



**Quasar**SDS

# Enterprise Storage Platform

## Technical Architecture & Capabilities Report

High-Performance software-defined NVMe-oF data mobility, disaster recovery, and bare-metal Kubernetes-native storage engine.

---

|                       |                                                        |
|-----------------------|--------------------------------------------------------|
| <b>Document Type:</b> | Technical Architecture Report (Enterprise Edition)     |
| <b>Date of Issue:</b> | June 2026                                              |
| <b>Version:</b>       | v2.0 (Consolidated Specifications)                     |
| <b>Audience:</b>      | CTOs, IT Infrastructure Architects, System Integrators |

# Table of Contents

---

Reference index for architectural validation, platform evaluation, and specifications.

- 1. Executive Summary** ..... 3
- 2. The Business Problem & Architectural Pressures** ..... 4
- 3. Platform Overview** ..... 5
- 4. Core Architecture & Component Topology** ..... 6
- 5. Enterprise Use Cases** ..... 8
- 6. High Availability & Failover Mechanics** ..... 10
- 7. Disaster Recovery & FinOps Optimization** ..... 13
- 8. Zero-Downtime Cloud Migration & Bidirectional Rollback** ..... 15
- 9. Kubernetes CSI Integration & Performance** ..... 16
- 10. Technical Validation & Cryptographic Data Integrity** ..... 19
- 11. Technical Specifications Summary** ..... 20

# 1. Executive Summary

---

Modern enterprise infrastructure demands storage engines capable of keeping pace with high-velocity cloud-native workloads, automating multi-site disaster recovery protocols, and delivering bare-metal physical performance without locking data into closed, proprietary storage arrays. **QuasarSDS** is a Kubernetes-native, software-defined storage (SDS) platform designed precisely to satisfy these three fundamental requirements.

By eliminating traditional structural trade-offs between speed and administrative flexibility, QuasarSDS delivers **sub-millisecond P99 latency** and **millions of concurrent IOPS** utilizing native NVMe over Fabrics (NVMe-oF TCP). It runs entirely on commodity x86/ARM hardware while integrating seamlessly with production orchestration layers via a standard Container Storage Interface (CSI) driver.

Beyond raw, physical IOPS performance, QuasarSDS provides a unified and secure data lifecycle pipeline. This comprises **synchronous replication** for absolute high availability (RPO=0), **asynchronous deduplicated backups directly to Object Storage (S3)** for optimized disaster recovery, and an incremental, zero-downtime **cloud migration path** to AWS with a native bidirectional fallback option.

**Absolute Data Sovereignty:** QuasarSDS is a proprietary, enterprise-grade storage software platform, but your data always belongs to you. Our backup and migration pipelines store blocks in open, unproprietary RAW format guided by clear, logical JSON manifests. This ensures that in any catastrophic scenario, your data can be audited, mounted, or restored using native Linux tools without requiring the QuasarSDS software itself, eliminating any risk of vendor lock-in.

## 2. The Business Problem & Architectural Pressures

Infrastructure engineers and enterprise IT organizations currently face several converging operational bottlenecks. Traditional storage options fail to satisfy modern cloud-native standards across multiple vectors:

| Operational Vector | Legacy Approach                                                                 | Inherent Architectural Problem                                                                    |
|--------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| High Availability  | Proprietary hardware-based SAN/NAS controllers.                                 | Extreme CAPEX, complex custom fabrics, and heavy vendor lock-in.                                  |
| Kubernetes Storage | Generic CSI plugins mapping network file systems or standard cloud volumes.     | Heavy kernel I/O taxes, severe IOPS degradation, and high inter-zone traffic fees.                |
| Disaster Recovery  | Periodic hypervisor snapshots or traditional file-level scheduled backups.      | Hours of data loss (high RPO), complex multi-step recovery runbooks, and massive manual overhead. |
| Cloud Mobility     | Block-level off-line disk copies or heavy application-level network migrations. | Extensive production maintenance windows and high risks of data corruption during transit.        |
| FinOps Efficiency  | Pre-purchased, over-provisioned local physical storage arrays.                  | Massive capital tied to idle capacity and unpredictable expansion costs.                          |

QuasarSDS replaces these disjointed mechanisms with a single unified, lightweight, and programmable storage plane.

## 3. Platform Overview

---

QuasarSDS is a **open-architecture, Kubernetes-native software-defined storage (SDS) platform** that aggregates local NVMe physical SSDs into a high-performance, distributed storage fabric. The platform is managed entirely through standard declarative APIs, making it natively compatible with GitOps automation pipelines.

