SNIA

Education

File Systems for Object Storage Devices

Paul Massiglia & Tushar Tambay
Symantec Corporation

SNIA Legal Notice

Education SNIA

- The material contained in this tutorial is copyrighted by the SNIA.
- 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
  - The 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 the SNIA Education Committee.
- 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.

Abstract

- ## File Systems for Object Storage Devices

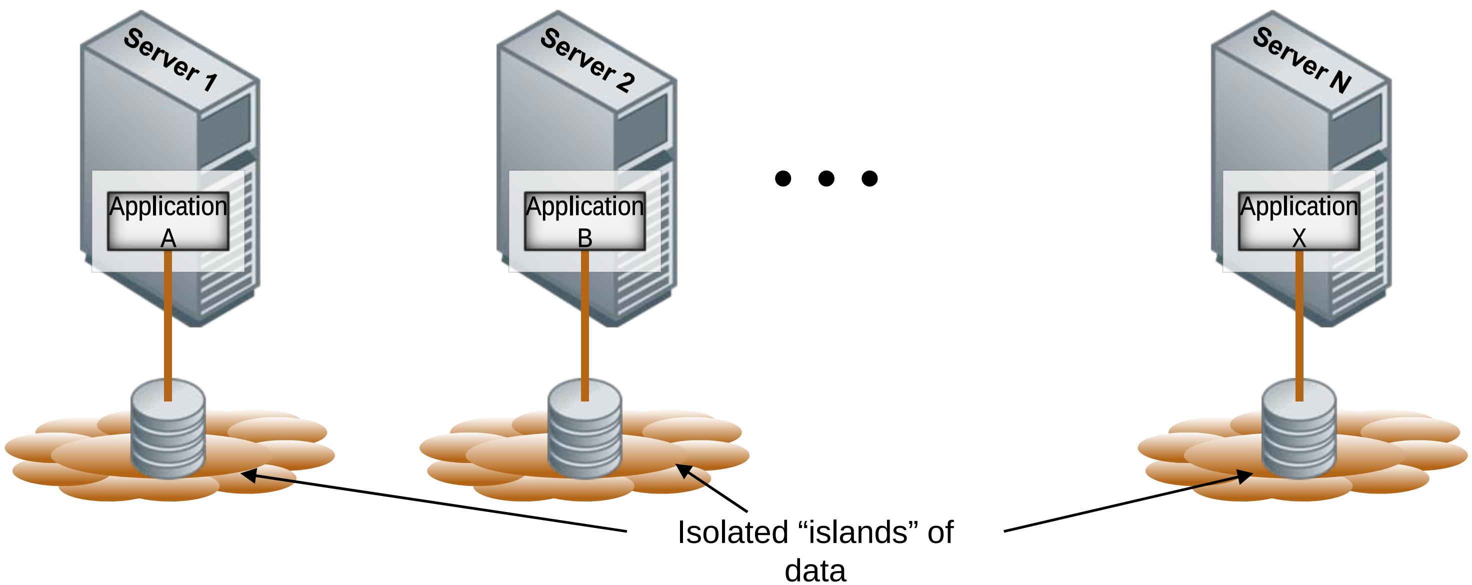
Object-based storage devices (OSDs) may well be the "next big thing" in file-oriented data storage. Already popular in the high-performance computing arena, they are poised to enter more general enterprise computing environments. By distributing storage management and enabling secure data transfer between storage devices and clients, OSDs promise significant improvements in scaling and administrative simplicity. But making effective use of OSDs requires a new breed of file system—one that makes use of the new devices effectively to deliver the promised benefits. This tutorial will describe the salient properties of OSDs, explain how file system technology is evolving to exploit the scalability and administrative simplicity they offer, identify the mature and emerging segments of the OSD-based file storage market, and show how technology that has been successful in HPC can be beneficially employed in the general data center environment. Standardization activities, notably the parallel NFS (pNFS) protocol for addressing OSDs will be discussed.

Outline

- Limitations of current file storage system designs
- Object Storage Devices (OSDs)
  - What they are
  - How they help overcome the limitation
- File systems for OSDs
  - Basic architecture
  - Adding value via performance and availability
- So, you think you want an object-based file storage system…

The root problem

- Increasing integration of business processes (aka "applications")
  - Business processes run on separate servers
  - Result: a need to share massive amounts of data ***in real time***
  - Consequence: a need for file servers of unprecedented scale

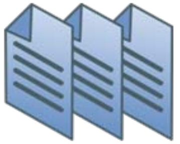


The ideal enterprise file storage system

- **Extreme capacity**
  - Millions of files, petabytes of data, thousands of clients
- **Performance**
  - High bandwidth and low latency with near-linear scaling
- **Universal access**
  - Data sharing among all data center computing platforms
- **Security**
  - Protect files against unauthorized access while sharing
- **Flexibility**
  - Easy administration and incremental growth

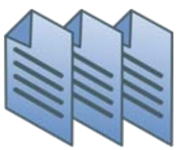
Teasing the problem apart…

- People deal with "business objects"
  - aka "files"
  - Names, sizes, access rights, lifetimes…



Teasing the problem apart…

- People deal with “business objects”
  - aka “files”
  - Names, sizes, access rights, lifetimes…



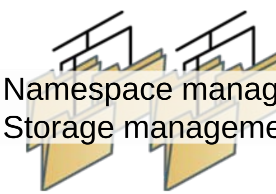
- Computers deal with storage objects
  - aka “blocks”
  - Locations (“addresses”)…and not much else

Teasing the problem apart…

Education SNIA

- People deal with "business objects"
  - aka "files"
  - Names, sizes, access rights, lifetimes…

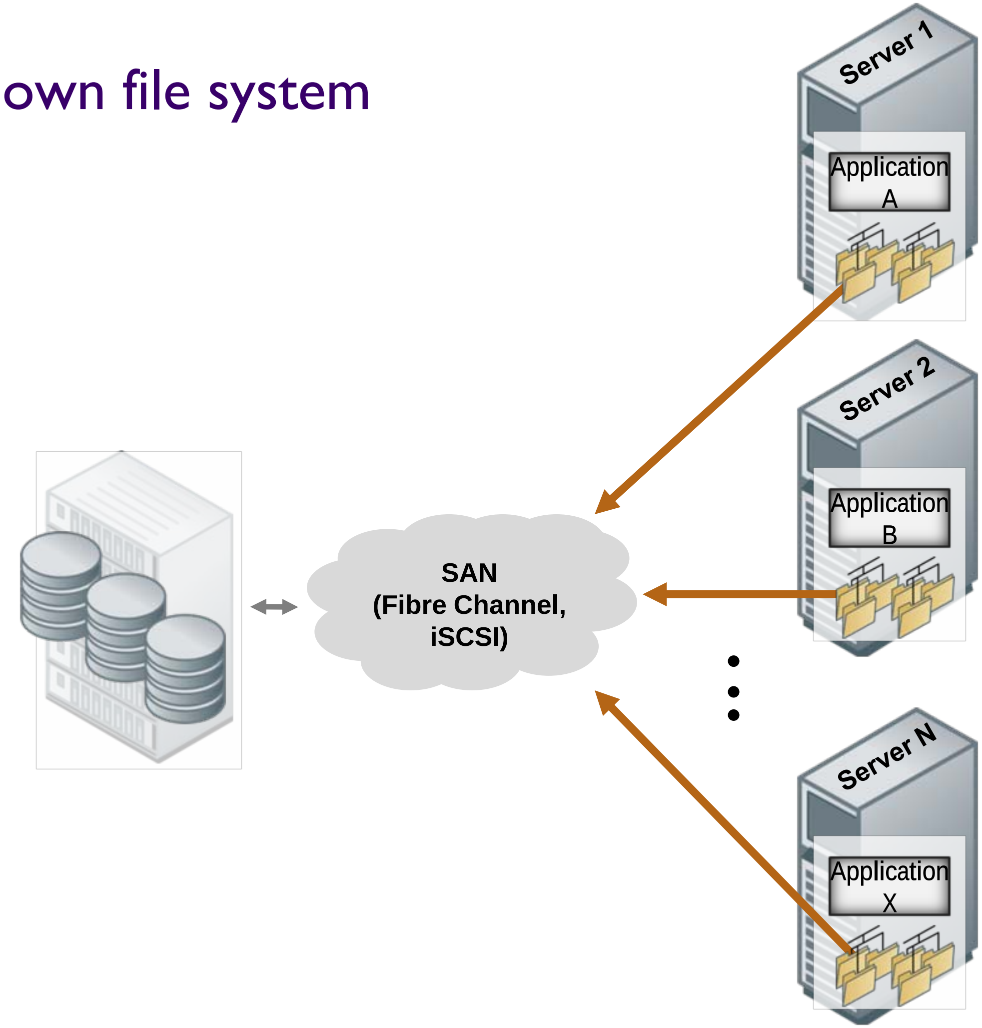
- In between: "file systems"
  - Who is allowed to access `/year2010/march/…/results`?
  - What free space is available for new files ?
  - Where is the data for `/year2010/march/…/results` stored ?



