SNIA | Cloud Storage Technologies

Architecting AI Data Foundations: Object Storage Patterns for Scale, Access, and Longevity

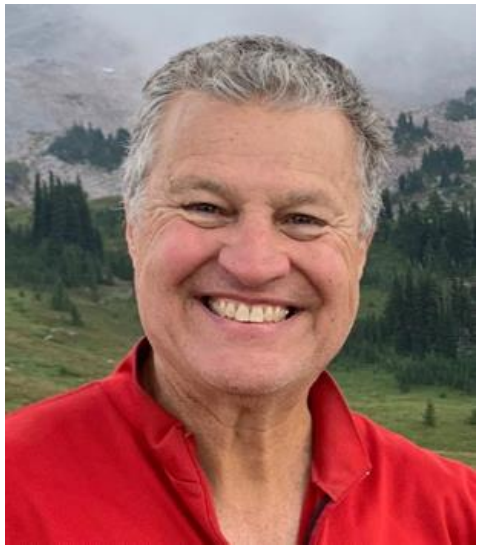
Live Webinar
July 29, 2026
10:00 am PT / 1:00 pm ET

Masood Noori Seagate

Kalyan Gunda Dell

Michael Hoard
SNIA CST

Today's Presenters



Michael Hoard
Moderator
SNIA CST Chair

Masood Noori
Staff Analyst, Cloud Marketing
Seagate

Kalyan Gunda
Chief Architect,
Dell Technologies

SNIA | CST
A SNIA Community

SNIA Cloud Object Storage Plugfests sponsored by CST

SNIA | Cloud Storage Technologies

https://snia.org/groups/cst

Committed to the adoption, growth and standardization of **intelligent data storage usage** in cloud infrastructures

- New simplified SNIA Membership Model
- Pay one fee for SNIA Membership – Participate in ANY group!
- Join SNIA groups at https://www.snia.org/member_com

The SNIA Community

200
industry leading organizations

2,000
active contributing members

50,000
IT end users & storage pros worldwide

SNIA | CST A SNIA Community

SNIA Legal Notice

- The material contained in this presentation is copyrighted by SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Agenda

01 **AI data representations**
How AI transforms enterprise data

02 **Object storage goes AI-native**
Iceberg, S3 Tables, Vectors & On-Prem

03 **Scaling the foundation**
Capacity, cost & the role of SMR

04 **Moving data efficiently**
Accelerated Object I/O, KV-cache offload & DPUs

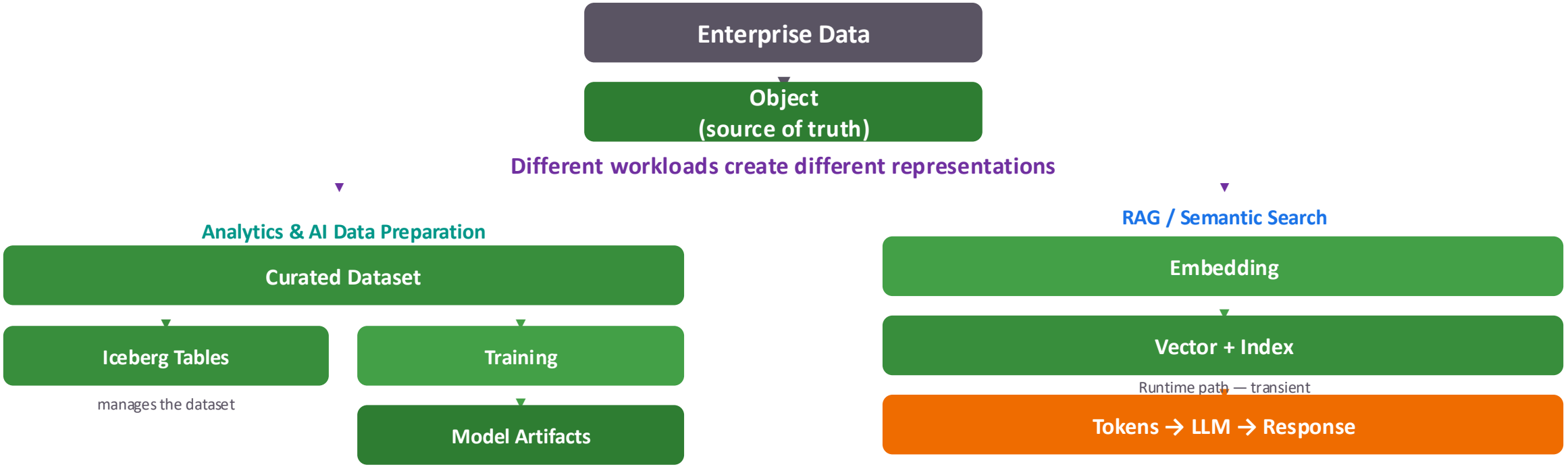
05 **Pipeline & industry direction**
The AI data pipeline & SNIA Storage.AI