Storage targets are presented directly to the node environment via two key interfaces:

1. **NVMe-oF TCP Protocol:** Exposes raw block devices directly to bare-metal servers, hypervisors, and standalone VMs.
2. **Kubernetes CSI Plugin:** Automatically provisions, attaches, and mounts persistent volumes directly inside active container pods.

## 4. Core Architecture & Component Topology

---

The architectural foundation of QuasarSDS enforces a strict separation of concerns by decoupling the control plane from the active data plane. The platform is comprised of four lightweight, containerized components:

| Component         | Role & Operational Scope                                                                                  | Deployment Model                                |
|-------------------|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| <b>Agent</b>      | Direct physical disk management, SPDK user-space initialization, replication logic, and backup streaming. | Kubernetes DaemonSet on storage provider nodes. |
| <b>Controller</b> | Global orchestration operator, state management, and cluster-wide volume provisioning.                    | High-Availability Deployment.                   |
| <b>Witness</b>    | Stateless consensus arbiter used to prevent split-brain states in HA failover events.                     | 3-Replica Deployment.                           |
| <b>CSI Driver</b> | Implements standard CSI specifications to map, mount, and manage lifecycle events of volumes.             | Kubernetes DaemonSet across worker nodes.       |

The data plane executes entirely in **User-Space** utilizing the **Storage Performance Development Kit (SPDK)**. By bypassing the Linux kernel's storage layers, interrupt handlers, and context switches, QuasarSDS achieves bare-metal efficiency, routing data directly from applications to physical NVMe media via polling-mode drivers.

