



Education

The File Systems Evolution

Presenter: Craig Harmer
Senior Technologist, Hitachi Data Systems

Author: Christian Bandulet
Principal Engineer, Oracle

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individuals may use this material in presentations and literature under the following conditions:
 - ◆ Any slide or slides used must be reproduced without modification
 - ◆ The SNIA must be acknowledged as 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 nor should be construed as legal advice or opinion. If you need legal advice or legal opinion please contact an attorney.
- The information presented herein represents the Author's personal opinion and current understanding of the 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.

➤ The File Systems Evolution

- ◆ File Systems impose structure on the address space of one or more physical or virtual devices. Starting with local file systems, over time additional file systems appeared focusing on specialized requirements such as data sharing, remote file access, distributed file access, parallel files access, HPC, archiving, security etc. Due to the dramatic growth of unstructured data, files as the basic units for data containers are morphing into file objects. This presentation will categorize and explain the basic principles of currently available file systems (e.g. Local FS, Shared FS, SAN FS, Clustered FS, Network FS, Distributed FS, Parallel FS, ...). It will also explain technologies like Scale-Out NAS, NAS Aggregation, NAS Virtualization, NAS Clustering, Global Namespaces, Parallel NFS. All of these files system categories and technologies are complementary. They are all being enhanced in parallel with additional value added functionality. New file system architectures will also be developed and some of them will be blended with these architectures in the future.

Check Out Other Tutorials



Check out SNIA Tutorial:
pNFS & NFSv4.2: A Filesystem
for Grid, Virtualization, and
Database



Check out SNIA Tutorial:
Leveraging Flash Memory in
Enterprise Storage



Check out SNIA Tutorial:
Primary Data Optimization:
What It Should Be



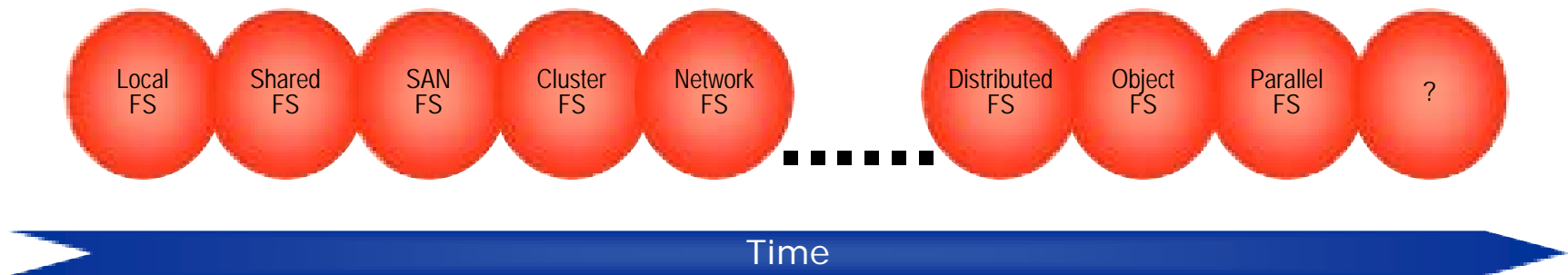
Check out SNIA Tutorial:
Practical Secure Storage: A
Vendor Agnostic Overview



Check out SNIA Hands-On Lab:
Advanced Data Protection and Reduction

The File Systems Evolution

- File systems have evolved over time to scale with storage systems and to fulfill new requirements
- Starting with local file systems, over time additional file systems appeared focusing on specialized requirements such as data sharing, remote file access, distributed file access, parallel files access, HPC, archiving, etc.

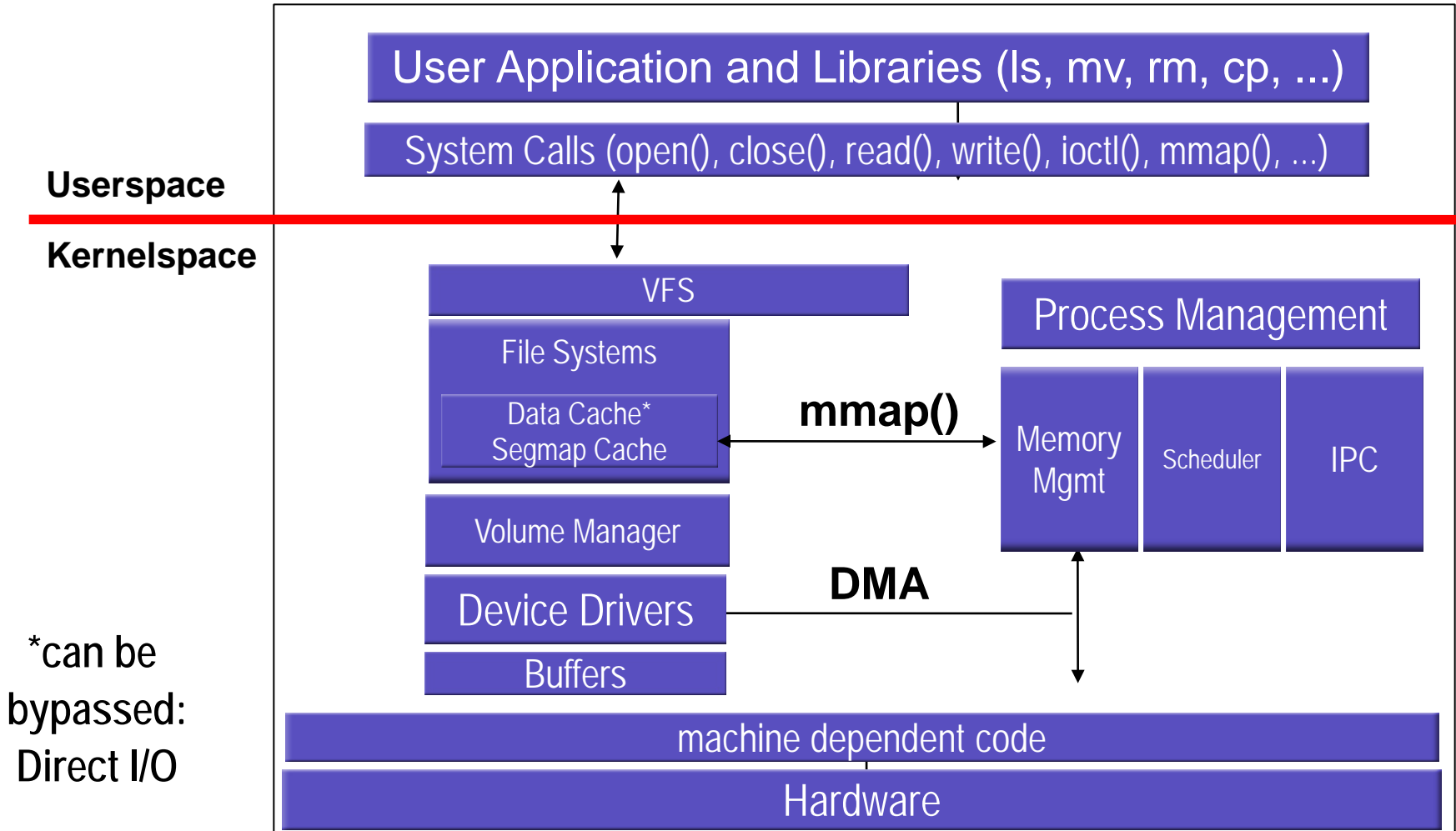


Note: The picture above does not reflect the exact sequence in which the files system types appeared. Some of them actually appeared in parallel. In particular, It is also not the intention to indicate that a new file system replaces its predecessors. Instead they are targeting complimentary objectives.

Agenda

- **File System Basics**
- File Systems Taxonomy
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- Distributed FS
- Distributed Parallel FS
- Scale-Out NAS
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments

File System & Operating System

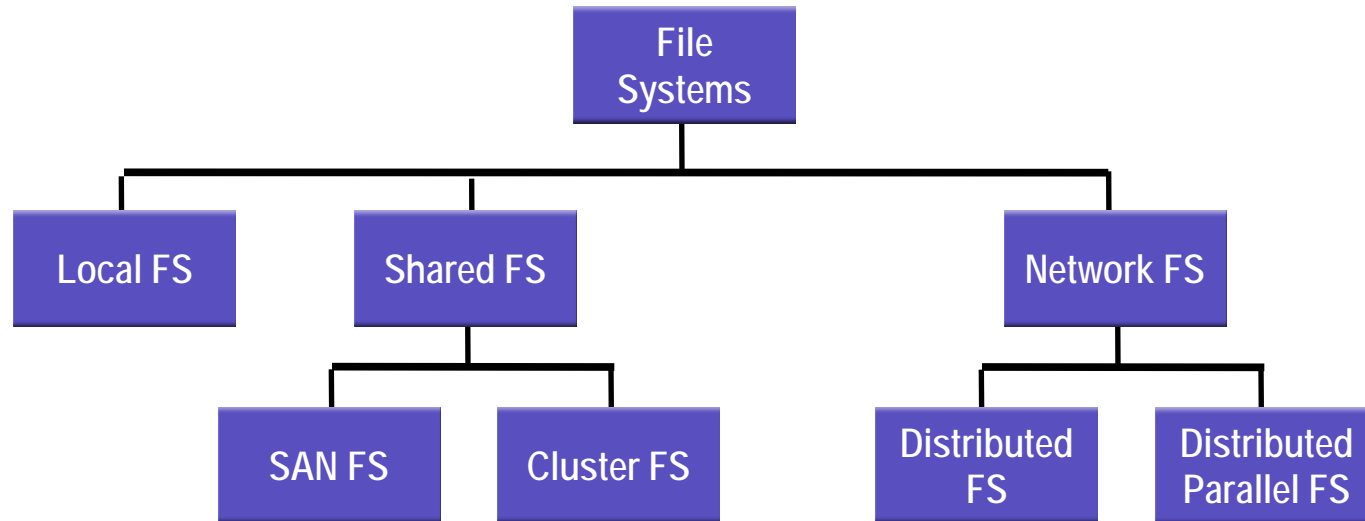


- File System Basics
- **File System Taxonomy**
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- Distributed FS
- Distributed Parallel FS
- Scale-Out NAS
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments

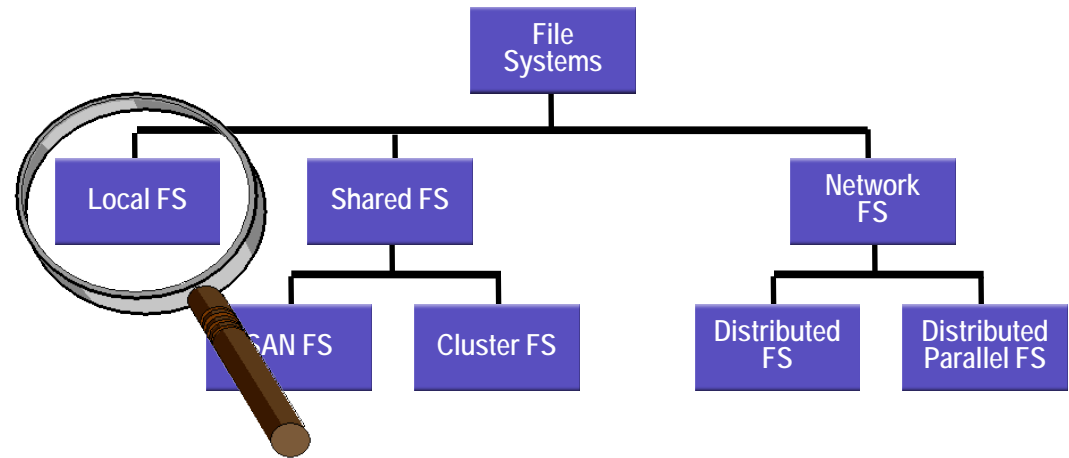
A Plethora of File Systems

- There's a file system for every letter in the alphabet and then some!
- Some you may recognize: AFS, CFS, GFS, JFS, NFS, UFS, ZFS
- There's a more extensive list in the Appendix

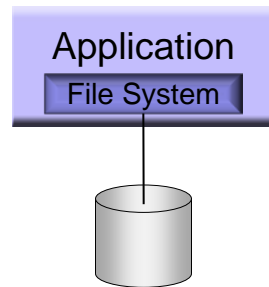
File System Taxonomy



- File System Basics
- File System Taxonomy
- **Local FS**
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- Distributed FS
- Distributed Parallel FS
- Scale-Out NAS
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments



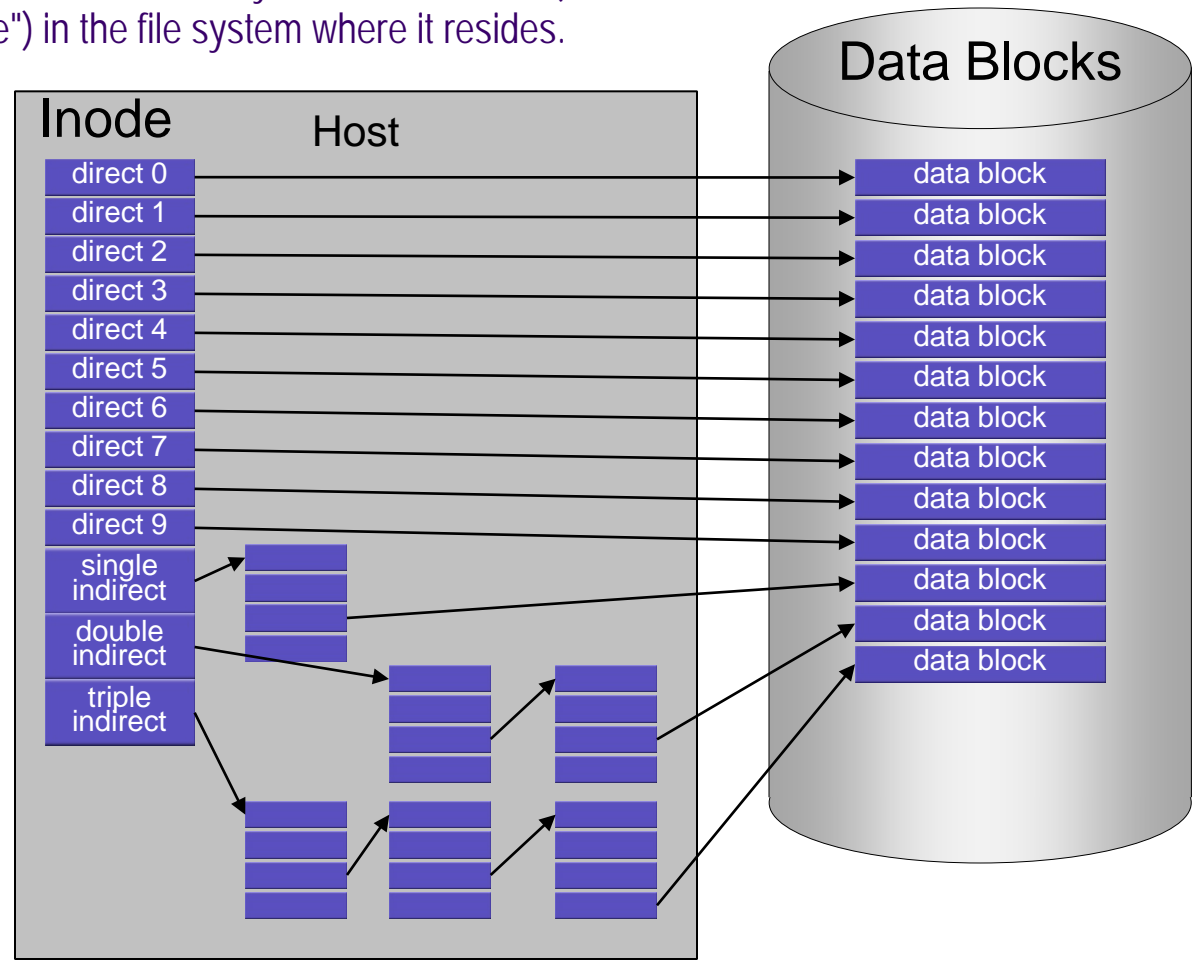
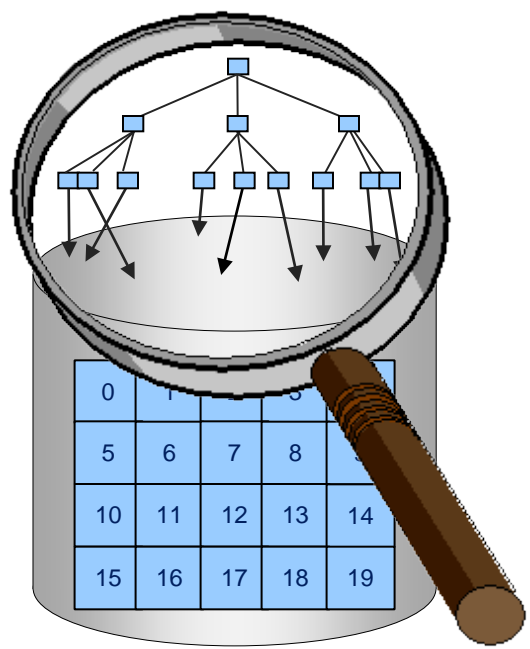
Local FS