Section 01

The AI Data Transformations/Representations

How Enterprise Data Becomes AI-Ready

Different workloads derive different persistent representations



Not every dataset follows every path. Object storage manages every persistent representation.

SNIA | CST A SNIA Community

Persistent Representations and Runtime State

Persistent

Object | Table | Embedding (stored as vector) | Model Artifacts

Runtime

Tokens | KV Cache

Representation	Optimized For
Object	Durability — the source of truth
Table	Reproducibility — governed, versioned datasets
Embedding	Retrieval — semantic search by meaning
Model Artifacts	Deployment — weights, config, tokenizer
Tokens	Execution — runtime input/output units; typically transient
KV Cache	Inference — reusable state; transient by default, optionally persisted

SNIA | CST
A SNIA Community

Tables & Vectors

Object Storage & AI Data Abstractions

Role of Tables in Analytics and AI

Tables turn raw enterprise data into reusable, governed datasets

Analytics and Data Engineering

- Query, filter, join, and aggregate structured data
- Combine transactions, telemetry, logs, business records
- Build reusable datasets for BI, reporting, and ops analysis

AI and ML Data Workflows

- Prepare and curate training and fine-tuning datasets
- Manage features, labels, evaluation results, monitoring data
- Associate metadata with source, ownership, and lineage

The role of tables

- Give data a schema and consistent structure
- Make datasets reusable across analytics, training, evaluation, and governance
- Provide a governed dataset layer above raw objects

Tables organize enterprise data into reusable, governed datasets for analytics and AI.

The open table format for modern analytics & AI datasets

Engine-agnostic	Spark, Trino, Flink, Athena, Presto, StarRocks — one format, every engine
ACID on object storage	**Atomic metadata commits** — every update either succeeds completely or not at all
Schema & partition evolution	Change columns or partitioning without rewriting terabytes of data
Time-travel	Query any historical snapshot — rollback bad writes, reproduce ML training runs
Open & vendor-neutral	Apache-governed, no lock-in — swap engines without migrating data

Anatomy of an Iceberg Table

Three layers — and only one remains an infrastructure responsibility

CATALOG

The coordination layer: resolves table names and commits the current metadata location

A small, atomic catalog update completes each commit.

▼

METADATA

Snapshots, manifest lists, manifest files (JSON + Avro)

Immutable files in object storage. Grows with every commit.

▼

TABLE DATA

Parquet data files — immutable, written once, never moved

Standard S3 objects. Already lives in your object store.

Data & metadata → solved — they live in object storage.
Catalog operations → the remaining infrastructure responsibility — requires separate infrastructure to run.

The Catalog Challenge

Infrastructure, operations, and scale — the hidden cost of self-managed Iceberg

Infrastructure

- Compute (K8s or VMs)
- Metastore (Hive / Nessie / Polaris)
- RDBMS for catalog state
- Network, security, IAM

Maintenance

- Snapshot expiration
- Orphan file cleanup
- Manifest compaction
- Data file compaction

Scale

- Connection exhaustion
- Commit contention
- Metadata explosion
- Multi-engine coordination

At scale, catalog operations and table maintenance often become major operational responsibilities.
What if the storage platform absorbed it?

Tables in Practice: Cloud & On-Prem

Industry pattern: embed the catalog into the storage platform

Cloud: S3 Table Buckets

- ✓ REST Catalog built into the storage platform
- ✓ No external metastore or RDBMS
- ✓ Automatic compaction & snapshot cleanup
- ✓ Multi-engine: Spark, Trino, Athena, EMR
- ⚠ AWS-only implementation — vendor lock-in

On-Prem: Your Infrastructure

- ✓ REST Catalog API embedded in object store
- ✓ Compatible interface with minimal application changes
- ✓ Background scanners for maintenance
- ✓ NVMe tier for high metadata IOPS
- ✓ Full data sovereignty — nothing leaves your DC

Same architectural pattern. Different infrastructure. Collapse catalog operations into the storage platform.

