

REFERENCE ARCHITECTURE

WEKA NeuralMesh with Backblaze B2 as the data repository

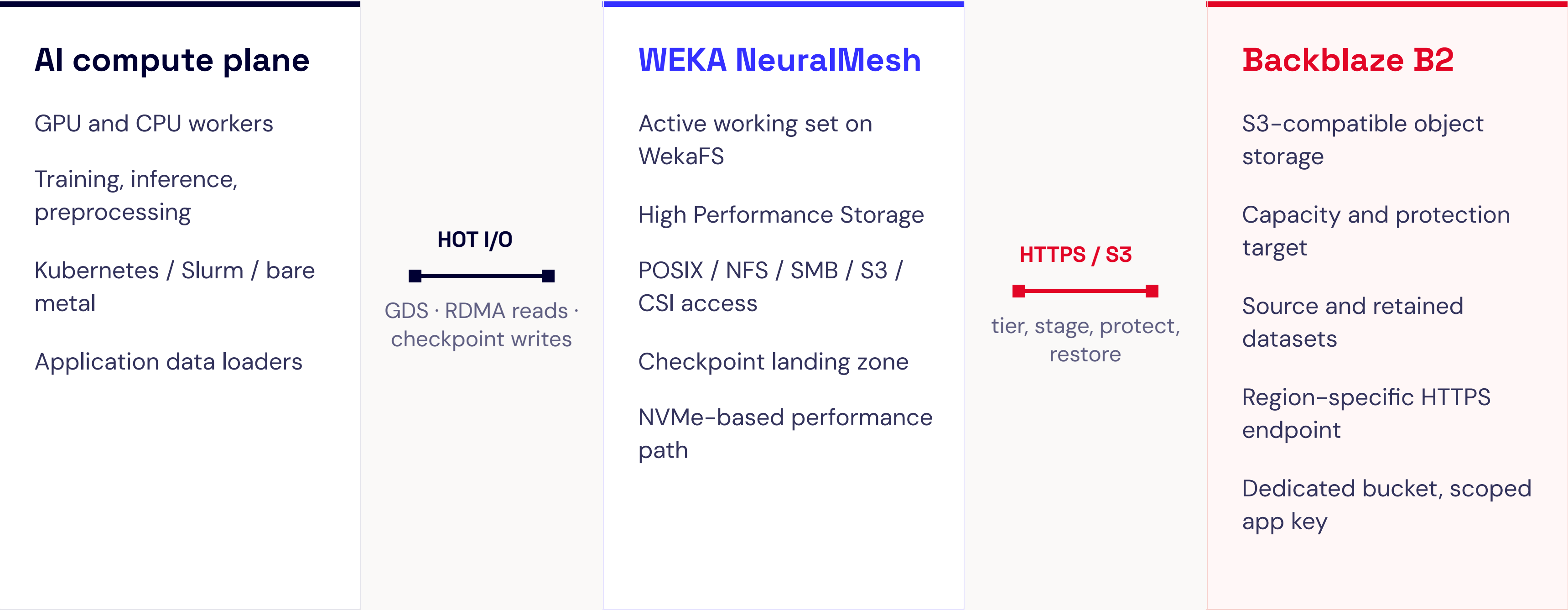
A high-performance active data tier next to the GPUs, with a durable S3-compatible capacity layer for retained datasets, protected checkpoints, and recoverable copies.



Backblaze B2 × WEKA NeuralMesh

Reference pattern

— Hot path — Capacity / protection — Control



Placement, policy, and observability

Dataset catalog and job scheduler · Lifecycle, version-expiry, and snapshot policy · Integrity verification · End-to-end metrics

Integration patterns

A1 — WEKA TIERING

Bucket as filesystem capacity

Cold extents live in the bucket in WEKA format; metadata stays on NVMe. The bucket is part of the live filesystem — not a copy.

Continuous delete-and-rewrite reclaims space — requires a version-expiry lifecycle rule; Object Lock is incompatible.

Select when capacity economics are the objective and the namespace should exceed NVMe by design.

A2 — SNAP-TO-OBJECT

Bucket as protection and mobility

Point-in-time snapshots — data plus metadata — independently restorable to another cluster, in any environment.

Namespace is usable on metadata restore, then rehydrates on access; retention lives in its own bucket and policy.

Select when filesystem-level protection, cross-cluster mobility, and namespace recovery are the objective.

B — EXTERNAL ORCHESTRATION

Bucket as a managed copy

An external mover copies data in a layout the customer chooses — the only pattern that yields application-readable objects.

Integrity, retries, and metadata fidelity are explicit in the mover; it competes with training for client resources.

Select when application-readable objects, portability, or per-dataset control are required.

A1 and A2 are different mechanisms, frequently deployed together. Objects written by the WEKA-native paths are internal format — not application-readable.

