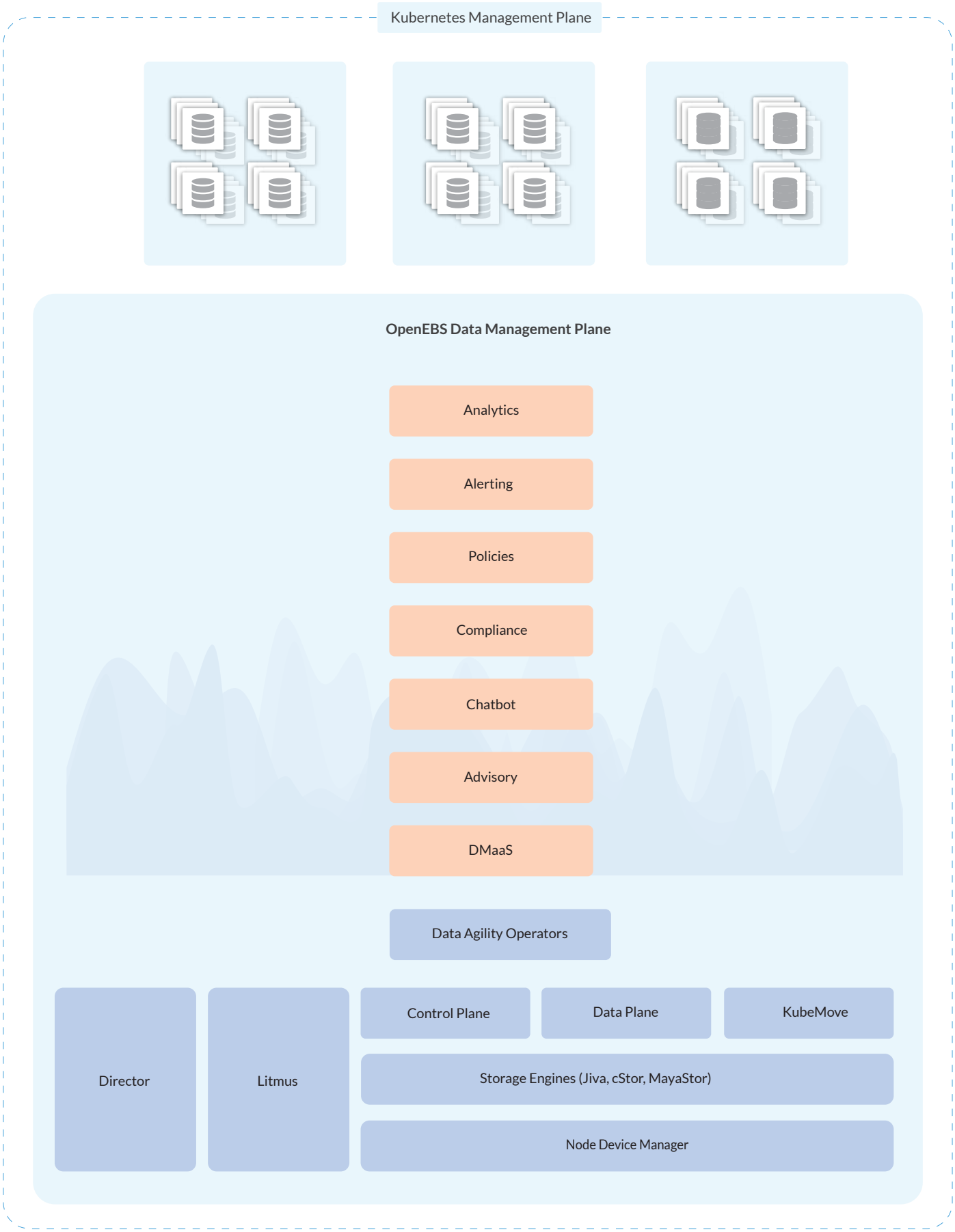



**MayaData's own dev team** uses this same tool to test every OpenEBS build.

## OpenEBS Kubernetes-Native DataOps

With a unified experience and declarative interface to manage the storage/data services, data engineers can interact with Data Infrastructure in a standard way. Building on the CSI foundation, projects like OpenEBS, Velero, standards like KubeMove, SODA Foundation (aka Open Data Autonomy/OpenSDS) are focusing on implementing Easy-To-Use Kubernetes Storage Services for on-prem and cloud and are pushing forward standardization of Declarative Data Plane (aka DDP).