- Computers deal with storage objects
  - aka "blocks"
  - Locations ("addresses")…and not much else

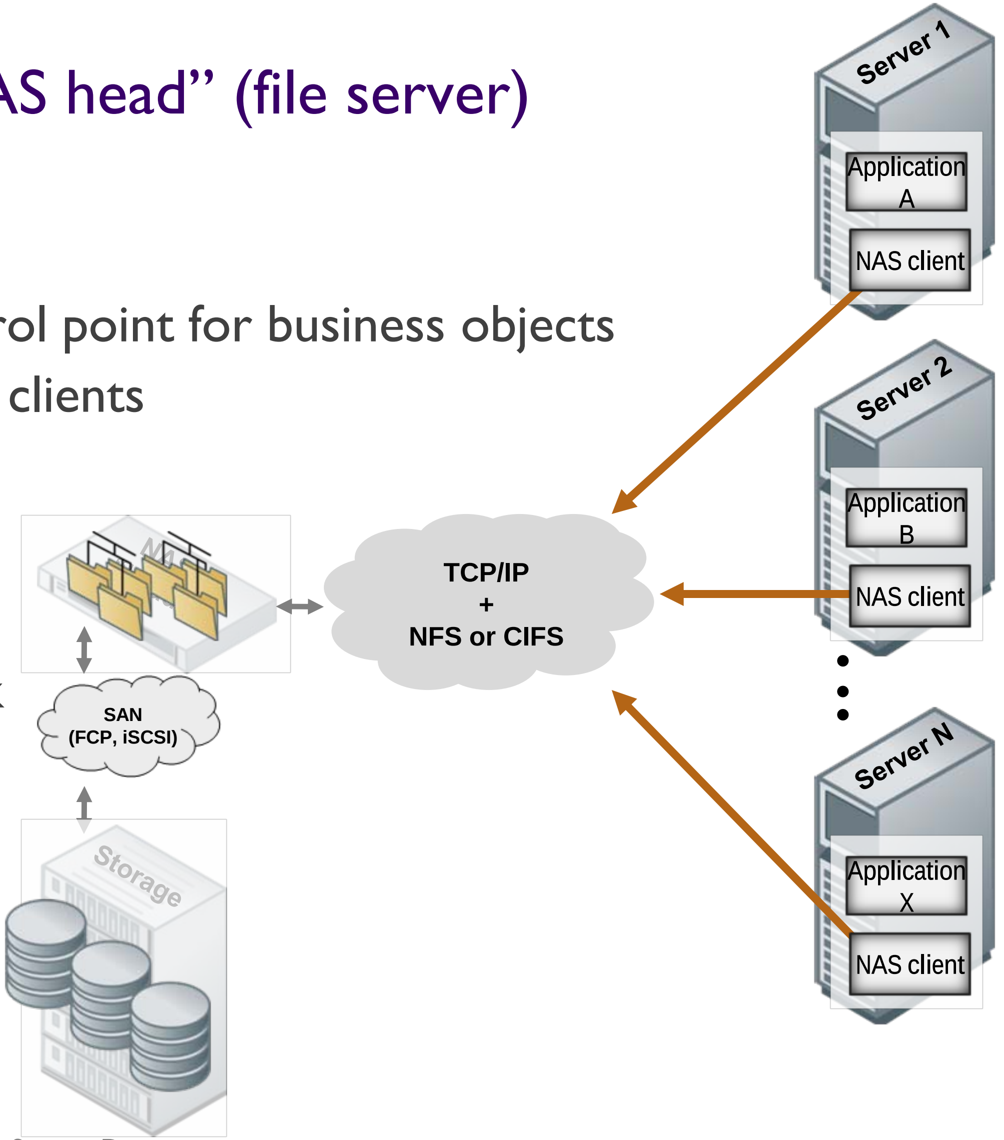
SANs and file systems

- Every client computer has its own file system
- That's good
  - Lots of parallelism
  - Short I/O paths
- …and not so good
  - Lots of file systems to manage
  - Lots of storage access rights to coordinate
- But it works…
  - Cluster file systems
  - SAN file systems



NAS and file systems

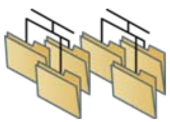
- One file system in the "NAS head" (file server)
- That's good…
  - Single name space and control point for business objects
  - Consistent semantics for all clients
- And not so good…
  - High latency protocols
  - The "NAS head" bottleneck
- But it works…
  - Clustered NAS
  - NAS aggregators



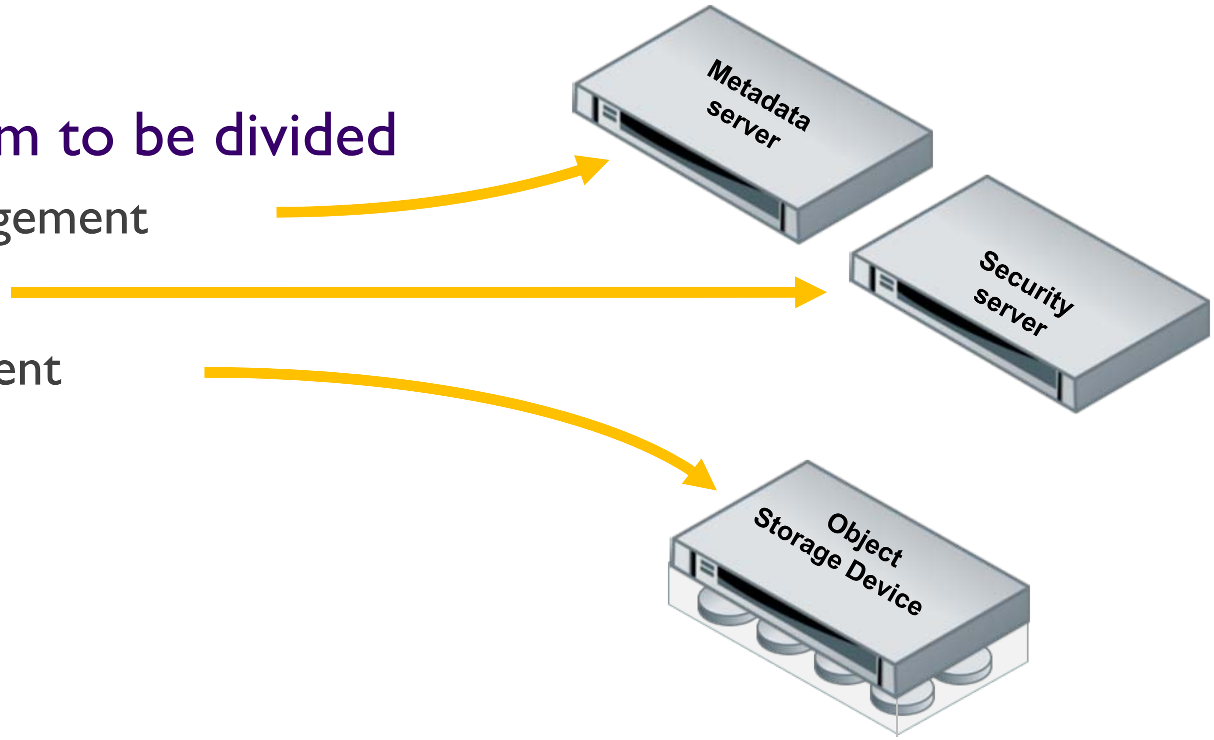
OSDs: a step closer to the ideal

Education SNIA

- The **Object Storage Device** (OSD)
  - Hybrid between a disk and a file server