ONE REPOSITORY, ANY DEPLOYMENT

Deployment topology

Backblaze B2

Durable S3 object repository

Source and retained datasets

Checkpoints and snapshots

Dedicated bucket per deployment

Region-specific HTTPS endpoint

HTTPS
/ S3

HTTPS
/ S3

HTTPS
/ S3

NeoClouds

Dedicated NeuralMesh

NeuralMesh Axon

GPU clusters

Hyperscaler clouds

AWS · Azure · GCP ·
OCI

Dedicated NeuralMesh

NeuralMesh Axon

GPU instances

On-premises datacenters

Customer-owned ·
Colocation

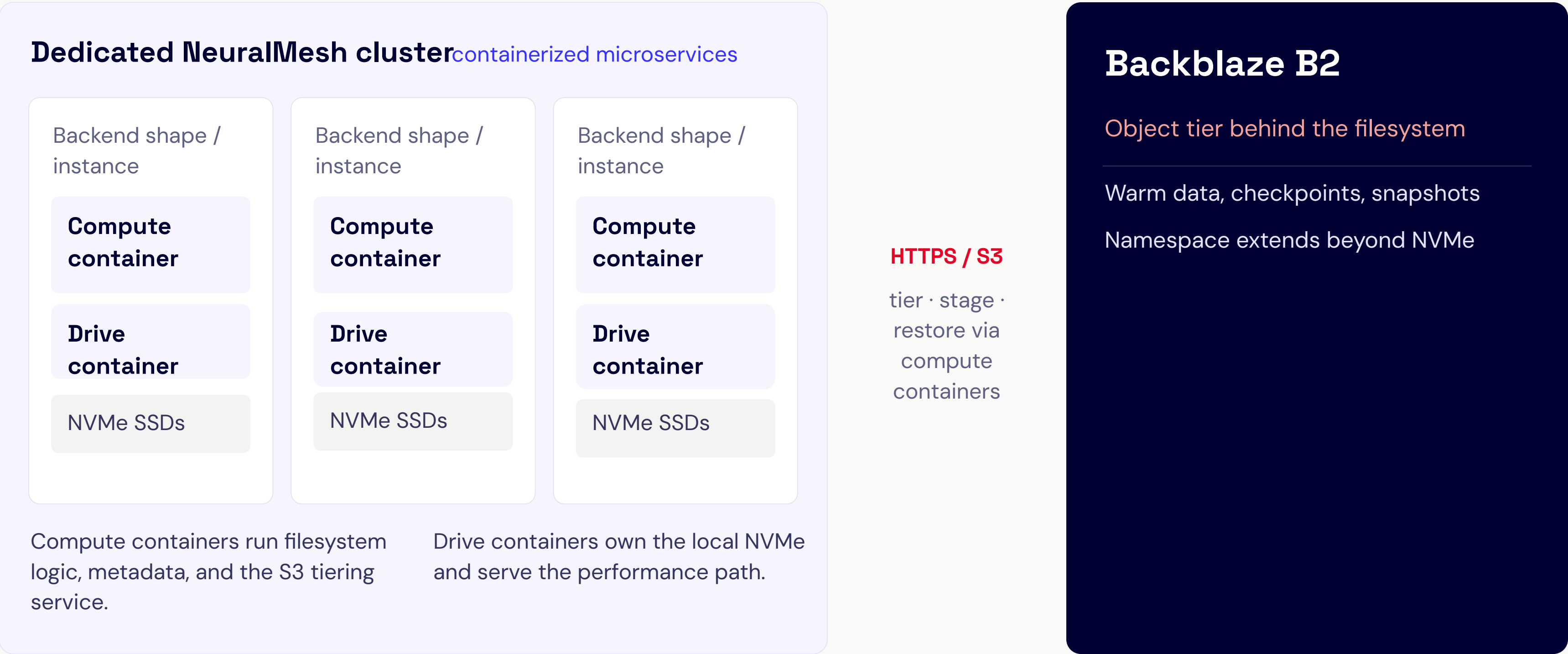
Dedicated NeuralMesh

NeuralMesh Axon

GPU servers

Compute follows GPU availability and price. Each environment runs a dedicated NeuralMesh cluster or converged NeuralMesh Axon — the same bucket hydrates either.

NeuralMesh backend topology



Every backend participates in tiering — parallel HTTPS/S3 streams to B2 scale with the cluster, not through a single gateway.

Bucket I/O mechanics

PUT — fixed 64 MiB objects

write path

4 KiB blocks → 1 MiB extents → **64 MiB object** → **PUT**

Objects are packed WekaFS extents — up to 128 extents per object, 64 MiB hard max.

Each compute process assembles and uploads its own objects; one object can hold data from many files.

Objects are opaque to S3 clients — reads and restores always go through NeuralMesh.

GET — parallel 1 MiB range reads

read path

64 MiB object → 1 MiB byte-range GETs × **64 threads / process**

Rehydration issues 1 MiB byte-range GETs against the 64 MiB objects, plus readahead.

