SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

Zeroing in on the Storage Sweet Spot: Navigating the Forces of Scale, Performance and Management

J Metz, Ph.D
Cisco Systems, Inc.

SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

A Plea For Floor Awareness*

*In dancing, it's important to know what is going on around you. Keeping an eye on not just what you're doing, but what's going on around you, prevents people from getting hurt.

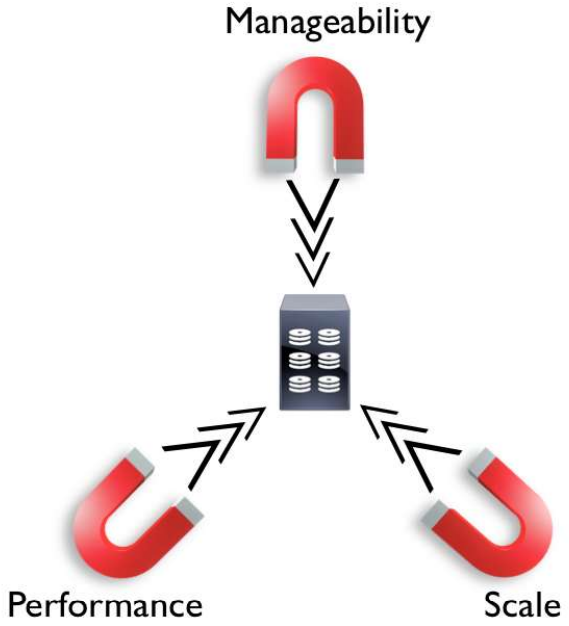
Agenda

- Storage Philosophy and Perspective
- Examining "Vertical" Storage
- Putting It All Together

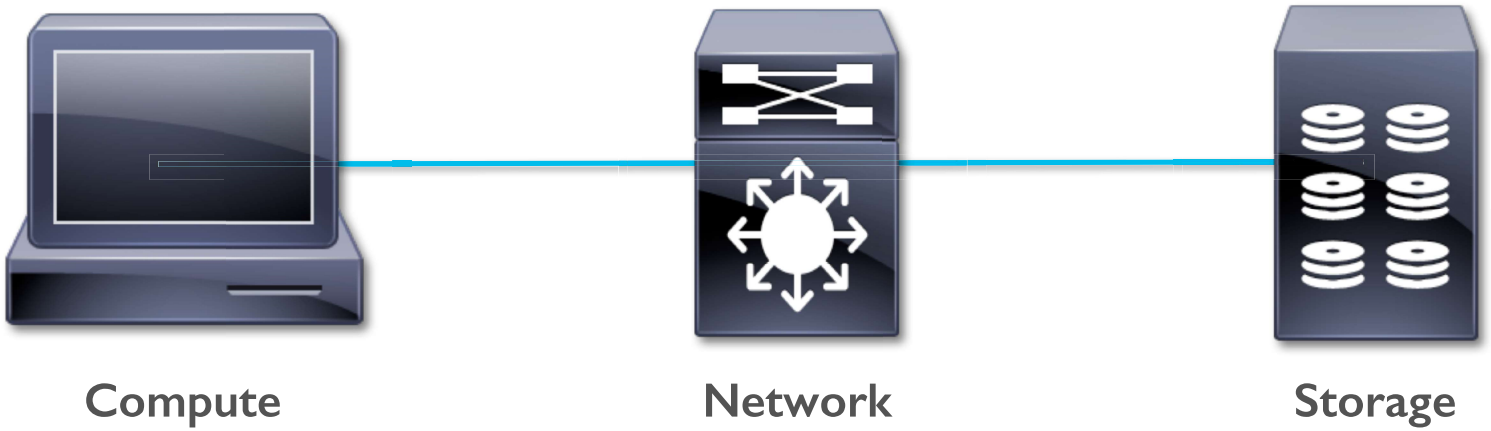
Part I: Horizontal v. Vertical

Storage Perspective

- There is a “sweet spot” for storage
  - Depends on the workload and application type
  - No “one-size fits all”
- What is the problem to be solved?
  - Deterministic or non-deterministic?
  - Highly scalable or highly performant?
  - Level of manageability?
- Understanding “where” the solution fits is critical to understanding “how” to put it together



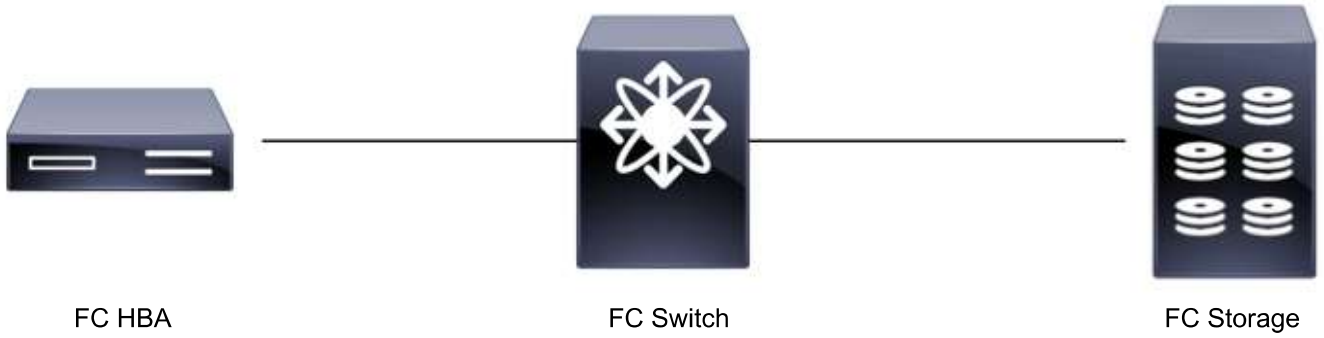
Horizontal View



- Many presentations look like this
- Focus on connectivity, *not* solutions

Example: Fibre Channel

Fibre Channel

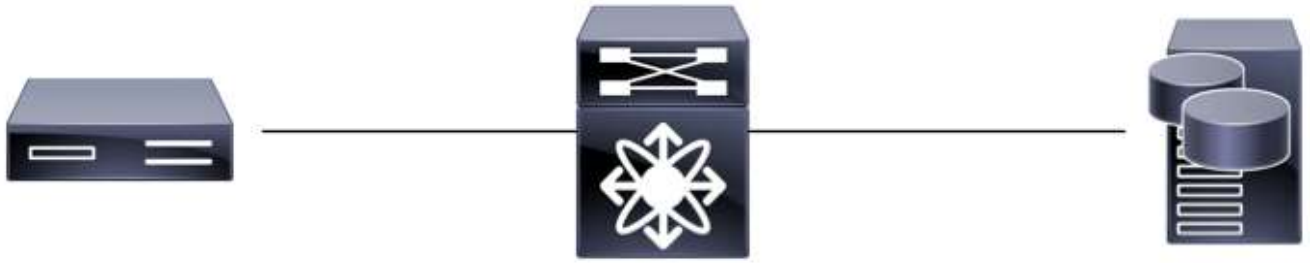


Bandwidth:
IOPS:
Latency:

Insert Hero Numbers Here

Example: iSCSI

iSCSI



iSCSI HBA/
Initiator

Ethernet Switch

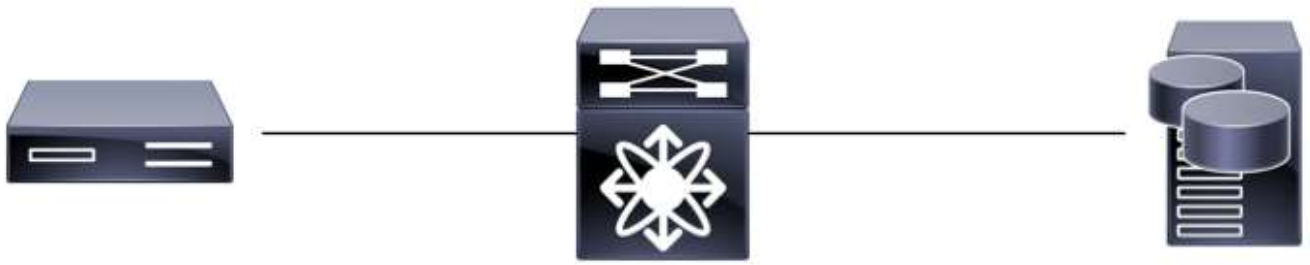
Ethernet Storage

Bandwidth:
IOPS:
Latency:

Insert Hero Numbers Here

Example: NVMe-oF

NVMe-oF



NVMe-oF Initiator

Ethernet Switch

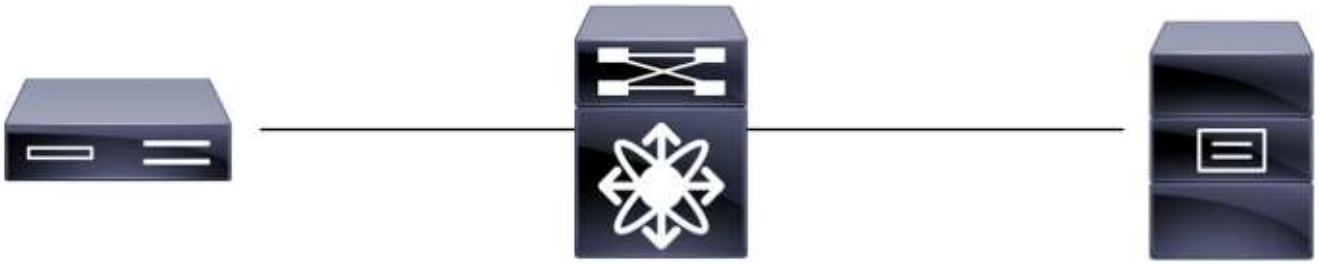
NVMe-oF Storage

Bandwidth:
IOPS:
Latency:

Insert Hero Numbers Here

Example: NFS

NFS



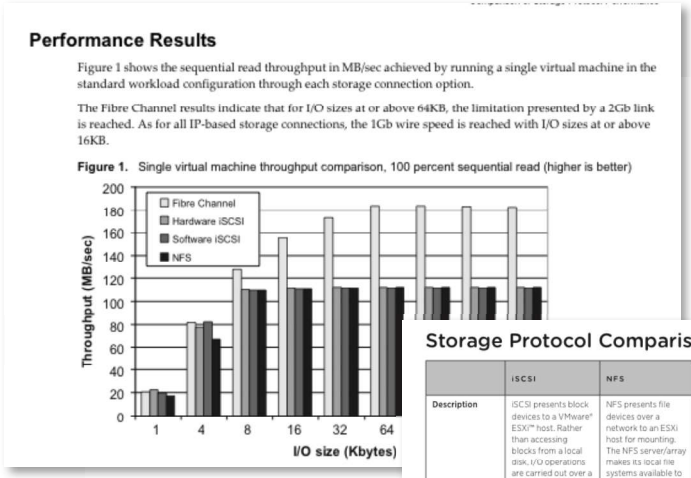
Ethernet NIC | Ethernet Switch | NFS Storage

Bandwidth:
IOPS:
Latency:

Insert Hero Numbers Here

Charts!!