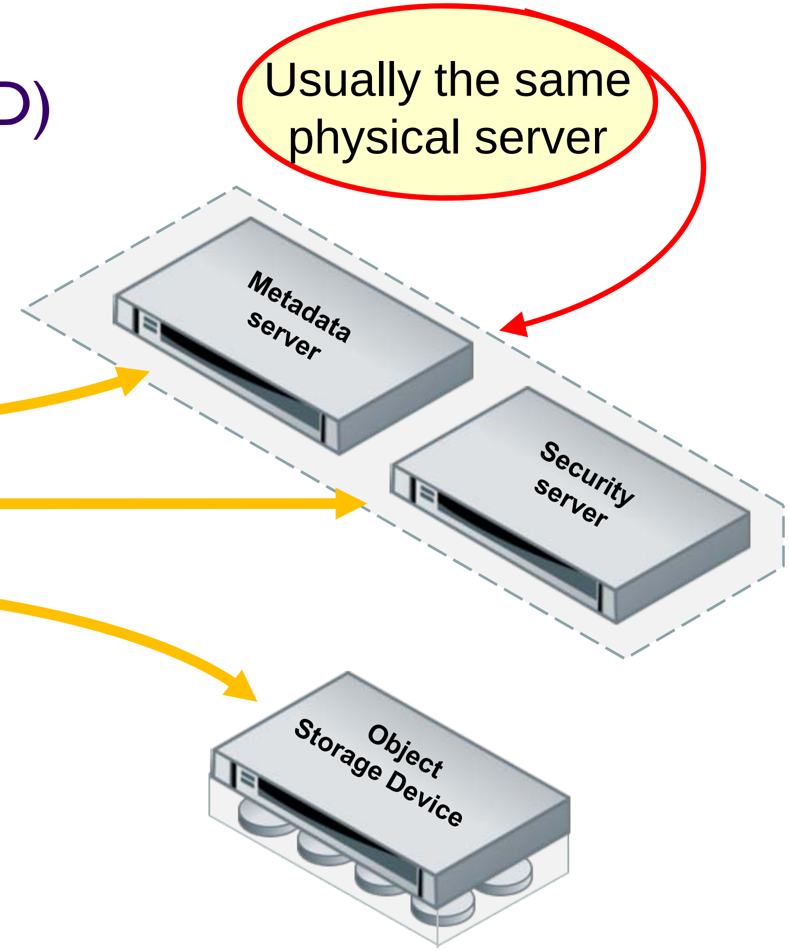
- Allows the problem to be divided
  - Namespace management
  - Access security
  - Storage management



- Enables
  - Extreme scaling
  - High performance
  - Robustness

OSDs relieve the limitations

- The **Object Storage Device** (OSD)
  - Hybrid between a disk and a file server
- Allows the problem to be divided
  - Namespace management
  - Access security
  - Storage management
- Enables
  - Extreme scaling
  - High performance
  - Robustness



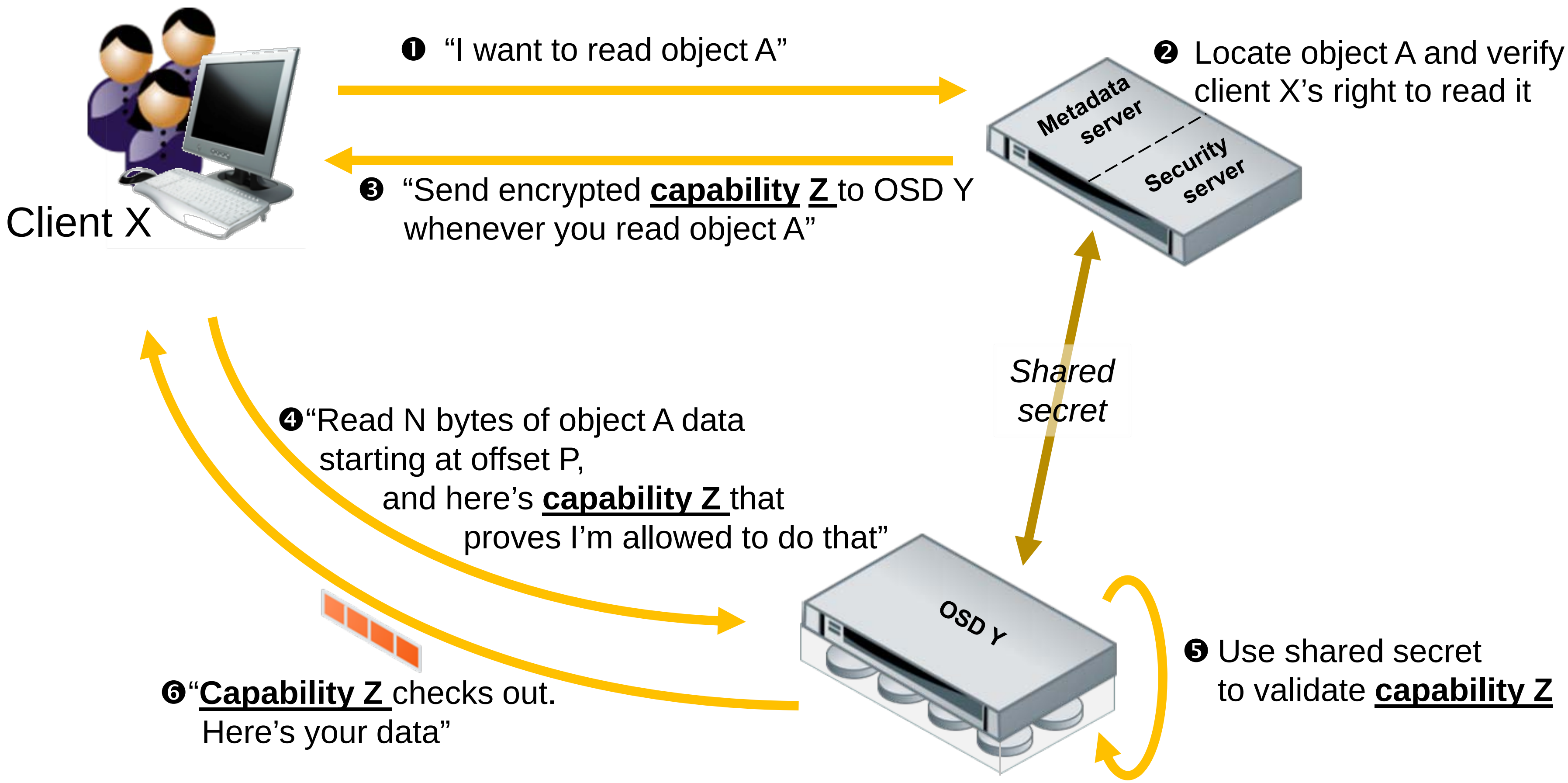
Like a disk, only different

	Disk	OSD
Model	Array of blocks • Number never changes • Size never changes	"Objects" • Created and deleted • Appended and truncated
Operations	Read Write } Disk block range	Create/delete Read/write } Object
Security	Zoning, LUN masking • Applies to entire device	"Capability" • Applies to each IOP
Typical transports	Fibre Channel, SCSI, iSCSI	iSCSI, ***<u>TCP/IP-RPC</u>***

Like a file server, only different

	File server	OSD
Model	Files	Objects
Naming	Hierarchical directory tree Human-readable names	Flat "partitions" Binary "names"
Operations	File: create, delete File block range: read, write, append, truncate	Object: create, delete, Object block range: read, write, append, "punch"
Security	User | group | world × `rwx` Access control lists • Apply to initial file access	Digitally signed "***capabilities***" • Apply to every I/O request