➤ FS is **co-located** with application server

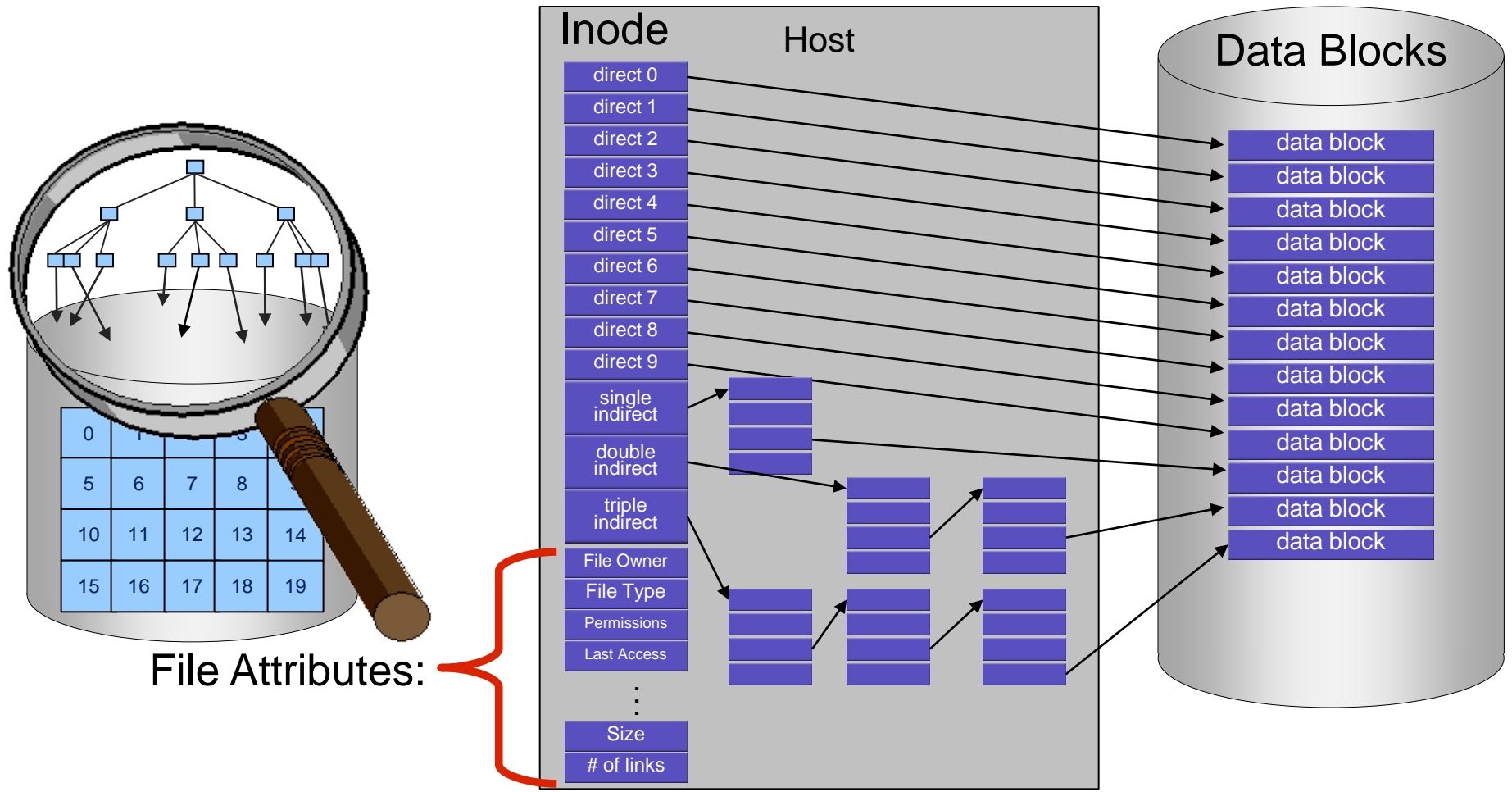
Traditional File System - Inode

When a file system is created, data structures that contain information about files are created. Each file has an inode and is identified by an inode number (often referred to as an "i-number" or "inode") in the file system where it resides.

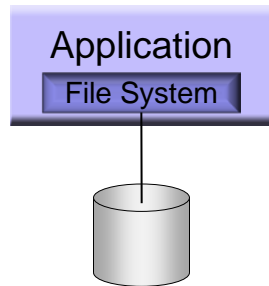


Traditional File System - Inode

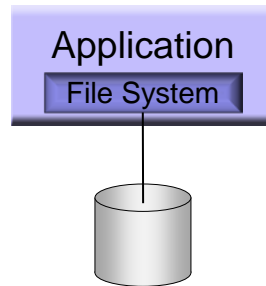
The inode also contains file attributes...



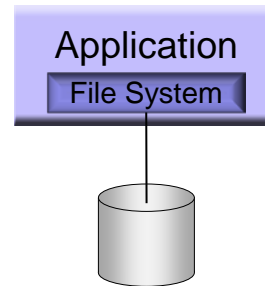
Local FS



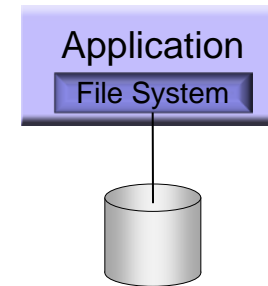
Local FS



Local FS

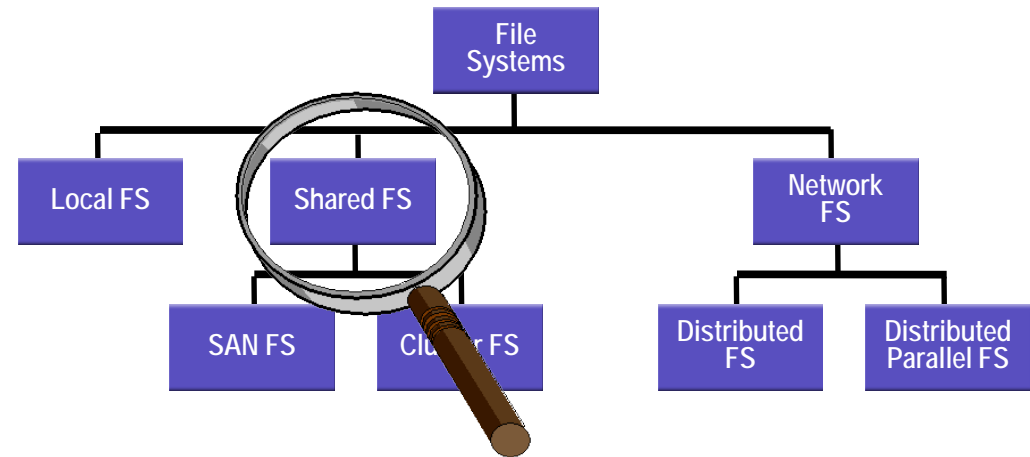


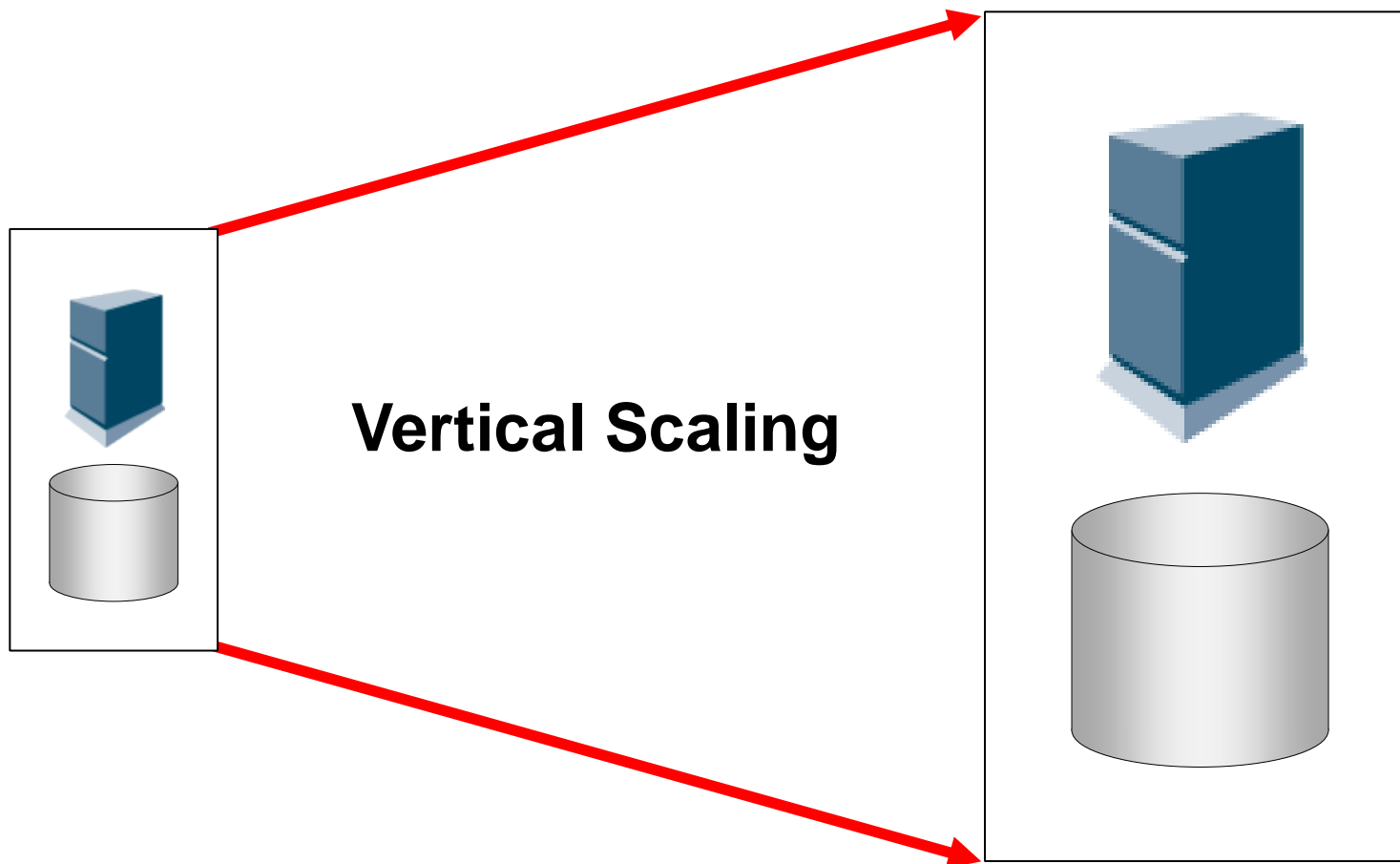
Local FS



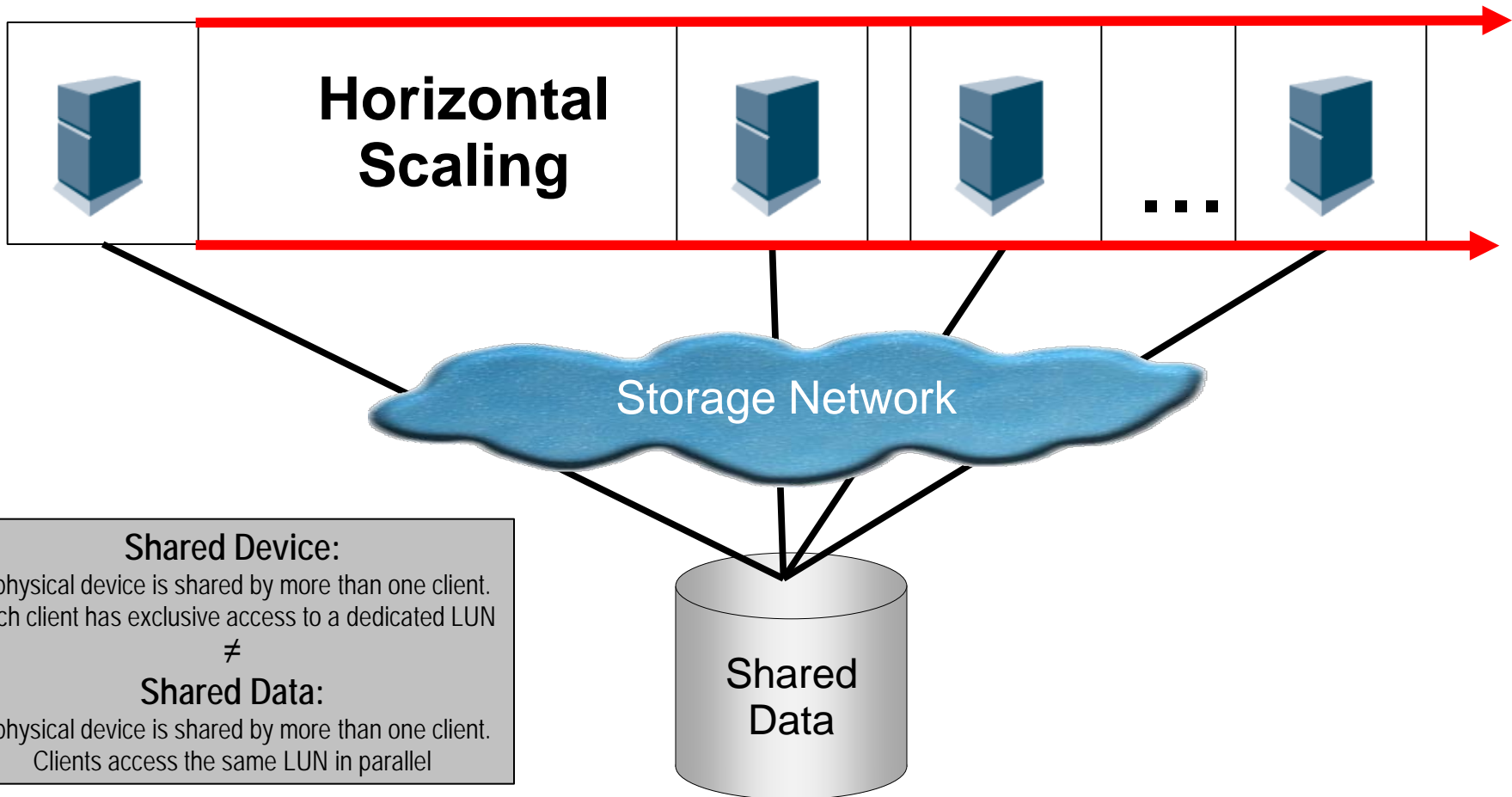
➤ **Islands of storage** (no data sharing)

- File System Basics
- File System Taxonomy
- Local FS
- **Shared FS / Global FS**
 - ◆ **SAN FS, Cluster FS**
- Network FS
- Distributed FS
- Distributed Parallel FS
- Scale-Out NAS
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments



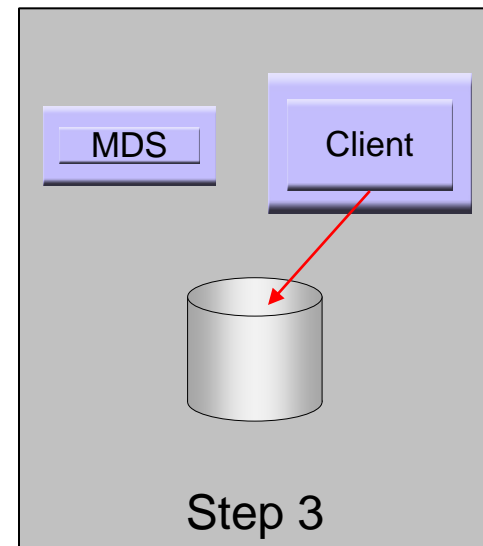
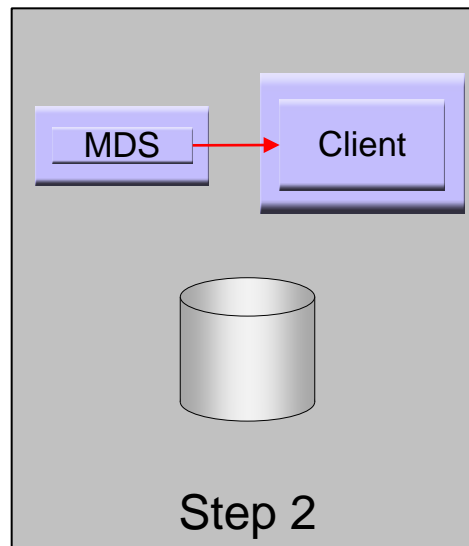
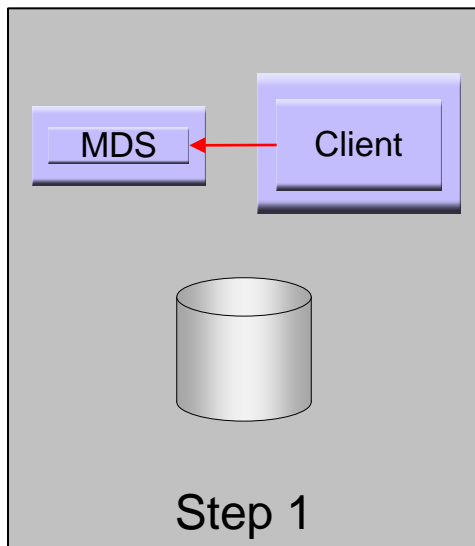


Scale-Out with Shared FS

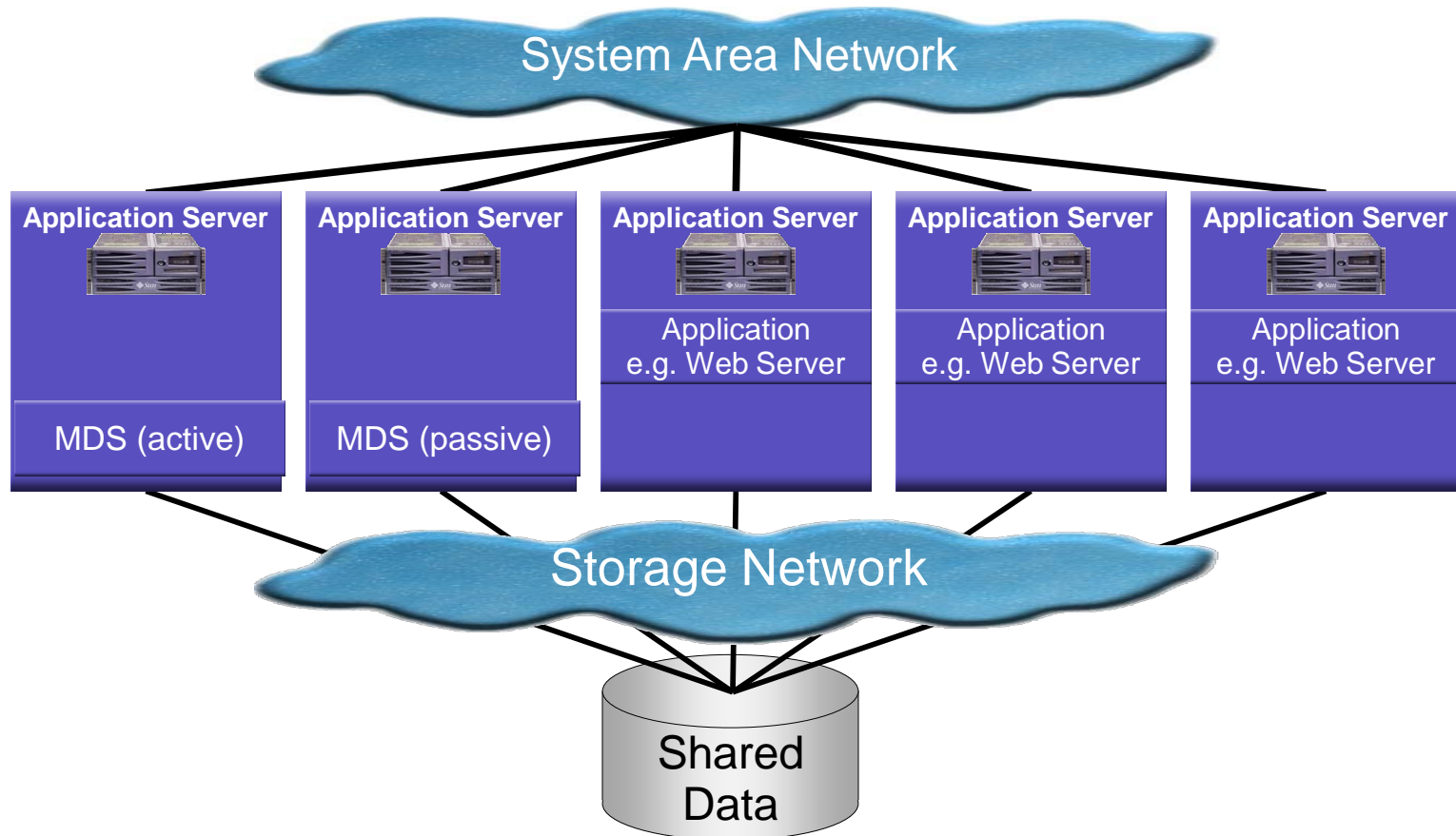


Shared FS / Global FS Data Access

- Separation between logical and physical placement
- Separate Metadata Server (MDS)
- File access is a three-step transaction...