On-prem object storage already has strong consistency, conditional writes, NVMe, scanners, EC, and geo-rep. Just embed the catalog.

Tables Answer Precisely Structured Questions.

Vectors Answer Semantically Expressed Questions.

Precisely Structured

```
SELECT * FROM Orders
WHERE Customer = 123
```

Semantically Expressed

Find documents similar
to this error message

Traditional storage primarily managed placement and durability. AI storage must also manage representation and retrieval.

Why Vectors — and Where They Come From

Embedding models convert enterprise data into semantic representations that enable similarity search

RAG
Ground LLM answers in your data

Semantic Search
Find by meaning, not keywords

AI Agent Memory
Retain & search past interactions

↳ or via curated tables (optional)

Embeddings are derived representations; without the source object, they lose provenance and interpretability

Data gravity: the index should live where the objects already are

Embedding the vector index into object storage makes architectural sense — data and its semantic representation stay together.

S3 Vectors: Index Built into Storage

The same industry pattern applied to vector embeddings

The vector problem today

✗ Separate vector DB: Pinecone, Milvus, Weaviate, pgvector

✗ Memory-bound: HNSW requires massive RAM as indexes grow

✗ Infra cost can exceed LLM inference cost itself

✗ Operational complexity: keeping embeddings fresh as source data changes

S3 Vectors eliminates it

✓ Native vector search built into the object store

✓ Billions of vectors per index, sub-second queries

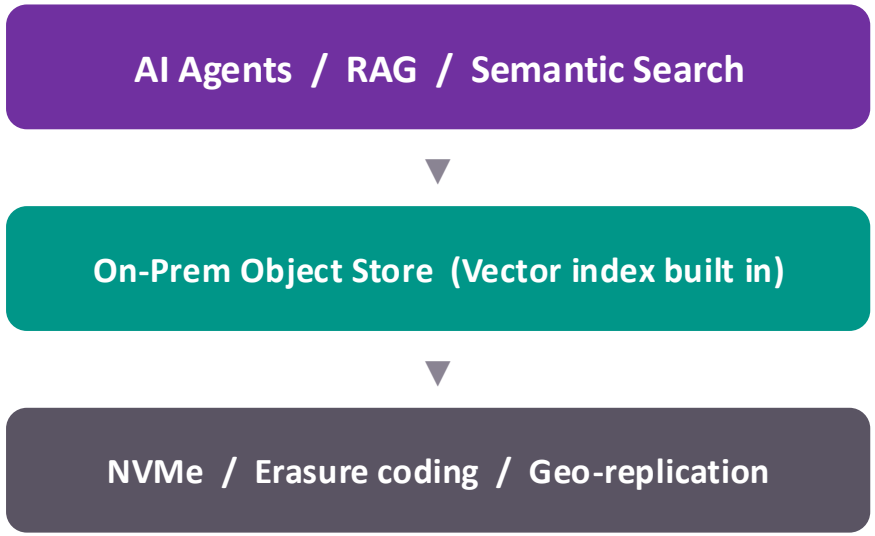
✓ Significant cost reduction vs dedicated vector DBs

✓ LangChain connector works out of the box (RAG pipelines)

Data gravity: embeddings are derived from objects already in your storage — the index should live with the data.

On-Prem Vectors: Your Data, Your Index

The same architectural pattern applies on-prem



Why this works

✓ DiskANN on NVMe: PQ codes in RAM, full vectors on SSD

✓ Compatible S3 Vectors API pattern — preserves the LangChain integration model

✓ NVMe-oF disaggregation: scale storage independently

✓ Background optimizer keeps index fresh as data changes

✓ Embeddings never leave your data center

Developer story: LangChain → S3 Vectors API → On-Prem Object Store. Compatible API preserves the same application pattern, your infrastructure.

What Must Storage Do Natively?