- Hero numbers make great charts!



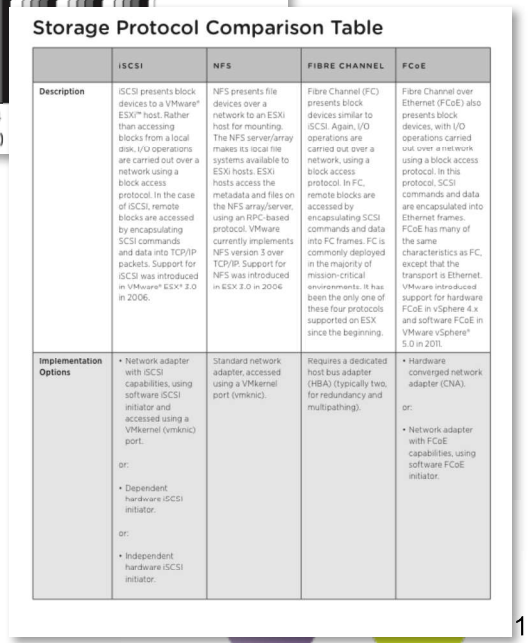
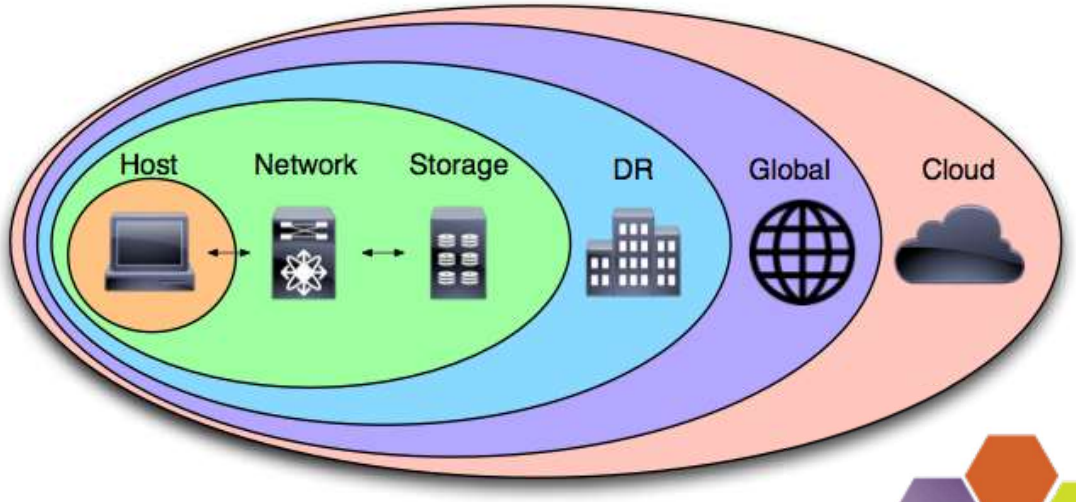


Image Sources:
Upper: Comparison of Storage Protocol Performance: ESX Server 3.5 (2008). VMware
Lower: Storage Protocol Comparison White Paper. (2012). VMware.

Advantages of the Horizontal View

- Explains the relationships between devices for a specific technology
- Easy to break down the component parts and “zoom in” to different sections



Disadvantages of the Horizontal View

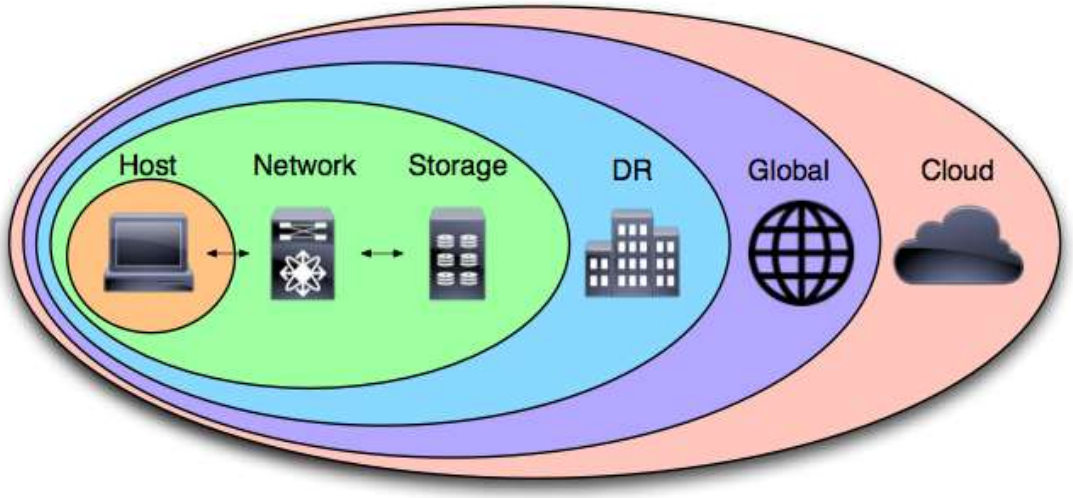
- Starts off in the middle; assumes you already know *why* you want to use that kind of technology
  - How do I know which one I should use?
- People will learn how to connect their favorite technology, without asking whether or not it's the right one to use

Example: What Problem Needs Solving?

- Problem:
  - I need to have very fast access for my Database application
- Problem:
  - I need to have multiple users share a single storage device for their home directories
- Problem:
  - I need to store a lot of images all over the world