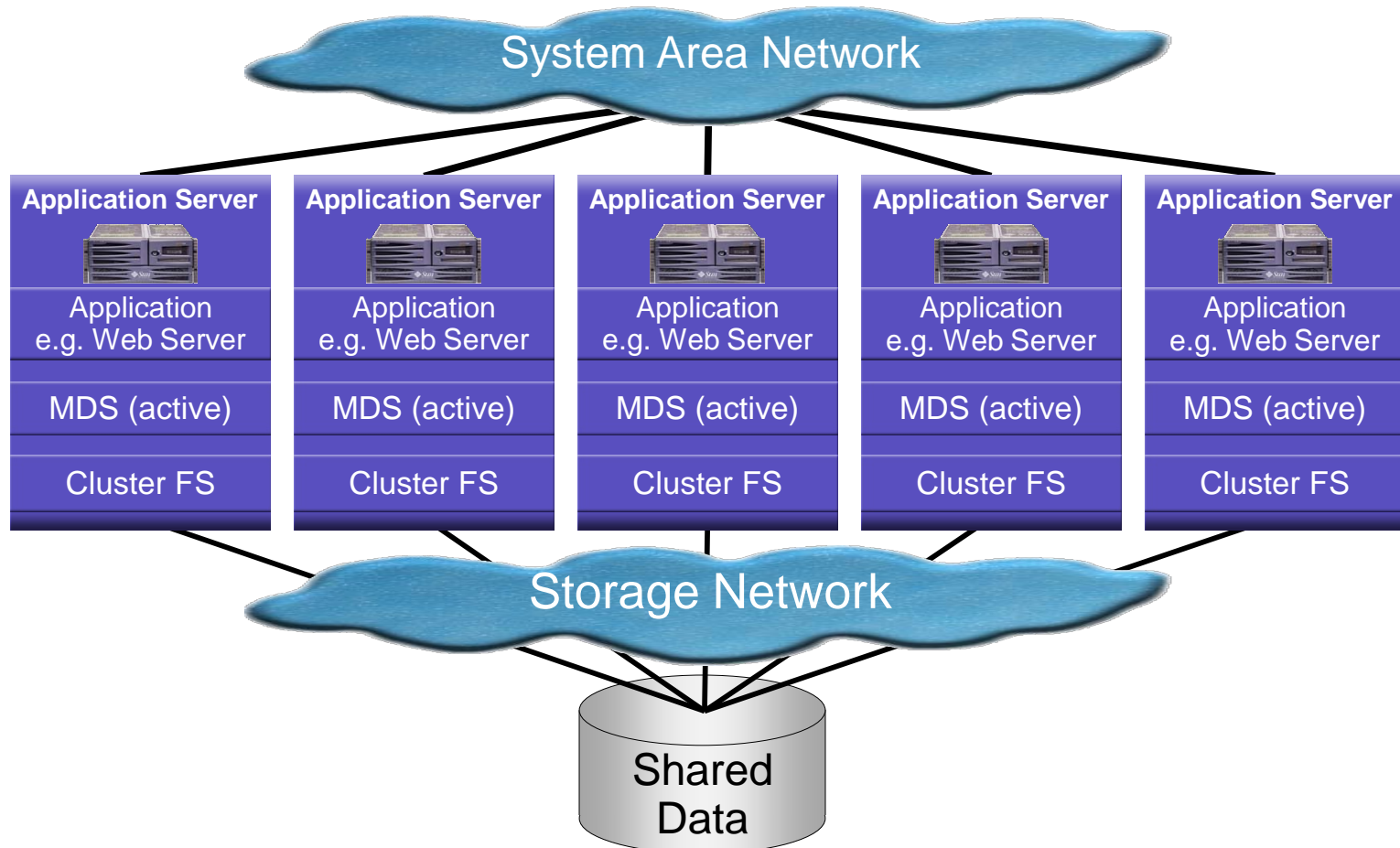


Shared FS / Global FS – SAN FS



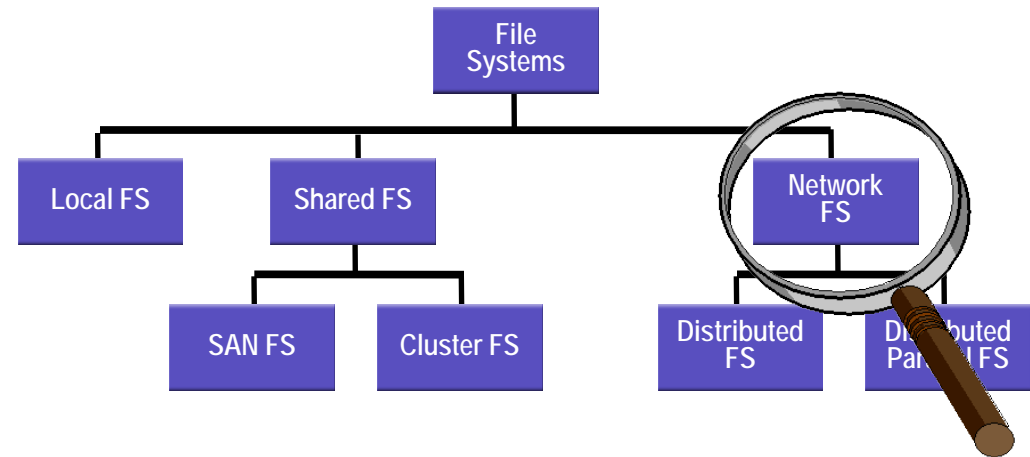
- MDS is not part of each node (i.e. **master/slave - asymmetric**)
- Typically homogenous with **number of nodes limited only by size of MDS**
- Distance between nodes limited only by size of SAN

Shared FS / Global FS – Cluster FS



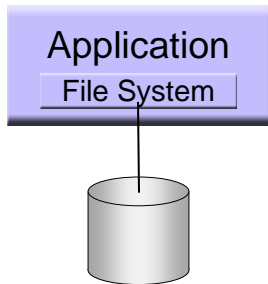
- MDS is part of each (cluster) node (**i.e. peer-to-peer - symmetric**)
- **Typically homogeneous with somewhat limited number of nodes**
- **Limited distance** between (cluster) nodes

- File System Basics
- File System Taxonomy
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- **Network FS**
- Distributed FS
- Distributed Parallel FS
- Scale-Out NAS
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments

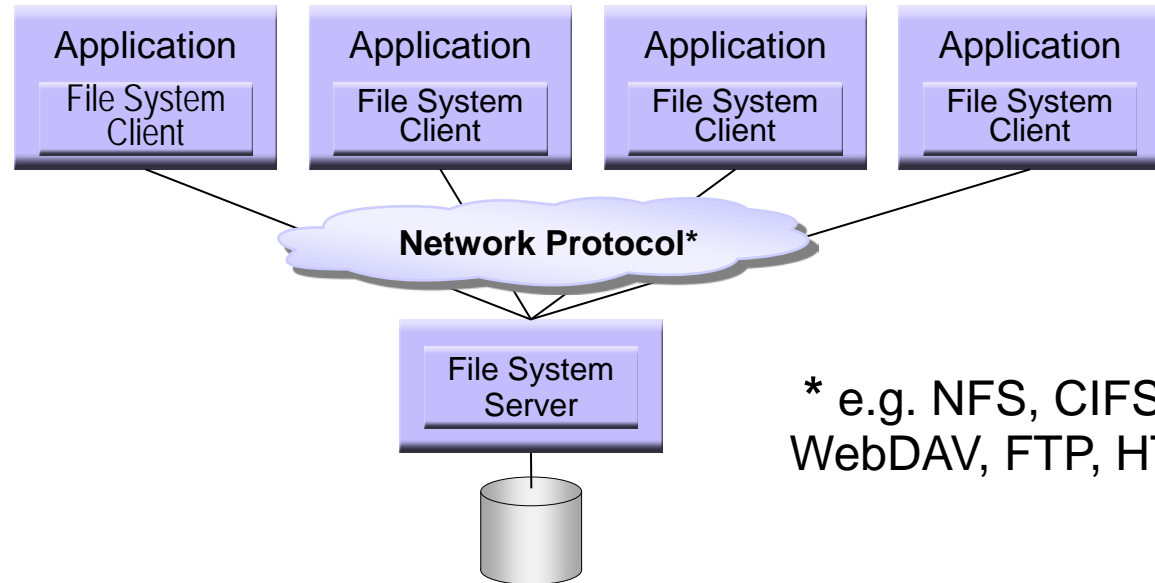


Network File Systems - aka Proxy FS

Local FS



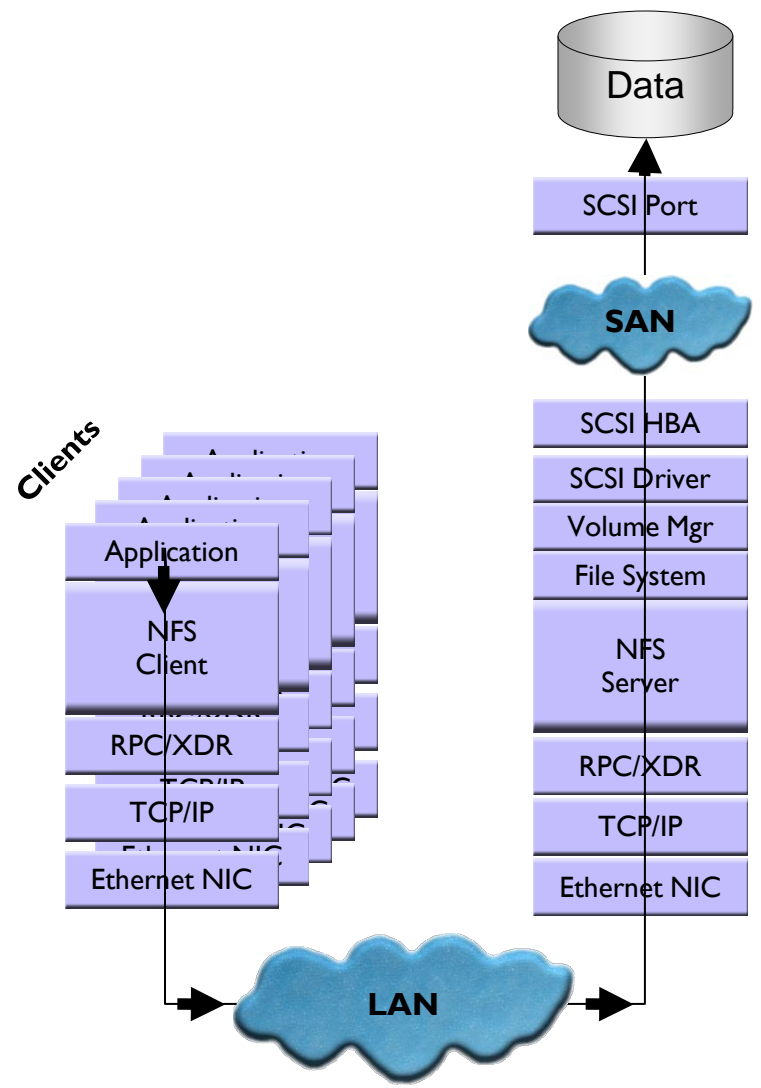
Network FS



* e.g. NFS, CIFS, AFP, WebDAV, FTP, HTTP, ...

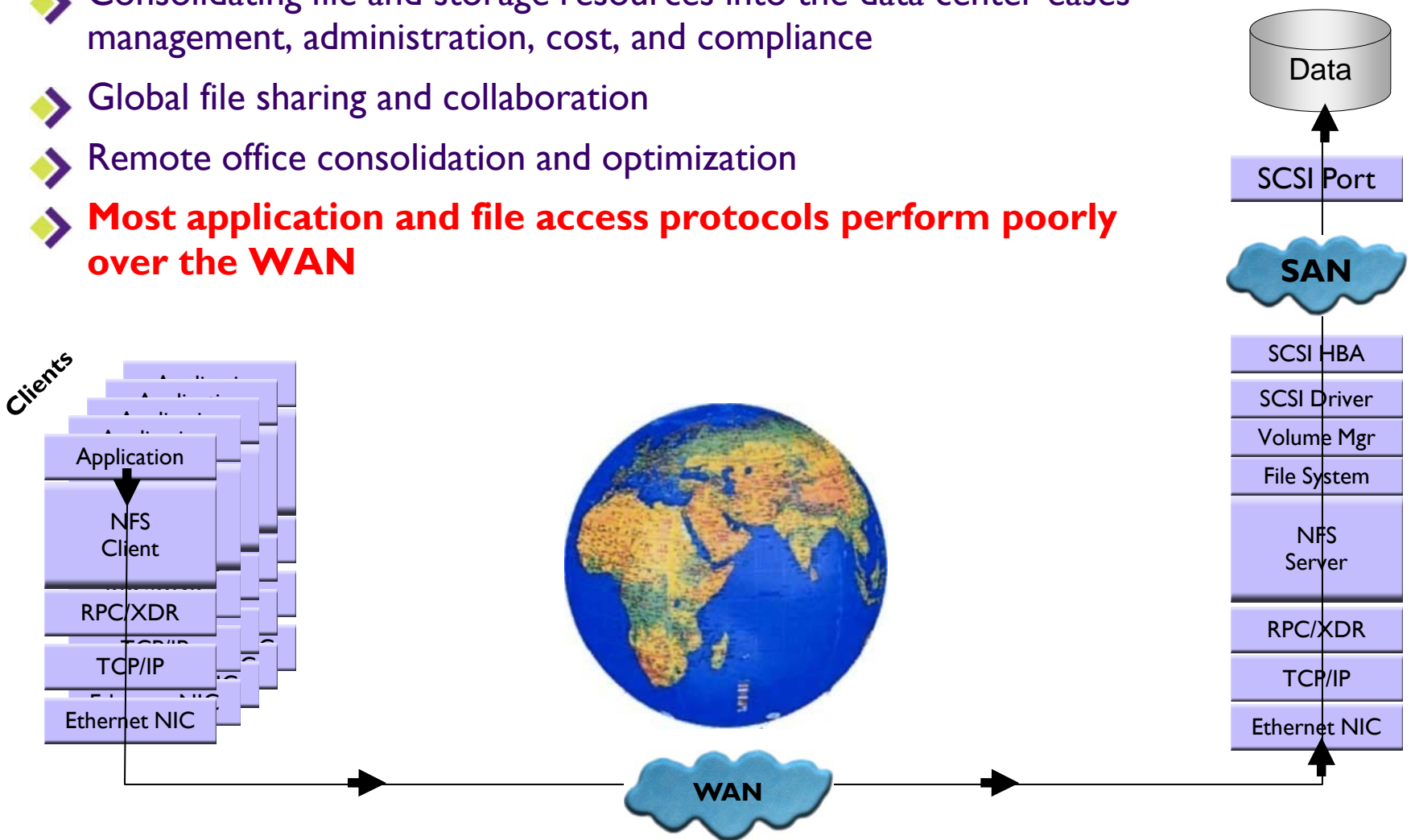
- A network file system is any file system that supports **sharing of files over a computer network protocol** between one or more file systems clients and a file system server

Network FS Stack (e.g. NFS)