Capabilities



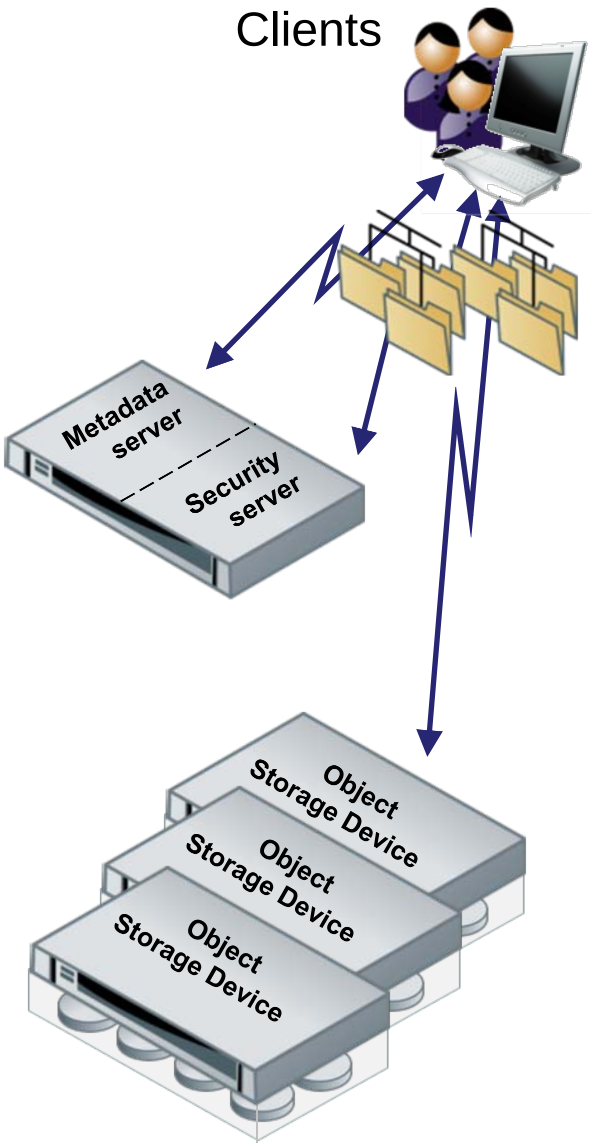
File systems for OSDs

Why not just OSD = file system ?

- Scaling
  - What if there's more data than the biggest OSD can hold ?
  - What if too many clients access an OSD at the same time ?
  - What if there's a file bigger than the biggest OSD can hold ?
- Robustness
  - What happens to data if an OSD fails ?
  - What happens to data if a Metadata Server fails ?
- Performance
  - What if thousands of objects are accessed concurrently ?
  - What if big objects have to be transferred really fast ?

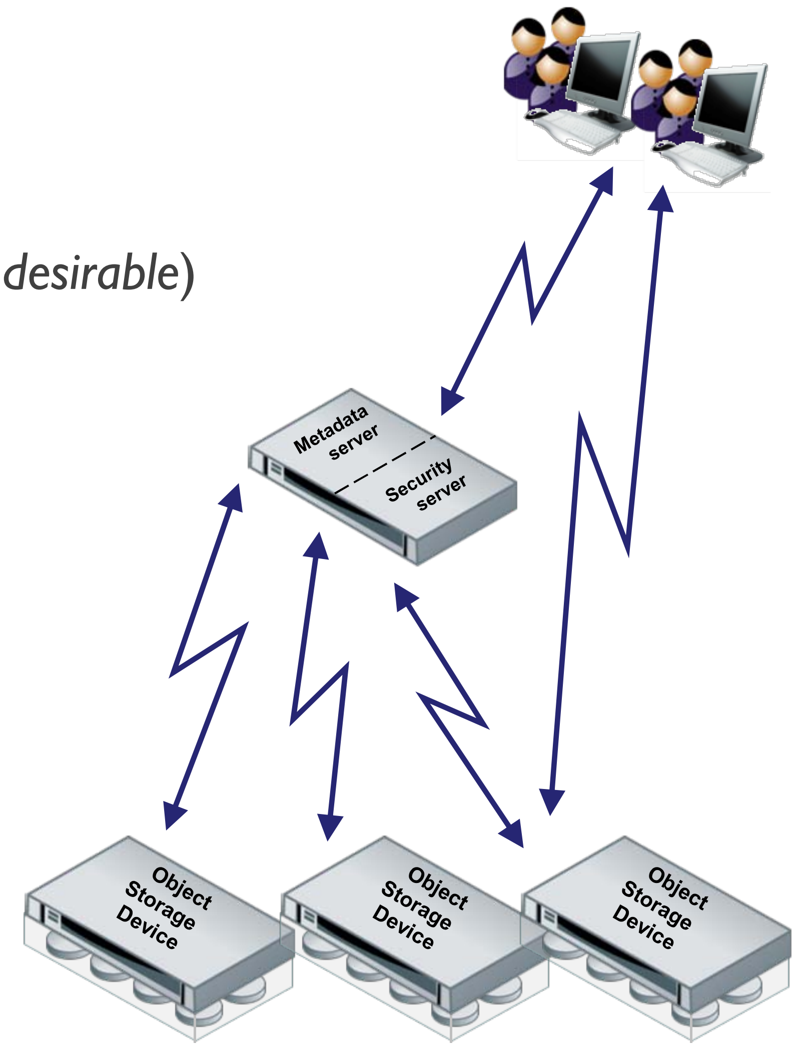
General principle

- Architecture
  - File = one or more *groups of objects* (*usually on different OSDs*)
  - Clients access Metadata Servers to locate data
  - Clients transfer data directly to & from OSDs
- Addresses
  - Scaling
  - Robustness
  - Performance



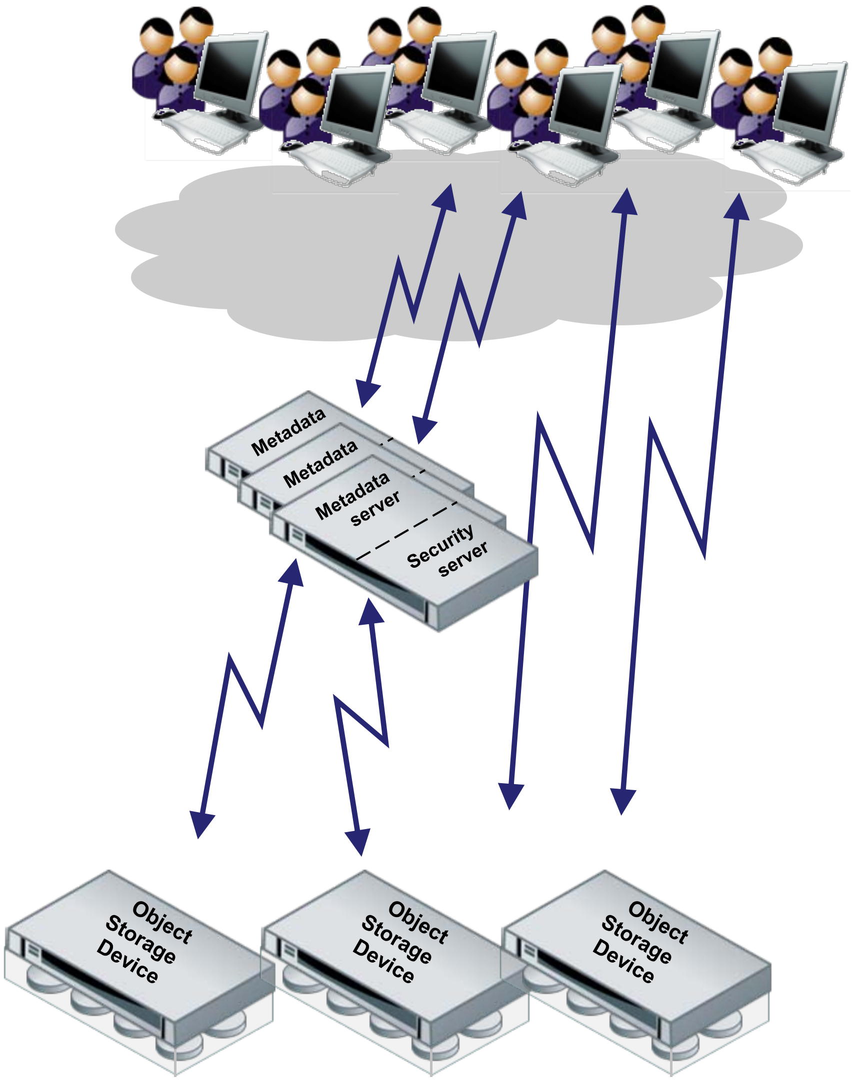
Scaling

- Add OSDs
  - Increase total system capacity
  - Support bigger files:
    (*files can span OSDs if necessary or desirable*)



