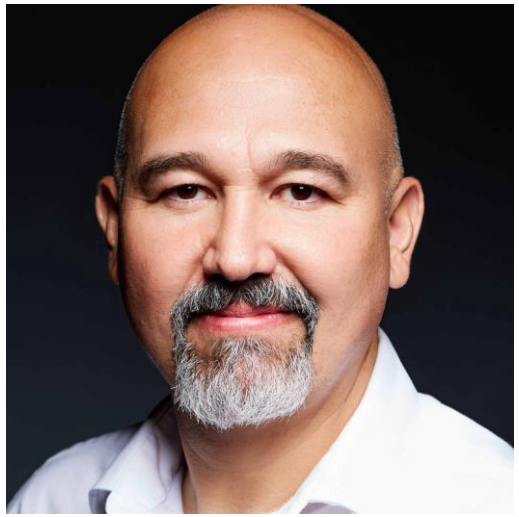
SNIA | Data, Storage & Networking

Deconstructing the Storage Software Stack: From Legacy Silos to AI-Scale Architecture

Live Webinar
May 14, 2026
10:00 am PT / 1:00 pm ET

Today's Presenters



Luis Freeman
Power Systems HDD TCEM & Project
IBM Systems
Moderator

Rohan Mehta
Member of Technical Staff - Systems Performance Architect
Micron Technology

Jayanthi Ramakalanjiyam
Software Engineering Leader
Celestica

The SNIA Community

200
industry leading organizations

2,000
active contributing members

50,000
IT end users & storage pros worldwide

SNIA | DSN A SNIA Community

SNIA | Data, Storage & Networking

What We Do

Drive the awareness and adoption of a broad set of technologies, including:

- Storage Protocols (Block, File, Object)
- Traditional and software-defined storage
- Disaggregated, virtualized and hyperconverged
- AI, including storage and networking considerations
- Edge implementation opportunities and factors
- Storage and networking security
- Acceleration and offloads
- Programming frameworks
- Sustainability

How We Do It

By delivering:

- Expert webinars and podcasts
- White papers
- Articles in trade journals
- Blogs
- Social Media
- Presentations at industry events

Logistics

- The slides are available under the attachments tab at the bottom of your console.
- Questions are welcome!
- Please rate the session and provide feedback!
- Want more sessions like this or other topics, let us know!
  - JOIN US! We meet on Thursday mornings at 11:00 AM eastern.
  - Email dsn-chair@snia.com if you have questions.

SNIA Legal Notice

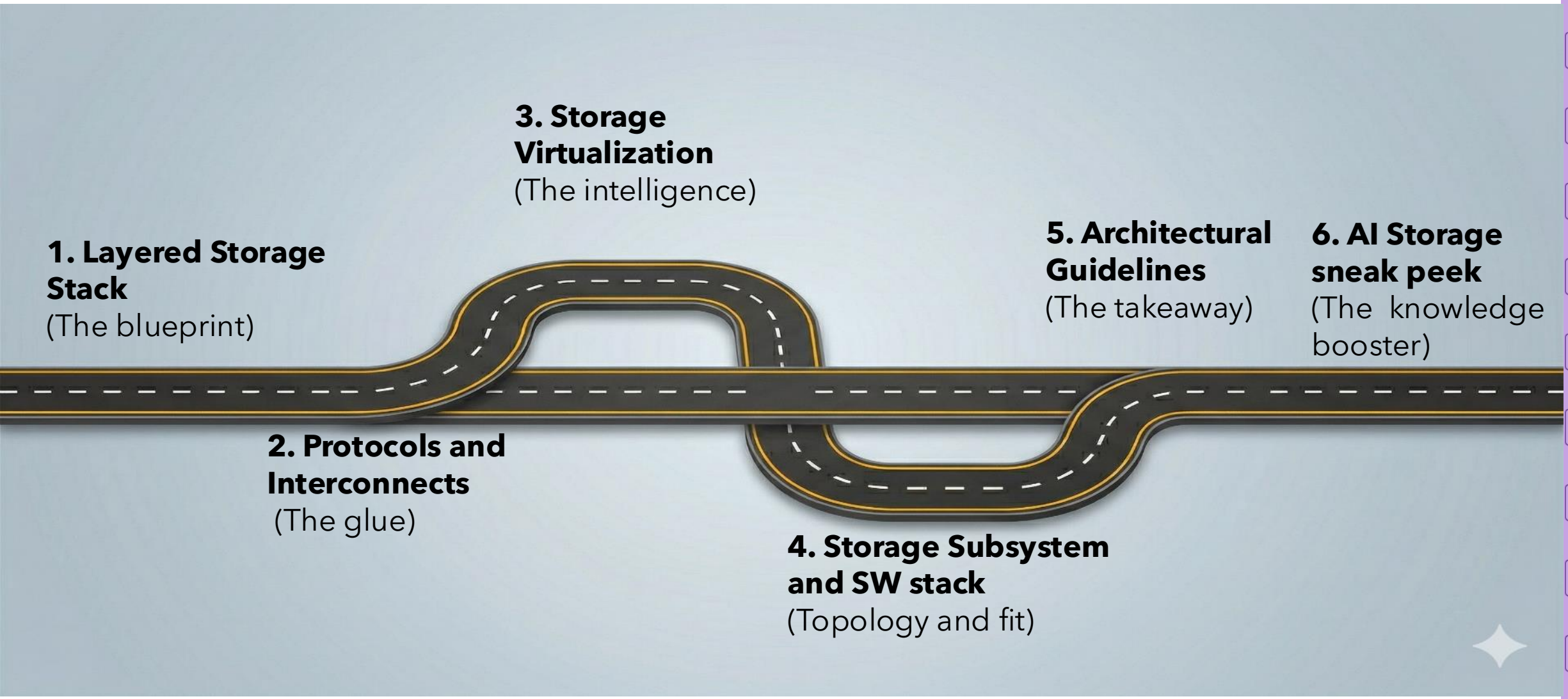
- The material contained in this presentation is copyrighted by SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - 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.

Disclaimer

- **Terminology:** “SW” encompasses both software and non-SSD firmware.
- **Context:** The stacks shown in this deck are simplified representations; real-world implementations involve specific technical nuances.
- **Focus :** This webinar focuses on the SW aspects not getting into the details of HW

Our Journey For the Next 50 Minutes…



SNIA DSN A SNIA Community

Quick Poll

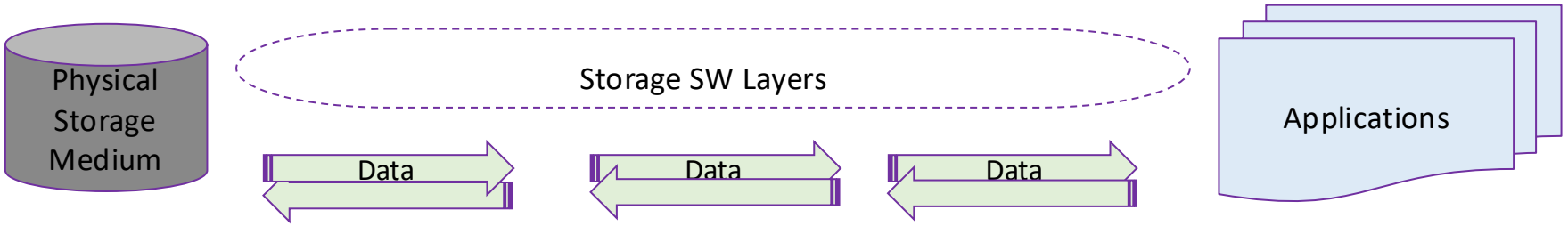
1. How familiar are you with this topic?
   A. I’m new at this
   B. I'm working with storage software architecture, but looking forward to learning more
   C. I live with this day and night