## OpenEBS Kubernetes-Native DataOps



The DDP delivers several architecturally important elements for the next generation of distributed applications DataOps:

- Enabling autonomous teams to manage their own storage
- Scalable polyglot big data storage
- Encryption for data at rest and in motion
- Compute and data locality
- Compliance for GDPR or HIPPA
- Multi-Cloud and Hybrid-Cloud deployments
- Backup and Migration
- Analytics and Visibility into usage for various teams

The DDI project is backed by Infrastructure Teams in large Enterprises that have already adopted Kubernetes and are using Kubernetes and projects like OpenEBS to deliver:

- Etcd As a Service
- ElasticSearch As a Service
- PostgreSQL As a Service
- ML pipelines of many types (one promising one is MELTANO from GitHub)

These implementations show that enterprise customers are taking full advantage of the capabilities delivered by a Declarative Data Infrastructure. These enterprises are leveraging the significant architectural enhancements that DDI at the data layer provides to deliver faster, better and competitively-differentiating enterprise analytics.

## OpenEBS Kubernetes-Native Security

OpenEBS integrates into and deploys within any deployment that follows Kubernetes security best-practices. K8s deployments must address four categories of security:

- Identity-based access,
- Multiuser Collaboration & Operations,
- Secrets Management, and
- Data Protection (encryption)

Any security deployment will rely on items 1-3 to deliver item 4, application layer Data Protection. The principles involved in Data Protection are to:

- Apply encryption at the earliest possible place
- Enable high-granularity security policy
- Protect data as it moves around

Best practice is to use a Kubernetes KMS encryption provider to manage encryption operations. KMS typically use three new concepts to dramatically improve K8 security: envelope encryption, data-specific keys, and automated, time-limited access control. Typically, a KMS is a separate K8 deployment recommended to be deployed somewhere other than the OpenEBS admin node. The KMS stores secrets in memory only and becomes the source of truth of access tokens for any data encrypted on disk. From a developer pov, kubectl combined with the kms-plugin simplifies key management problem by hiding its complexity behind a simple KMS fetch call. For more detail on this topic see Using a KMS provider for data encryption.

- Given an encryption key provided by a KMS, there are five ways that OpenEBS can apply these keys to the data that runs through a Container Attached Storage controller
- Encrypt/decrypt data in transit on the network
- Encrypt/decrypt data at rest in a volume,
- Encrypt/decrypt data at rest in a volumes rule set
- Encrypt/decrypt data at rest in volumes in a container deployment set 5. to encrypt/decrypt data at rest in volumes created by a user
- Encrypt/decrypt data at rest in volumes on an OpenEBS cluster When Vault updates encryption keys (e.g. in case of revocation or for time-based key rotation) the corresponding data encryption key should be automatically updated in OpenEBS volumes.

Given this encryption flexibility, OpenEBS integrates cleanly and seamlessly into any enterprise security environment designed and deployed for Kubernetes-native applications.