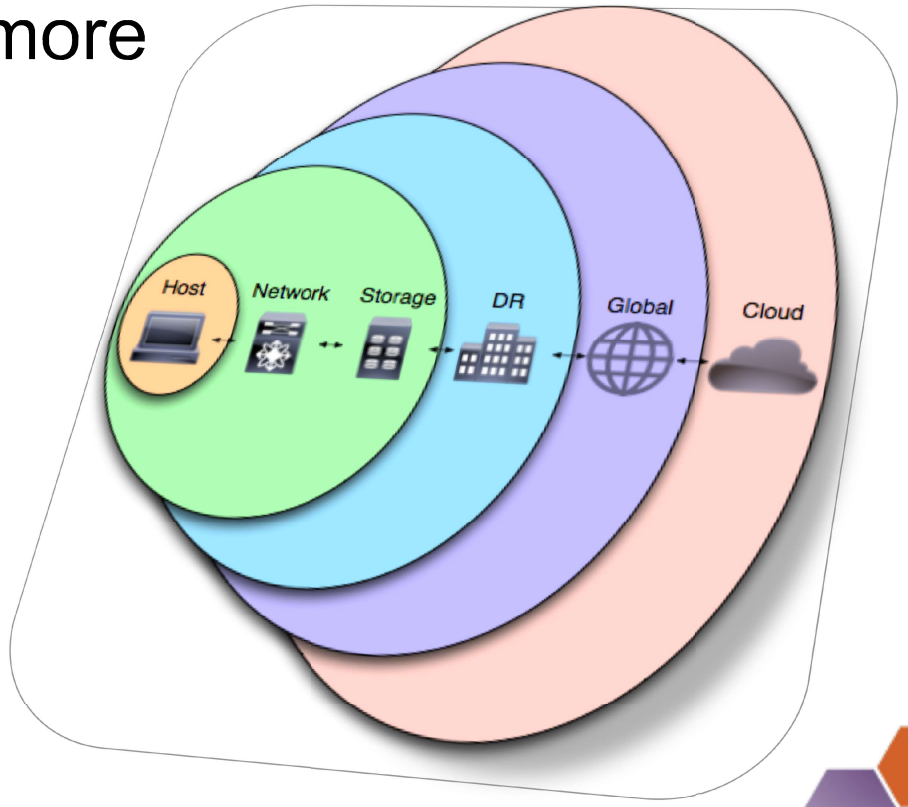
Changing the View

- Take the horizontal approach…



Vertical Approach

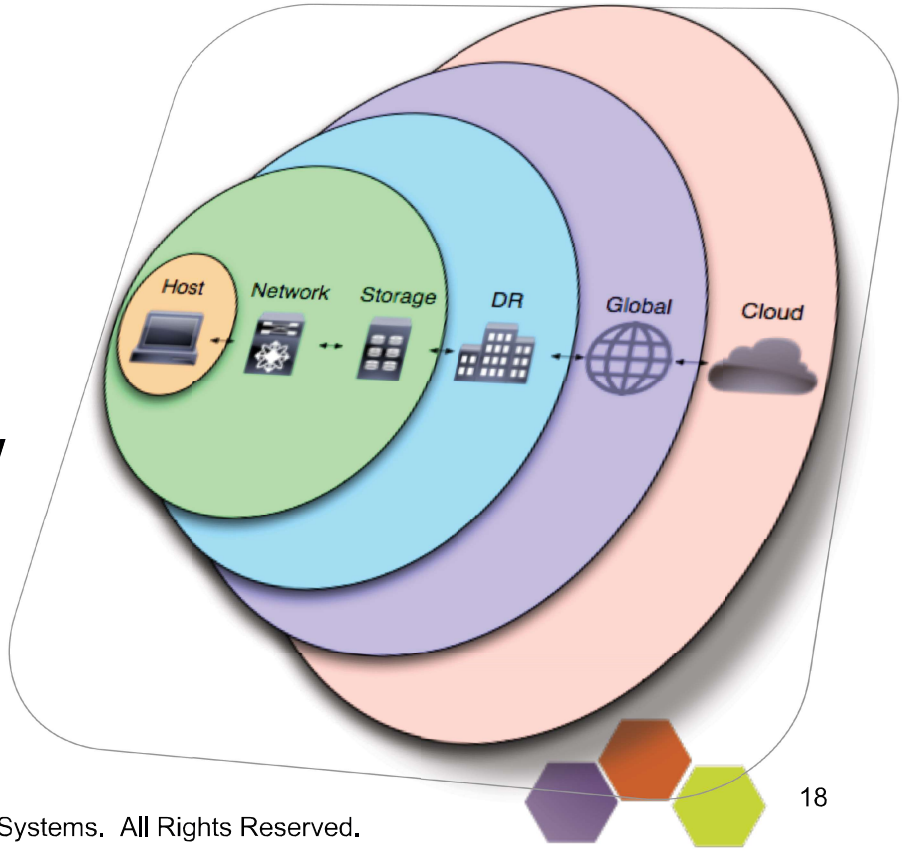
- … and look at it more vertically



Part II: Examining the Vertical

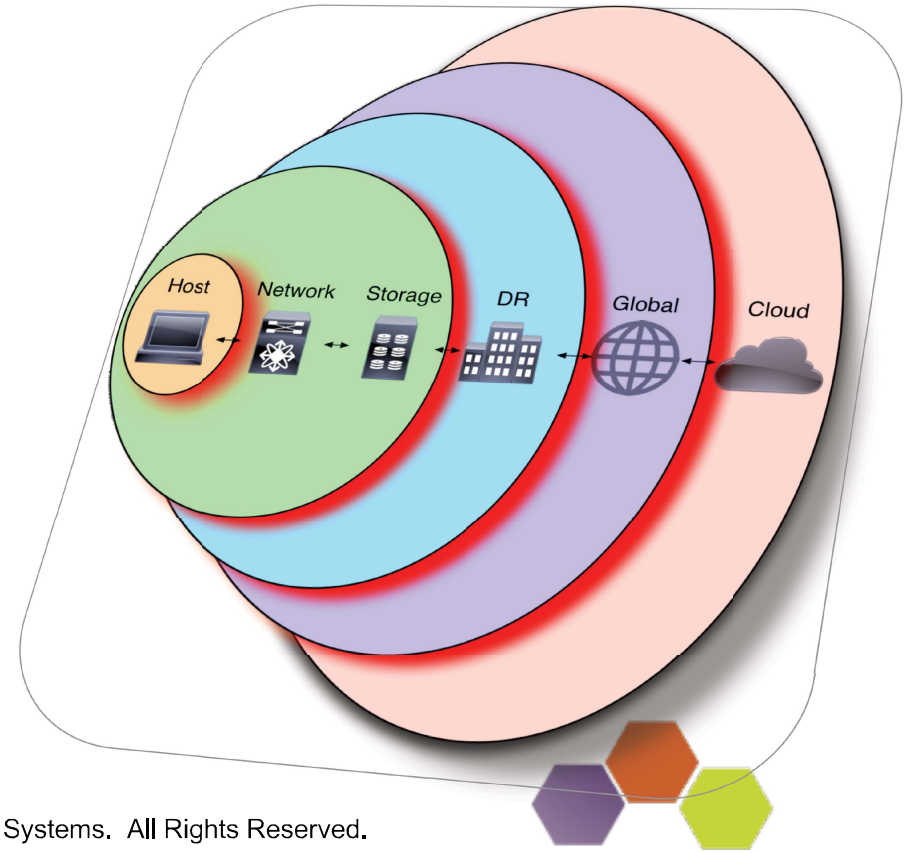
Big Picture

- Many ways to solve a problem
- Lots of overlap
  - Can easily get confused about which to choose
  - If two different approaches can do the same thing, how do you know what to do?



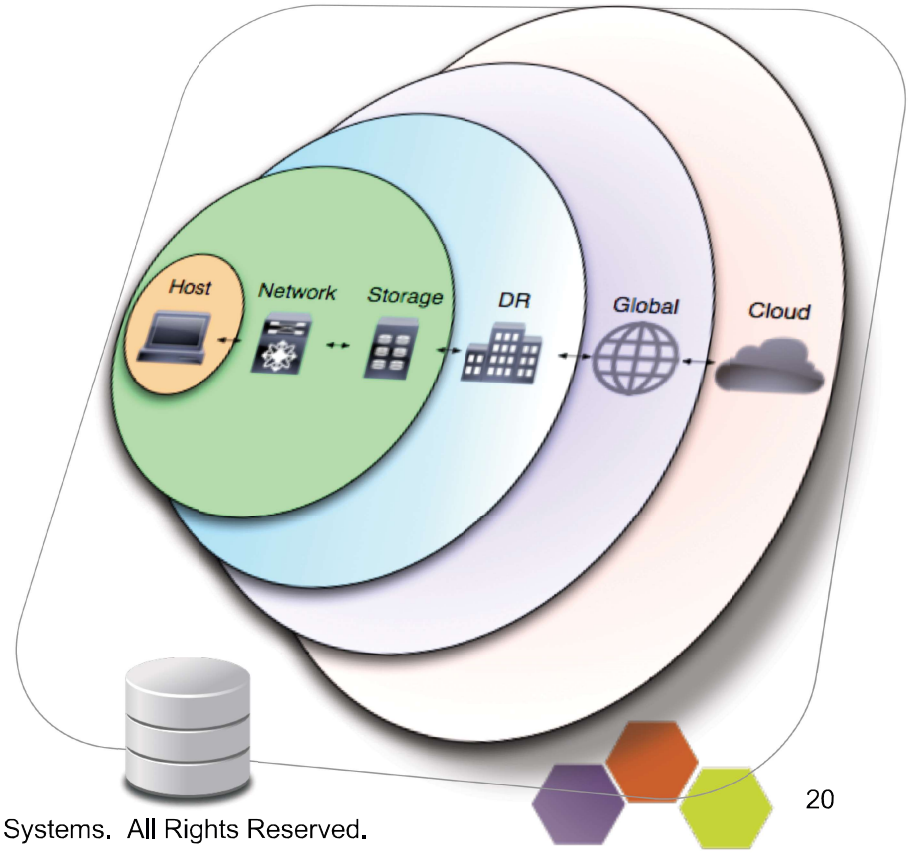
Big Picture

- When you miss the sweet spot, you risk major problems
  - Careful of the “Danger Zones”



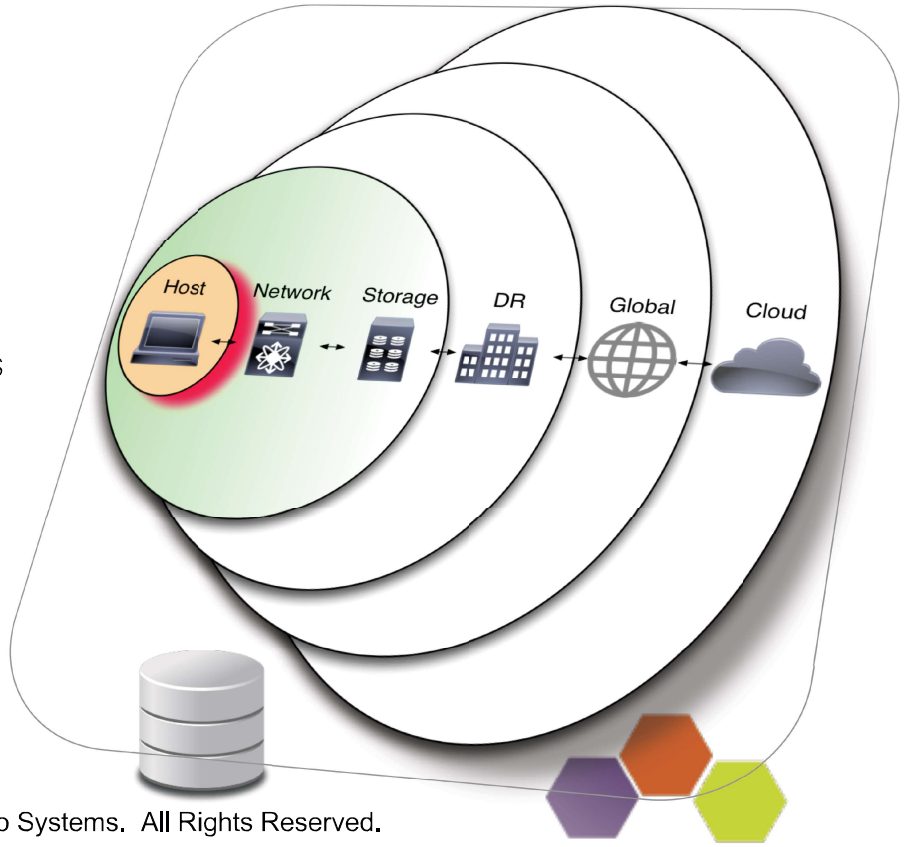
Block Storage Scope, Vertically

- Host and Storage are very close together
- Latency-sensitive
- Rigid Architectures
- High Performance, designed for high transaction, rapidly changing data
- Distance is for Disaster Recovery and Backup only



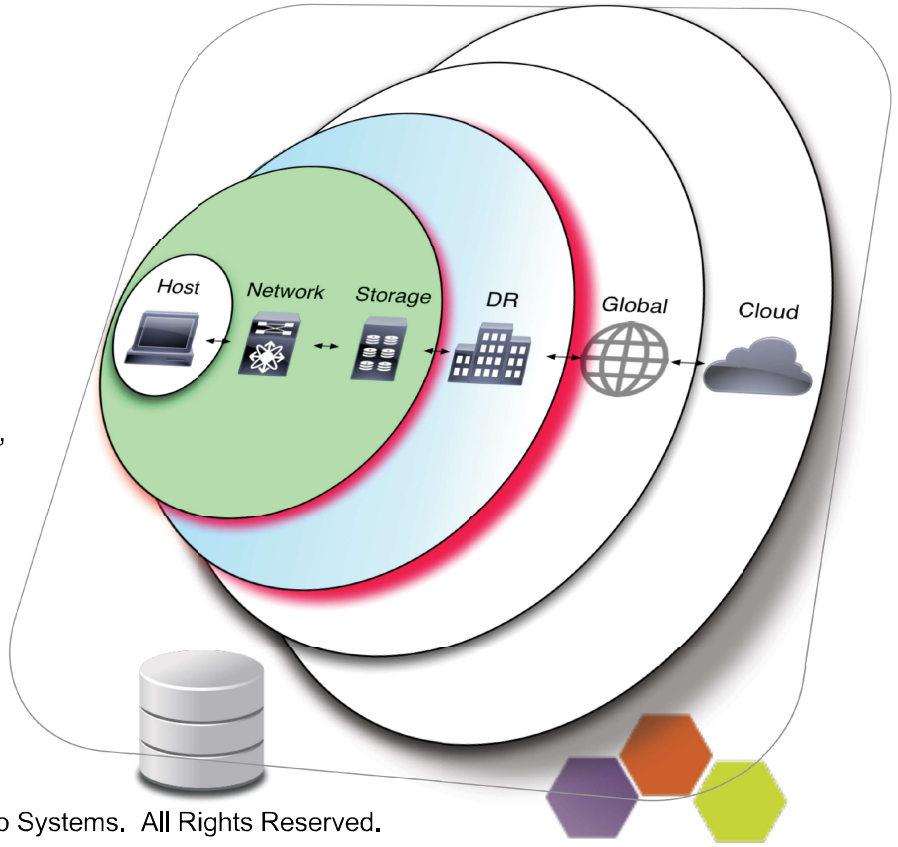
Intra-Host (PCIe/NVMe)