Capabilities required to support tables, semantic indexes, and model artifacts

For Iceberg Tables

Atomic CAS:	Compare-and-swap for metadata pointer commits
Strong consistency:	Immediate read-after-write for metadata
REST Catalog API:	Native HTTP endpoint, Iceberg REST spec
High metadata IOPS:	NVMe-backed, low-latency small I/O
Auto-maintenance:	Compaction + snapshot cleanup scanners

For Vector Embeddings

Vector APIs:	Put, Query, Get, Delete vectors natively
Index engine:	DiskANN (NVMe) or HNSW (in-memory)
Similarity search:	Cosine, Euclidean, dot product built-in
Tiered storage:	DRAM (hot) → NVMe (warm) → Hard Drive (cold)
Quantization:	768-dim (3 KB) → 32 bytes via PQ in RAM

For Model Artifacts

Large sequential reads: Sustain high-throughput model loading into accelerated compute tiers

Checkpoint writes: Sequential, high-throughput, versioned

Artifact versioning: Track weights, config, tokenizer together

Architecture Comparison

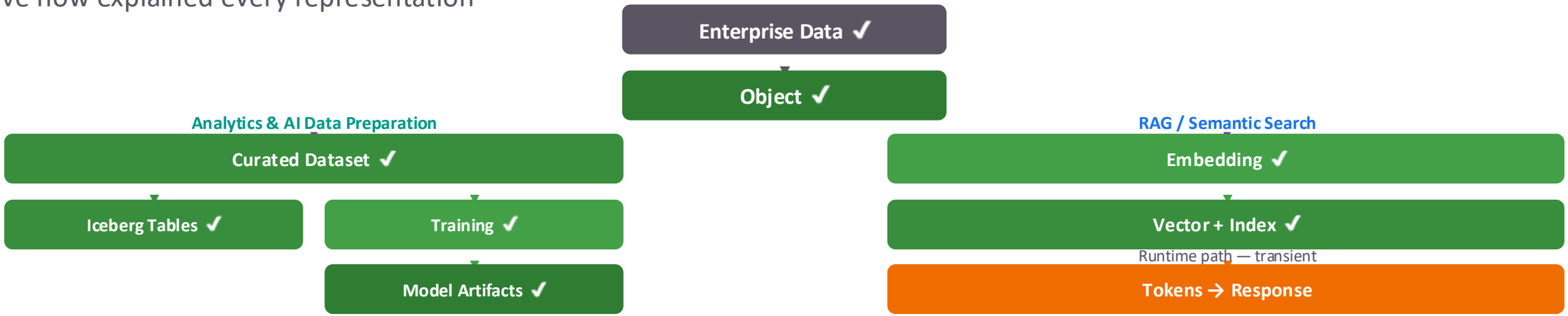
Self-managed vs cloud-native vs on-prem embedded

	Self-Managed	AWS (S3 Tables + S3 Vectors)	On-Prem (Built-in Catalog + Index)
Table Catalog	External (Polaris + Postgres + K8s)	Built into Table Bucket	Built into storage platform
Vector Index	External (Milvus / Pinecone)	Built into Vector Bucket	Built into storage (DiskANN on NVMe)
Extra Infrastructure	K8s + HA DB + Vector DB cluster	Minimal customer-managed	Reduced operational overhead
Maintenance	Manual Spark jobs + index rebuilds	Fully automated by AWS	Automated via scanner framework
Vendor Lock-in	None	Platform-specific	Open-interface portability
Data Sovereignty	Depends on deployment	AWS regions only	Your data center

On-prem object storage with built-in catalog and vector index delivers the S3 Tables + S3 Vectors experience — on your hardware, with your data, under your control.