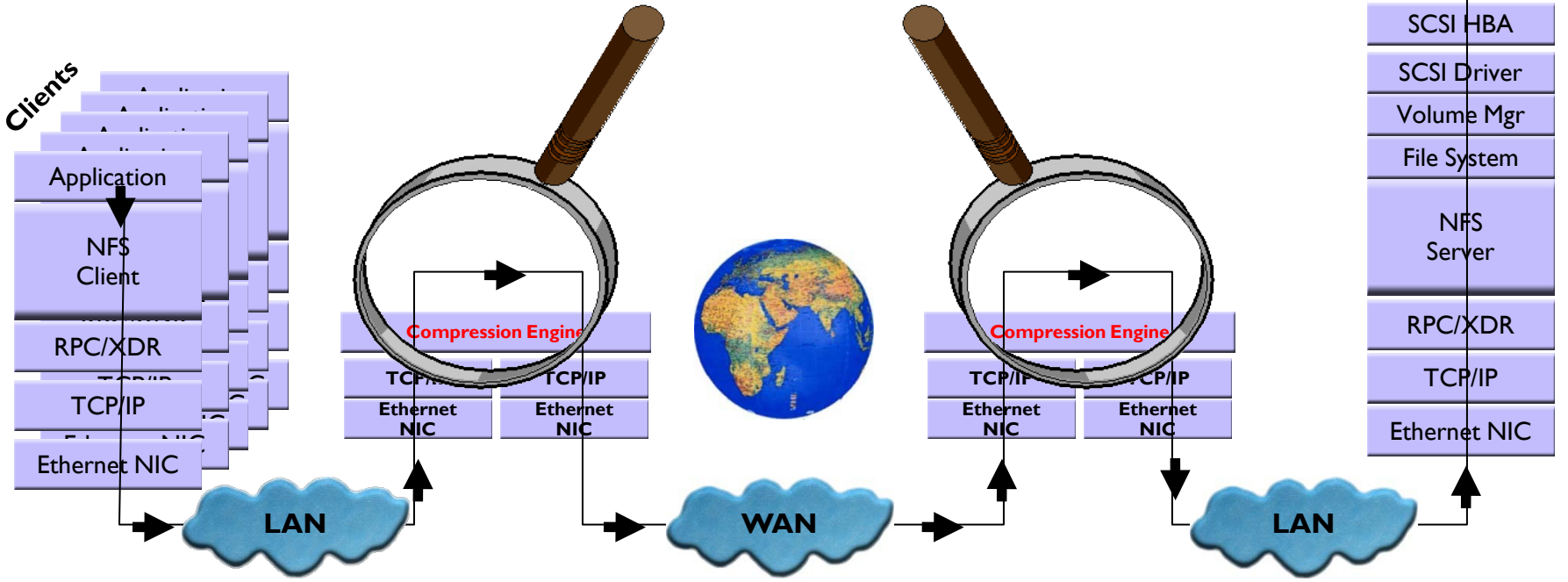
Network FS in a Distributed World

- Consolidating file and storage resources into the data center eases management, administration, cost, and compliance
- Global file sharing and collaboration
- Remote office consolidation and optimization
- **Most application and file access protocols perform poorly over the WAN**

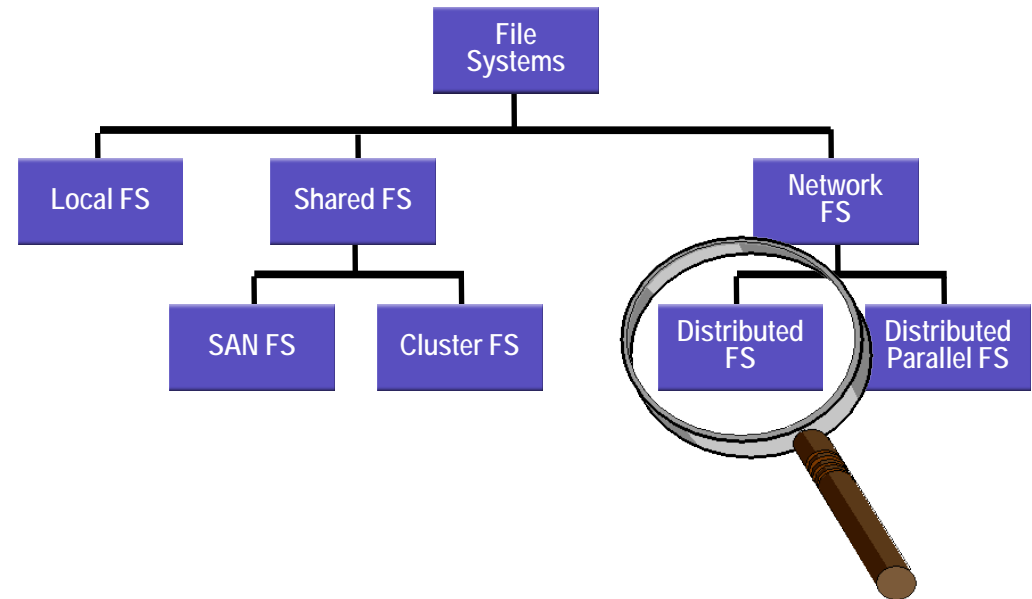


Network Compression

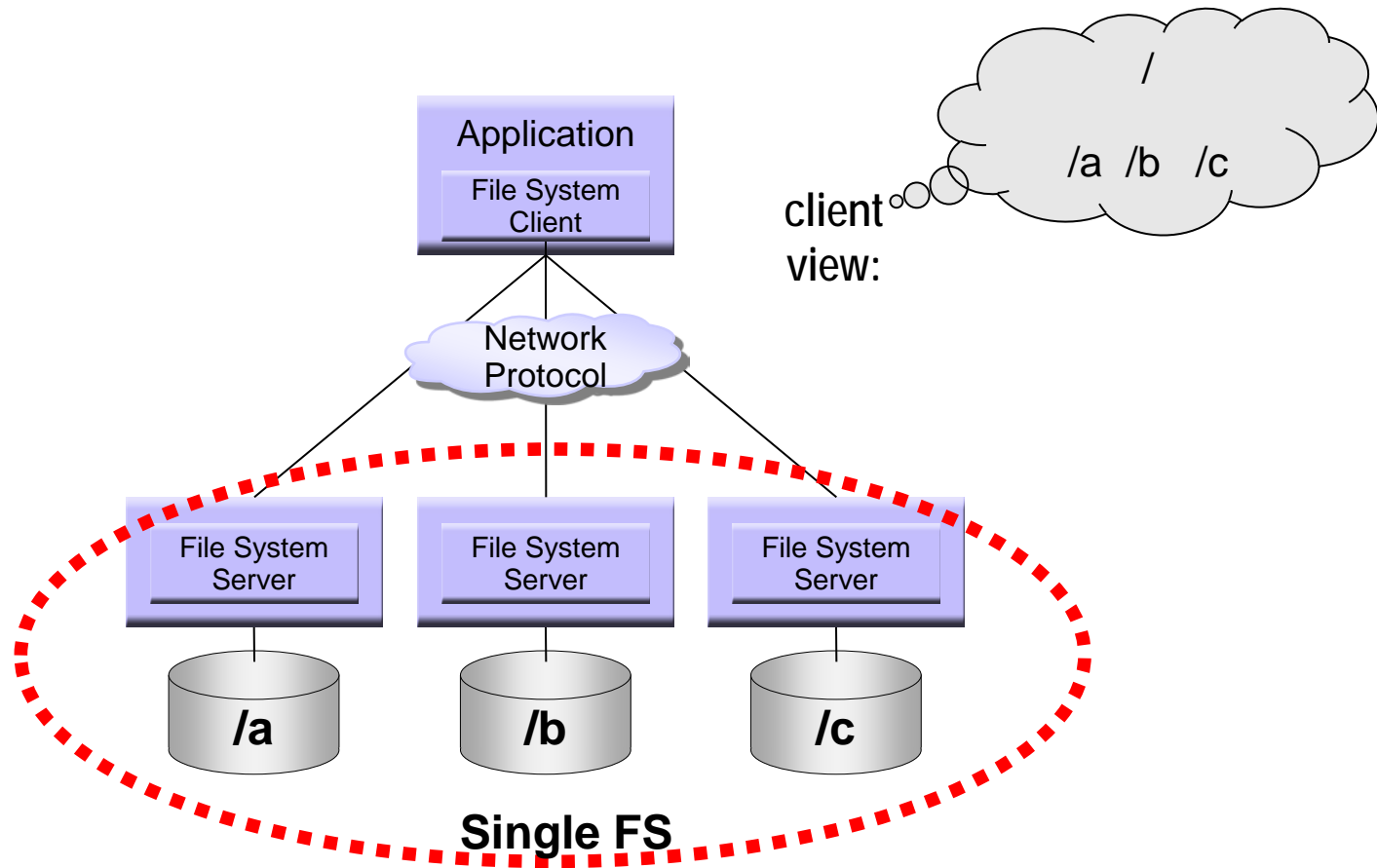
- Application-specific optimization: email, document management, SQL, ...
- Protocol-specific optimization: HTTP, NFS, CIFS, WebDAV, FTP, TCP/IP, ...
- Transport acceleration: TCP accelerators
- Intelligent caching: read-ahead, deferred write, coherency, ...
- Data compression: compression algorithms, file-aware differencing, data aggregation, I/O clustering, chunk based de-duplication, cross-protocol data reduction, ...



- File System Basics
- File System Taxonomy
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- **Distributed FS**
- Distributed Parallel FS
- Scale-Out NAS
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments

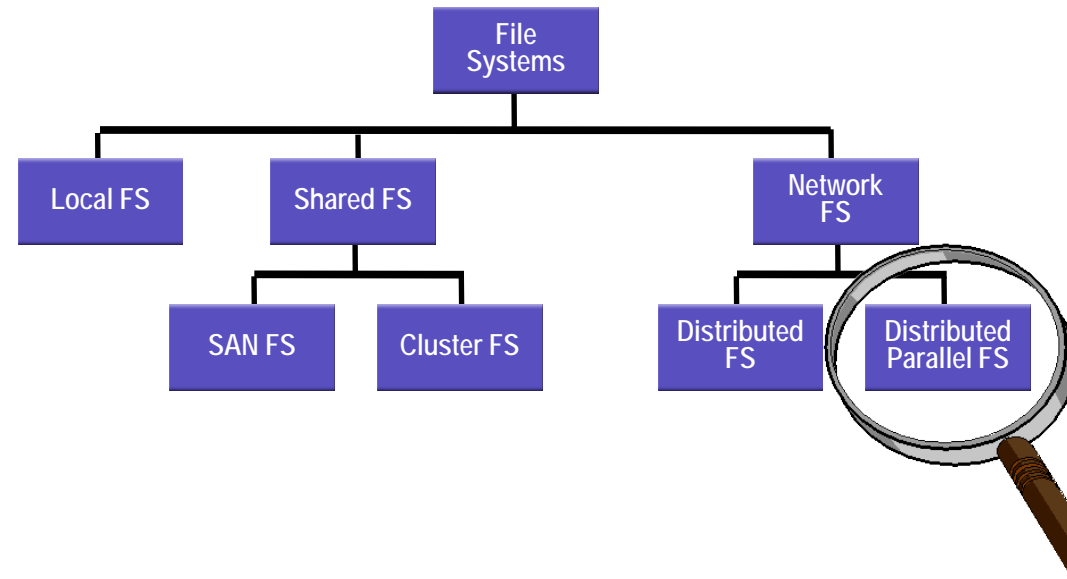


Distributed File System (DFS)



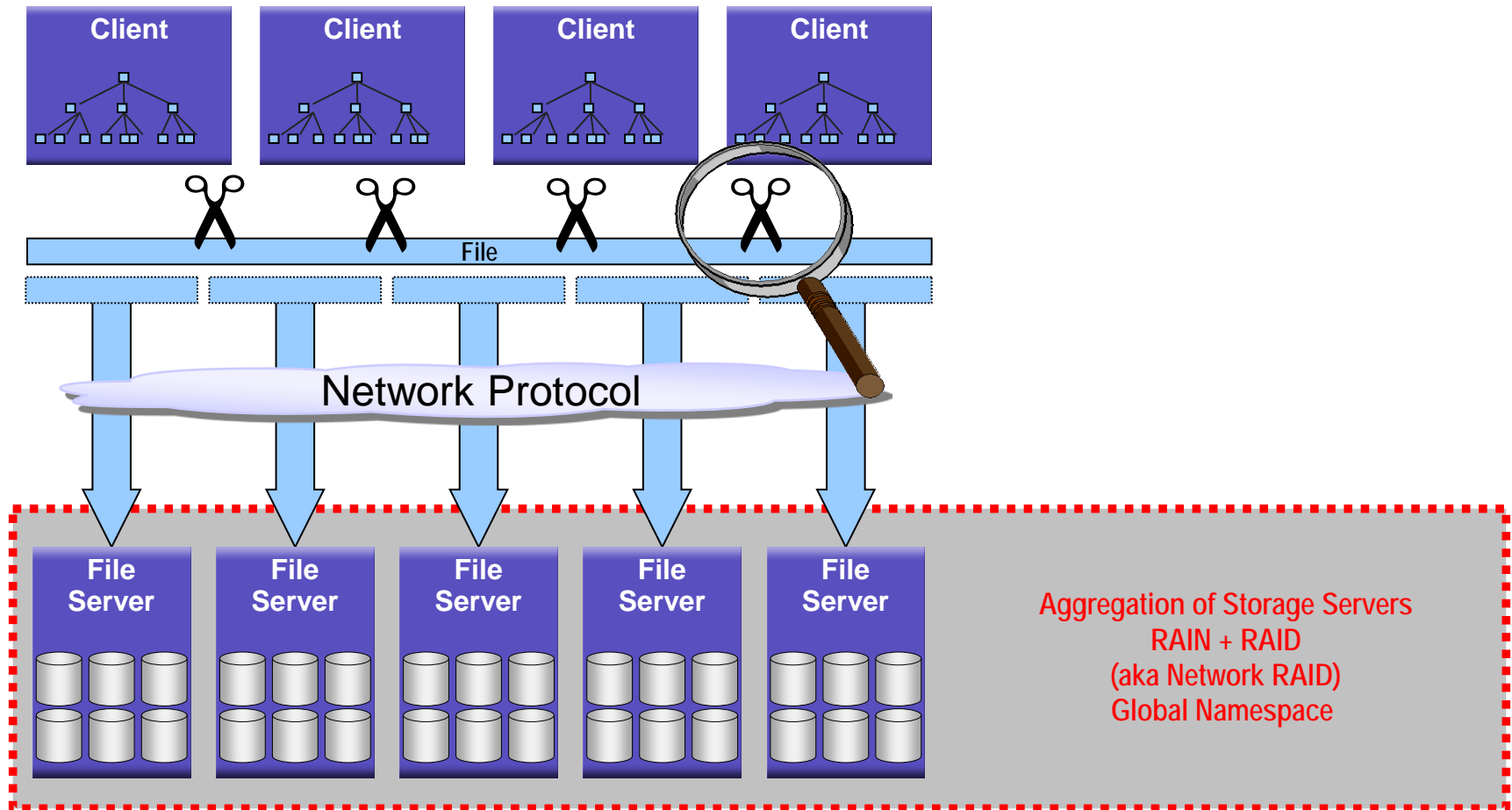
- **A distributed file system is a network file system** whose files are dispersed across file servers (\neq Parallel FS)

- File System Basics
- File System Taxonomy
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- Distributed FS
- **Distributed Parallel FS**
- Scale-Out NAS
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments

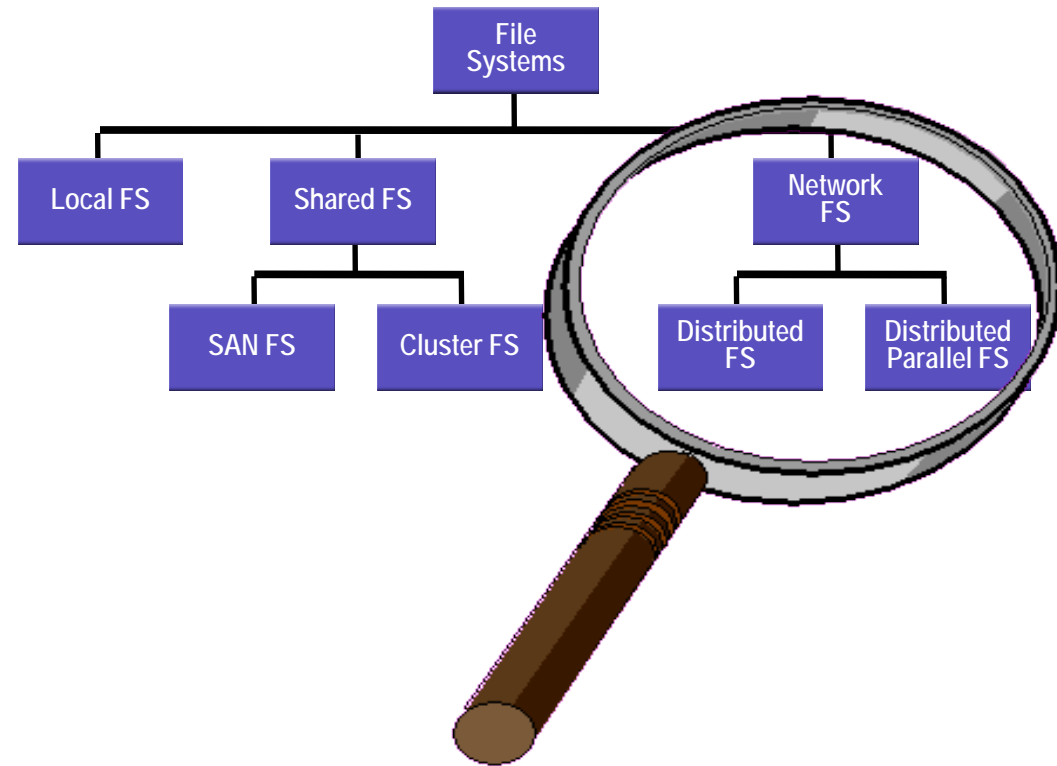


Distributed Parallel File System

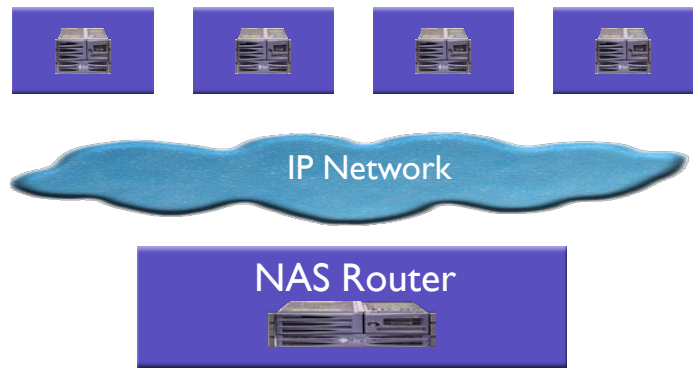
- ▶ File Segments distributed across storage nodes
 - ▶ Allows parallel I/O to individual files (striping)