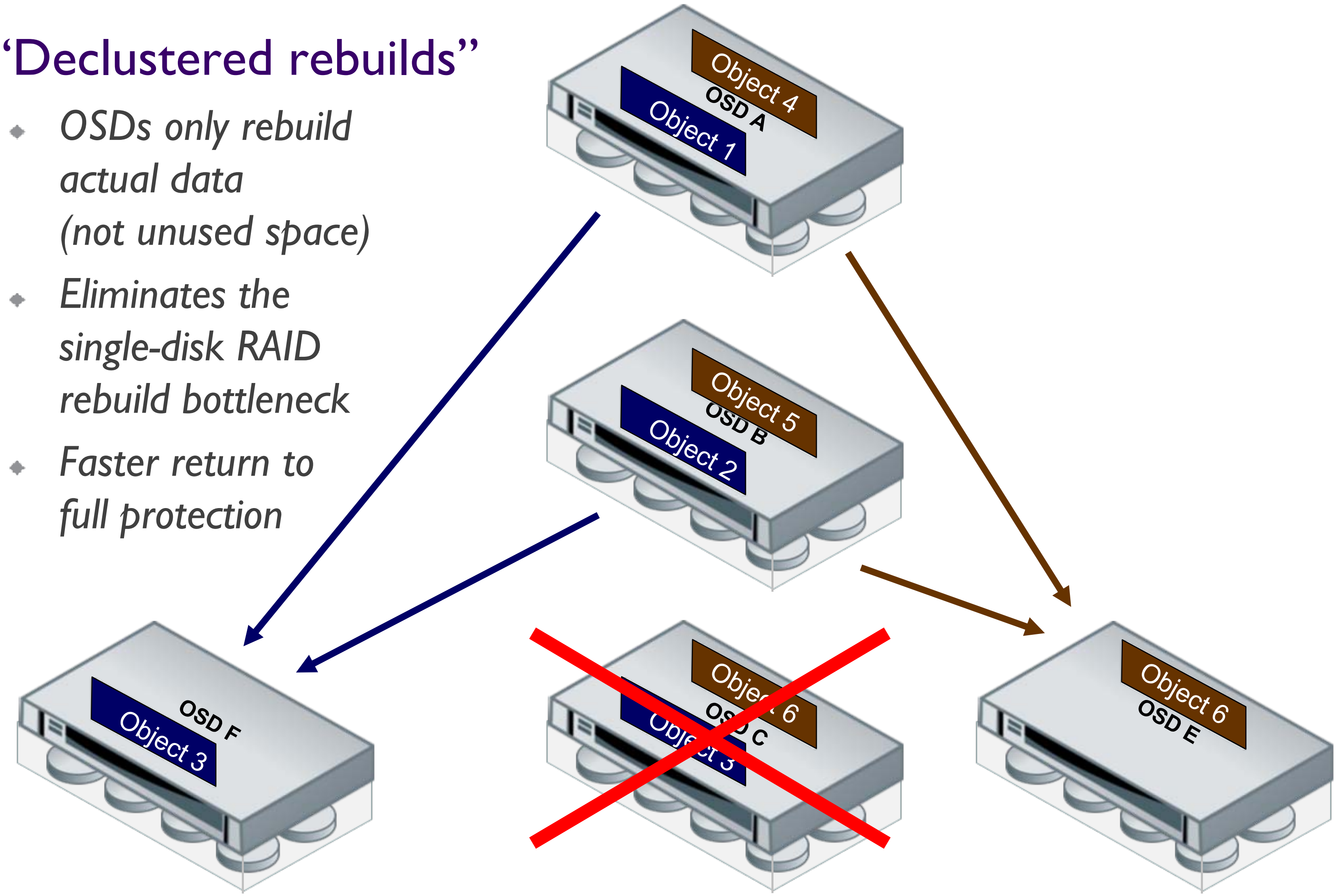
Robustness

- Add metadata servers
  - Resilient metadata services
  - Resilient security services
- Add OSDs
  - Failures affect smaller percentage of system resources
  - Inter-OSD mirroring and RAID
  - Near-online file system checking



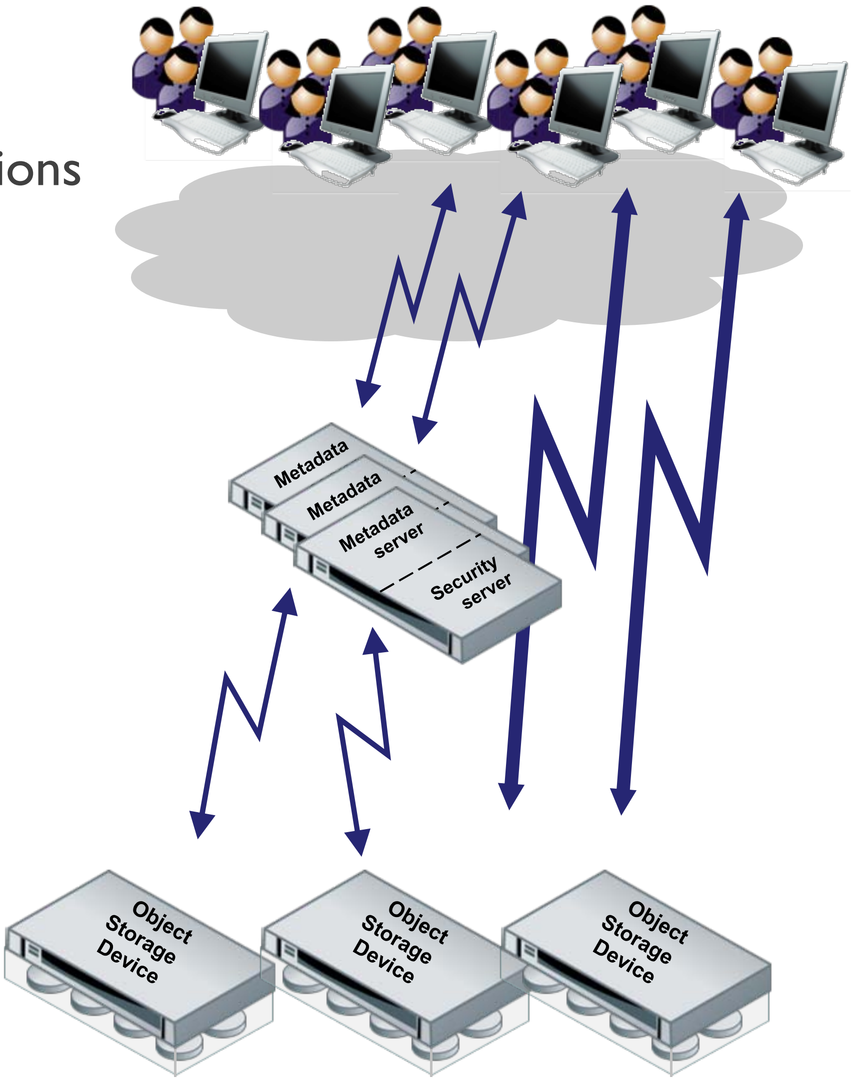
An important advantage

- "Declustered rebuilds"
  - *OSDs only rebuild actual data (not unused space)*
  - *Eliminates the single-disk RAID rebuild bottleneck*
  - *Faster return to full protection*