SNIA | DSN
A SNIA Community

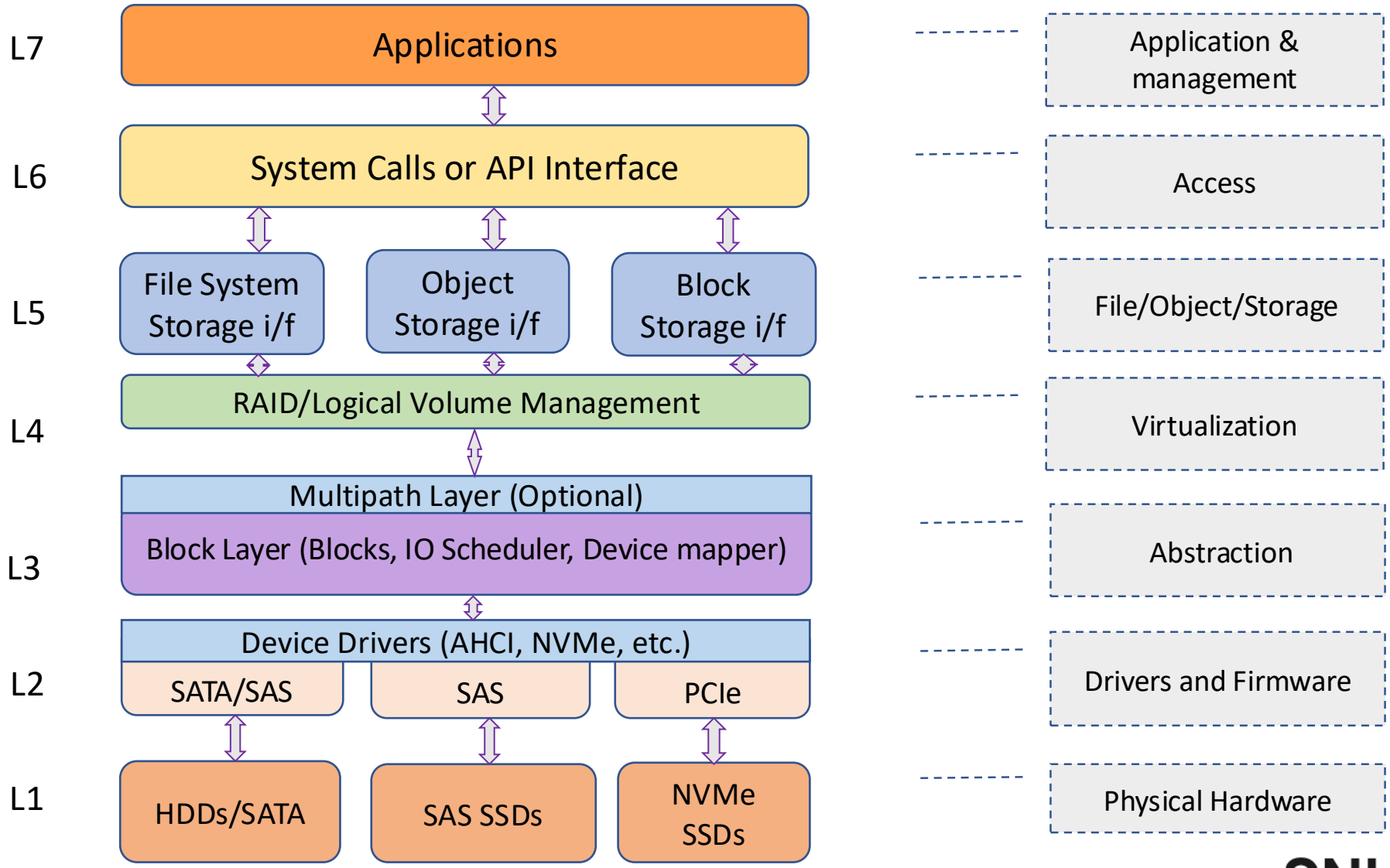
Layered Storage Stack

Storage Software Stack Layers - Simplified

- Layer 7 - Applications and Management
- Layer 6 - Access Protocols (Native vs. Over the network)
- Layer 5 - File System/Object/Block Storage
- Layer 4 - Virtualization: Scale, Security, Elasticity
- Layer 3 - Abstraction: OS layer
- Layer 2 - Drivers, Firmware
- Layer 1 - Physical Hardware



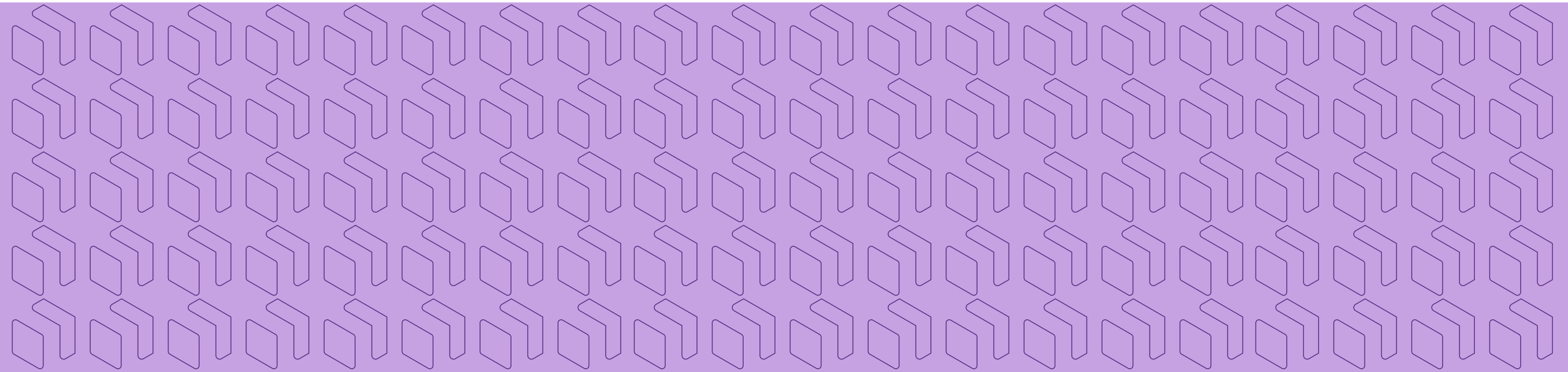
Standalone Server - Software Stack



SNIA DSN A SNIA Community

SNIA | DSN
A SNIA Community

Protocols & Interconnects



Storage Protocols and Networking Interconnects

Storage Types

Block Storage	File Storage	Object Storage
• FC, iSCSI, FCoE, NVMe, NVMe-oF	• NFS, SMB (CIFS), pNFS	• API Interface