- PCIe
  - Co-Located
  - Device limitations
  - Remote storage access
    - Extending the bus architecture poses risks
    - Developing bus technology into a fabric poses risks
  - Distance is a non-starter



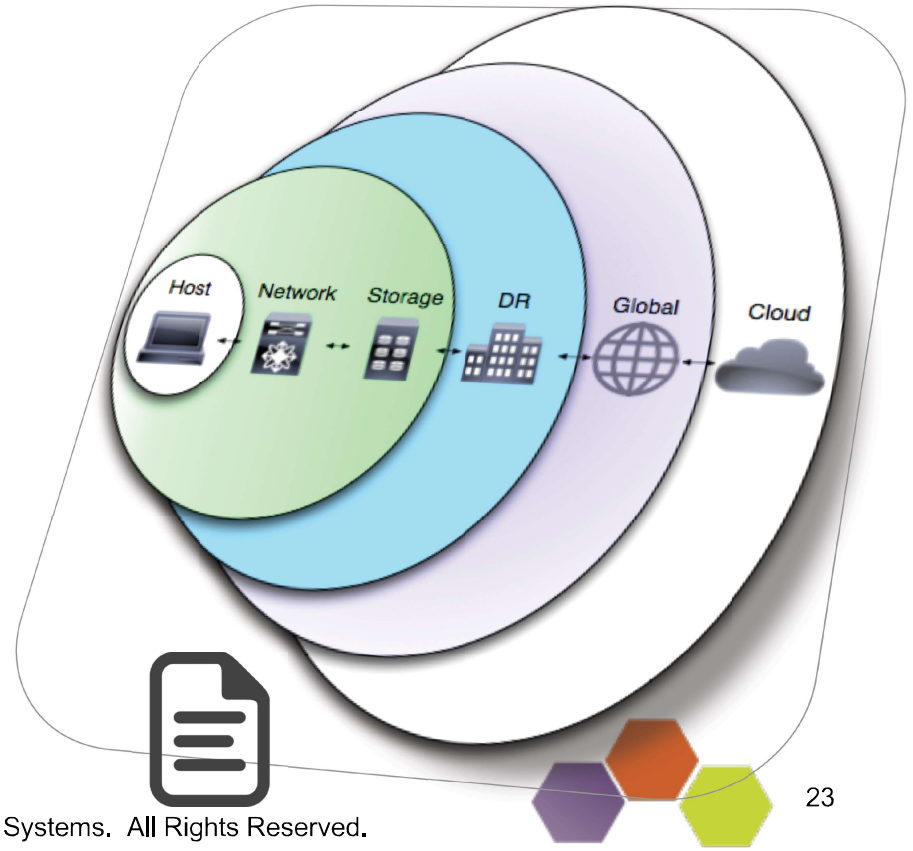
Intra-Data Center

- Fibre Channel, FCoE, InfiniBand & iSCSI
  - Not Co-Located
  - Primary usage: Intra-Data Center
  - Distance considerations
    - Extending the architecture poses risks
    - Host-to-storage is not a good idea
    - Global distance is not impossible for FC/iSCSI, but requires extra considerations



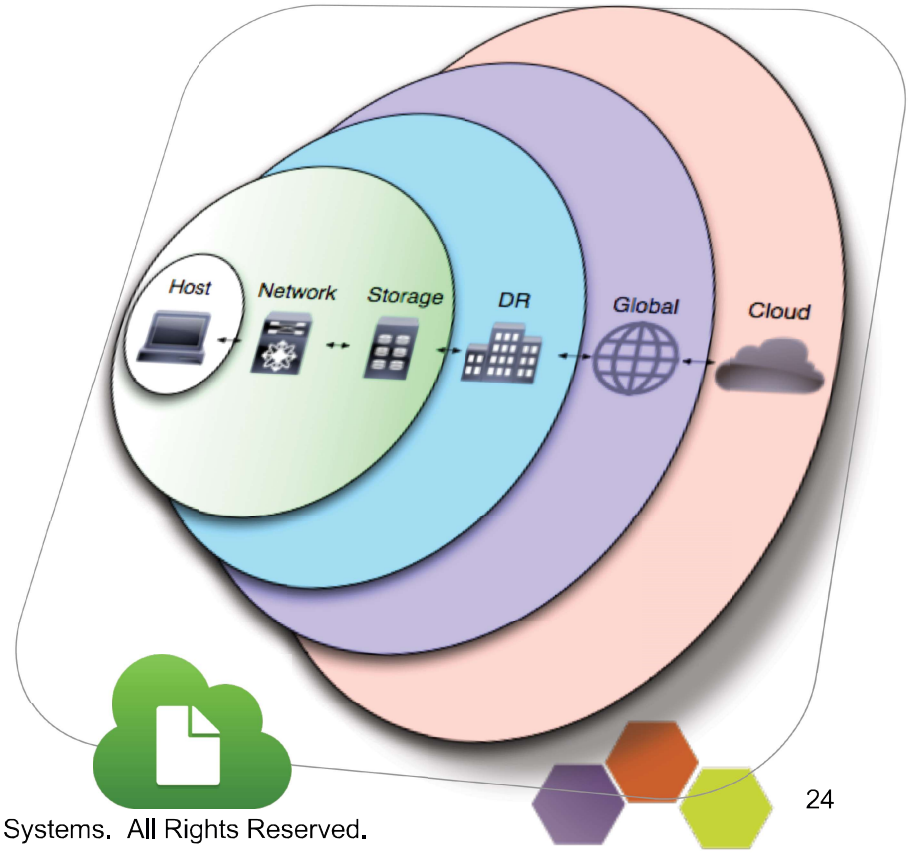
File Storage Scope, Vertically

- Less rigid architecture, less performant than Block
- Inside and Outside Data Center
- Designed for sharing data among clients at scale
- Distance can be for normal operations, Disaster Recovery, and Backup

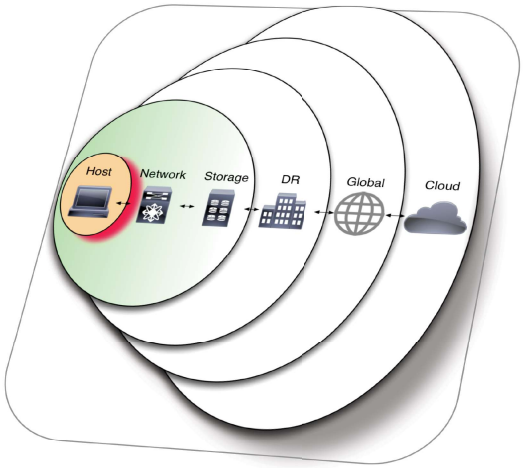


Object Storage Scope, Vertically

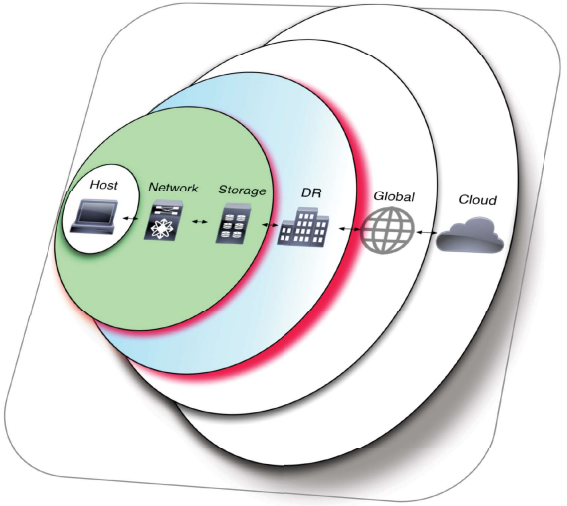
- Object Storage
  - Least performant
    - For data that doesn't change much, if at all
  - Designed for scale and distance - access from anywhere



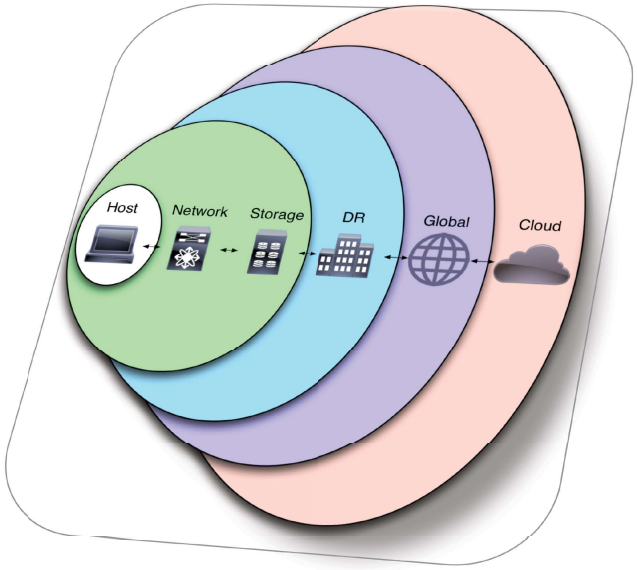
Scope Comparison



PCIe

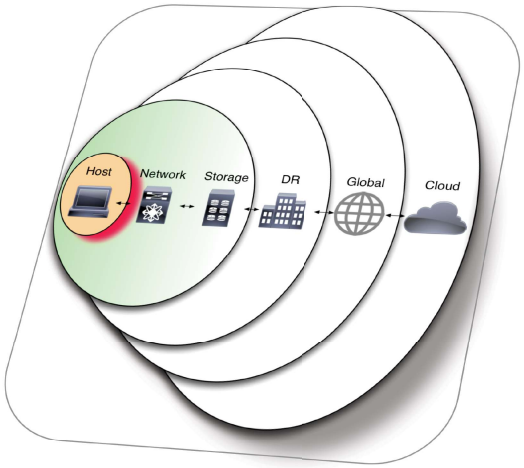


Fibre Channel
Ethernet (FCoE, iSCSI, iSER, NVMe-oF)
InfiniBand

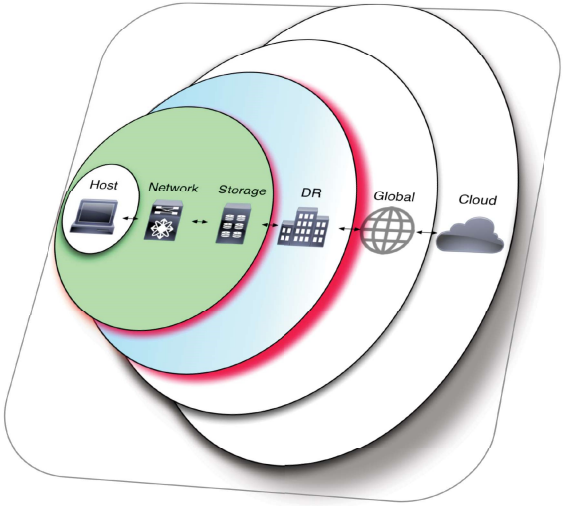


Ethernet (NFS, SMB, Object)