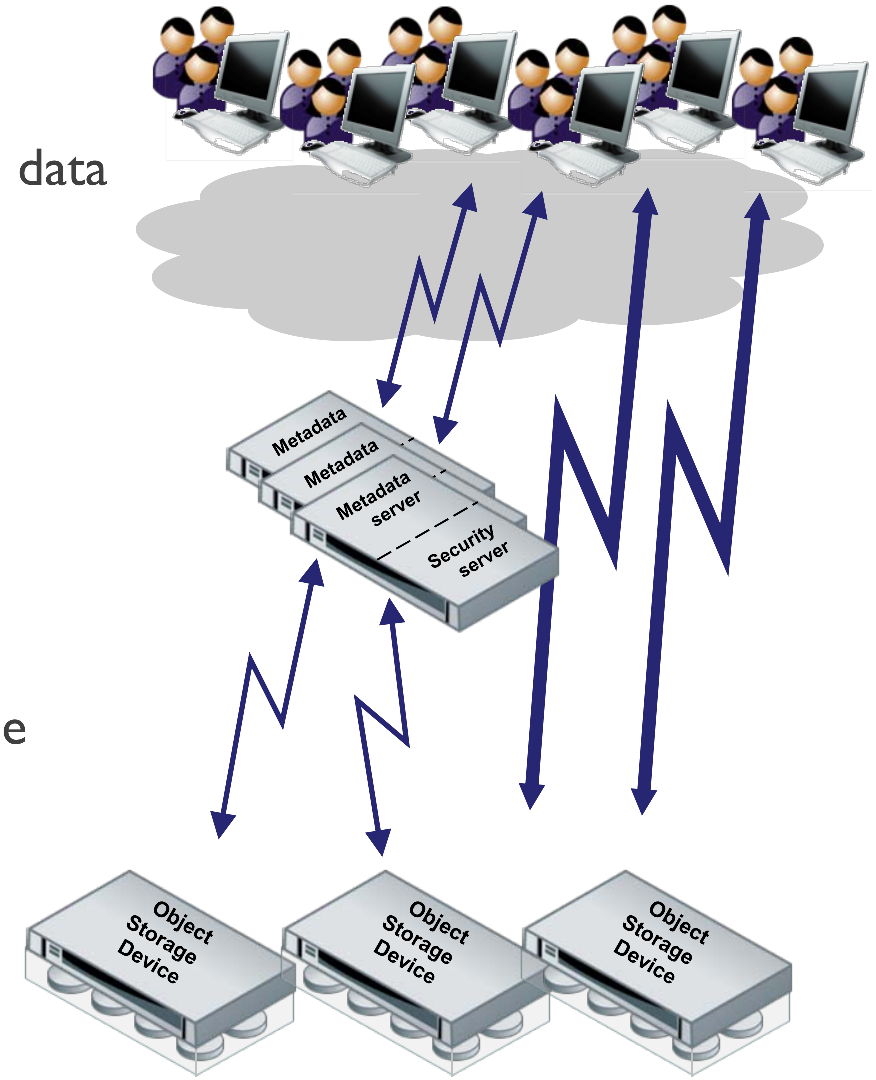
Performance

- Add metadata servers
  - More concurrent metadata operations (**`Getattr`**, **`Readdir`**, **`Create`**, **`Open`**,…)
- Add OSDs
  - More concurrent I/O operations
  - More bandwidth directly between clients and data



Additional advantages of OSD systems

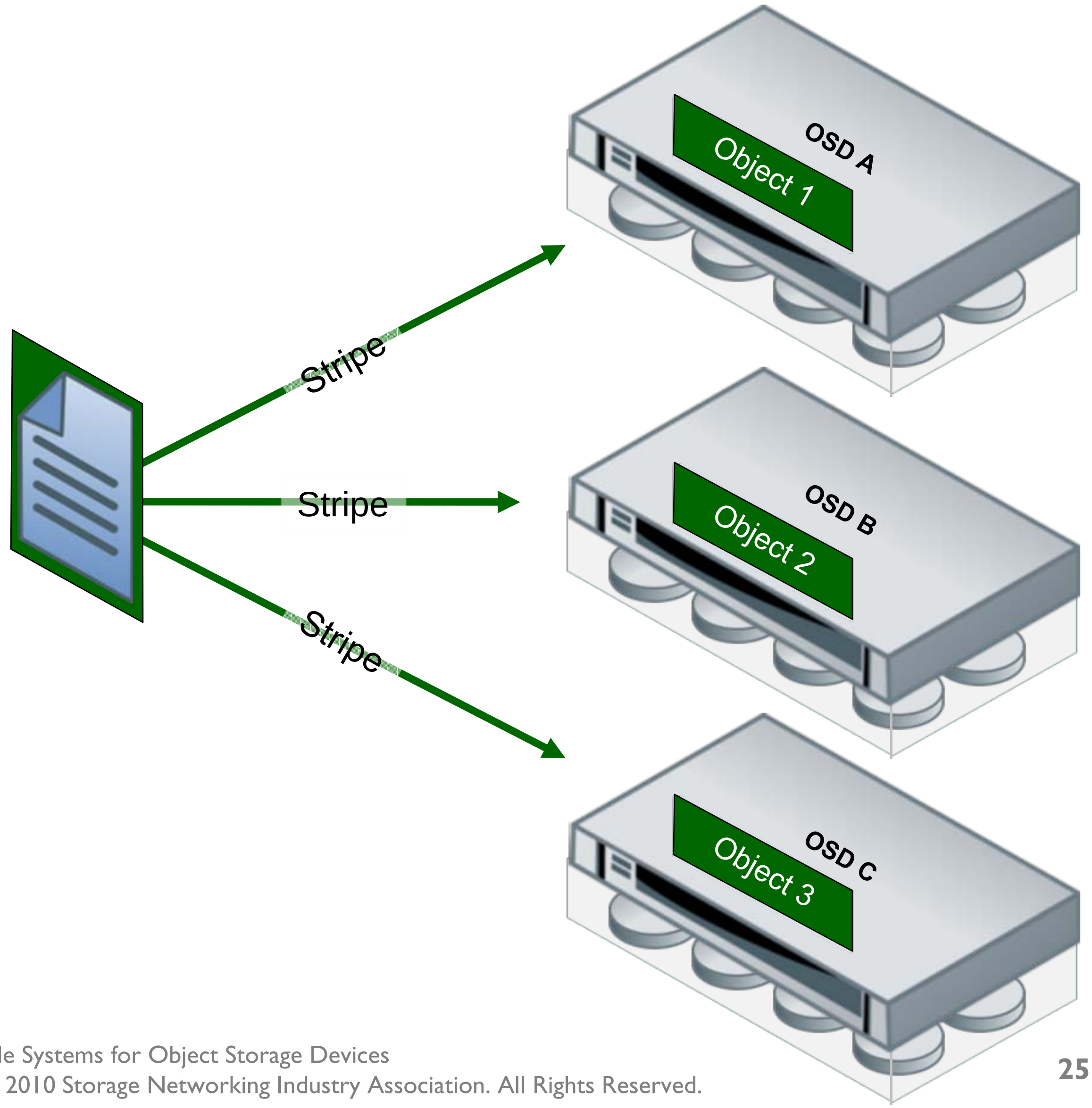
- Optimal data placement
  - Within OSD: proximity of related data
  - Load balancing across OSDs
- System-wide storage pooling
  - Across multiple file systems
- Storage tiering
  - ***Per-file control*** over performance and resiliency