Memory Expansion

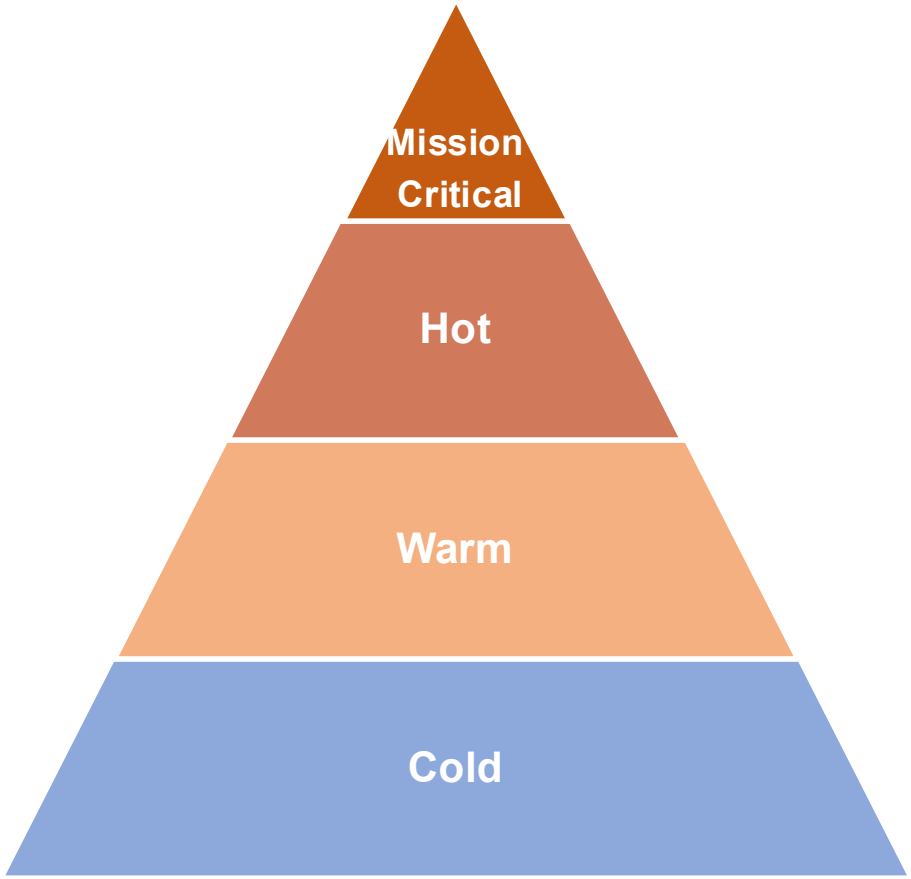
Local	Remote

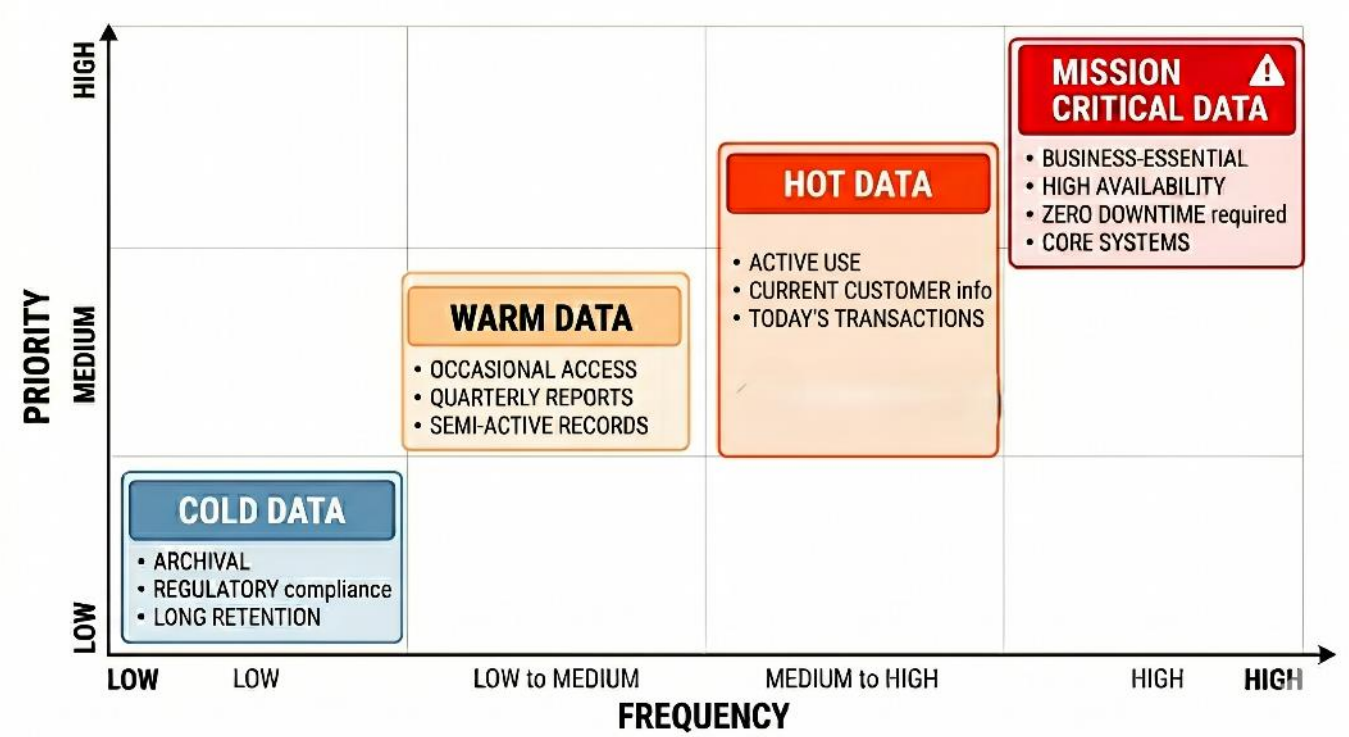
• CXL (becoming popular) for increased system memory	• RDMA
	  • RoCEv2
	  • iWARP

Storage Location

Local Access	Remote Access Interconnects
• PCIe/SAS protocols/ Links	• FC HBA, SAS HBA, iSCSI HBA, etc. (Traditional) • Ethernet NIC (with RoCE - RDMA over Converged Ethernet capabilities) • InfiniBand • Omni-Path (for High Performance Computing (HPC)) • UALink (Ultra Accelerator Link for AI accelerators)

Types of Data





Note: Type of data is the deciding factor to choose the physical storage medium, networking interconnects and the access protocols

SNIA | DSN
A SNIA Community

Virtualization

Why is Resource Virtualization Important?