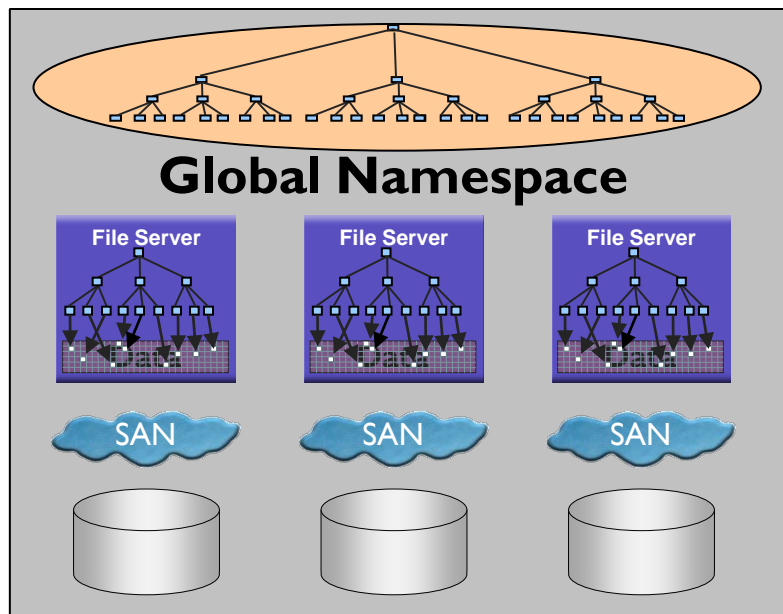
- File System Basics
- File System Taxonomy
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- Distributed FS
- Distributed Parallel FS
- **Scale-Out NAS**
 - ◆ **NAS Aggregation**
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments



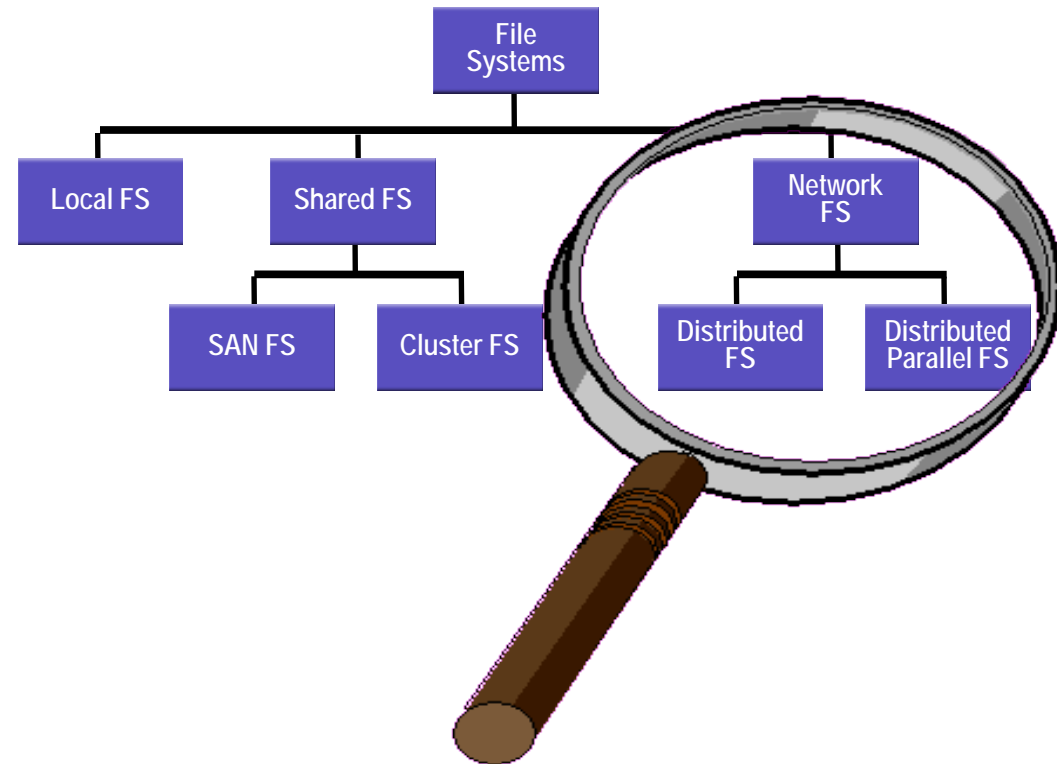
NAS Aggregation & Global Namespace



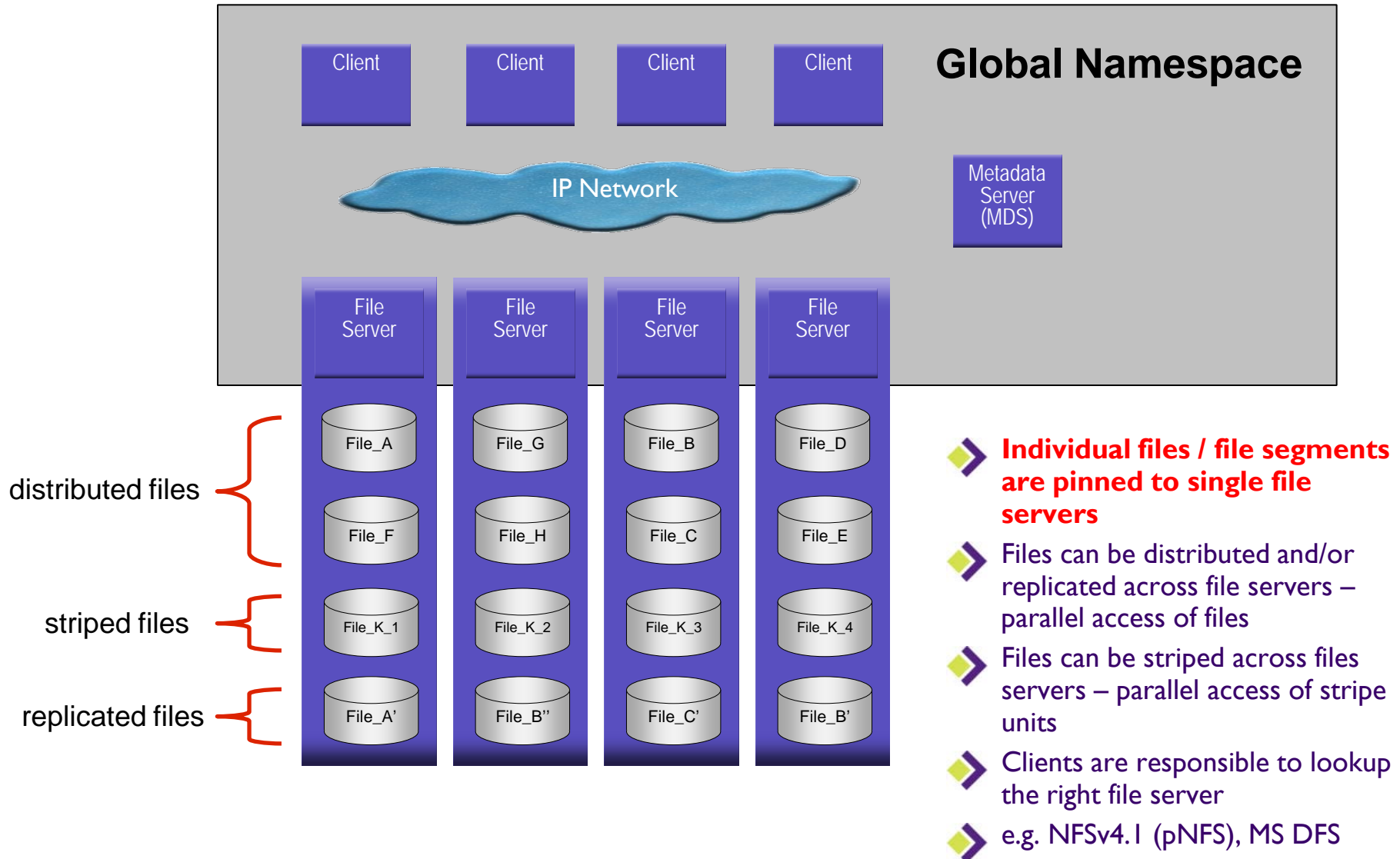
- In-Band Solution
- Aka NAS Router



- File System Basics
- File System Taxonomy
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- Distributed FS
- Distributed Parallel FS
- **Scale-Out NAS**
 - ◆ NAS Aggregation
 - ◆ **NAS Virtualization**
 - ◆ NAS Cluster / NAS Grid
- FS Future Developments

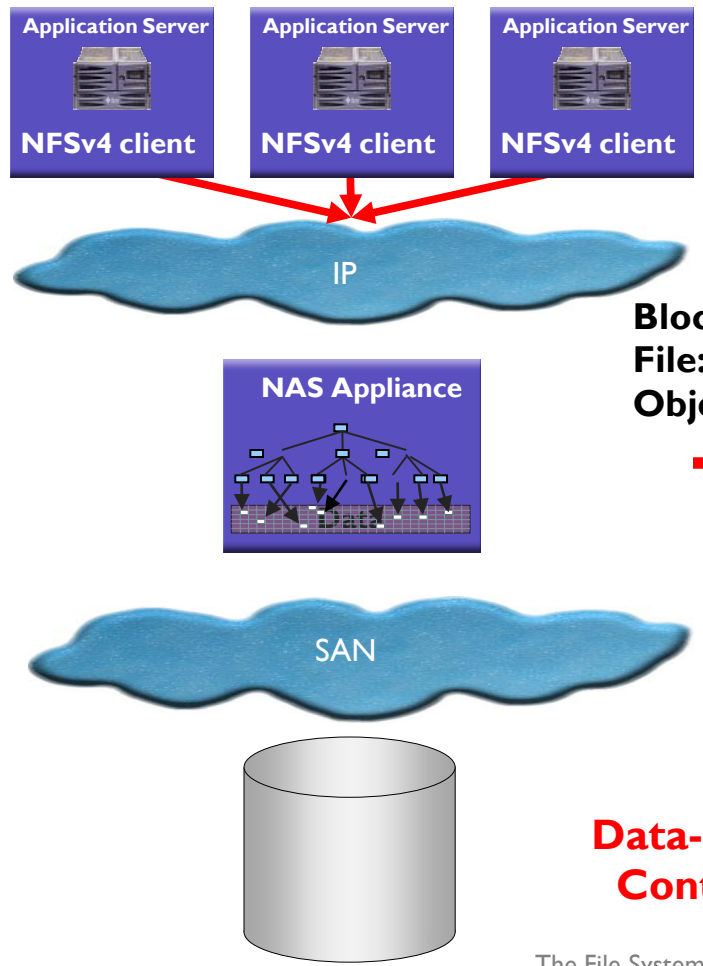


NAS Virtualization - Out-of-Band



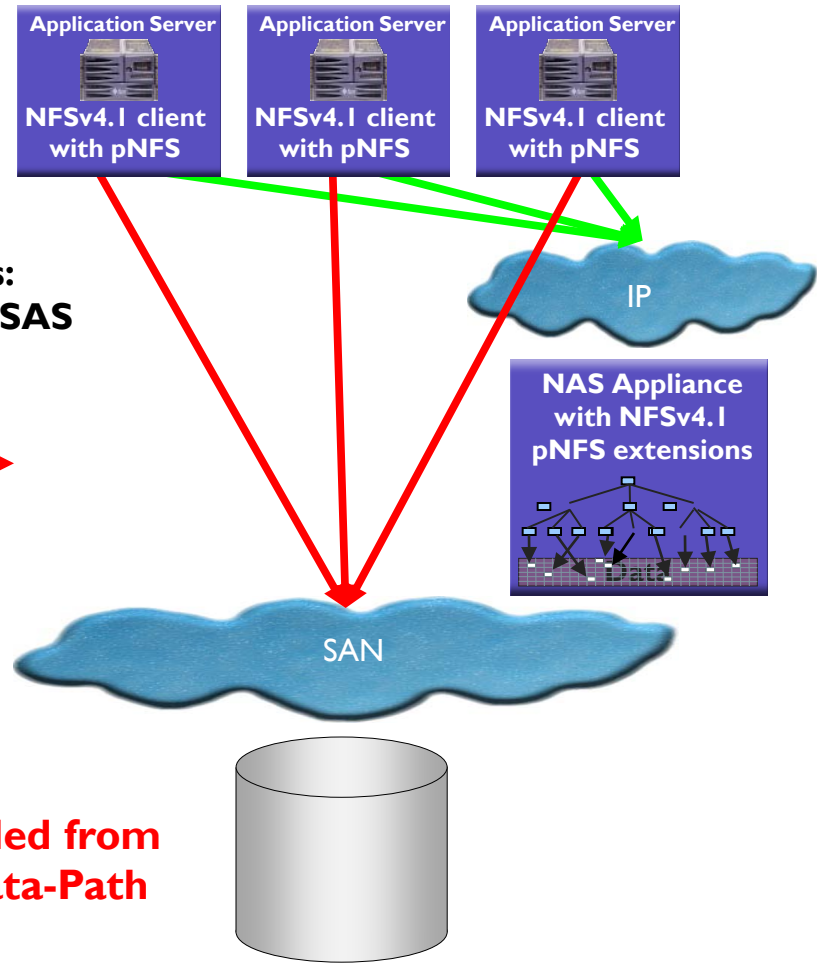
NAS Virtualization – NFS4.1 pNFS

In-Band NAS:



Out-of-Band NAS:

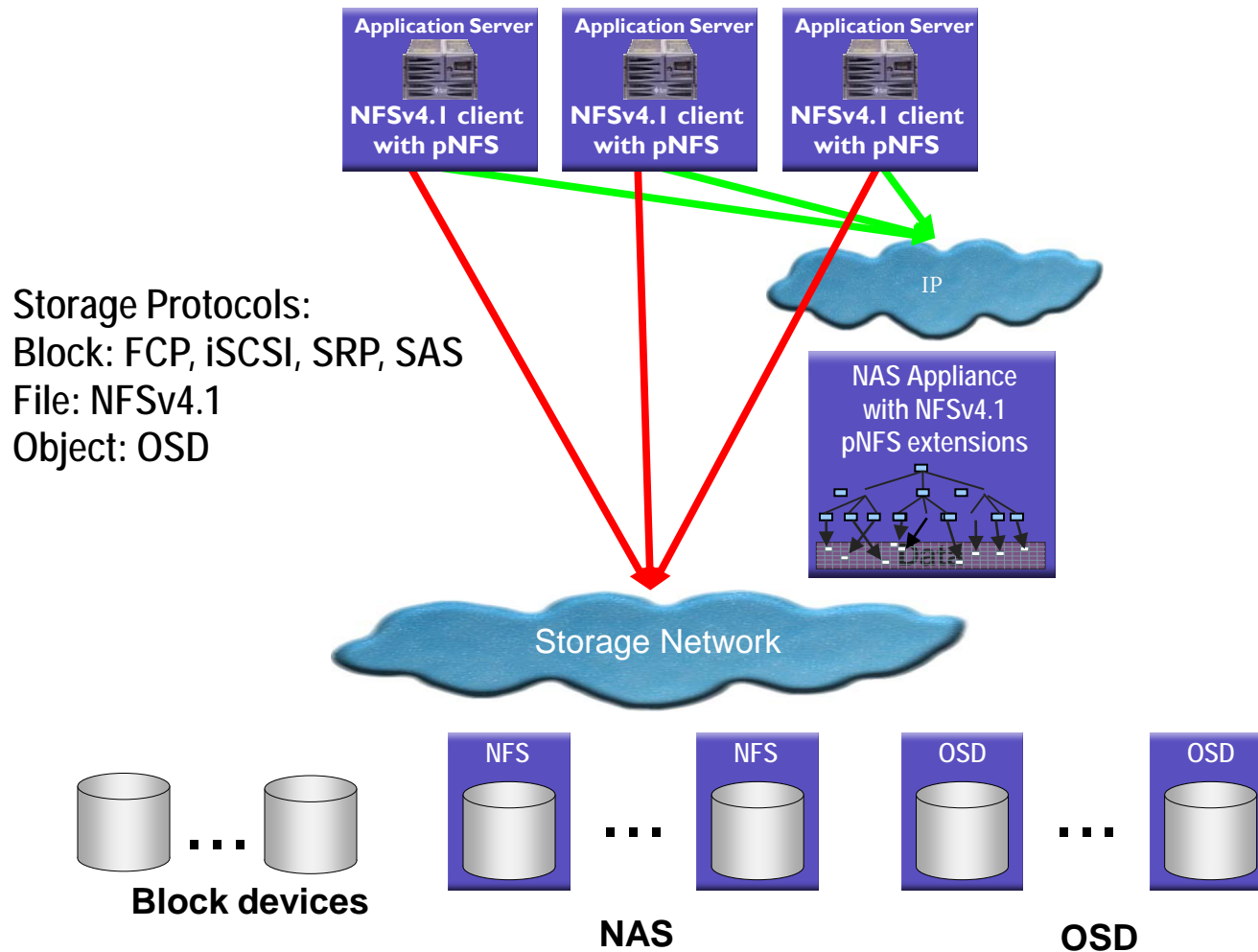
Storage Protocols:
Block: FCP, iSCSI, SRP, SAS
File: NFSv4.1
Object: OSD