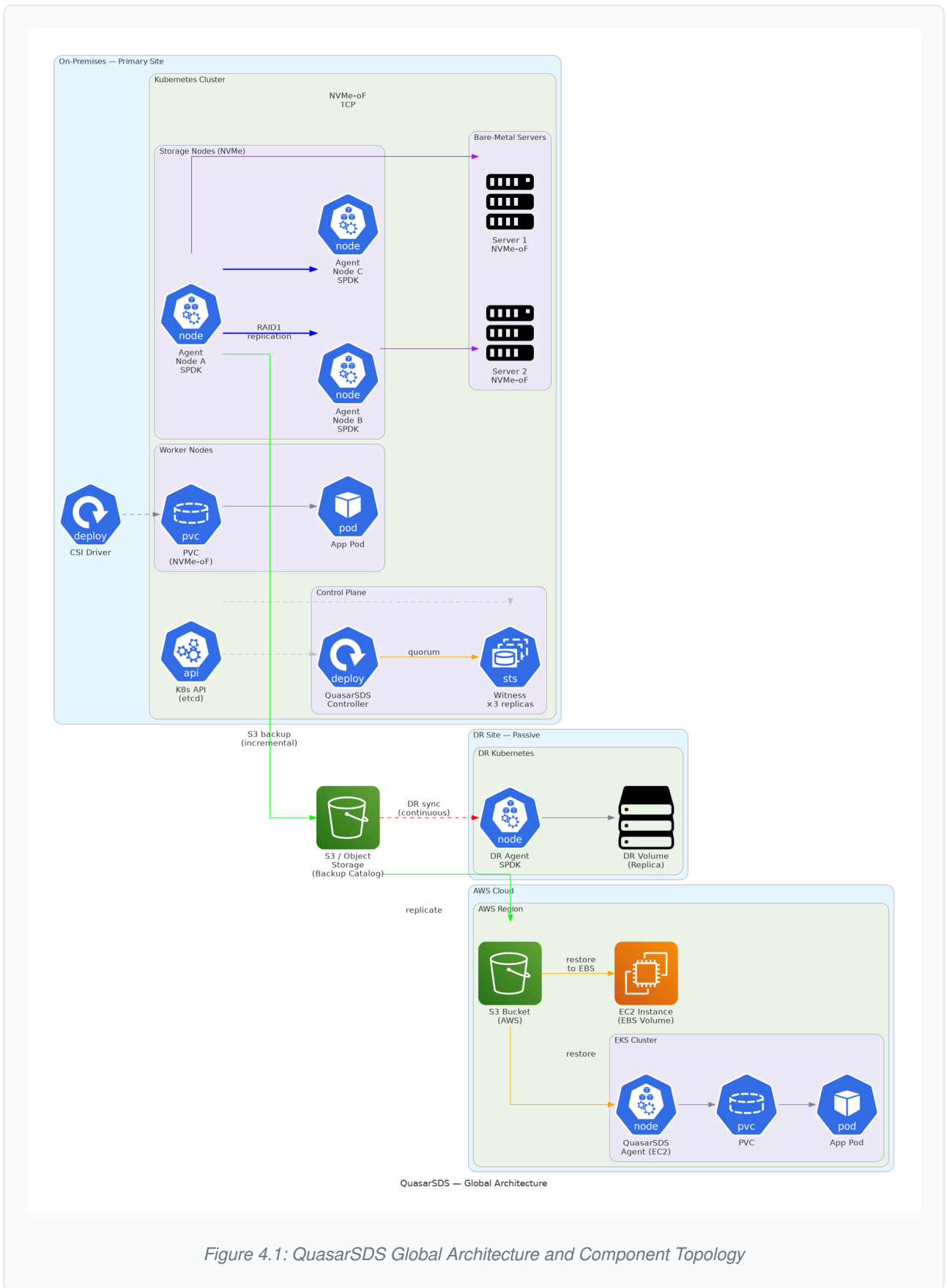


Figure 4.1: QuasarSDS Global Architecture and Component Topology

## 5. Enterprise Use Cases

---

To guide IT decision-makers, CTOs, and Infrastructure Architects, the following production blueprints demonstrate how QuasarSDS solves real-world bottlenecks, ranging from state-of-the-art AI clusters to legacy physical storage array modernization.

### 5.1 AI/ML Infrastructure & High-Speed GPU Feeding

**The Challenge:** Legacy storage architectures like shared NAS (NFS/SMB) and Object Storage (S3/HTTP) are highly unsuited for active AI model training pipelines. File systems (NAS) introduce massive kernel translation overhead and lock contention when thousands of training workers attempt to access multi-gigabyte datasets concurrently. Object storage (S3) lacks low-latency random-access block operations, forcing engines to download full objects or deal with highly latent API parsing, leading to severe **GPU Starvation** and idle compute cycles.

**The QuasarSDS Block-Level NVMe-oF Solution:** QuasarSDS bypasses the limitations of both File and Object storage by delivering high-performance, raw storage at the **Block Level** consumed via the optimized **NVMe over Fabrics (NVMe-oF TCP)** protocol.

- **Raw Block-Level Efficiency over Networks:** By presenting datasets as raw block devices over standard Ethernet via NVMe-oF, QuasarSDS eliminates the virtual file system (VFS) layers, directory metadata operations, and kernel context switches. It behaves exactly like local PCIe NVMe media, delivering data directly to training nodes at absolute wire speeds.
- **Massive Parallelism (64K Queue Depth):** Traditional NFS serializes read operations, creating immediate bottlenecks under heavy concurrency. NVMe-oF supports up to 64,000 independent command queues with up to 64,000 commands per queue. This massive architectural parallelism allows massive GPU clusters to stream data in parallel from the same underlying QuasarSDS fabric without performance degradation.
- **Bypassing the Host Kernel (SPDK Polling):** Operating entirely in user-space, QuasarSDS pulls blocks from NVMe disks and pushes them directly to the network interface card (NIC) without host CPU interrupts or context switches, reducing P99 latency to  $<150 \mu\text{s}$  to prevent GPU idle cycles.
- **Sub-Second Model Checkpointing:** Leveraging SPDK multi-channel writes, PyTorch or Ray frameworks can write model checkpoints at sequential throughput speeds of **>12.5 GB/s**, shortening model-state pause times from minutes to fractions of a second.
- **Kubernetes-Native Orchestration (CSI):** Exposes high-performance block volumes dynamically to Kubernetes AI orchestrators (e.g., KubeFlow, Ray, Volcano) via the QuasarSDS CSI driver, mounting bare-metal block storage directly inside active containerized workspaces.

## 5.2 Kubernetes-Native Stateful Applications (Databases at Bare-Metal Speed)

**The Challenge:** Relational and non-relational enterprise databases (PostgreSQL, MongoDB, Cassandra) require ultra-high transactions per second (TPS) and predictable latencies in the cloud, but traditional cloud storage limits IOPS or charges prohibitive premium fees.