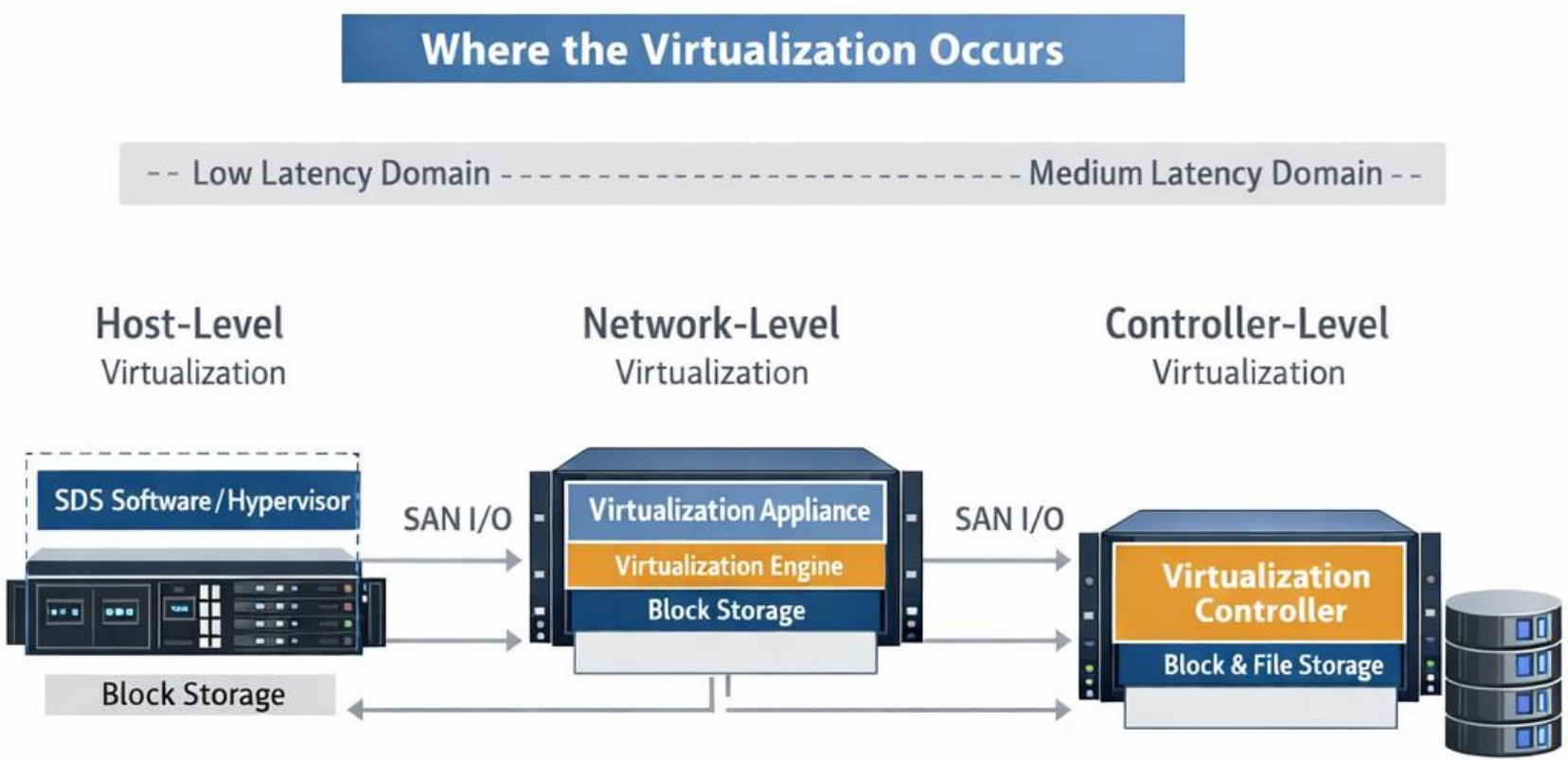
In the storage context, Virtualization

- Breaks the rigid link between where the data is stored, how it is accessed and managed
- Improves the efficiency, Agility, Resilience and cost benefits

Storage Virtualization

- Advancements in Software Defined Storage (SDS) solutions and Networking interconnects help to decentralize the functionalities.
- When performance requirements are met, physical data location can become transparent to the application.
- In the SDS context, “virtualization techniques” can be viewed along two dimensions:
  - **Implementation based virtualization -** host level, within a storage controller (array), or in the network
  - **Access based virtualization -** block-level, file-level, or object-based storage

Implementation-based Virtualization



Notes: Storage HW infrastructure/ Architecture is abstracted and virtualized

Implementation-based Virtualization - Host Level

- The virtualization logic is handled by software running on a server (the “host”).
- Aggregates diverse physical storage like, internal drives, DAS, or networked LUNs, into a unified virtual pool presented to applications or VMs.
- Key Characteristics
  - **Resource Usage:** Consumes host CPU and memory cycles.
  - **Management:** Decentralized; each host manages its own stack.
  - **Reduced latency**
- Primary Use Case: The foundation of Hyper-Converged Infrastructure (HCI) and cloud-native storage.

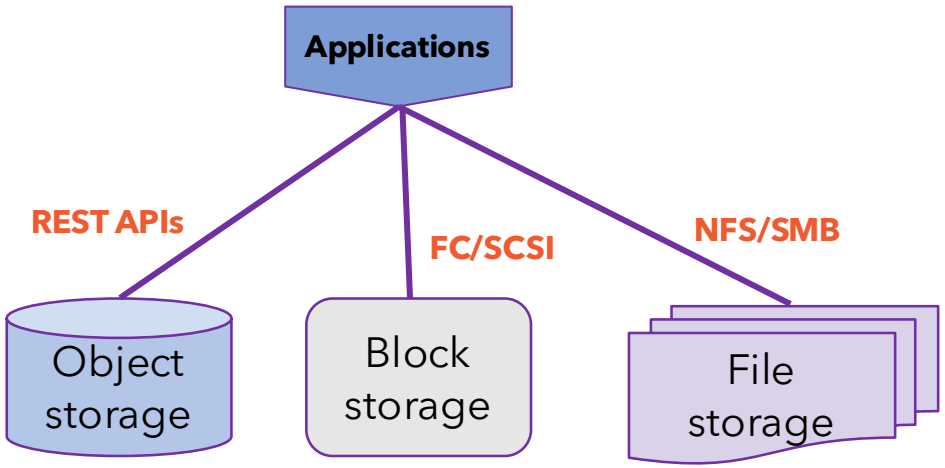
Implementation-based Virtualization - Controller Level (Array based)