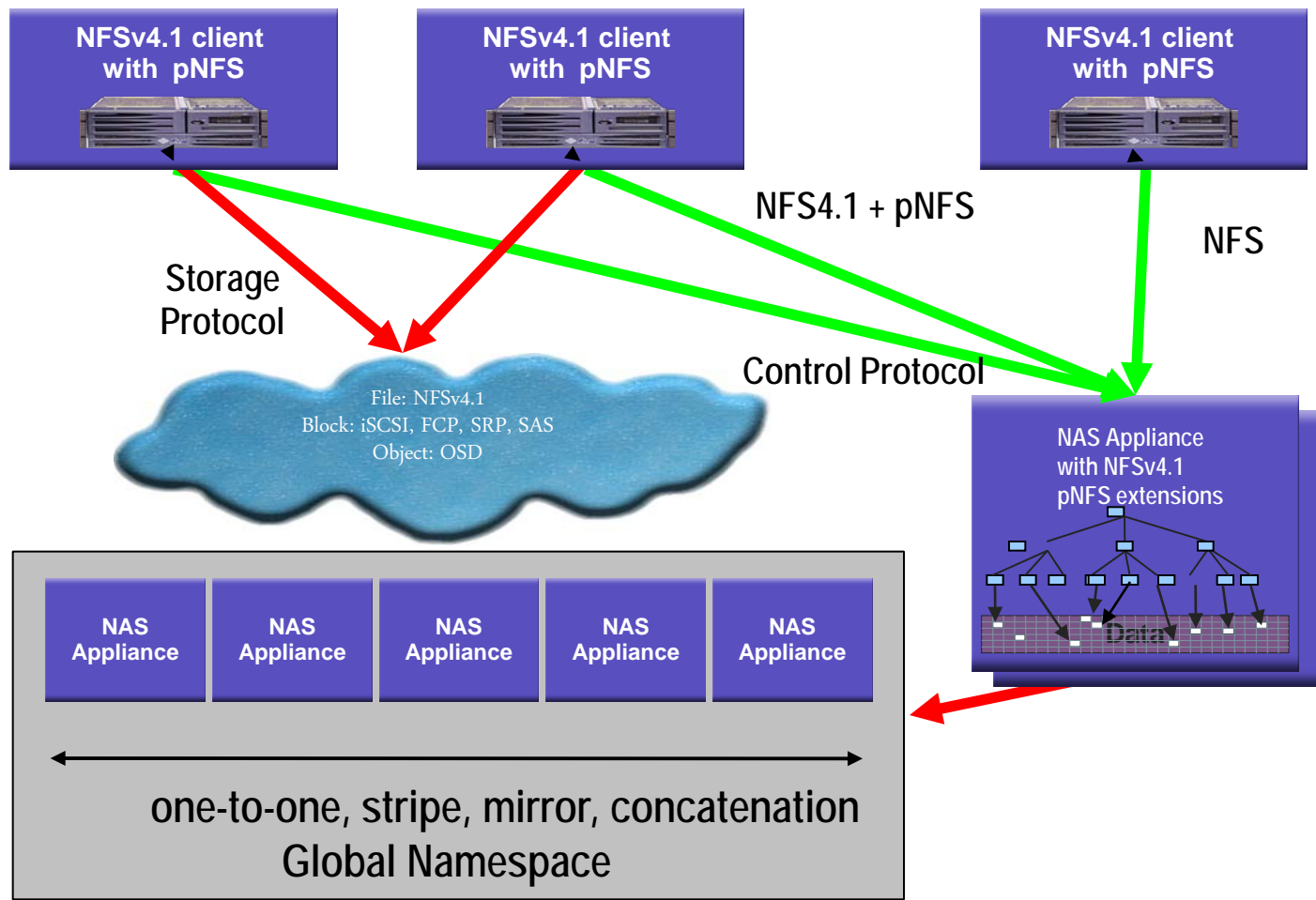
Data-Path is de-coupled from Control- and Metadata-Path

NAS Virtualization – NFS4.1 pNFS

Out-of-Band NAS:



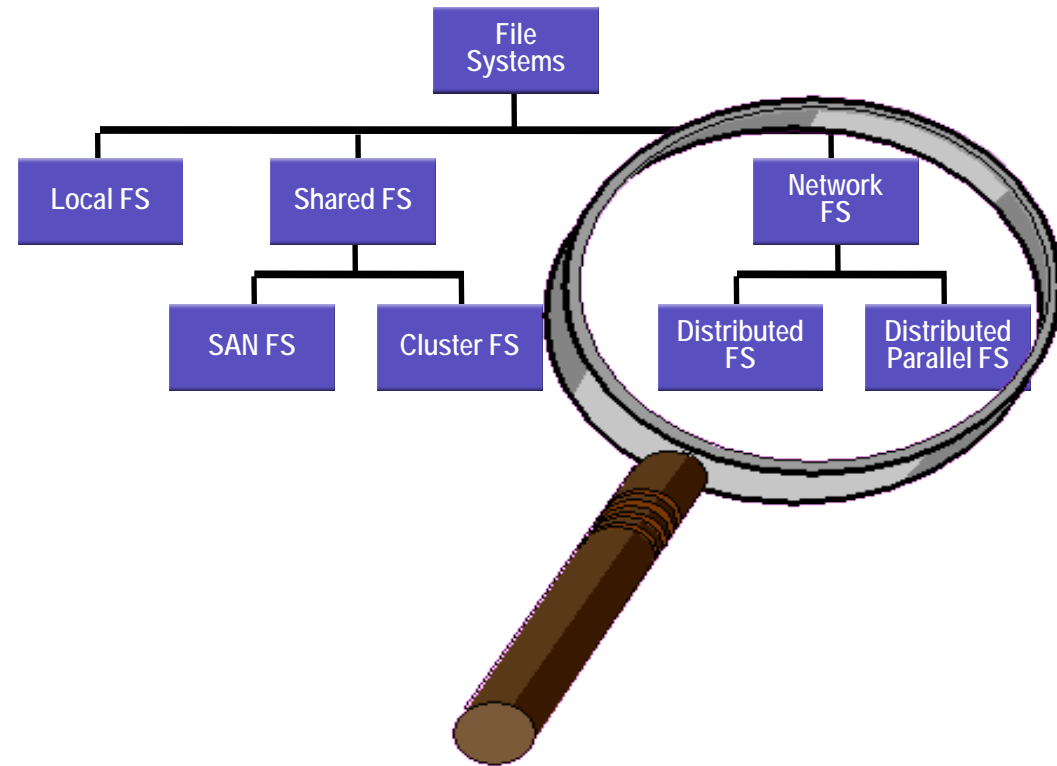
NAS Virtualization – NFS4.1 pNFS



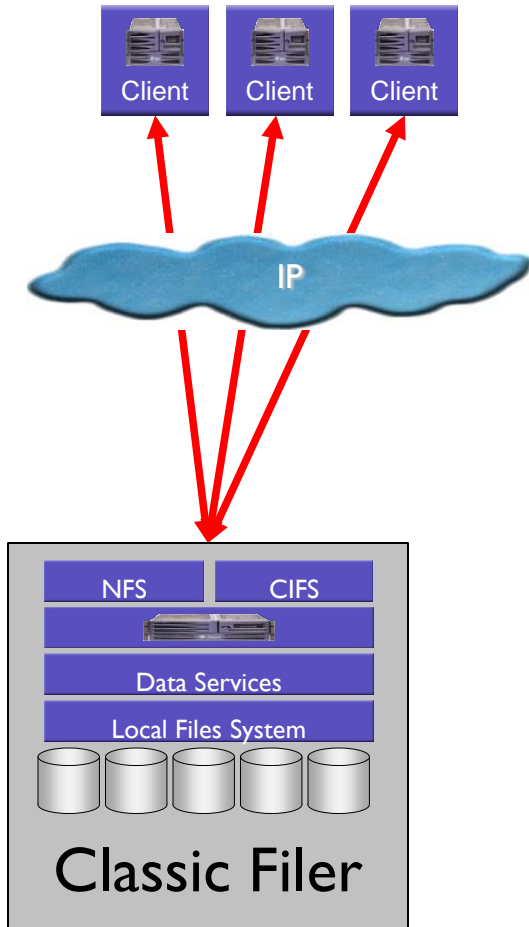
MDS acts as proxy for clients not pNFS enabled

MDS creates Global Namespace

- File System Basics
- File System Taxonomy
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- Distributed FS
- Distributed Parallel FS
- **Scale-Out NAS**
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ **NAS Cluster / NAS Grid**
- FS Future Developments

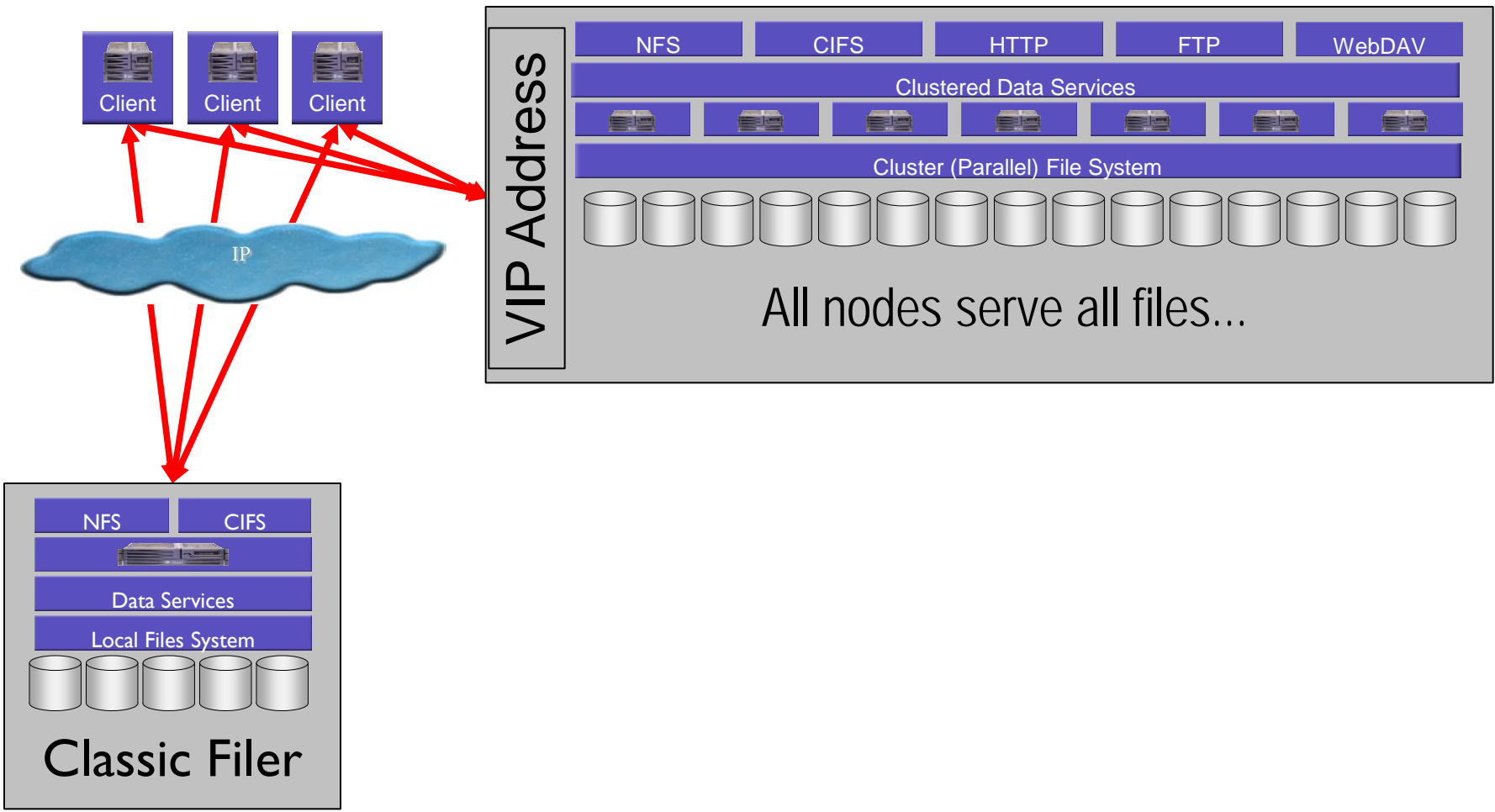


NAS Cluster / NAS Grid (1)



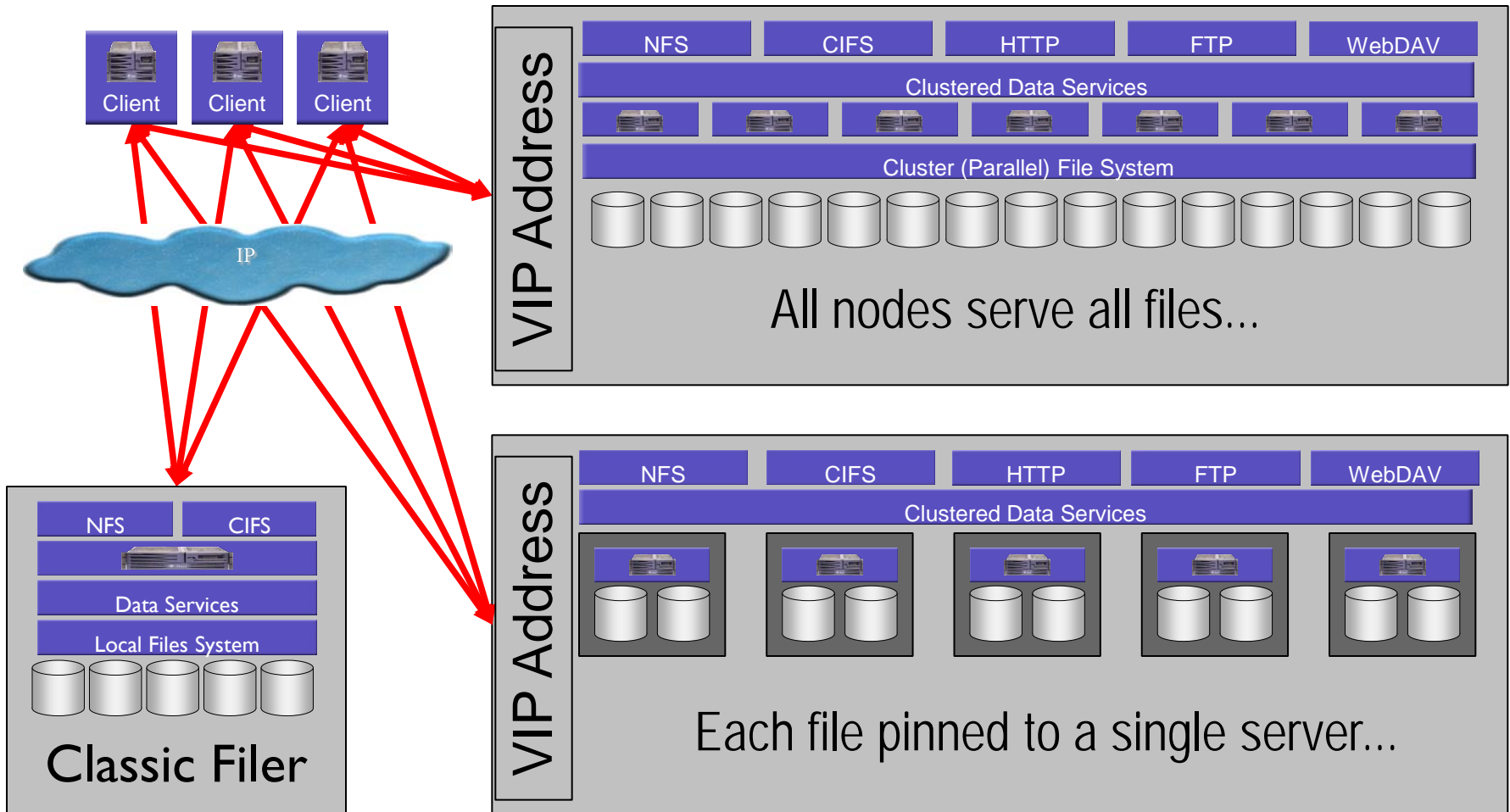
NAS Cluster / NAS Grid (2)

Two variants:



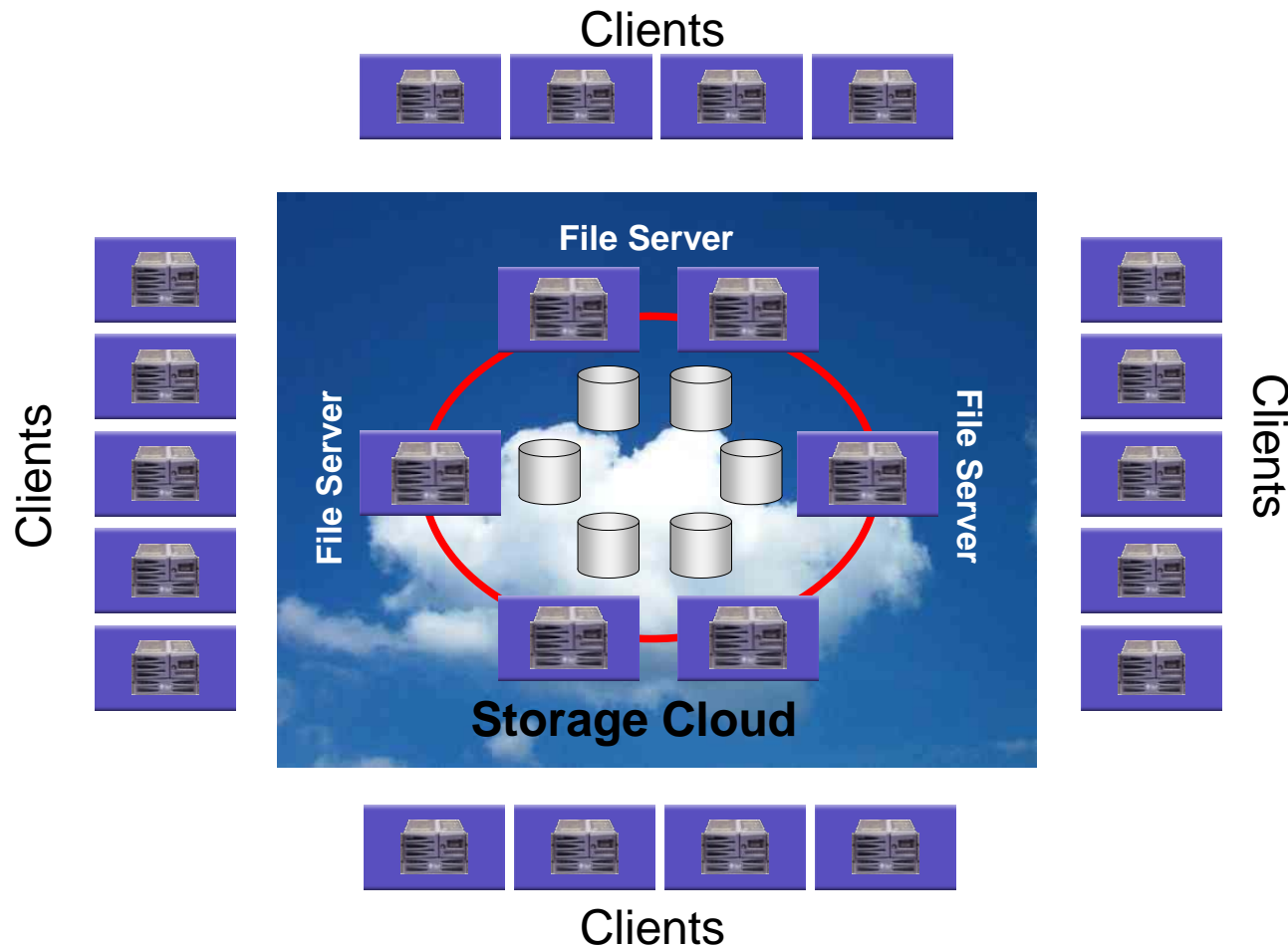
NAS Cluster / NAS Grid (3)