**The QuasarSDS Solution:**

- **Bare-Metal Performance in Containers:** QuasarSDS provides transactional databases with local physical NVMe performance while retaining cloud-native advantages such as automated failover in ~6 seconds.
- **Transactional Consistency with RF3:** Three-way synchronous replication (RF3) guarantees that every database commit is acknowledged across three physical storage nodes, achieving full ACID compliance without latency penalties thanks to SPDK user-space optimization.

## 5.3 Legacy SAN Modernization & Hardware Lifecycle Extension

**The Challenge:** Organizations hold legacy, highly capital-intensive SAN/NAS storage arrays that cannot communicate natively with Kubernetes or fail to scale for dynamic microservice workloads, but cannot yet write off or retire this hardware due to amortization cycles.

**The QuasarSDS Solution:**

- **Modernizing Legacy Storage:** QuasarSDS can act as a high-performance software-defined storage front-end on top of legacy SAN arrays.
- **Unified CSI Abstraction:** Exposes legacy SAN physical resources directly to Kubernetes as dynamic, orchestratable NVMe-oF block volumes, extending hardware lifetime by up to 300% while transitioning to a modern, cloud-native operational model.

## 5.4 Hybrid Cloud Mobility & FinOps Cost Reduction

**The Challenge:** High data egress fees, vendor lock-in, and static cloud storage performance tiers severely restrict hybrid cloud architectures and drive up infrastructure operational costs.

**The QuasarSDS Solution:**

- **Active Multicloud Strategy:** Stream historical cold block data to low-cost S3 object storage while dynamic cloud workloads hydrate AWS EBS volumes on-demand using direct EBS APIs.
- **FinOps Financial Optimization:** Reduce cloud storage bills by up to 60% by eliminating over-provisioned premium cloud block devices, substituting them with dynamic, siphoned QuasarSDS volumes matched síncronamente to actual workloads.

## 6. High Availability & Failover Mechanics

---

### 6.1 Synchronous Multi-Node Replication (Zero RPO)

To ensure maximum durability, QuasarSDS replicates block write commands synchronously across multiple physical nodes using a software RAID-1 mechanism:

- **Replication Factor 2 (RF2):** Every block write is synchronously committed to two independent nodes before acknowledging completion. One physical node can fail with absolute zero data loss.
- **Replication Factor 3 (RF3):** Every write is committed to three independent nodes. This setup survives two simultaneous node outages, ideal for critical databases.

Because write operations are committed in a blocking, synchronous fashion, the standby secondary replica remains perfectly identical to the active primary replica at any given CPU instruction, maintaining a **Recovery Point Objective (RPO) of absolute zero**.