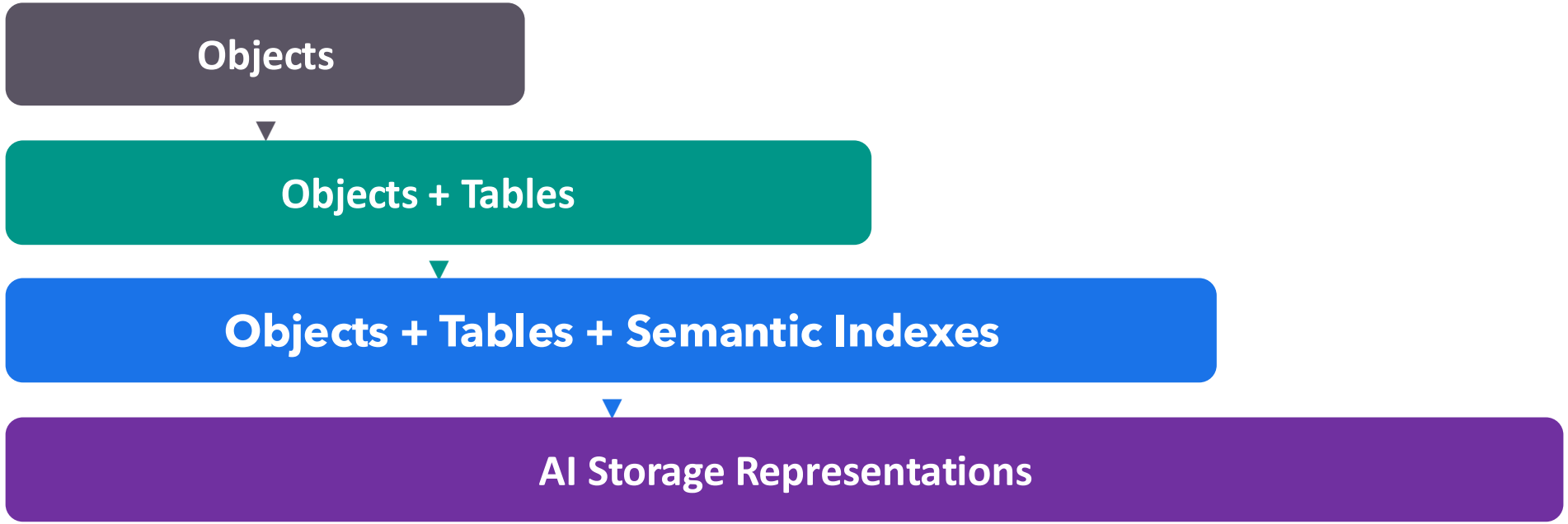
Remember This Picture?

We've now explained every representation



Object Storage Evolution

Managing multiple representations of enterprise knowledge



Object storage is evolving from managing objects alone to managing multiple AI data representations.

Section 02

Architecting Durable AI Data Foundation

Placement, lifecycle, movement, and economics across tiers

Architecting the AI Data Foundation

Where does the data live once the object store becomes the application layer?

KALYAN

Object goes AI-native

Tables · Vectors · Metadata · S3 APIs

- Application-facing data representations
- Queryable object layer
- Cloud + on-prem patterns

MASOOD

The architecture lens

acceleration + durable capacity

- SSD / NVMe where latency binds
- Hard drive / object where capacity scales
- Policy + lifecycle make data reusable

OUTCOME

AI-ready object storage pattern

- Data use maps to media tier
- Object policy drives lifecycle
- Standards keep the path open

Best-practice architecture — not product positioning. Build once. Reuse often. Retain economically.

AI is generating more data than ever – **and it all needs to be fast and cheap and reusable at the same time.**
Today you must pick two

Policy & Metadata Become Architecture

For AI, object storage isn't just capacity — it's the control surface for the data lifecycle.

Notifications

Trigger ingest, prep, retrain and downstream steps when objects land or change.

Metadata & Tags

Label datasets, lineage, model versions, embeddings, governance and ownership.

Versioning

Preserve dataset iterations, processing outputs, checkpoints, prompts and artifacts.

Lifecycle

Move data based on reuse, retention and cost.

Replication

Place data where compute, DR, sovereignty or collaboration require it.

Encryption

Keep policy and keys aligned to application, tenant or trust domain.

Design rule — treat policy, metadata, events and lifecycle as architecture requirements — not optional features.

Place Data by Workload

Latency Tier vs. Capacity Tier — start with the workload, not the media.

LATENCY TIER — latency / throughput binds

SSD / NVMe / accelerated object

- Model load & fast startup
- Training feed — keep the GPUs fed
- Vector search & RAG index traversal
- Latency-sensitive KV-cache restore

Goal: keep compute busy

CAPACITY TIER — durability / cost binds

Hard-drive-backed object storage

- Raw & prepared datasets
- Long-lived checkpoints & model artifacts
- RAG corpora & historical embeddings
- Logs, outputs, compliance & archive

Goal: retain & reuse economically

Placement principle — SSD for the latency path; hard drive / object for durable scale.

The Durable System of Record

Hard-drive-backed object is the persistent foundation behind accelerated AI pipelines.

Datasets

Raw, curated, versioned and retained data.

Checkpoints

Sequential writes, model history, restart points.

RAG Corpora

Documents, embeddings, context and governance.

Archive

Long-lived, immutable, cost-optimized retention.