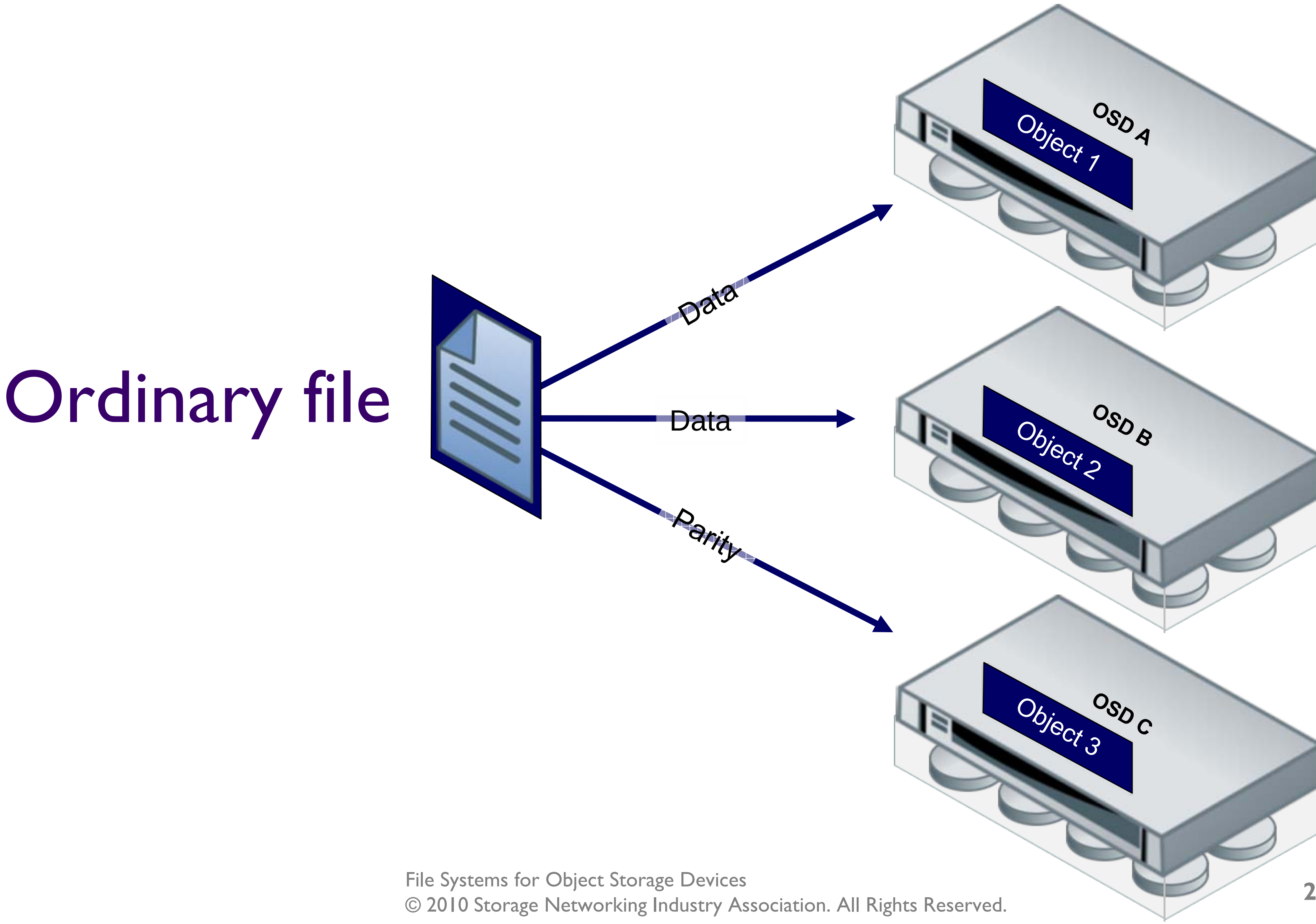
Per-file control

Education SNIA

Scratch file

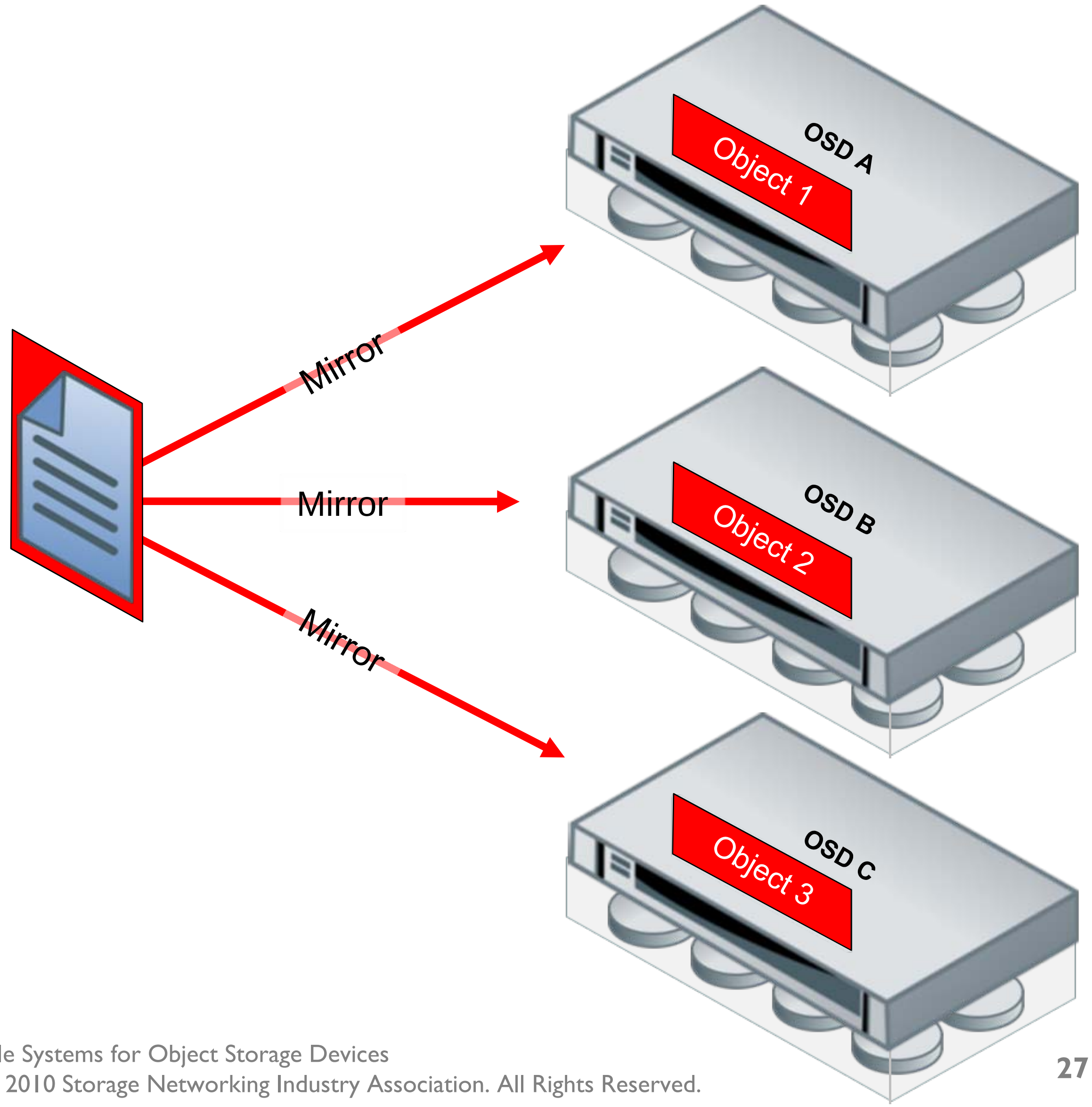


Per-file control: RAID



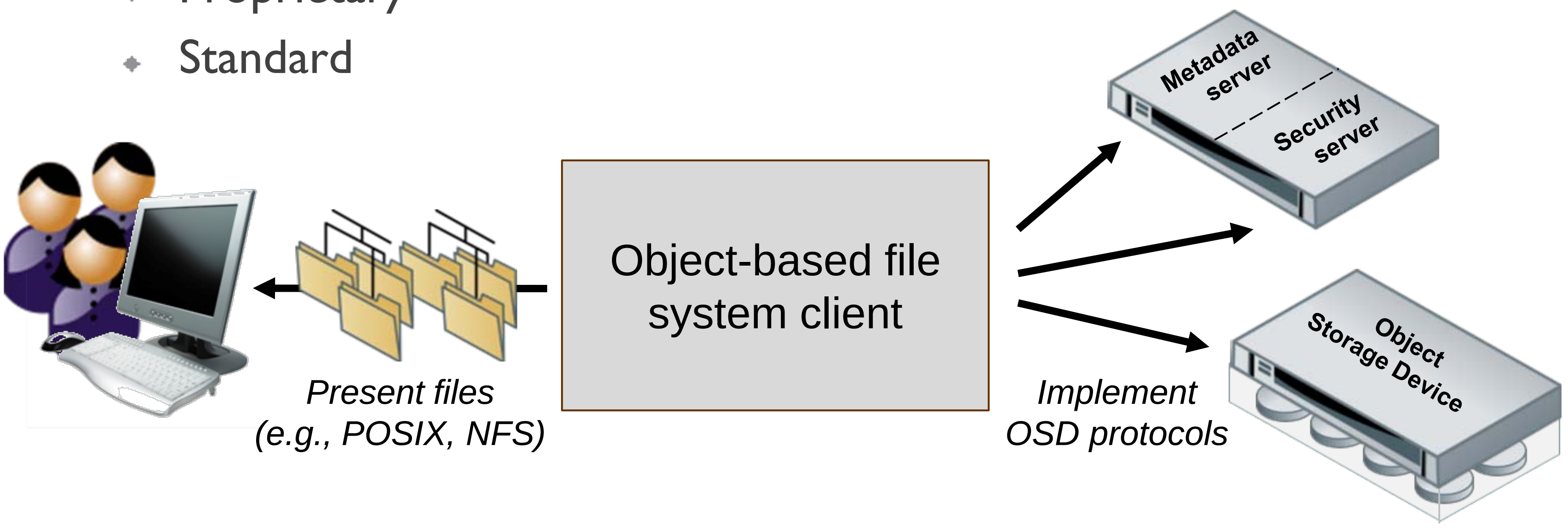
Per-file control: mirroring

Critical file



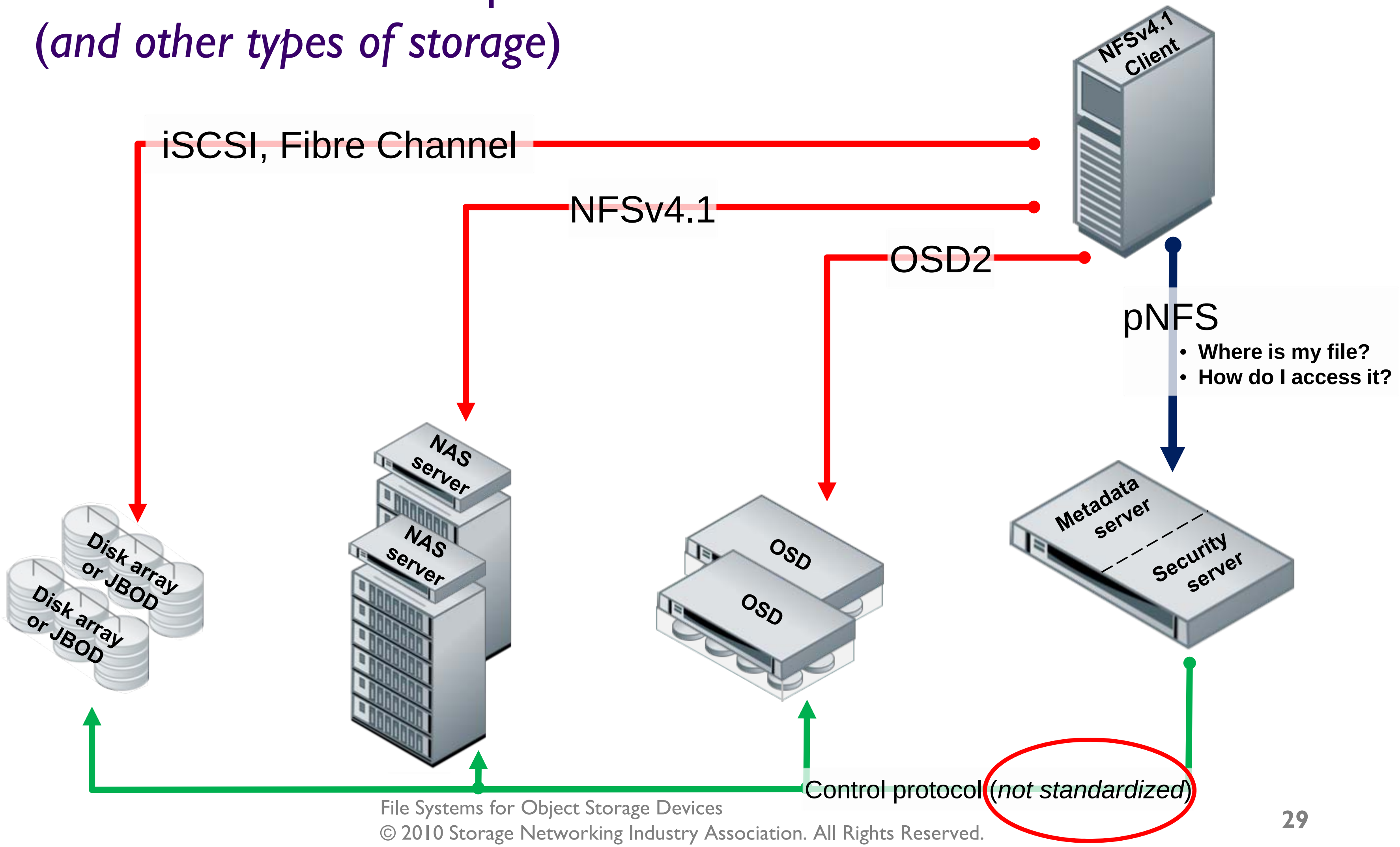
Accessing OSD-based file systems

- It's not
  - SCSI
  - NFS/CIFS
- Needs a "client component"
  - Proprietary
  - Standard



pNFS

- A standard file access protocol for OSD
  (and other types of storage)



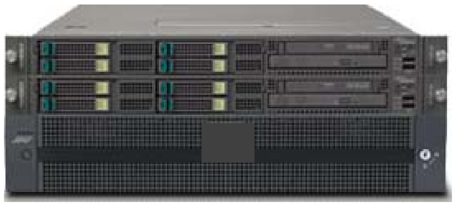
OK, so you buy into the concept

Evaluating vendors and products

- Two basic classes of OSD-based storage systems

  - Server with embedded disks
  - Software-based OSD functionality

  - Purpose-built OSD "bricks"
  - Highly-integrated system



Which type is optimal depends on what's important to you

Look before you leap

- Be clear about the problem(s) you are trying to solve
  - Massive capacity ?
  - Very large single name space ?
  - Secure distributed access to files ?
  - File integrity (AV, DLP,…) ?
  - Extreme resiliency ?
  - Network simplification ?
  - Performance you can't get from your current architecture ?
  - Administrative simplicity you can't get from your current architecture ?

Look before you leap

- Be clear about your IT organization's capabilities
  - Favor turn-key solutions vs. integration of components ?
  - Conservative or open to new technologies and techniques ?
  - Experience with large-scale computing and data ?
  - SAN or NAS oriented ?
  - Network expertise ?
  - Cross-data center integration ?

OSD storage system checklist

Vendor / product maturity	How does the vendor rate ?	How important is it to me ?
Years of field experience		
Number of versions / updates / component refreshes		
Number of installations		
Largest (smallest) installation		
Geographic coverage		
Dominant applications / data access profiles		
Types of client platforms installed		
Types of backbone networks installed		

OSD storage system checklist

Flexibility	How does the vendor rate ?	How important is it to me ?
Largest (and smallest) supported configuration		
Increments of expansion (e.g., *capacity, cache, metadata processing, bandwidth*)		
Support for "mix-n-match" components (e.g., *different generations, disk capacities,…*)		
Standards compliance and alternate component sources		

OSD storage system checklist

Education SNIA

Availability / data protection	How does the vendor rate ?	How important is it to me ?