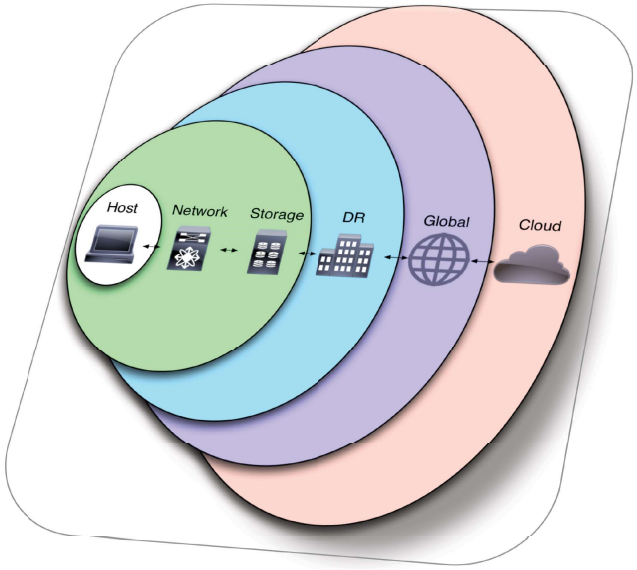
Scope Comparison



NVMe
Appliance
Rackscale/TOR

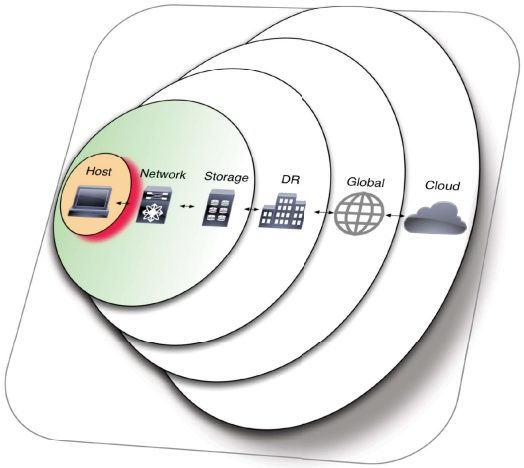


NVMe-oF
Traditional Arrays
Software-Defined Storage
Hyperconvergence
Appliance/TOR

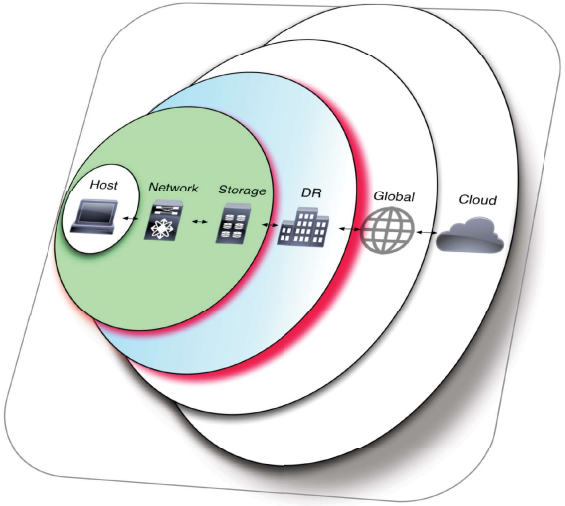


DR/BC/Archive
Cloud Data Stores

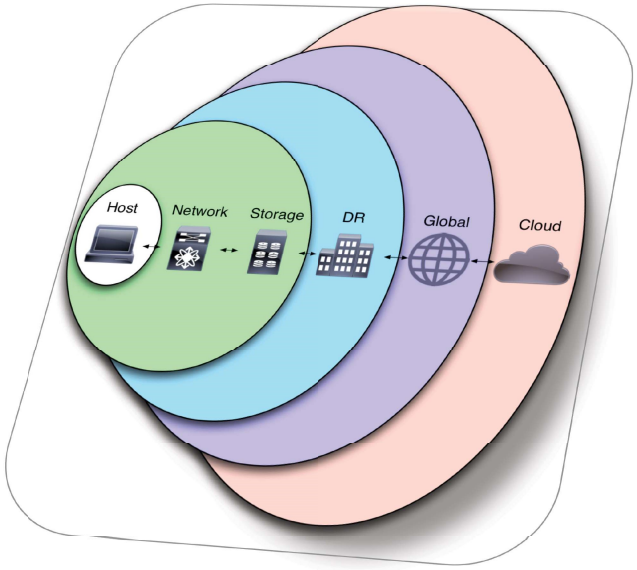
In the Future (to hear people talk)