System of Record · Hard Drive / SMR / lifecycle-managed tiers

Design goal — make drives invisible — expose durable capacity through object semantics and lifecycle policy.

Reusable AI State Is Persistent Data

AI creates reusable state; preserving it saves recompute but grows persistent data.

CREATE	REUSE	PERSIST	RETAIN
Tokens, embeddings, features, outputs	RAG, restart, prefill, fine-tune	Checkpoints, KV state, corpora	Governance, IP, audit, retrain

KV cache is one example, not the only one — once state has value, it becomes durable data.

Capacity driver — a fast growing consumer isn't raw data — it's reusable AI state.

SMR: More Capacity per Drive

Shingled Magnetic Recording overlaps write tracks to pack more TB into every drive and rack slot.

>50%

of nearline hard-drive exabytes projected SMR by 2028

+10–20%

areal density vs. CMR — more capacity per drive & per slot

What SMR gives you

- Earliest access to highest-capacity platforms
- Sequential, zoned (host-managed) writes
- More TB per drive and per rack slot

Why it fits AI

- Ingest, checkpoints, corpora & cold KV are append-friendly
- Enabled in Linux today: dm-zoned, f2fs / btrfs / zonefs
- Ideal for the warm / cold tiers of the AI data lake

SMR adoption projection: IDC. Architecture concepts are industry-standard (INCITS T10/T13 ZBC/ZAC).

SMR for Append-Friendly AI Data

SMR is strongest when objects are large, sequential, immutable and lifecycle-managed.

GOOD FIT — APPEND & SEQUENTIAL

- Ingest objects — large, immutable versions
- Checkpoints — sequential dump + retained history
- RAG corpora — bulk ingest, mostly read / reuse
- Cold KV & model state — lifecycle-managed
- Archive & compliance — long retention

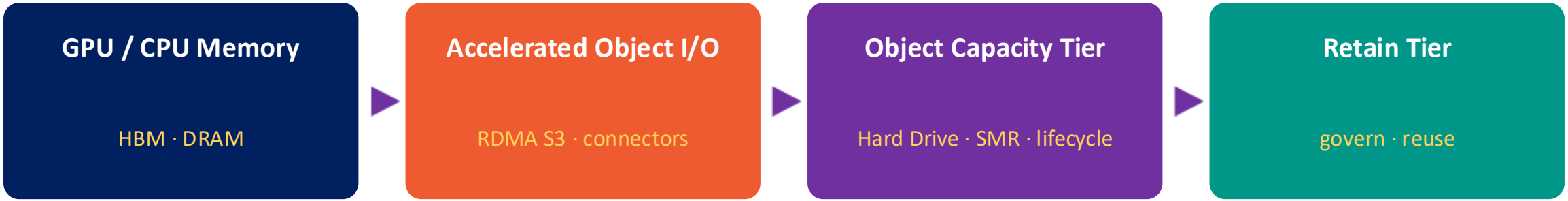
SMR DESIGN CHECKLIST

- Use immutability + versions, not in-place mutation
- Batch writes into large objects or manifests
- Let lifecycle move colder data to denser tiers
- Keep heavily used metadata / index paths on SSD
- Expose simple S3 semantics to applications

SMR best practice — map media to write pattern and lifecycle. Design for zones, not random overwrite.

Design for Data Movement

As accelerated object I/O improves, the persistent tier behind it matters more.



Performance

Can the path feed compute without starving GPUs?

Capacity

Can the foundation sustain persistent growth economically?

Standards

Can the access layer stay interoperable & vendor-neutral?

Movement principle — faster access increases the value of durable capacity.

The Mixed-Fleet Object Architecture

Expose object semantics on top; place each data class on the right tier below.

AI Applications — Training · Inference · RAG · Feature engineering · Observability

Object / Data Access Layer — S3 APIs · tables · vectors · metadata · events · lifecycle policy

Acceleration Tier
NVMe SSD · cache · prefetch · high-throughput staging

Durable Capacity Tier
Hard Drive / SMR object · datasets · checkpoints · corpora · archive

Operations & Trust — monitoring · placement · replication · encryption · attestation-ready data path

Object semantics above. Workload-aware placement below. Developers never care where the bytes live.

Design Rules for AI-Ready Storage

Place data by latency, reuse, lifecycle and economics.