## OpenEBS Data Agility Feature Summary

**OpenEBS Data Management** includes the following features:

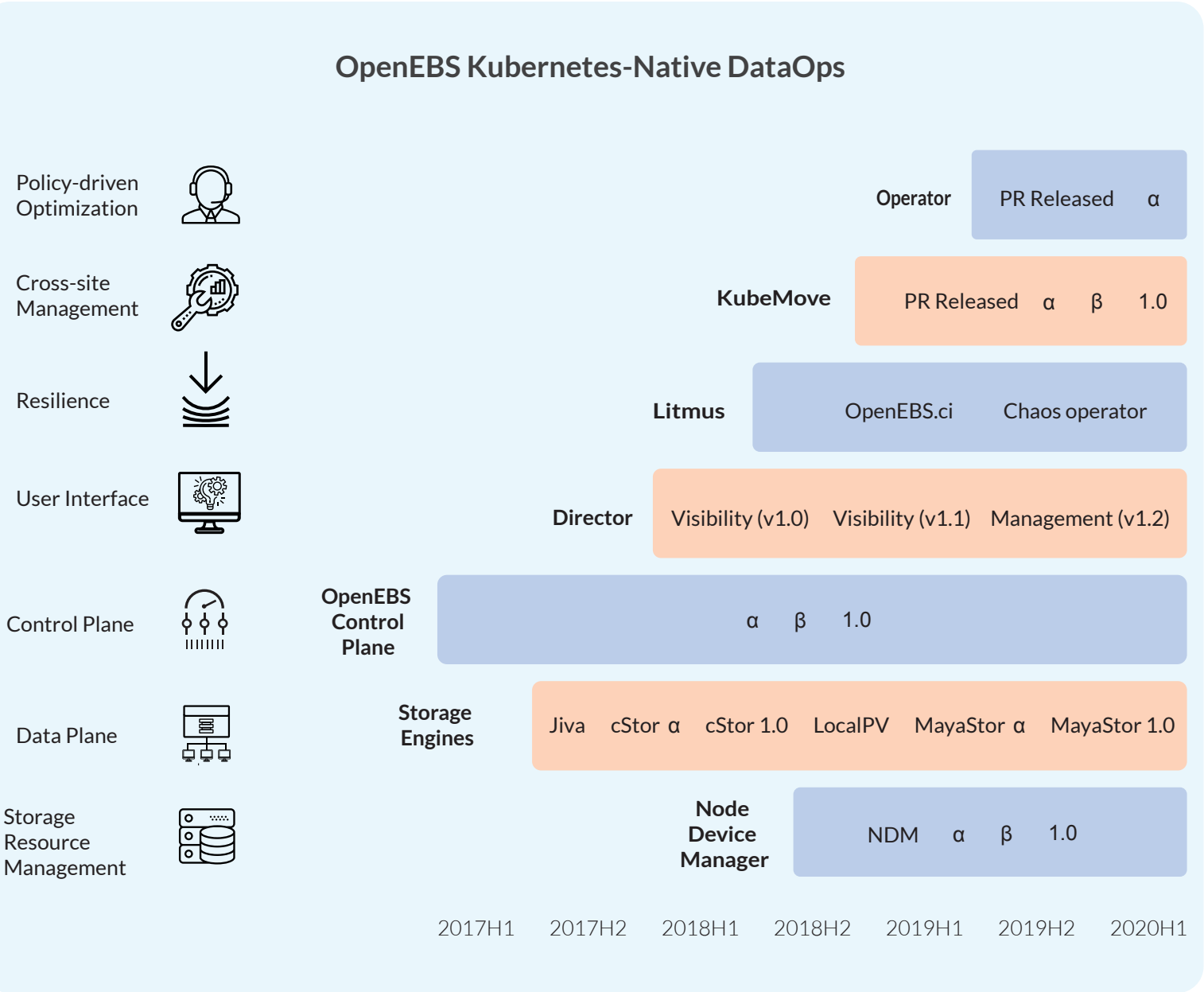
- **Data management**
  - Rule-driven, dynamic volume resizing
  - Rule-driven, dynamic provisioning of LocalPVs in a host path or in disk mode
  - Built-in thin-provisioning, storage pooling and compression management
  - Rule-driven synchronous volume replication across clusters, sites, zones and clouds
  - Fine-grained, Kubernetes-style multi-tenancy isolation by name spaces
  - Per workload data provisioning and copy management
  - Optional software-defined RAID
- **Resiliency management**
  - Rule-driven provisioning, monitoring and inventory management of storage resources
  - Data persistence across nodes, availability zones and cloud providers
  - Full snapshot and clone across media types, hosts and clouds
  - Automated data hydration of replacement volumes in the case of volume failure
  - Fine-grained, Kubernetes-style multi-tenancy isolation by name spaces
  - Automated chaos operator probe for discovering deployment weakness (Litmus)