K/V



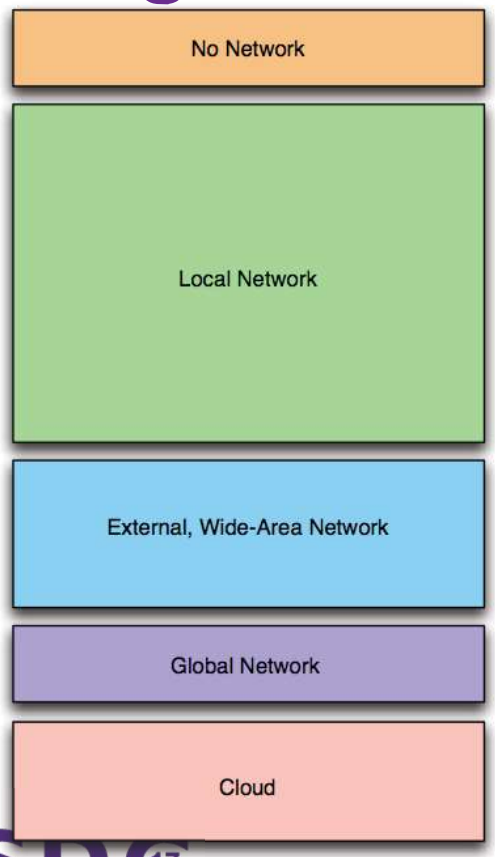
K/V

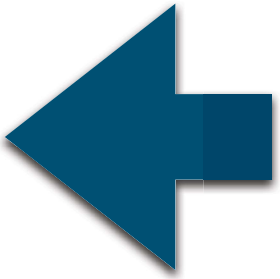


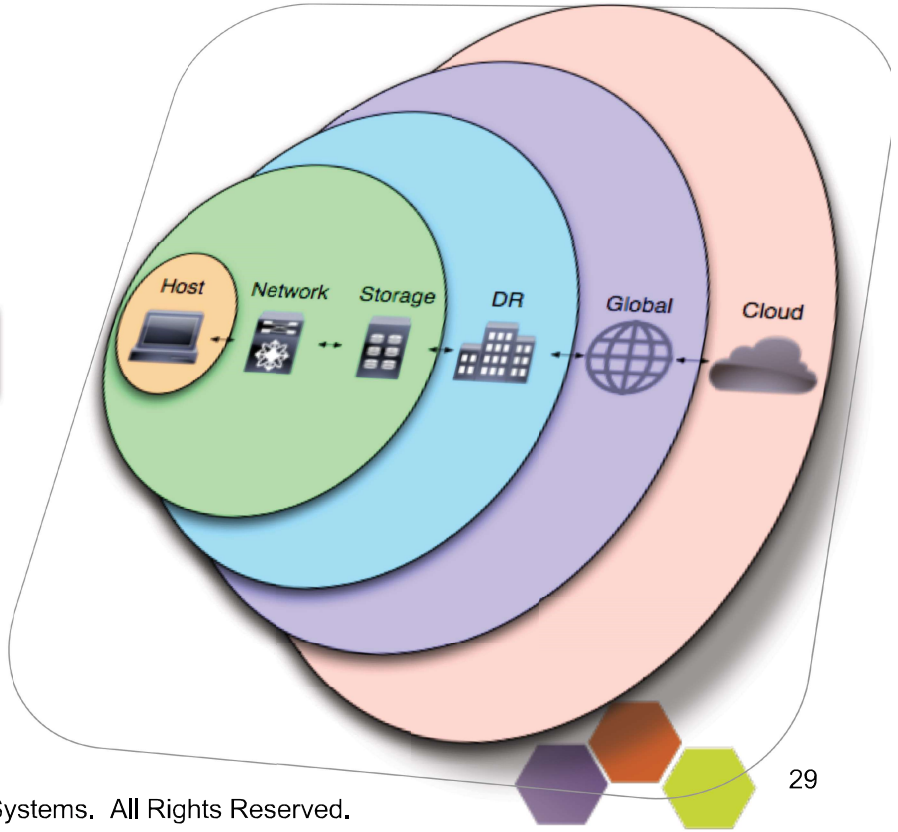
K/V

Part III: Putting it Together

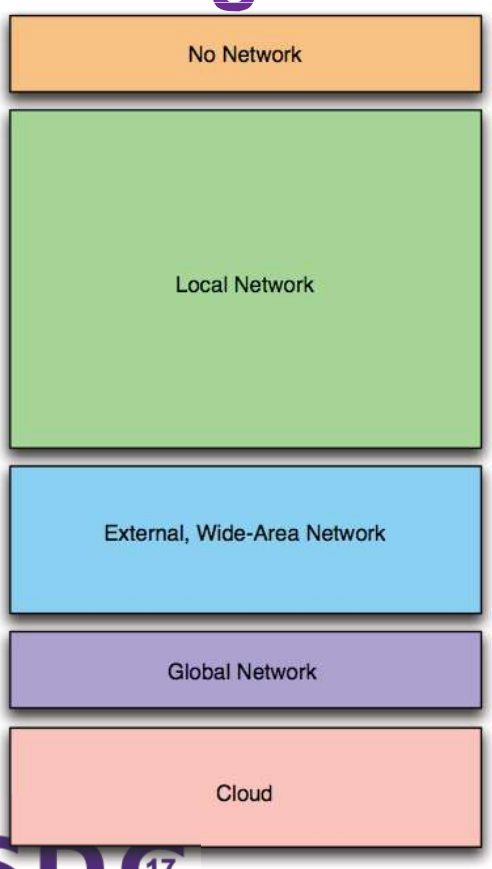
Seeing Vertically





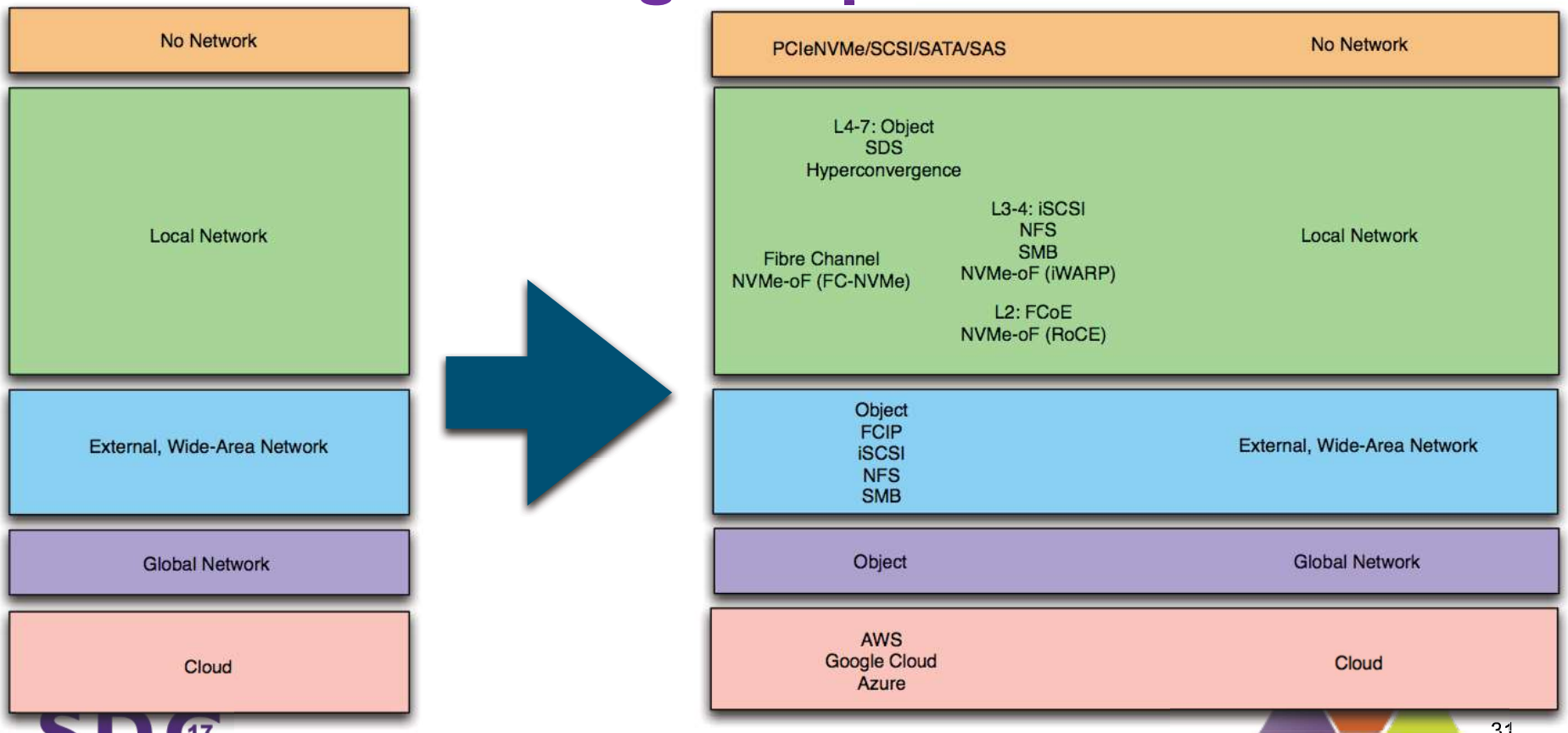


Thinking Vertically



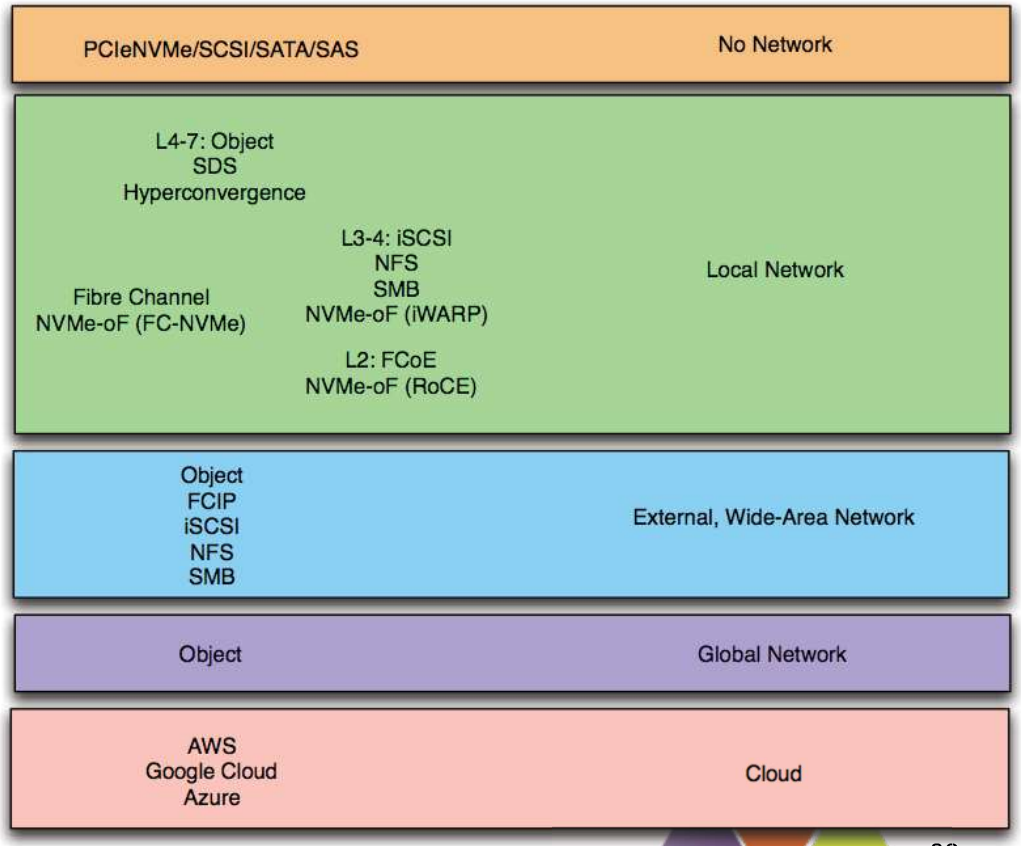
- Internal Storage
  - Buffering & Caching; Load/Store Operations; Local I/O operations
- Local Network
  - Inside a Data Center
  - Hosts communicate with storage targets or network-attached storage
- Wide-Area Network
  - Backup and Disaster Recovery, Remote Offices, Distributed sharing systems (e.g., Drop-Box)
- Global Network
  - Global object data/storage
- Cloud
  - Catch-all storage for “storage not located here”

The “Vertical Storage Map”



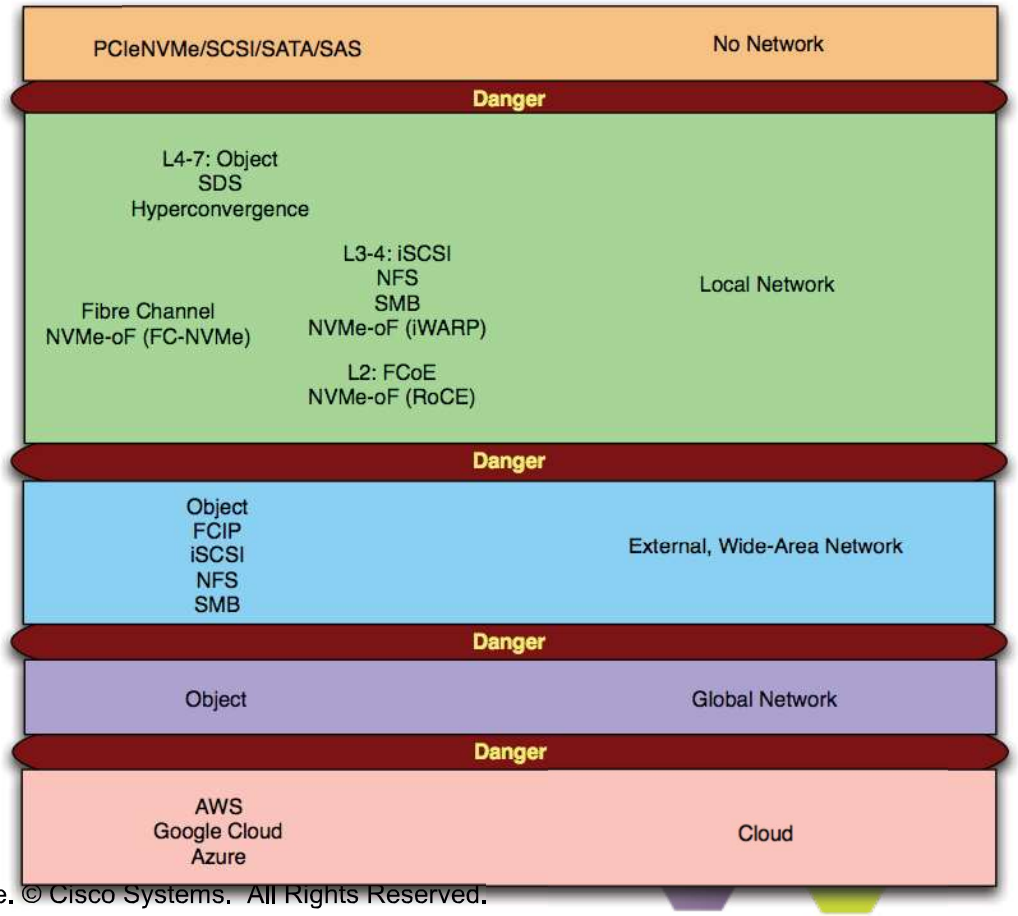
The Vertical Map

- A general guide to understanding "where" technologies play
- From the local (top of the chart) to the global (bottom) you can marry the needs of workloads and applications
- *Most* storage solutions focus in the green area
- Notice how the same technologies fit into multiple layers
- Boundaries are not rigid: new technologies move the goalposts



Know the Danger Zones

- There is no storage technology that is a “direct replacement” for another storage technology
  - There are always trade-offs
- The risk is where these concepts overlap
  - The trades are more “off” than on