Default 64 GET/PUT/DELETE threads per compute process — concurrency scales with the cluster.

Example: 8 backends × 30 compute processes × 64 threads = 15,360 potential parallel reads.

64 MiB static PUT object size

1 MiB parallel GET range size

64 threads per compute process

Size B2 for many small parallel range reads, not large sequential GETs — throughput scales with request concurrency.

CONVERGED ON THE GPU SERVERS

NeuralMesh Axon

Backblaze B2

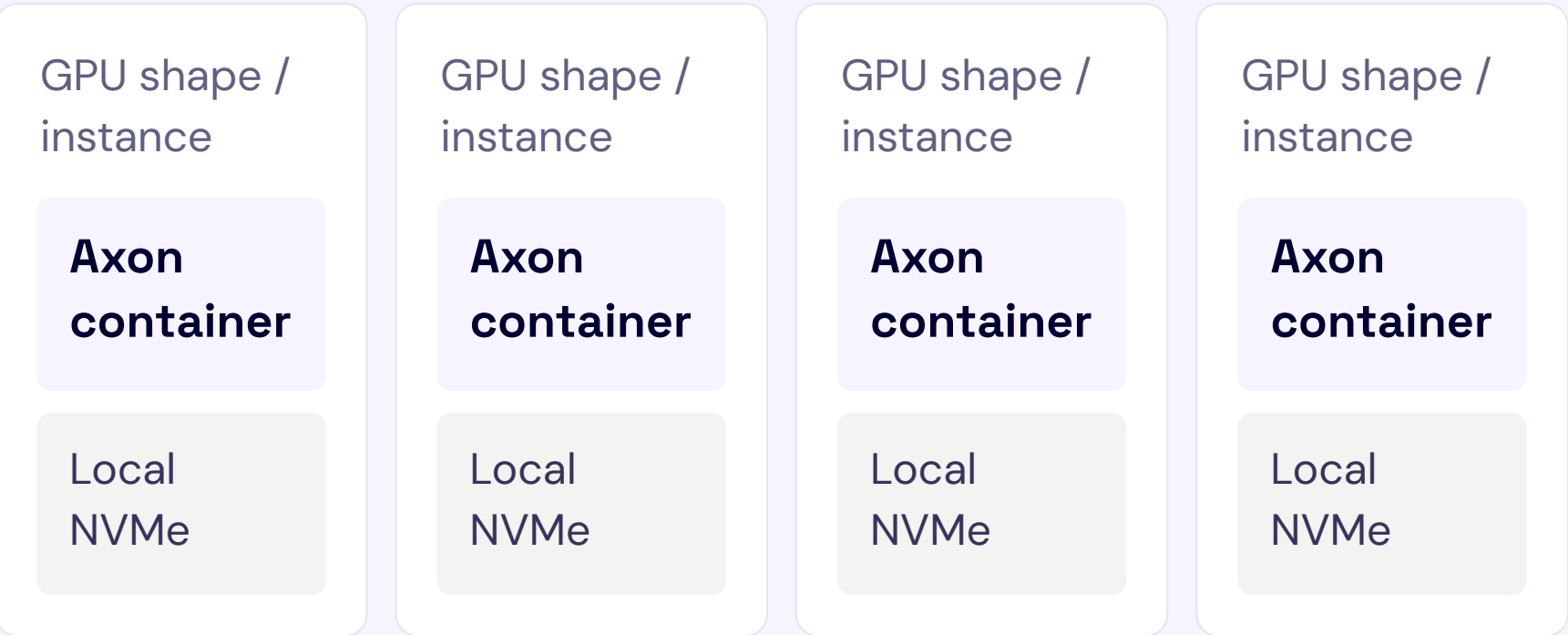
S3 bucket with training datasets

Checkpoint and snapshot target

Scoped application key

S3 LOAD
hydrate · tier
back

GPU cluster with embedded NeuralMesh Axon no external storage tier



Axon fuses each node's NVMe into one shared, POSIX-compliant namespace — GPUs read and checkpoint at local-NVMe speed while datasets load straight from the B2 bucket.

Data lifecycle

01 — LAND

Write through NeuralMesh

Clients write to the filesystem; NeuralMesh persists data to the B2 bucket — the system of record.

02 — STAGE

Hydrate NeuralMesh

The working set loads over HTTPS/S3 into the deployment running the job.

03 — RUN HOT

Train and infer

GPUs read and checkpoint on the NVMe path — GDS, RDMA, POSIX.

04 — PROTECT

Tier back to B2

Checkpoints and snapshots tier back; cold data reclaims NVMe capacity.

OPEN ITEMS

Integration pattern (A1 tiering, A2 Snap-To-Object, or B external mover), object layout, supported releases, and recovery semantics are open items pending WEKA's object store qualification of B2.



WEKA × BACKBLAZE · ARCHITECTURE