- **Usage management**
  - Fine-grained data usage visibility and management (Grafana dashboards)
  - Node affinity and anti-affinity for explicit resiliency management
  - **cMotion**-managed stateful application migration across on-prem & cloud sites
  - Slack-integrated smart alerts (MuleBot)
  - Visual cluster state navigation and introspection tool (Director)
- **Performance scalability**
  - Fully distributed, Kubernetes-managed scalability with bare metal performance and per-application performance monitoring
  - Dedicated software-defined storage controller per workload supports massive scalability and avoids the I/O blender effect
  - Easy, scalable, Kubernetes-managed storage policy via tagging
  - Architecture fully leverages Intel multicores to deliver maximum distributed throughput
  - Native Intel spdk support for optimal I/O performance on every node
- **Security**
  - Kubernetes-native security integration

## OpenEBS Product Roadmap

The OpenEBS product roadmap is focused on delivering a complete, intelligent and self-optimizing data agility platform for the Enterprise in order to simplify developers lives, ease DevOps To Do lists, and cost- and performance-optimize data management automatically

The MayaData product roadmap is building towards the vision of automated data agility.




Starting with the delivery of OpenEBS control in 2017, MayaData has sequentially delivered and enhanced three data management engines that span storage environments across multiple clouds, on-premise data centers and out to the edge: Jiva, cStor and MayaStor.
The introduction **of Node Device Manager (NDM)** in 2018 simplified and unified device management across all of those environments.
Also in 2018, **MayaData Director** delivered easy-to-use graphical management of Kubernetes clusters, and **MayaData Litmus** delivered automated resilience testing and readiness certification of data agility clusters.
In 2019 H1, **KubeMove** began to address cross-cloud/cross-premises container and storage migration.
Building on all of these components in 2019 H2, MayaData's **Data Operator** will deliver policy-driven, cross-site and cross-zone automated optimization of the entire data agility stack.

## OpenEBS Customer Testimonials

**CTO evaluating OpenEBS:**

“I love OpenEBS Architecture; it helps remove the layers from the application delivery process. There is no need for a dedicated storage administrator or team. I can easily scale up and down this solution.”

**Artem Larmoliuk, Lead Solution Architect, Levi9**

"We use OpenEBS because it helps us move faster to build reliable solutions for our enterprise clients that include stateful workloads on Kubernetes. It is by far the easiest to deploy and operate container attached storage we 've tried - and MayaOnline and Litmus are promising ways to save time for users running OpenEBS at scale."

**Henry Baltazar, the 451 Group**

"The OpenEBS CAS offered by MayaData provides users with granular control over individual workloads... enabling non-storage administrators to manage storage effectively. Also, MayaData has already established a growing open source user community as well as multiple partnerships"

**Kelvin Wong, Senior Devops Engineer, Streetline**

“At Streetline our speed of feature delivery and our ability to reliably operate systems at scale and reasonable cost are mission critical. OpenEBS on a vanilla Kubernetes running on Ubuntu VMs in vSphere is helping us deliver on these key responsibilities.”