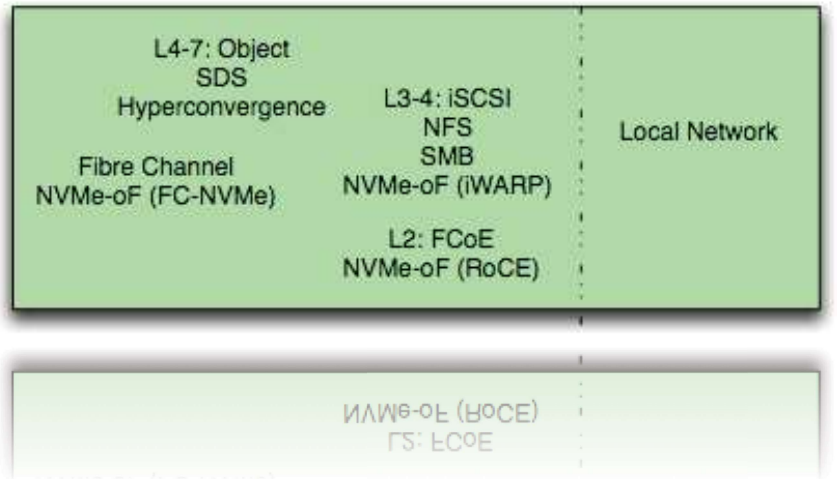


Local Storage

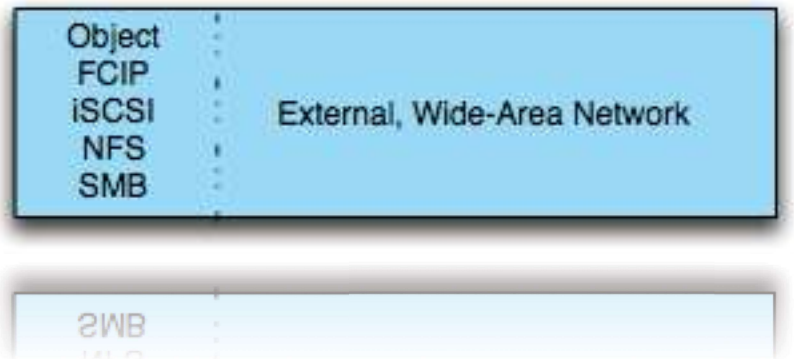
- Common Usage
  - Internal to servers, bus-based architectures
  - Recent emphasis on PCIe-based storage solutions
- All-Block solutions
  - Can build SDS and Hyperconvergence solutions with this foundation
- Risk
  - Adding layers of software and abstraction changes the fundamental architecture
  - Trying to extend internal storage solutions (e.g., PCIe extensions) can have unintended consequences

Remote Storage (Intra-Data Center)

- Common Usage
  - Connecting hosts with storage targets inside a Data Center
  - Different connecting strategies emphasize or de-emphasize storage
  - Deterministic vs. Non-deterministic storage approaches
- Block, File (and some Object) solutions
  - Some solutions trade off scale and performance for manageability
- Risks
  - Assuming that “all storage is equal” and is just an I/O problem
  - Not understanding that “using Ethernet for storage” covers a lot of ground with massive margin for error



Long-Distance Storage



- Common Usage
  - Backup, disaster recovery, infrequent access to
- Block, File and Object usage
  - Block storage is only used for backup - never connect a host to a storage device across a wide-area network!
  - File and object storage is used for backup, but also for low-frequency, small size storage access
- Risks
  - Expecting block storage to work the same over distances
  - Placing the wrong workload on long-distance links

Global Storage

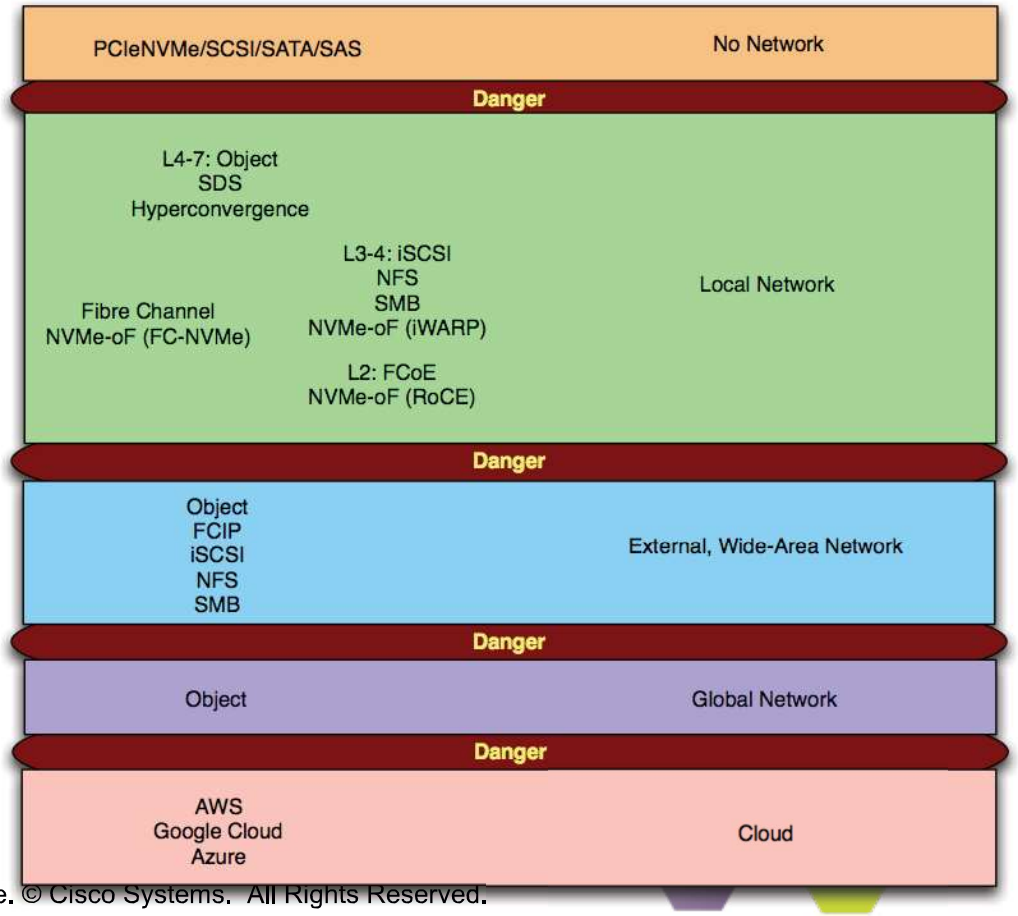
- Common Usage
  - Data replication services, Backup and Recovery, eventually consistent data
- Object Usage Only
- Risks
  - Workload/Distance mismatch

Cloud Storage

- Common Usage
  - Varied: Workloads completed “in the cloud” use a
- Can be Block, File, or Object
  - But distributed to client machines via File or Object
- Risks
  - Choosing the wrong storage type for your workload can be fundamentally hazardous to your employment future
  - Costs can skyrocket

Big Picture

- Need to focus on not just the “sweet spots,” but also the overlapping areas
- How does this affect:
  - Management?
  - Lifecycle?
  - Budgeting?
  - Upgrades?
  - High Availability and Reliability?



Pop Quiz!

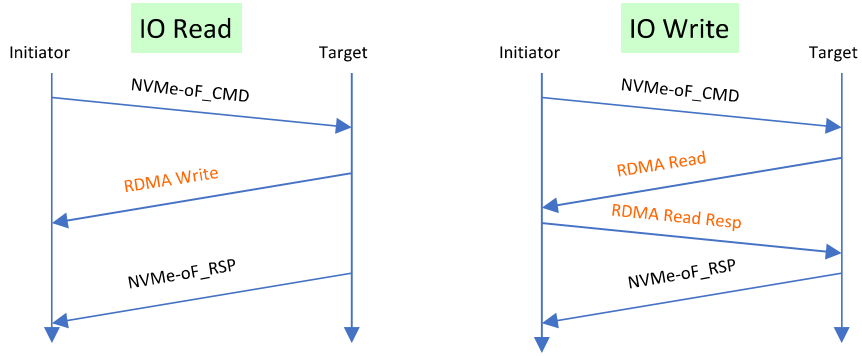
NVMe-oF

- **“What’s the difference between using Fibre Channel versus RDMA-based Ethernet Transports?”**
  - How do you go about answering such a question?
  - What is the *real* question we are trying to answer?

Zero-Copy

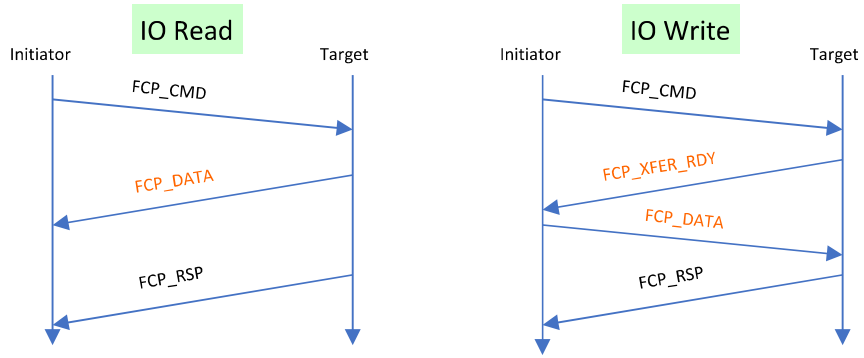
- Zero-copy
  - Allows data to be sent to user application with minimal copies
- RDMA is a semantic which encourages more efficient data handling, but you don't need it to get efficiency
- FC has had zero-copy years before there was RDMA
  - Data is DMA'd straight from HBA to buffers passed to user
- Difference between RDMA and FC is the APIs
  - RDMA does a lot more to enforce a zero-copy mechanism, but it is not required to use RDMA to get zero-copy

NVMe-oF using RDMA



- NVMe-oF over RDMA protocol transactions
  - RDMA Write
  - RDMA Read with RDMA Read Response

NVMe-oF using FC-NVMe



- FCP Transactions look similar to RDMA
  - For Read
    - FCP_DATA from Target
  - For Write
    - Transfer Ready and then DATA to Target