Data protection models *(e.g., mirror, RAID5-6,…)*		
Fault protection domains *(e.g., disks, OSDs, network links & switches…)*		
Sustainable combinations of faults		
Monitoring, fault detection and notification mechanisms		
Self-healing		
Component hot-swap		
Online hardware upgrade		
Rolling software upgrade		

OSD storage system checklist

Functionality	How does the vendor rate ?	How important is it to me ?
Client access networks and features (e.g., *link aggregation*)		
Client access protocols (e.g., *NFS, REST,…*)		
Application semantics (e.g., *POSIX, NFS,…*)		
Automatic load / capacity balancing		
Advanced functions (e.g., *backup integration, archiving, duplicate elimination, anti-virus, data loss prevention, data classification,…*)		
Integration with system management frameworks		
Disaster recoverability (*remote replication and failover*)		

OSD storage system checklist

Security	How does the vendor rate ?	How important is it to me ?
Client authentication		
LDAP / AD support		
Per-client / per-user data access authorization		
Access control lists		
Digitally signed capabilities		
Protection against misbehaving clients		
Data encryption on the network		
Data encryption on media		

Summary

- OSDs: new technology that enables file storage systems of extreme
  - Scale
  - Robustness
  - Performance

  …based on custom or low-cost "commodity" components

- OSDs require specialized file systems
  - Metadata/security servers
  - Client components

- OSD-based storage system user community
  - Initially high-performance computing
  - Today: being adopted for data-intensive applications throughout the enterprise storage market
    - Financial services, telecom, biotech, oil & gas, aerospace, semiconductor

- As a promising new technology, it deserves a (careful) look

Q&A / Feedback

- Please send any questions or comments on this presentation to SNIA: trackfilesystems@snia.org

Many thanks to the following individuals for their contributions to this tutorial.

- SNIA Education Committee

Craig Harmer
Julian Satran
Rich Ramos
Erik Riedel
Mike Mesnier
Ralph Weber

Appendix

Further Reference

- Academic research
  - www.pdl.cmu.edu
  - www.dtc.umn.edu
- Standards work
  - www.snia.org/apps/org/workgroup/osd
  - www.t10.org/drafts.htm
  - www.ietf.org/dyn/wg/charter/nfsv4-charter.html
- Industry research & development
  - www.sun.com/lustre
  - www.opensolaris.org/os/project/nfsv41/
  - www.panasas.com