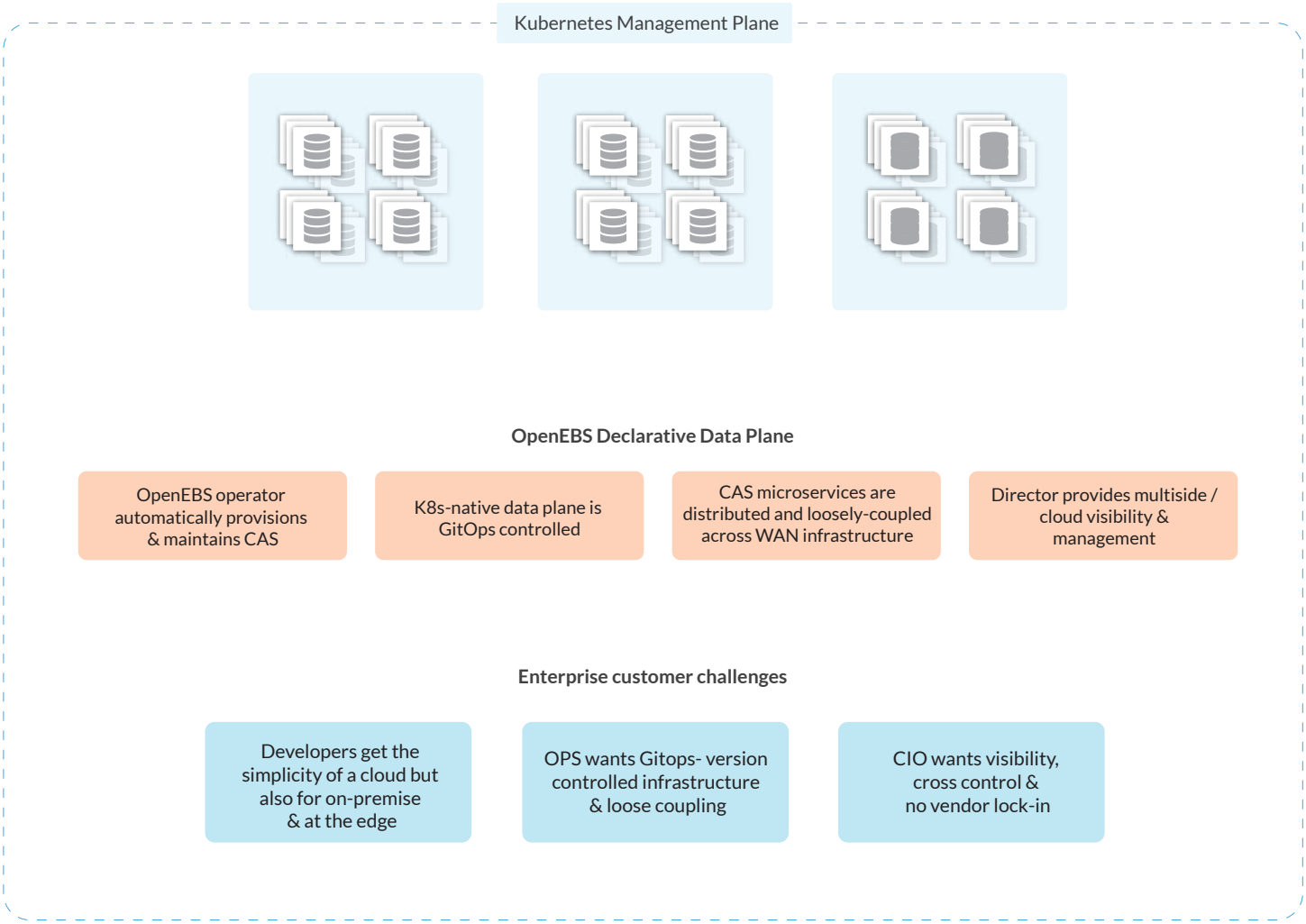
“OpenEBS was deployed so quickly that it initially seemed like something might have gone wrong. It just worked immediately out of the box. I was able to get the alternative system working as well; however it took a bit longer and I never felt the level of clarity about its operation - such as where the data was actually stored - as I did when I ran OpenEBS.”

## OpenEBS Summary

OpenEBS is a cloud-native data agility layer for any Kubernetes deployment that itself uses a Kubernetes-style, state-based, declarative operator to predictably manage resiliency, consistency and scalability of K8s data across all infrastructures and media types in an enterprise, simplifying backup, save & restore of cluster application state. Benefits include:

- The OpenEBS unified data agility infrastructure gives developers fine-grained, self-service autonomy and control of disaggregated storage but with the simplicity of cloud and seamless dev/test/prod deployment across sites.
- OpenEBS creates a loosely-coupled, decentralized but **unified data agility infrastructure** that can support the data management needs of K8s applications across data centers, multiple clouds and all the way out to ARM infrastructure at the 5G edge.
- OpenEBS gives **DevOps** distributed GitOps - loosely-coupled but git version-managed & cost-controlled data infrastructure across availability zones provides policy-driven data agility and provisioning that empowers distributed development teams to self-provision and work at their own pace while giving IT full visibility and control and avoiding cloud lock in.
- OpenEBS gives **CxOs** centralized visibility, cost control & no vendor lock-in.

OpenEBS aligns with Kubernete's fundamental architecture to deliver the last mile of Kubernetes deployments, a fully Kubernetes-native, CI/CD-controlled declarative data management platform that spans and simplifies the complex, heterogeneous data infrastructure encountered in the typical enterprise.




MayaData

linkedin.com/company/mayadata/

https://blog.mayadata.io

@mayadata_inc