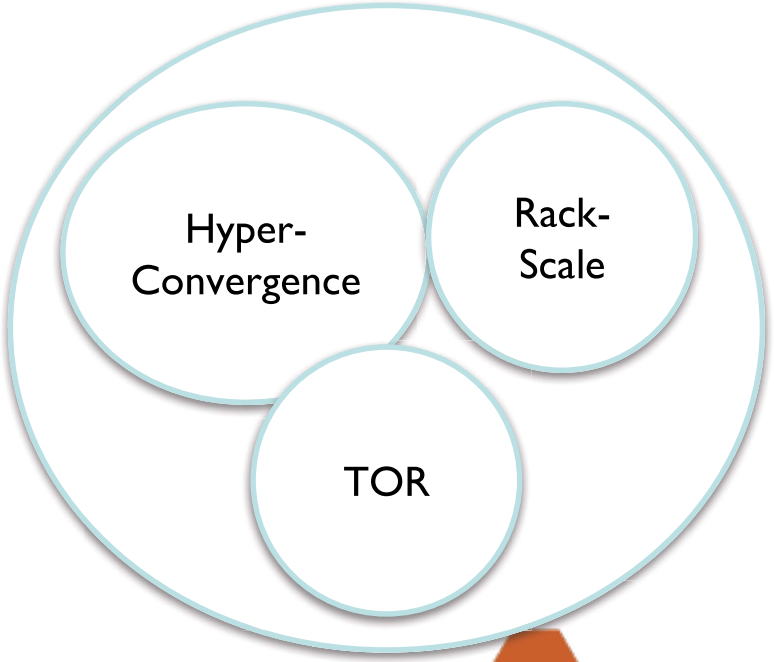
So…

- Looking at the key functionality requirements for Zero-Copy don't help
- Thinking Horizontally doesn't help
- Thinking in terms of connectivity doesn't help
- What helps?
  - Deterministic or Non-Deterministic environments
  - Not just data locality, but *transport* locality
  - "Outside in" or "Inside Out" management

Bottom Line

Beware The Law of the Hammer

- True for both developers, consumers, and architects
- Can you do the same thing with a different paradigm?
- Are we shoe-horning in "solutions" *just because we can?*
- *What happens when we become obsessed with "which technology will 'win'?"*

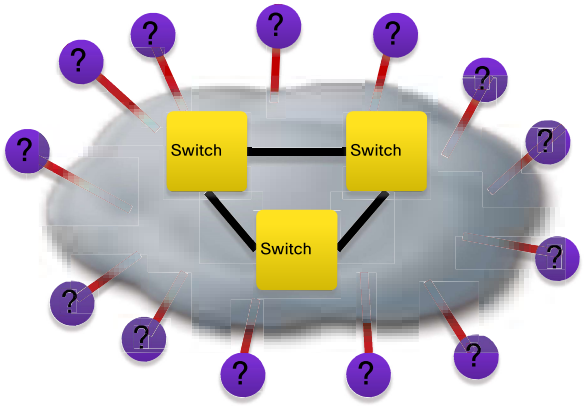


“Battle of the Boxes” or “Data on a Stick?”

- Application Developers rarely think of network topologies
  - “As long as I have access to *some* storage, I’m okay. How it gets there is not my problem.”
- Compute/Server developers rarely think of storage relationships
  - “My job is to get the I/O to the wire. After that it’s not my problem”
- Storage developers rarely think of management issues
  - “My job is to protect the bits at all costs. How people manage their boxes is not my problem.”

Network Determinism

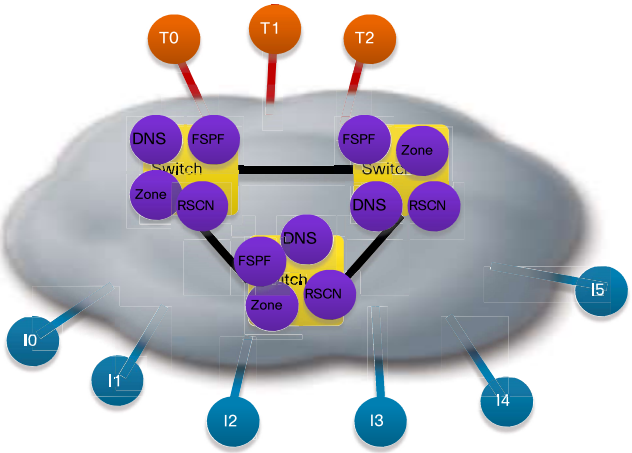
- Non-Deterministic
  - Provide any-to-any connectivity
  - Storage is unaware of packet loss – relies on ULPs for retransmission and windowing
  - Provide transport w/o worrying about services
  - East-West/North-South traffic ratios are undefined
- Examples
  - NFS/SMB
  - iSCSI
  - iSER
  - iWARP
  - (Some) NVMe-oF



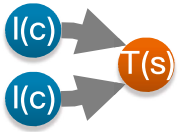
Fabric topology and traffic flows are highly flexible

Client/Server Relationships are not pre-defined

Network Determinism (cont.)



Fabric topology, services and traffic flows are structured



Client/Server Relationships are pre-defined

- Deterministic Storage
  - Goal: Provide 1:1 Connectivity
  - Designed for Scale and Availability
  - Well-defined end-device relationships (i.e., initiators/targets)
  - Only north-south traffic; east-west mostly irrelevant
- Examples
  - Fibre Channel
  - Fibre Channel over Ethernet
  - InfiniBand
  - RoCE
  - (Some) NVMe-oF

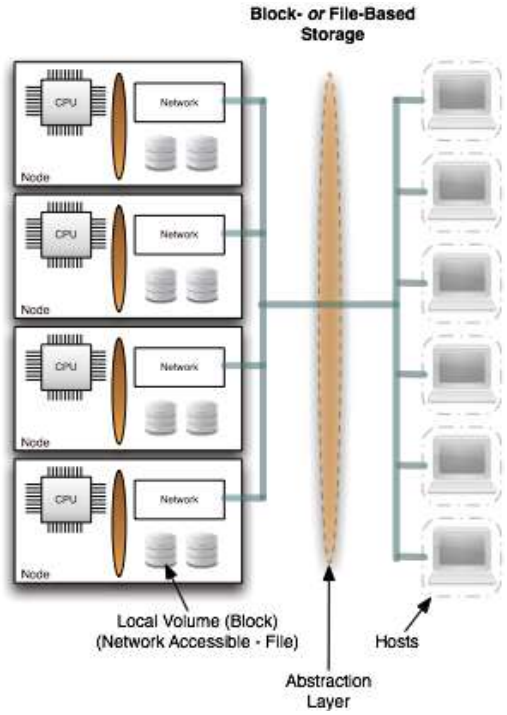
Comparison

	Ethernet	PCIe	Fibre Channel	InfiniBand
Intra-Host	No	Yes	No	No
Direct Attached (DAS)	*Yes*	Yes	Yes	Yes
Network Attached (NAS)	*Yes*	No	No	No
Storage-Area Network (SAN)	*Yes*	No	Yes	Yes
Deterministic Capability	*Yes*	Yes	Yes	Yes
Non-Deterministic Capability	*Yes*	No	No	No
Block Storage	*Yes*	Yes	Yes	Yes
File Storage	*Yes*	No	No	No
Object Storage	*Yes*	No	No	No
Global Distance	*Yes*	**Hell no**	No	No

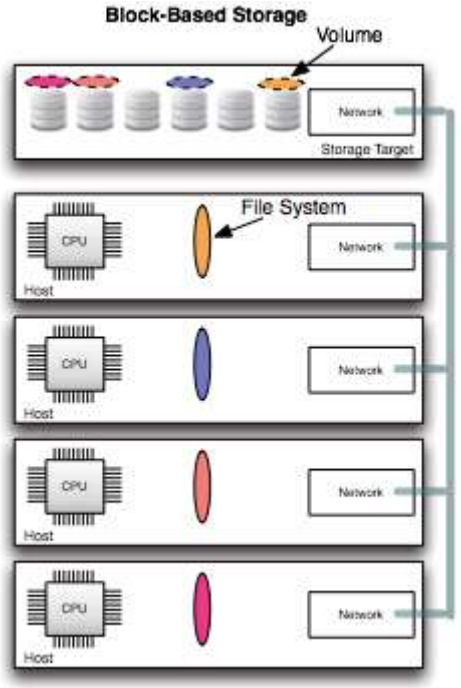
Comparison (cont…)

	Ethernet	PCIe	Fibre Channel	InfiniBand
Centralized Fabric Service	No	No	Yes	Yes
Consistent Performance at Scale	Maybe	N/A	Yes	Yes
Flexible Architectures	Yes	No	No	No
Open Source Ecosystem	Yes	No	No	No
End-To-End Qualification	No	No	Yes	Yes
Standards-based Solutions	Sometimes	Yes	Yes	Yes
Multiple Traffic Simultaneously	Yes	No	Yes	No
Codified Management	No	Yes	Yes	Yes

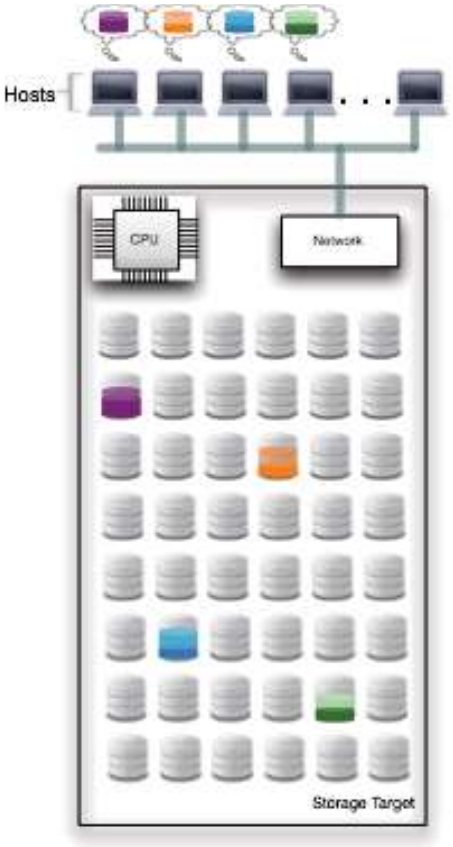
Example - Architectural Comparison



Hyperconverged



Top of Rack (TOR)



Array

Example - Architectural Comparison

- Considerations
  - Data protection/High Availability
  - Scale vs. Management
  - Performance Consistency
  - Growth/Migration
  - Ease of Backup/Restoration
  - Update Procedures

Hyperconverged

Top of Rack (TOR)

Array

<Technology> Is Dead! Long Live <other technology>!

- Tape is dead! Long live disk!
- Disk is dead! Long live SSDs!
- SSDs are dead! Long live NVMe!
- Mainframe is dead! Long live arrays!
- Arrays are dead! Long live Software-Defined!
- Fibre Channel is dead! Long live Ethernet!

Summary

Summary

- Understanding the reasons why you want to put a storage solution in place is *far more important* than how the parts connect together
  - First things first
- Understanding the strengths/weaknesses of each "layer" will help prevent massive failures in the danger zone
- Understanding the combination of technology "sweet spots" and usage requirements can make you a hero
- There is a need for everyone - consumers, developers, and architects, included - to gain just a bit more "floor awareness"