Two variants:



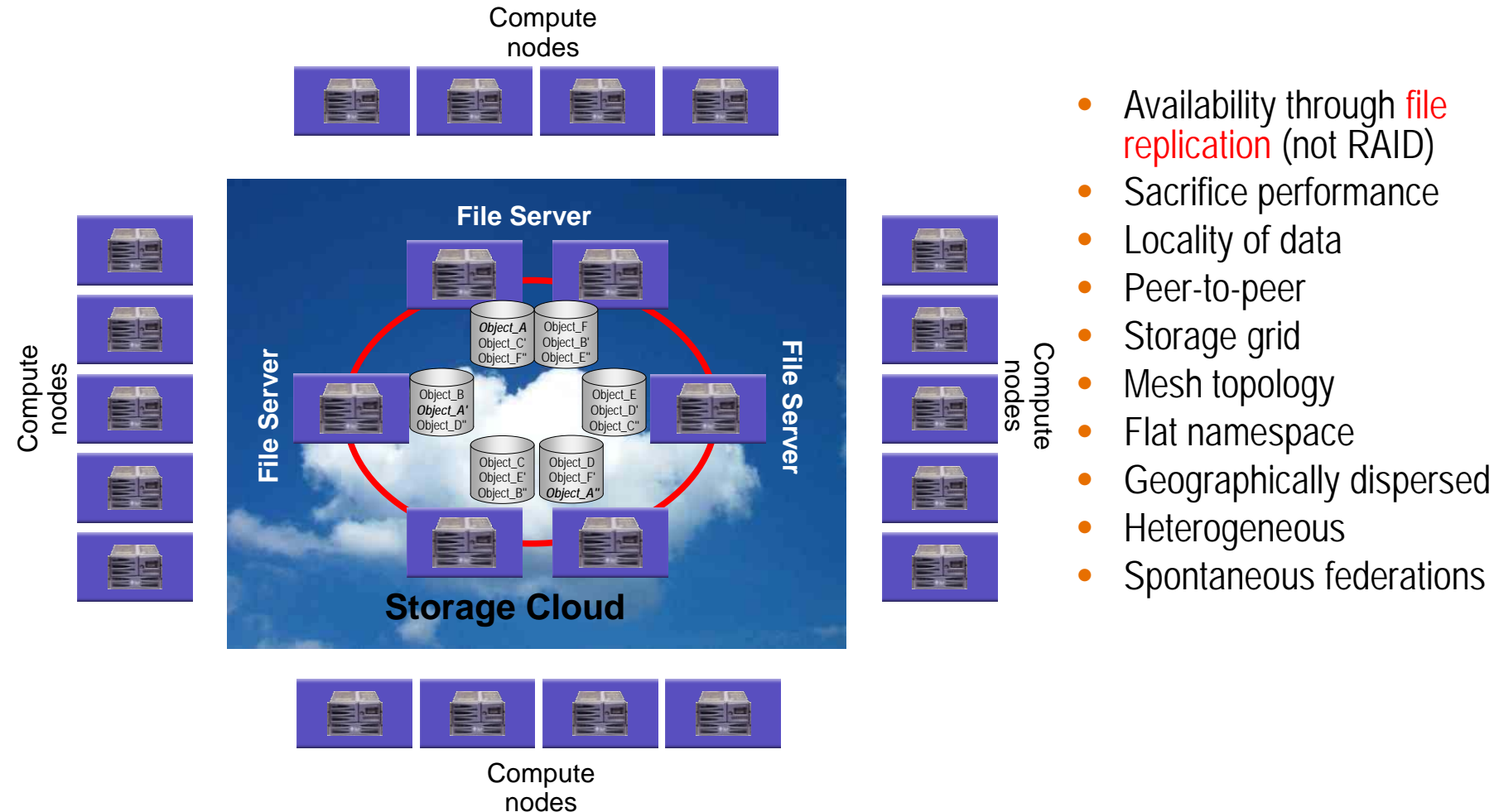
NAS Cluster/Grid & Cloud Storage

- NAS Cluster is effectively a storage cloud

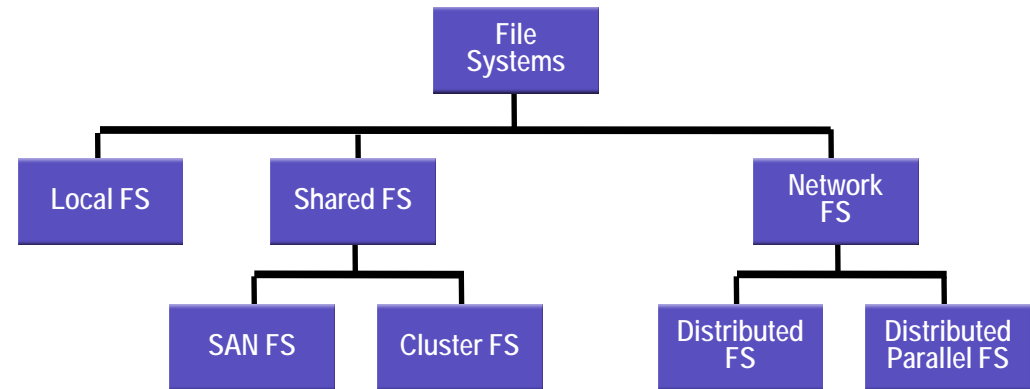


File Systems for Clouds

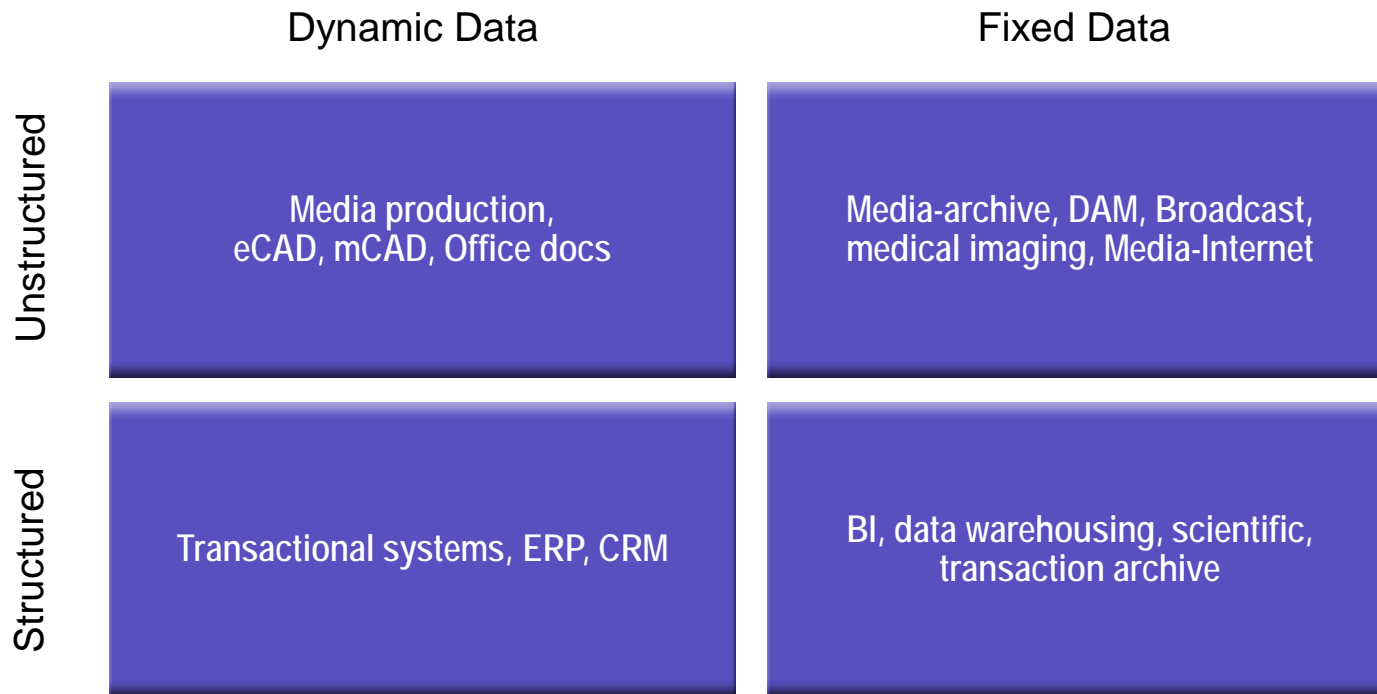
File Systems morphing into distributed object-based content repositories



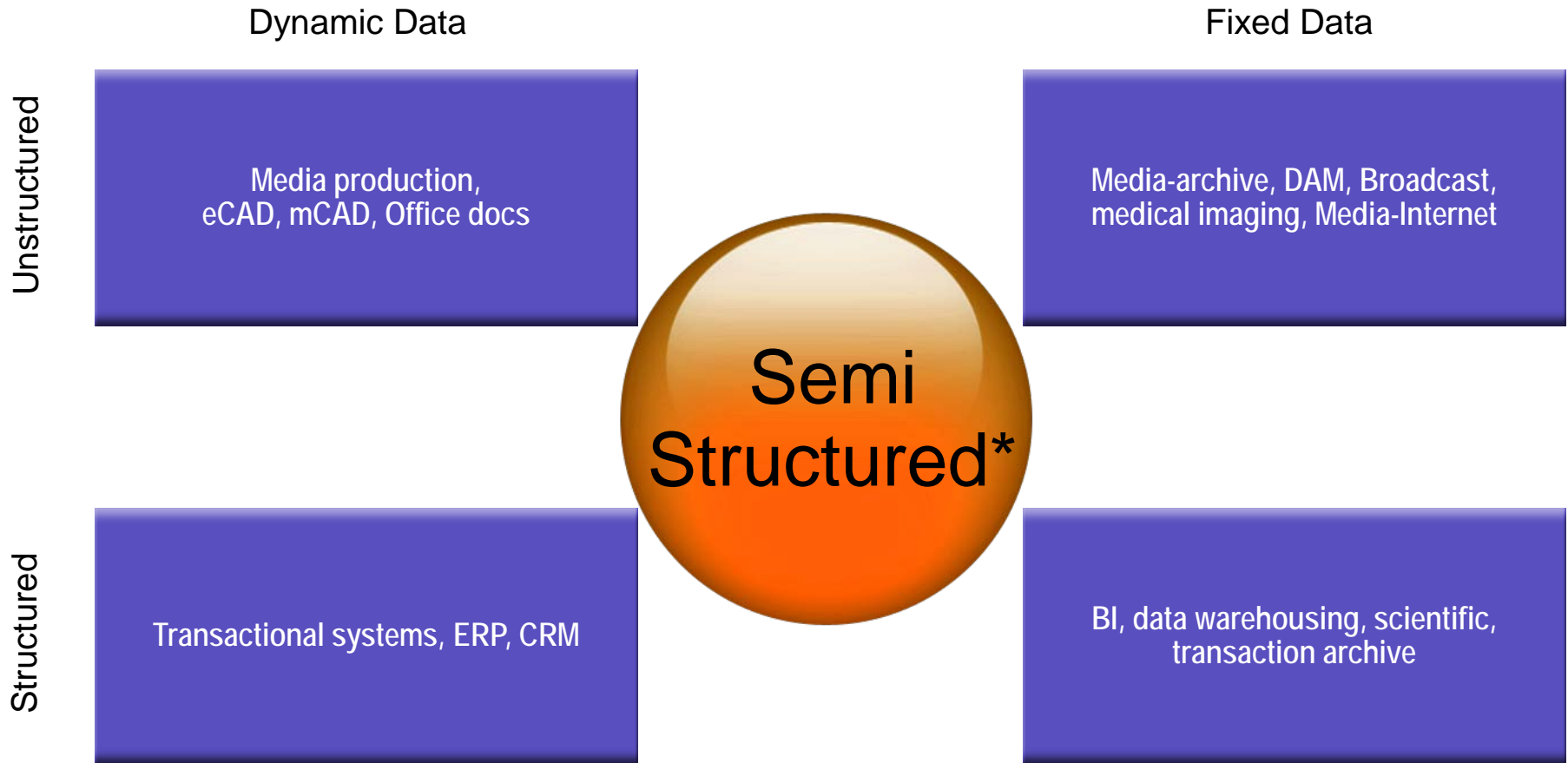
- File System Basics
- File System Taxonomy
- Local FS
- Shared FS / Global FS
 - ◆ SAN FS, Cluster FS
- Network FS
- Distributed FS
- Distributed Parallel FS
- Scale-Out NAS
 - ◆ NAS Aggregation
 - ◆ NAS Virtualization
 - ◆ NAS Cluster / NAS Grid
- **FS Future Developments**



Storage Market Segmentation



Storage Market Segmentation



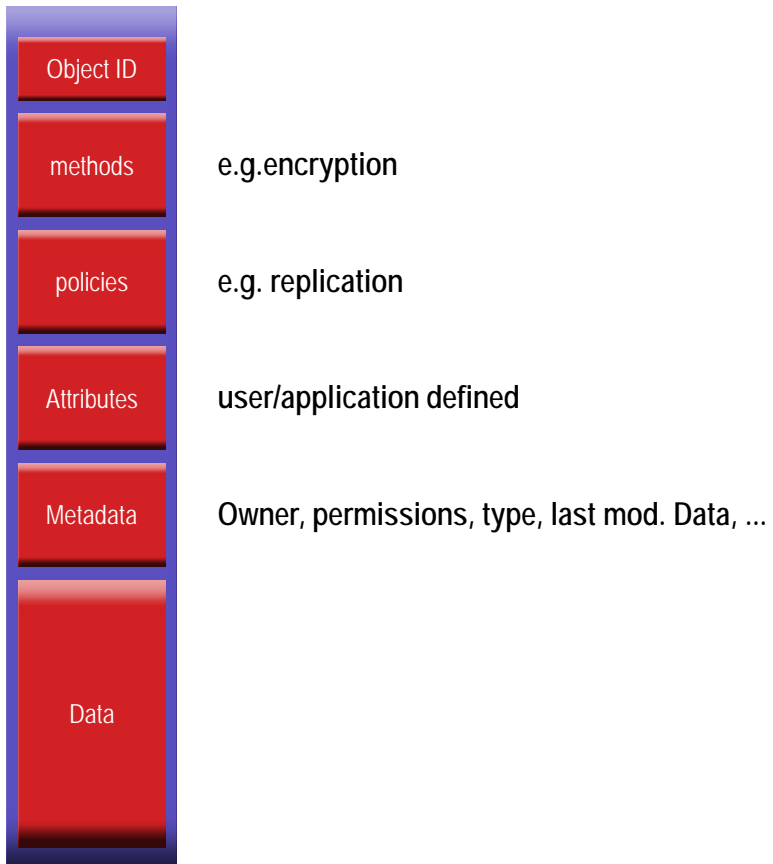
*Semi-Structured Data contains dynamic meta-data defined by users and/or applications

Traditional Files



Owner, permissions, type, last mod. Data, ...

Semi Structured Data



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

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

- SNIA Education Committee

**Christian Bandulet
Craig Harmer**

Reference Material

FS Names from wikipedia.org

- ADFS – Acorn's Advanced Disc filing system, successor to DFS
- BFS – the Be File System used on BeOS
- EFS – Encrypted filesystem, An extension of NTFS
- EFS (IRIX) – an older block filing system under IRIX
- Ext – Extended filesystem, designed for Linux system
- Ext2 – Second extended filesystem, designed for Linux systems
- Ext3 – Name for the journalled form of ext2
- FAT – Used on DOS and Microsoft Windows, 12, 16 and 32 bit table depths
- FFS (Amiga) – Fast File System, used on Amiga systems. This FS has evolved over time. Now counts FFS1, FFS Intl, FFS DCache, FFS2
- FFS – Fast File System, used on *BSD systems
- Fossil – Plan 9 from Bell Labs snapshot archival file system
- Files-11 – OpenVMS filesystem
- GCR – Group Code Recording, a floppy disk data encoding format used by the Apple II and Commodore Business Machines in the 5¼" disk drives for their 8-bit computers
- HFS – Hierarchical File System, used on older Mac OS systems



FS Names from wikipedia.org(2)

- ◆ HFS Plus – Updated version of HFS used on newer Mac OS systems
- ◆ HPFS – High Performance Filesystem, used on OS/2
- ◆ ISO 9660 – Used on CD-ROM and DVD-ROM discs (Rock Ridge and Joliet are extensions to this)
- ◆ JFS – IBM Journaling Filesystem, provided in Linux, OS/2, and AIX
- ◆ LFS – 4.4BSD implementation of a log-structured file system
- ◆ MFS – Macintosh File System, used on early Mac OS systems
- ◆ Minix file system – Used on Minix systems
- ◆ NTFS – Used on Windows NT, Windows 2000, Windows XP and Windows Server 2003 systems
- ◆ NSS – Novell Storage Services. This is a new 64-bit journaling filesystem using a balanced tree algorithm. Used in NetWare versions 5.0-up and recently ported to Linux.
- ◆ OFS – Old File System, on Amiga. Nice for floppies, but fairly useless on hard drives
- ◆ PFS – and PFS2, PFS3, etc. Technically interesting filesystem available for the Amiga, performs very well under a lot of circumstances. Very simple and elegant



FS Names from wikipedia.org(3)

- ReiserFS – Filesystem that uses journaling
- Reiser4 – Filesystem that uses journaling, newest version of ReiserFS
- SFS – Smart File System, journaled file system available for the Amiga platforms
- UDF – Packet based filesystem for WORM/RW media such as CD-RW and DVD.
- UFS – Unix Filesystem, used on older BSD systems
- UFS2 – Unix Filesystem, used on newer BSD systems
- UMSDOS – FAT filesystem extended to store permissions and metadata, used for Linux
- VxFS – Veritas file system, first commercial journaling file system; HP-UX, Solaris, Linux, AIX
- VSAM
- WAFL – Used on Network Appliance systems
- XFS – Used on SGI IRIX and Linux systems
- ZFS – Used on Solaris
- SAM QFS (Oracle)



FS Names from wikipedia.org(4)