01	**Start with the latency budget**	Place media by latency requirement, not by habit.
02	**Treat metadata & policy as architecture**	Events, tags, versioning, lifecycle and retention make AI data usable.
03	**Use SSD for acceleration, not everything**	Keep GPUs fed without making every retained byte all-flash.
04	**Use Hard Drive / SMR for durable scale**	Datasets, checkpoints, corpora, outputs and archive need density.
05	**Design for reuse, trust & standards**	Reusable state, secure paths and open interfaces make it durable.

Bottom line — hard drives aren't outside AI — they're the durable capacity layer that lets AI data compound over time.

Section 03

AI Data Infrastructure in Practice

The All-Flash Cost Wall

AI data is exploding — but flash economics can't scale to hold all of it.

~12×

Enterprise SSD $/TB vs. nearline hard drive — widening as NAND shifts to HBM

~40 PB

KV cache for one 70B model at 1M sessions × 128K context

<0.27 W/TB

Hard-drive power for warm / cold tiers at exabyte scale

THE DEMAND

- Training sets, checkpoints & retrain loops
- KV cache — the fastest-growing consumer
- Vector indexes & long-lived RAG corpora
- Sovereign / retained data for continuous learning

THE CONSTRAINT

- All-flash can't scale to this economically
- NAND shortage: capacity prioritized for HBM / AI
- Power & cooling limits in the datacenter
- Answer: tier Hard Drive + SSD — not flash-only

Sources: VDURA / Blocks & Files price tracking (2026); SNIA SDC StorageAI 2026; TrendForce NAND allocation.

Accelerated Object Storage & KV-Cache Offload

Keep GPUs busy — move data faster, and reuse context instead of recomputing it.

Accelerated Object Storage

- Zero-copy data movement straight into GPU / CPU memory
- Bypasses the HTTP / TCP stack; RoCEv2 on existing Ethernet
- SNIA Accelerated Object I/O TWG is building an interoperable RDMA-S3 spec

KV-CACHE OFFLOADING

- Reuse attention state instead of recomputing it on the GPU
- Tier the cache: GPU HBM → CPU DRAM → SSD → object store
- Orchestrated by NVIDIA Dynamo / NIXL across HBM, flash & object

PROOF POINTS

- Dell ObjectScale + RDMA S3 — **≈2× token throughput** vs. GPU-cache-only
- NVMe KV offload — **up to ~10× more concurrent users per GPU**
- VAST + NVIDIA ICMSP — **~10× faster prefill**

Sources: SNIA SDC StorageAI 2026 (Dell, HPE); NVIDIA Dynamo docs; NVIDIA ICMSP (CES 2026); VAST 2026.

The AI Data Pipeline

Every stage has a distinct storage signature — from cost-optimized capacity to GPU-fed throughput.

Ingest	Preprocess	Train	Inference	Archive
MEDIA **Object / Hard Drive**	MEDIA **SSD / NVMe**	MEDIA **pNFS + Hard Drive**	MEDIA **Vector + Object**	MEDIA **Hard Drive / Deep obj**
ACCESS **S3, REST**	ACCESS **NVMe-oF**	ACCESS **pNFS**	ACCESS **RDMA, gRPC**	ACCESS **S3, POSIX**
VALUE **Cost + scale**	VALUE **Throughput**	VALUE **Parallel GPU feed**	VALUE **RAG + KV cache**	VALUE **Retention**

Checkpointing is exploding — bigger models, higher frequency, tighter retrain loops → sequential writes + durable retention.

SNIA | CST A SNIA Community

Key Takeaways

01 **AI is storage-bound**

AI creates persistent representations and reusable runtime state that reshape storage requirements.

02 **Object storage is going AI-native**

S3 Tables, Vectors & Metadata turn the bucket into an active AI data layer.

03 **All-flash can't scale alone**

Tiered Hard Drive + SSD, with SMR unlocking capacity, is the economic foundation.

04 **Move data, don't just store it**

Accelerated Object I/O, KV-cache offload & DPUs keep GPUs busy without all-flash cost.

05 **Trust & standards make it durable**

Confidential computing + SNIA Storage.AI keep the foundation secure and interoperable.

Q&A

Thanks for Viewing this Webinar

- Please rate this webinar and provide us with feedback
- This webinar and a copy of the slides are available at the SNIA Educational Library
- A Q&A blog from this webinar will be posted at snia.org/blog
- Follow us on LinkedIn and X for future webinar dates

THANK YOU!