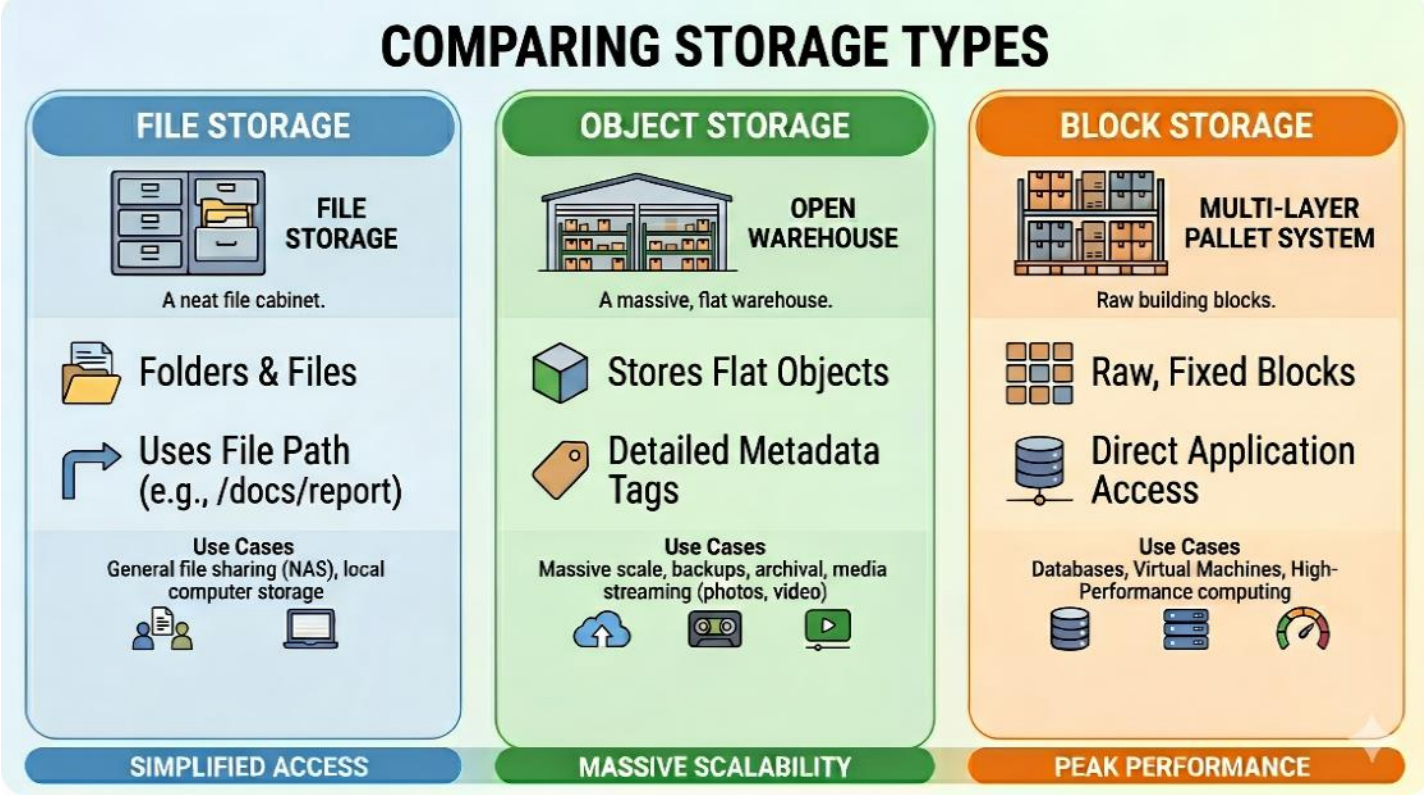
- The primary controller
  - Pools its own storage capacity with other connected arrays, often from multiple vendors.
  - Presents this aggregated capacity as a unified pool to all connected servers.
- Key Characteristics
  - **Centralized Pooling:** Consolidates multiple physical arrays into a single manageable resource.
  - **Vendor Agnostic:** Simplifies heterogeneous environments by masking hardware differences.
  - **Advanced Data Services:** Automated Tiering (SSD/HDD) and replication offloaded to the controller.
  - **Abstracted Complexity:** Servers see simple volumes; the controller handles the underlying “nitty-gritties”.
- Common Use Case
  - Enterprise SAN Environments: Critical for large-scale data centers requiring high availability and performance.

Implementation-based Virtualization - Network Level

- The most prevalent model, where intelligence resides in a switch positioned between servers (Initiators) and storage (Targets).
- This device intercepts and redirects I/O requests, pools disparate storage into a single virtual layer, abstracts the entire physical architecture from the host.
  - **Symmetric Virtualization:** The appliance sits directly in the data path.
  - **Asymmetric Virtualization:** The appliance manages the control path, but data moves directly between host and storage.
- Key Characteristics
  - **Thin Provisioning:** Just-in-time capacity allocation to reduce waste.
  - **Data Protection:** Native support for Snapshots, Clones, and Remote Replication for Disaster Recovery (DR).
  - **Automated Tiering:** Intelligent movement of “hot” data to SSDs and “cold” data to HDDs.
  - **Fabric-Wide Management:** Simplifies scaling across massive SAN environments.
- Common Use Case
  - High-Performance Workloads: Ideal for AI/ML pipelines that require massive, unified data pools and high-speed interconnects.

Access-based Virtualization





Notes: Access methods of different storage types of are abstracted and virtualized