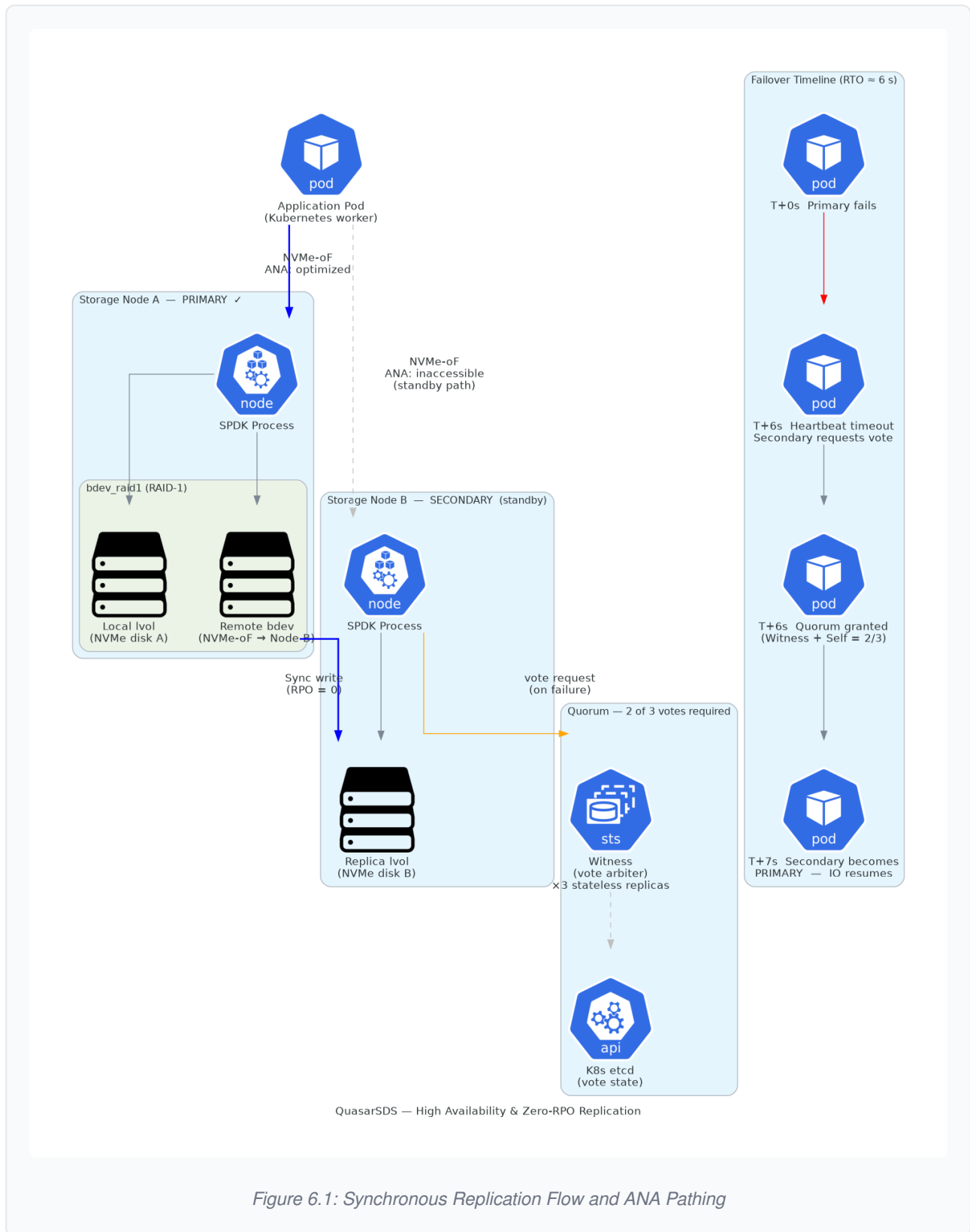


Figure 6.1: Synchronous Replication Flow and ANA Pathing

## 6.2 Automated Consensus and Path Failover

In the event of an active primary node outage, the surviving components automatically trigger a failover sequence:

```
T+0s Primary storage node experiences unexpected hardware failure.
T+6s gRPC heartbeat monitoring triggers a timeout detection on the standby node.
T+6s Quorum consensus sequence initiated. Standby requests Witness vote.
T+6s Witness evaluates etcd state via optimistic lock. Vote authorized.
T+6s Secondary replica promoted to active Primary role.
T+7s CSI clients execute path redirection via NVMe-oF Asymmetric Namespace Access (ANA) .
```

The application workload resumes I/O operations in approximately **6 seconds (RTO)**. The redirection occurs natively at the NVMe protocol layer, ensuring zero application crashes, zero required pod restarts, and zero configuration changes.

## 6.3 Split-Brain Prevention

To prevent network partitions from creating two active primary writers, QuasarSDS enforces strict **quorum consensus** requiring at least 2 out of 3 active votes. The three voters are the dos storage replica nodes plus the stateless **Witness** service. The Witness interacts with the Kubernetes control plane's `etcd` database using optimistic concurrency controls (`resourceVersion`), preventing race conditions in complex failover scenarios.

# 7. Disaster Recovery & FinOps Optimization

## 7.1 Continuous Incremental Block Streaming to S3

Rather than relying on resource-intensive VM snapshots, QuasarSDS streams changes continuously to any S3-compatible object store (Amazon S3, MinIO, Ceph RGW) using a **Full + Incremental Backup Chain**:

1. An initial **Full Backup** compresses and optional encrypts the entire volume structure into small block chunks.
2. Subsequent **Incremental Backups** stream only the specific blocks modified during that particular sync window, drastically reducing network usage.
3. A lightweight, machine-readable `catalog.json` manifest is written to the S3 bucket, listing the exact logical offsets of each incremental block to allow instant point-in-time recovery.