Quick Poll

2. Is this presentation meeting your expectations so far?

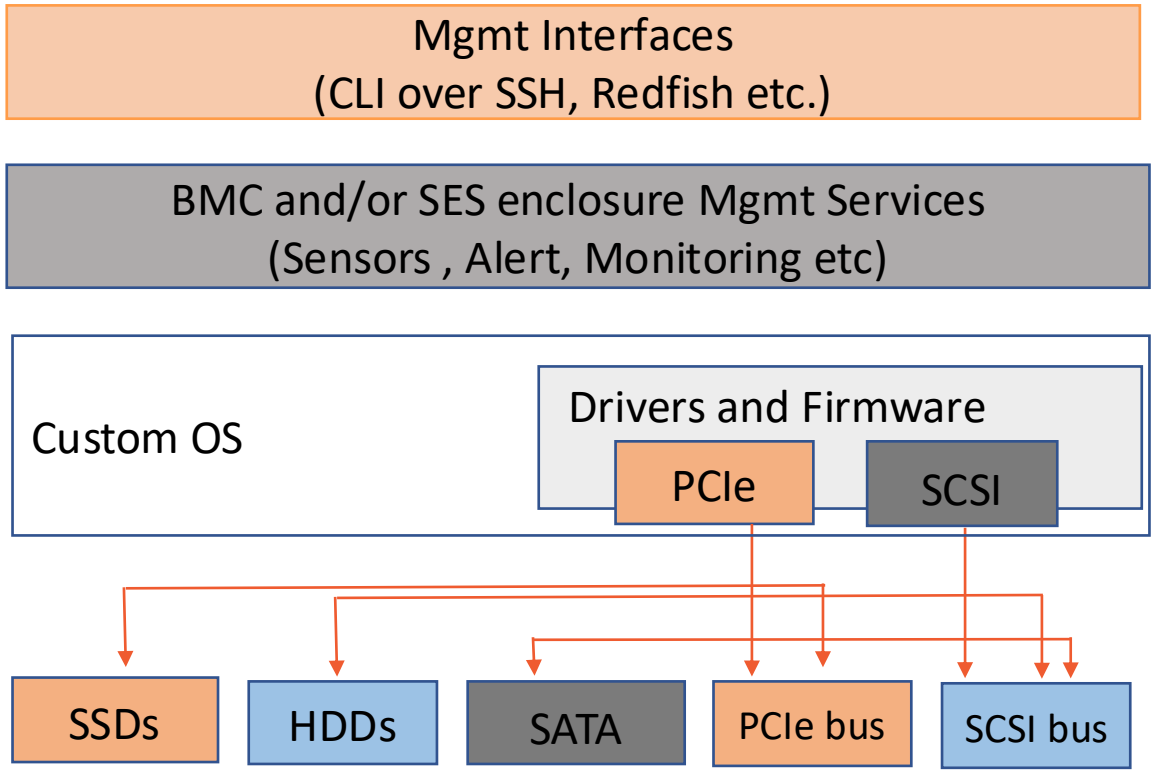
A. Yes

B. Somewhat

C. No

Putting it all together into the storage stack…

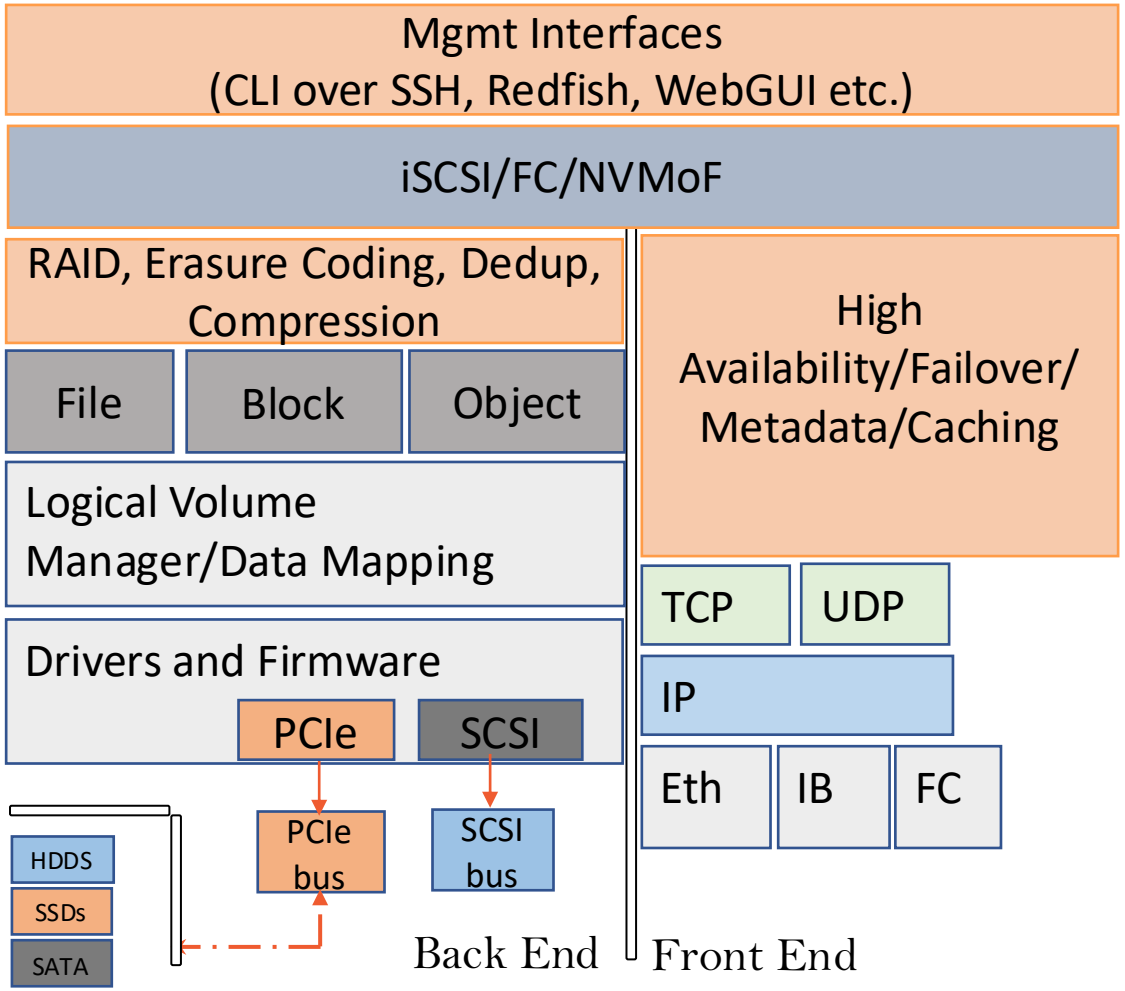
Storage Drive Enclosure



SW Stack Functionalities

- **Standard Formats:** Defined as **JBOD** (Disks) or **JBOF** (Flash); supports hybrid configurations.
- **Optimized Software:** Software stacks tailored to the specific storage medium (HDD vs. SSD).
- **Connectivity:** Utilizes **PCIe** and **SCSI** protocols for high-speed drive access.
- **Management:** Dedicated processor for **BMC** (Baseboard Management Controller) operations.
- **Fabric Integration:** Scalable to **"Over Fabric"** architectures via DPUs and specialized software.
- **Enterprise Scale:** Multiple units aggregate to form massive **Storage Arrays**.

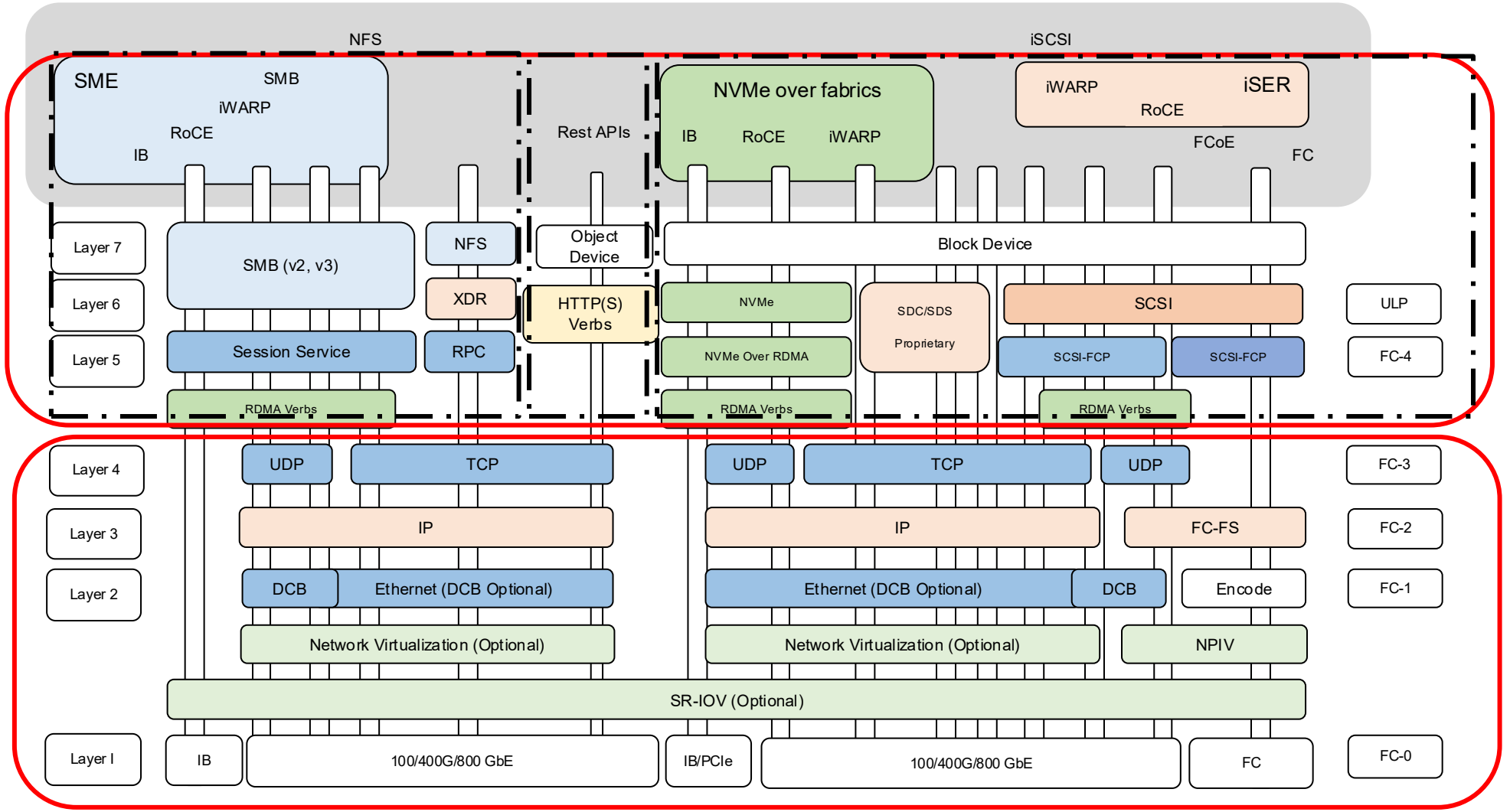
Storage Controller



SW Stack Functionalities

- **Front-End :** Connectivity, Low Latency, and Reliable Access.
  - **Services:** Multipathing, QoS, Access Control, and Metadata management.
- **Back-End :** Data Integrity, Security, and Media Longevity.
  - **Services:** Device discovery, health monitoring, and media-specific optimizations.
- **Unified Virtual Pool:** Location-agnostic storage volume
- **Advanced Resilience:**
  - **RAID:** Redundancy *within* the local storage domain.
  - **Erasure Coding:** Data resilience *across* distributed domains.
- **Efficiency Layer:** Deduplication, Compression, and **Auto-tiering** (Hot/Cold data)

Host Server - Software Stack



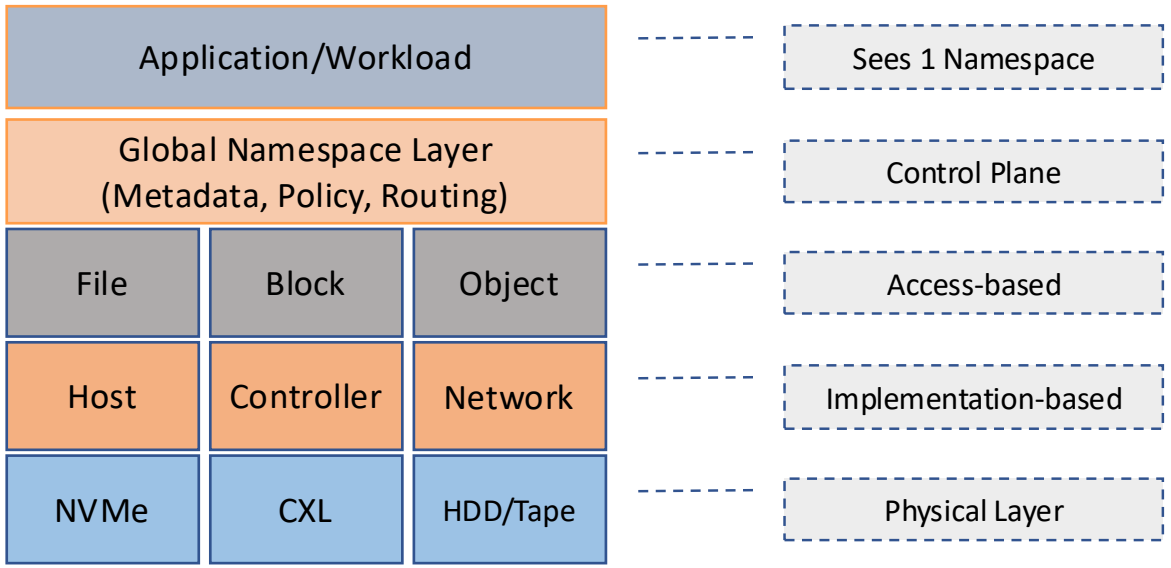
Thanks to Erik Smith for this diagram; also available here.

SNIA DSN A SNIA Community

Host Server - Software Stack - Summary

- **Application Layer:** Host servers run diverse applications requiring seamless, high-performance data access.
- **Storage Abstraction:** Decouples storage complexity and data services from the application layer.
- **Multi-Protocol Support:** Block: SCSI / NVMe, File: NFS / SMB, Object: S3
- **Hardware Agnostic:** Supports various physical media depending on specific deployment needs.
- **Performance Acceleration:** Utilizes Kernel/CPU Bypass via intelligent NICs and HBAs to minimize latency.

Application View: Global Namespaces (GNS)

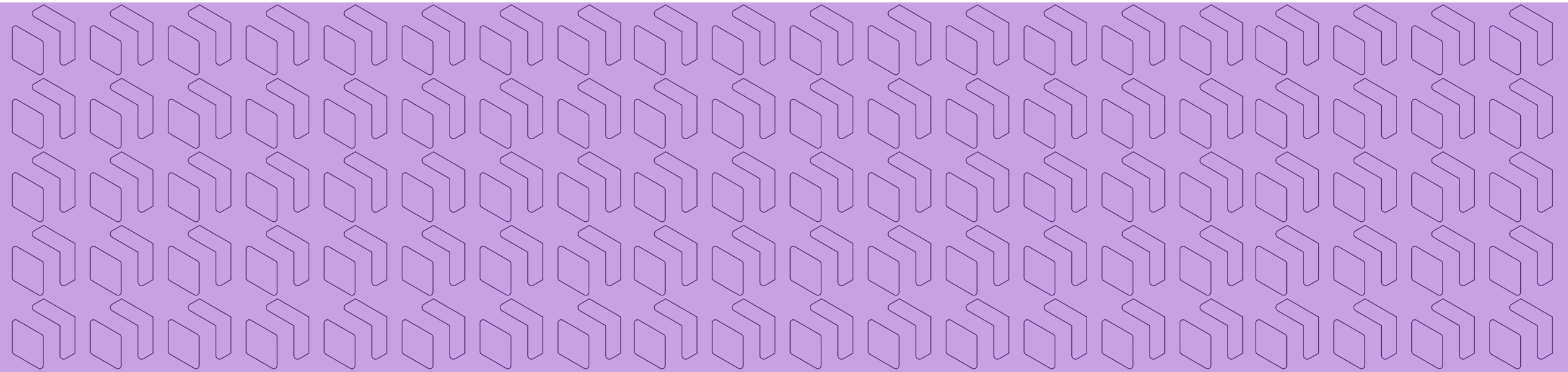


- Emerges from **combining** access-based and implementation-based virtualization by decoupling the logical path from physical data location.
  - Also known as separation of Control Plane and Data Plane
- **Single unified namespace** spans heterogeneous backends (NVMe, object, file) regardless of tier, node, or site
- **Location transparency:** same as before, i.e., when performance requirements are met, the application never needs to know where data physically lives
- Enables **dynamic data mobility:** data can move across tiers or nodes without changing the client-facing access path
- *GNS shifts the performance optimization burden from the application to the infrastructure. The performance ceiling is higher because the system can make globally-informed placement decisions. But realizing that ceiling requires a well-designed, distributed metadata layer that doesn't become the bottleneck.*

Notes: Data location is virtualized

SNIA | DSN
A SNIA Community

Key Architectural Guidelines



Simplified Performance “Lanes”

Lane 1: Kernel-bypass / RDMA / PCIe-like paths (fastest)

- **NVMeNVMe-oF (RoCE / iWARP / IB)**
- **RDMA verbs**
- **SR-IOV**
- **FC-NVMe / FCP**

Characteristics: Kernel bypass, Zero-copy (or near zero-copy), Low CPU per IO, Microsecond-scale latency

Lane 2: Kernel TCP/IP block paths (middle)

- **SCSI**
- **iSCSI**
- **TCP/IP**
- **NVMe-TCP**

Characteristics: Kernel involvement, Copy + context switches, Good compatibility, Still very fast, but higher tail latency

Lane 3: File / Object / User-space paths (slowest)

- **SMB / NFS**
- **RPC / XDR**
- **HTTP(S) / REST**
- **TLS**
- **Application-level semantics**

Characteristics: User-space processing, Serialization & metadata overhead, Best for sharing & scale, not latency

Key architectural insights

A grounding principle:

- Implementation-based virtualization and Access-based virtualization are mutually exclusive.
- Any one of the 3 implementation layers can expose more than one access type.
- Each of the 9 combinations can traverse multiple protocol paths in your stack.
- But not all paths are equal in latency, CPU overhead, or scalability.