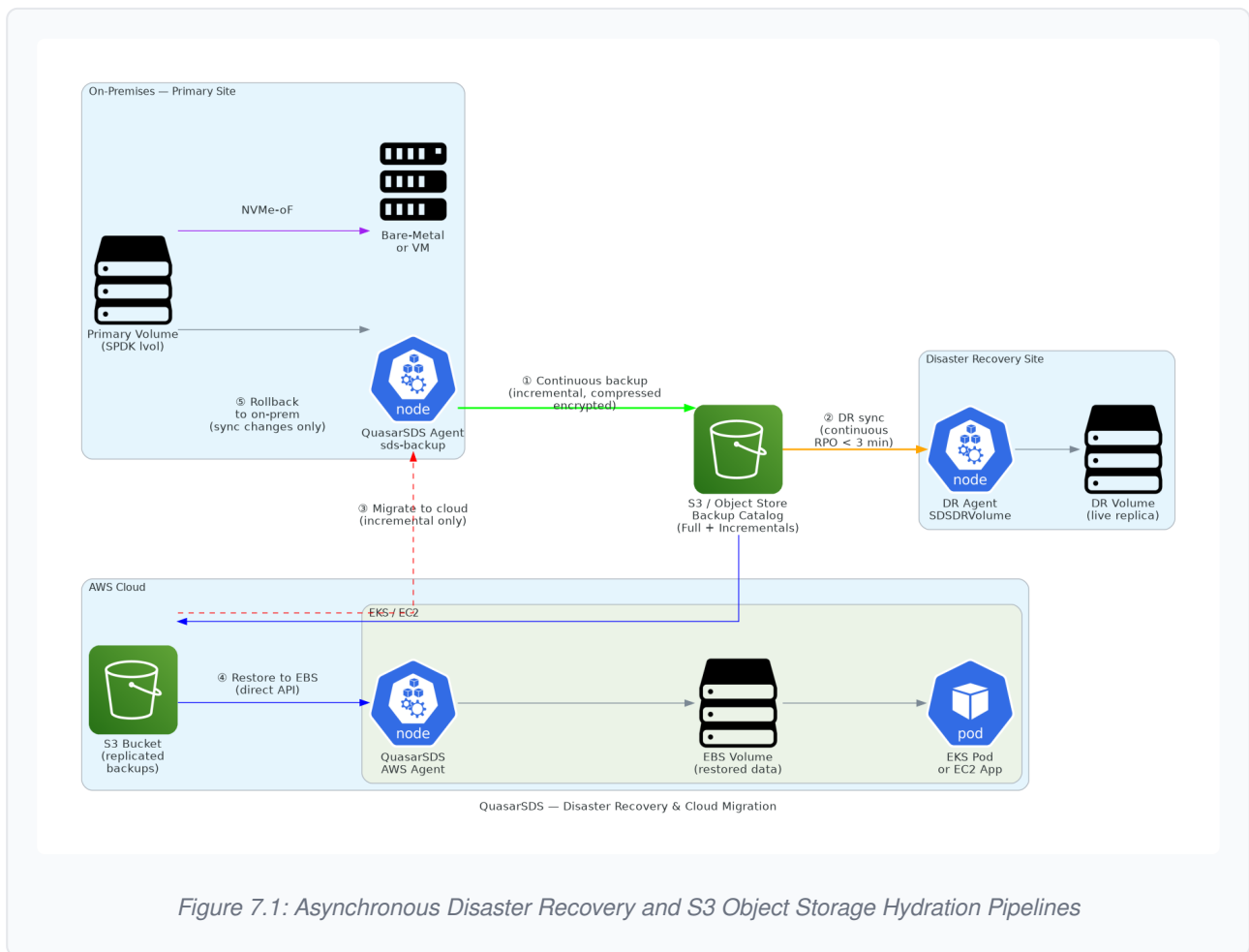


Figure 7.1: Asynchronous Disaster Recovery and S3 Object Storage Hydration Pipelines

## 7.2 Multi-Region Disaster Recovery

The declarative ``SDSDRVolume`` Custom Resource continuously monitors the target S3 catalog, retrieving, decompressing (via ``zstd``), and applying incremental blocks to a remote disaster recovery volume. This architecture supports 1-to-Many replication topologies, allowing multiple geographical targets to maintain read-only standby copies of a single production volume with minimal overhead.

## 7.3 Continuous RPO Validation

The DR controller continuously measures and exposes the ``lagSeconds`` metric (the age of the newest unapplied S3 block at the destination). These metrics are natively exposed via a Prometheus exporter, enabling operations teams to monitor real-time compliance with business SLAs on unified dashboards.

## 8. Zero-Downtime Cloud Migration & Bidirectional Rollback

---

The decoupled architecture of QuasarSDS enables seamless block-level data mobility across hybrid cloud topologies without requiring complex application re-architecting.