Virtualization Type	Ideal Path	AI Stage
Host + Block	Local NVMe / NVMe-oF (RDMA)	Inference
Host + File	SMB Direct (RDMA)	Training / Inference
Host + Object	HTTP over TCP (large I/O)	Training
Network + Block	FC-NVMe / NVMe-oF	Inference
Controller + Block	FC-NVMe	Inference
Controller + File/Object	NFS / SMB Direct (RDMA)	Training

Other combinations are possible, but often involve gateways, translation layers, extra metadata hops, and higher CPU cost. These solutions are typically not widely deployed at scale in the industry and hence considered as “first-class” or “custom” designer solutions only.

Key architectural insights - Key Takeaways

Latency is minimized when the virtualization boundary is closest to the CPU and the protocol is closest to PCIe semantics.

- Host-level + block + NVMe = fastest
- Network-level adds hops but adds flexibility
- Controller-level trades raw latency for predictability and services
- File & object always sacrifice latency for sharing and scale

Key architectural insights - Decision Matrix

Step-1
- Identify the workload's access pattern

Step-2
- Pick an implementation layer

Step-3
- Validate against the 9-combination matrix

Step-4
- Match the protocol to the path

Data Indirection and Redirection

- **Data Indirection**
  - Logical address to physical address mapping is maintained in a separate table.
  - Helps in Thin Provisioning, Flash Wear leveling, Snapshot, Deduplication with no disturbance to the application
- **Data Redirection**
  - Redirects the IO Requests to a different path
  - Helps in High availability/Failover, Snapshot (Existing data is not copied for snapshot, instead new data is redirected to a new drive)

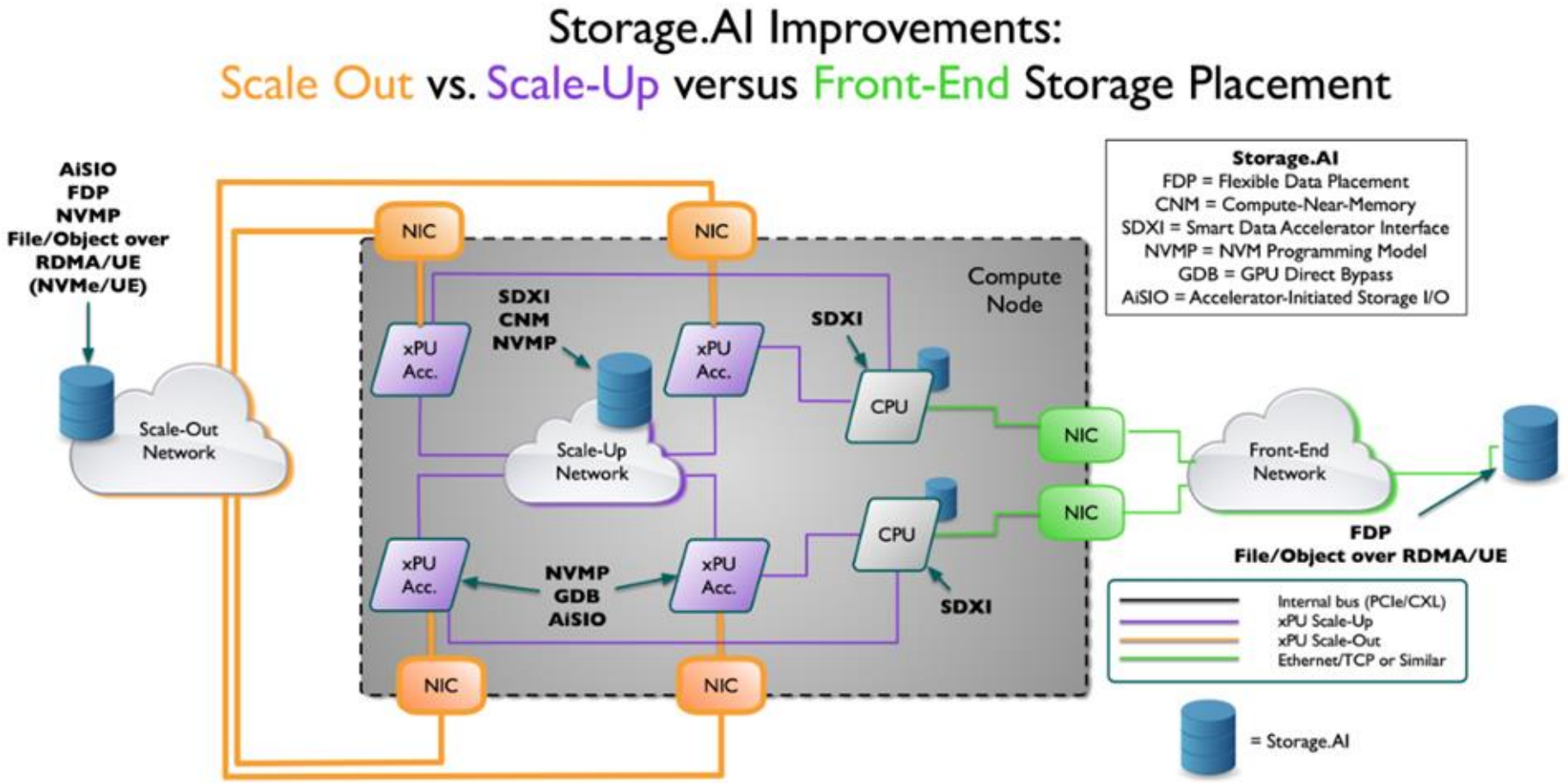
SNIA | DSN
A SNIA Community

StorageAI

Scale-up, Scale-out, and Scale-across

- Scale-up/Scale-in (Vertical Scaling):
  - Increasing the capacity of single server/rack by adding more compute, memory, storage, or networking resources.
  - Limited by the capabilities of the underlying hardware.
- Scale-out (Horizontal Scaling):
  - Increasing the capacity by adding more similar servers/racks to do parallel processing.
  - Theoretically infinite but limited by the synchronization and networking between the systems.
- Scale-across (Datacenter scaling):
  - Increasing the capacity by spreading across disparate environments and geographical locations.
  - Limited by many factors like latency, global laws and regulations, power, etc.

SNIA StorageAI™ Community



If you want to know more about SNIA's focus areas in the StorageAI™ community, please refer this link - https://www.snia.org/storage.ai, and consider attending the community meetings.

SNIA DSN A SNIA Community

Quick Poll

3. Do you feel better equipped and prepared to make storage software stack and architectural decisions now?

A. Yes
B. Somewhat
C. No

Recommended SNIA Webinars

- File vs Block vs Object Storage
- Everything You Wanted to Know About Throughput IOPs and Latency But Were Too Proud to Ask
- Everything You Wanted to Know About PCIe But Were Too Proud to Ask
- Everything You Wanted to Know About RDMA But Were Too Proud to Ask
- On-prem, Cloud, and Hybrid Storage Systems **– June 11, 2026**

References

- https://www.thomas-krenn.com/en/wiki/Linux_Storage_Stack_Diagram
- https://www.youtube.com/watch?v=XqSXgNC8M6M
- https://provandal.dev/protoviz/#/stacks

SNIA | DSN
A SNIA Community

Q & A

Like, Comment, Subscribe!

- Please rate this webinar and provide us with your feedback
- This webinar and a copy of the slides are available at the SNIA Educational Library snia.org/educational-library
- A Q&A from this webinar, including answers to questions we couldn't get to today, will be posted on our blog at sniablog.org
- Follow us on LinkedIn and X @SNIA

SNIA | DSN
A SNIA Community

Thank You!