### 8.1 On-Premises to AWS Migration

Migrations are performed online utilizing a non-disruptive, multi-phase hydration model:

- **Phase 1: Initial seeding** — A full backup is generated from the local physical volume and uploaded asynchronously to the target S3 bucket while the local application remains online.
- **Phase 2: Live replication** — Small, periodic incremental block updates are sent to S3 to keep the cloud catalog synchronized with active live transactions.
- **Phase 3: Cutover** — The AWS-native QuasarSDS agent reads S3 data and hydrates an active AWS EBS volume directly utilizing high-speed **AWS EBS Direct APIs**. The local application is paused, one final micro-incremental delta is applied, and the workload boots instantly on AWS.

### 8.2 Bidirectional Rollback Capability

Unlike tools that trap workloads in the public cloud, QuasarSDS maintains an **incremental, bidirectional data movement** reverse path. Backups can stream from the newly active AWS volume back to S3, allowing the on-premises site to act as a disaster recovery destination. If rollback is required, only the incremental blocks written during the cloud execution phase are copied back, minimizing egress costs and downtime.

## 9. Kubernetes CSI Integration & Performance

---

By eliminating intermediate file systems, block mapping abstractions, and operating system kernel contexts, QuasarSDS delivers extreme performance directly to containerized workloads.



## 9.1 Enterprise Performance Benchmark Comparison

| Performance Metric               | QuasarSDS Engine (SPDK) | Standard Enterprise SAN / NAS |
|----------------------------------|-------------------------|-------------------------------|
| Sequential Read/Write Throughput | > 12.5 GB/s             | 1.5 – 3.2 GB/s                |
| Random 4K Read/Write IOPS        | > 1,200,000 IOPS        | 80,000 – 250,000 IOPS         |
| P99 Transaction Latency          | < 150 $\mu$ s           | 1.2 – 4.5 ms                  |

## 9.2 Declarative Configuration Example (YAML)

Provisions a synchronous, high-availability, double-replicated storage class:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: quasarsds-high-availability
provisioner: quasarsds.io
parameters:
  poolName: nvme-pool-direct
  strategy: RF2 # Synchronous two-way mirroring
reclaimPolicy: Delete
volumeBindingMode: Immediate

---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgres-primary-pvc
spec:
  storageClassName: quasarsds-high-availability
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 250Gi
```

## 10. Technical Validation & Cryptographic Data Integrity

---

To guarantee complete, bit-perfect data integrity, QuasarSDS performs full cryptographic block-level verification at the completion of every block sync or cloud hydration loop. This process verifies that the source and target physical disks are mathematically identical.

This validation can be executed and audited transparently via the `qsdctl` command-line utility:

```
$ qsdctl verify --source local-spdk --target aws-ebs
[INFO] S3 Catalog synchronization complete.
[INFO] Running block-level cryptographic verification...

✓ Source Volume MD5: 4a8e9b10c23d4e5f6a7b8c9d0e1f2a3b
✓ Target AWS EBS MD5: 4a8e9b10c23d4e5f6a7b8c9d0e1f2a3b

[SUCCESS] 100% Data Integrity Verified. Bit-by-bit identical copy.
```

# 11. Technical Specifications Summary

| Technical Architecture Metric | Standard Supported Specification Value                            |
|-------------------------------|-------------------------------------------------------------------|
| Replication Methods           | Synchronous multi-node mirroring (RF1, RF2, RF3)                  |
| Target RPO / RTO Bounds       | Zero (0) RPO / ~6 Seconds Automated Cluster Failover              |
| Transport Storage Protocol    | NVMe over Fabrics TCP (NVMe-oF TCP) standard spec                 |
| Backup Formats                | Raw blocks guided by logical JSON catalogs, compressed via `zstd` |
| Active Data Storage Core      | Intel Storage Performance Development Kit (SPDK) in User-Space    |
| Supported Linux Platforms     | Red Hat Enterprise Linux (RHEL 9+), Ubuntu Server 22.04 LTS+      |
| Provisioning Abstractions     | Kubernetes Custom Resource Definitions (CRDs), fully GitOps-ready |

**Document Registry:** Architectural Specification Report v2.0 | **Last Evaluated:** June 2026

For deep-dive technical validations, architectural review sessions, or deployment partnership inquiries, please contact the QuasarSDS Core Engineering Group.

✉ [engineering@quasarsds.io](mailto:engineering@quasarsds.io)