- 9P The Plan 9 and Inferno distributed file system
- AFS (Andrew File System)
- AppleShare
- Arla (file system)
- Coda
- CXFS (Clustered XFS) a distributed networked file system designed by Silicon Graphics (SGI) specifically to be used in a SAN
- Distributed File System (DCE)
- Distributed File System (Microsoft)
- Freenet
- Global File System (GFS)
- Google File System (GFS)
- IBRIX Fusion™
- InterMezzo
- Isilon OneFS™
- Lustre (Oracle)

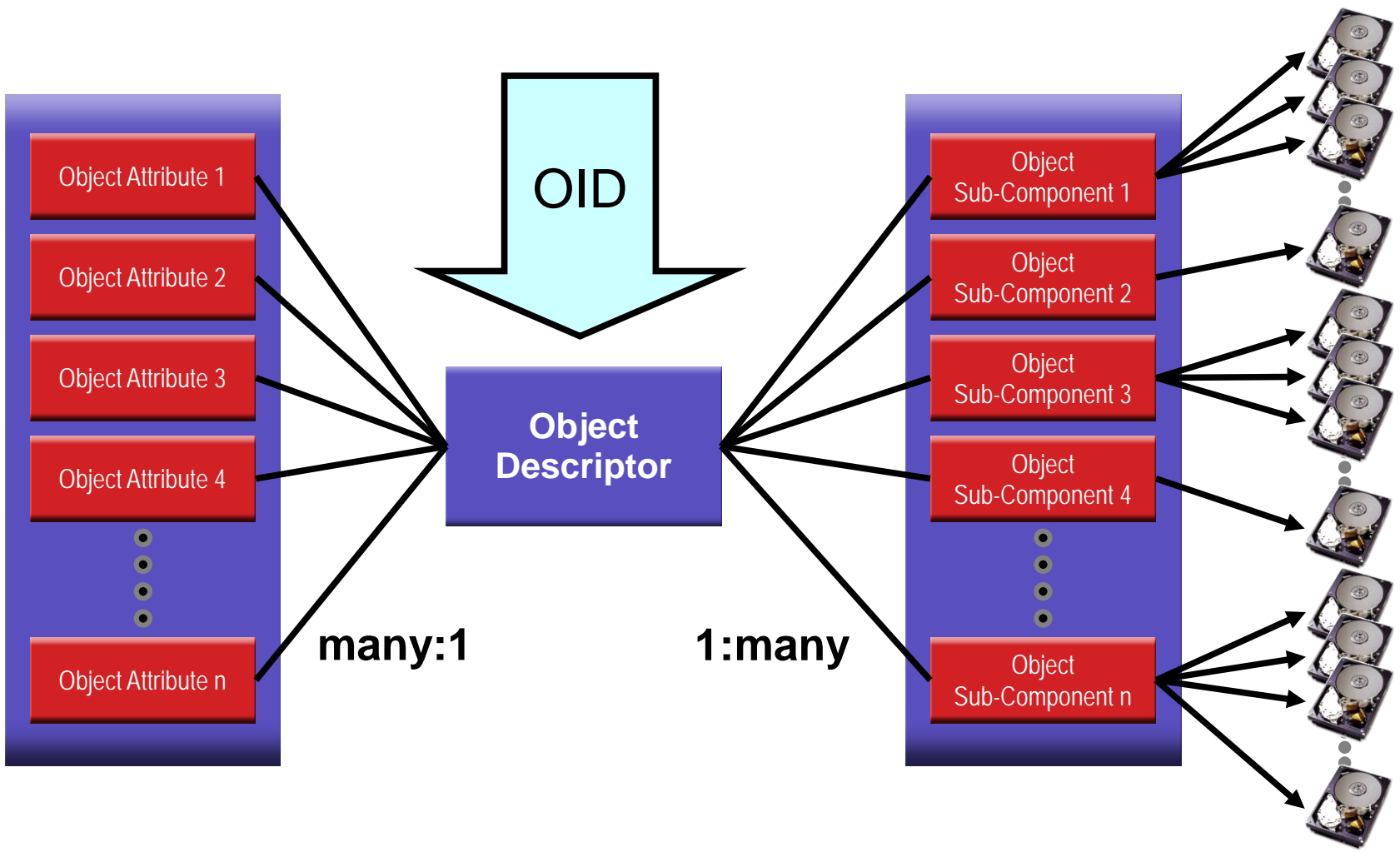


FS Names from wikipedia.org(5)

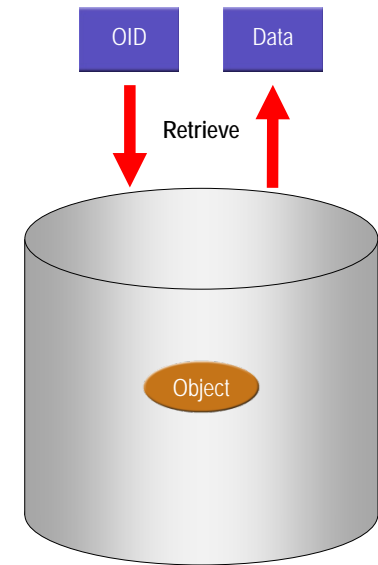
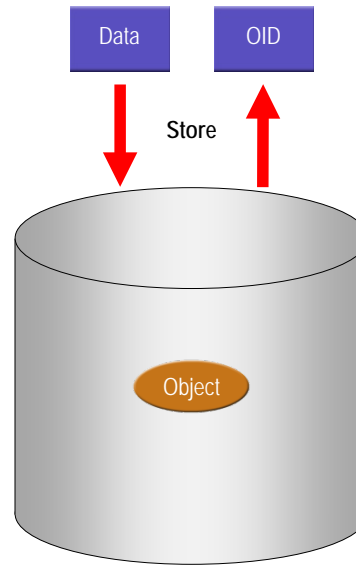
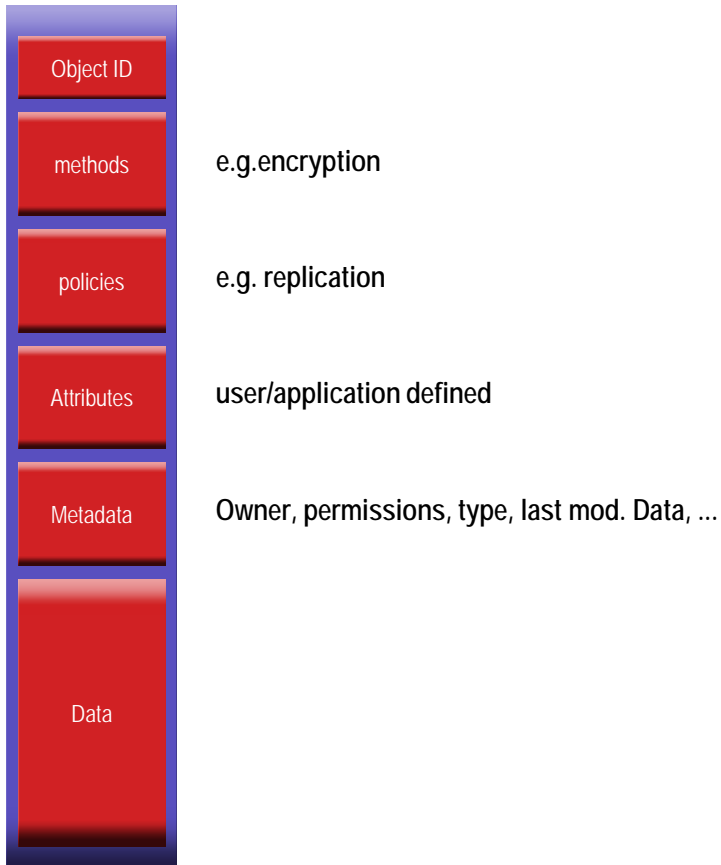
- NFS
- OpenAFS
- Server message block (SMB) (aka Common Internet File System (CIFS) or Samba file system)
- Xsan (a storage area network (SAN) filesystem from Apple Computer, Inc.)
- archfs (archive)
- cdfs (reading and writing of CDs)
- cfs (caching)
- Davfs2 (WebDAV)
- Devfs
- ftpfs (ftp access)
- fuse (filesystem in userspace, like lufs but better maintained)
- GPFS an IBM cluster file system
- JFFS/JFFS2 (filesystems designed specifically for flash devices)
- LUFs (replaces ftpfs, ftp ssh ... access)
- nntpfs (netnews)
- OCFS (Oracle Cluster File System)



The File Object Model (1)



The File Object Model (2)



The File Object Model (3)

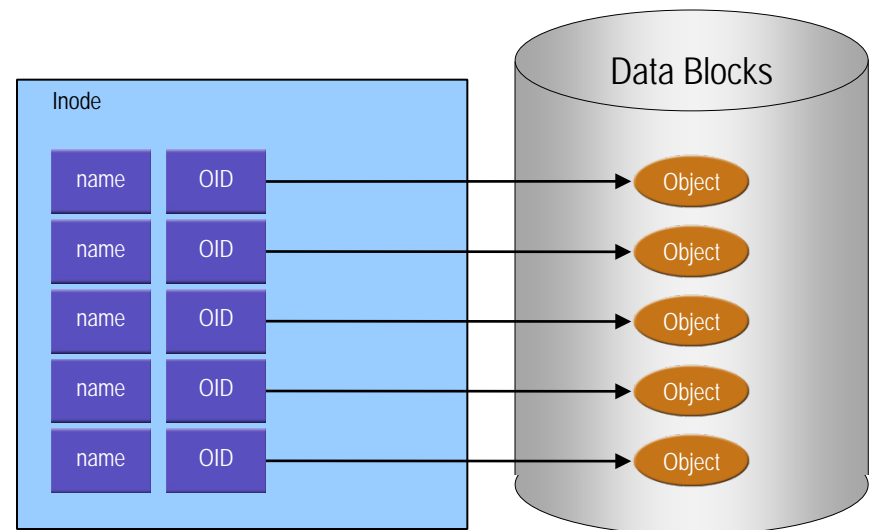
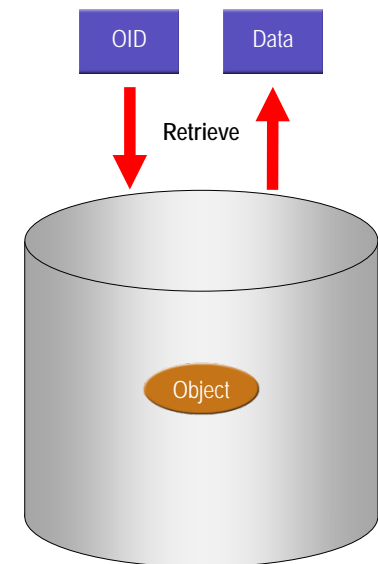
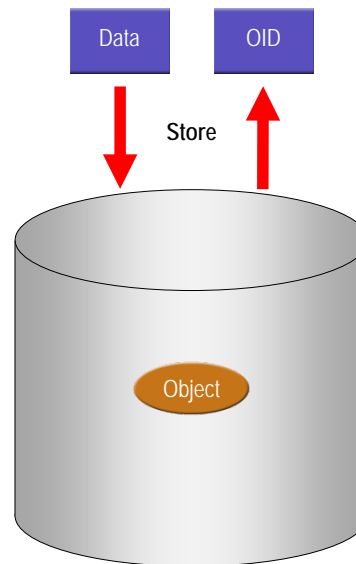


e.g. encryption

e.g. replication

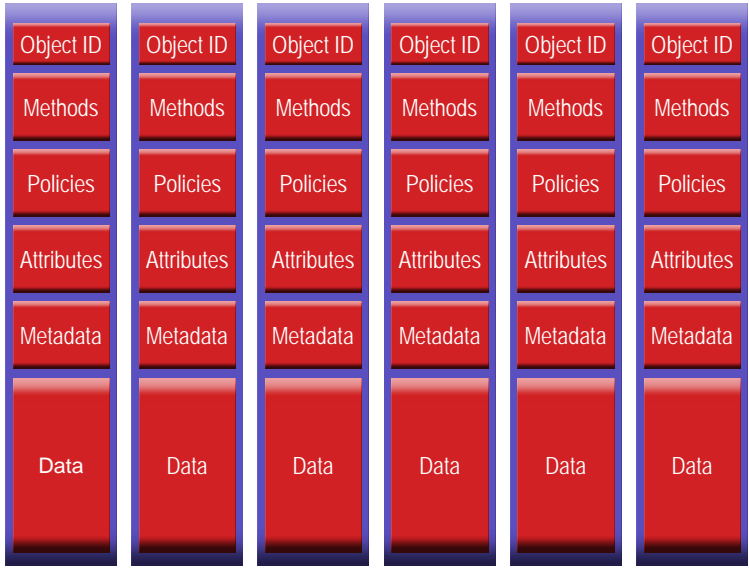
user/application defined

Owner, permissions, type, last mod. Data, ...

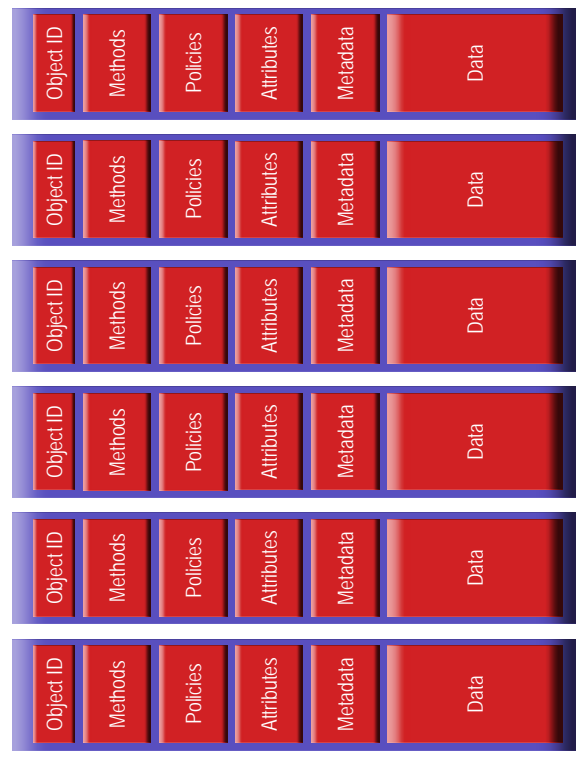


Managing File Objects (1)

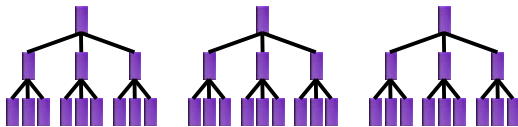
- File objects can be managed like records in a relational database with user data as Binary Large Objects (BLOBs)



Database Schema



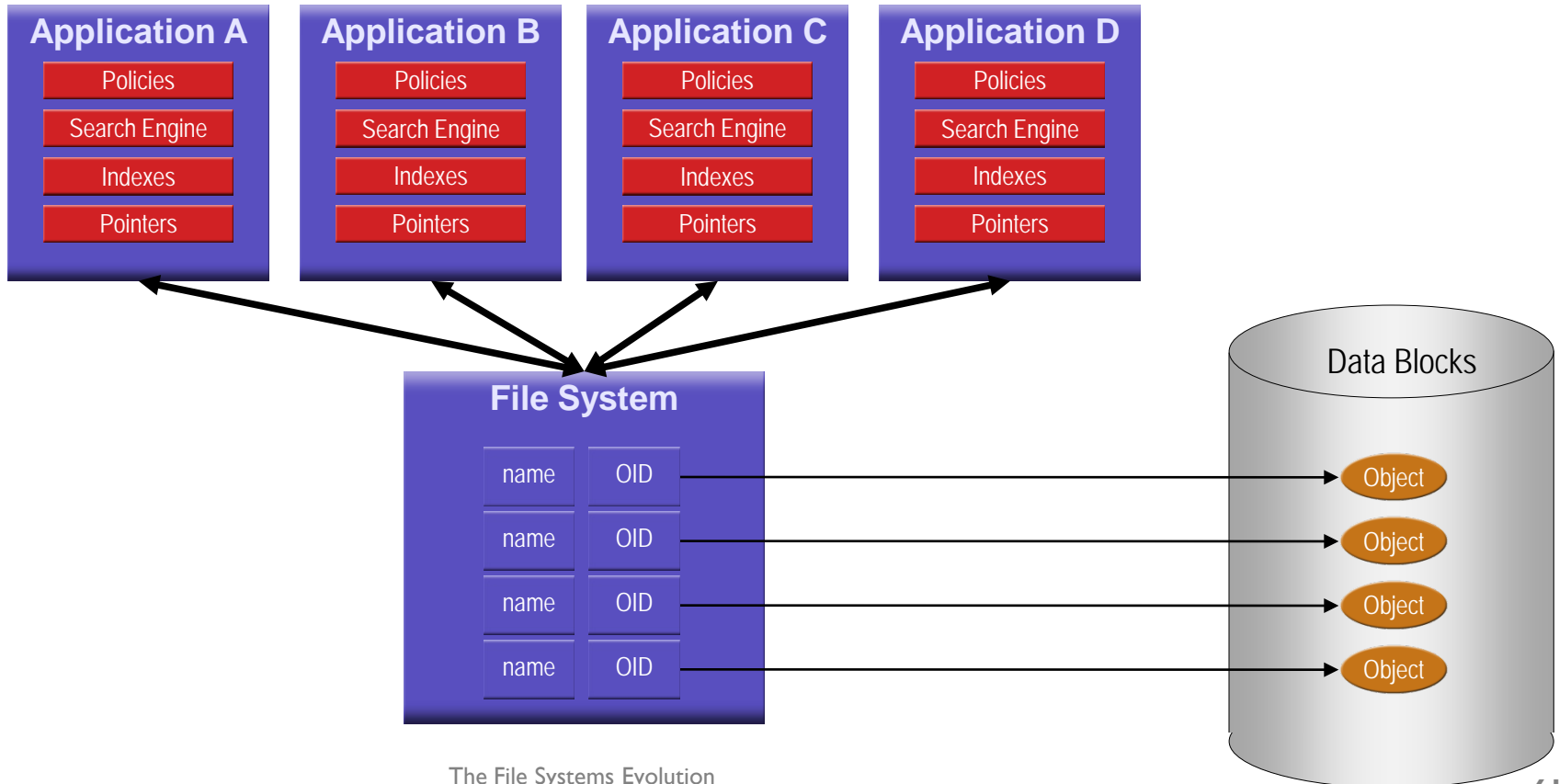
Managing File Objects (2)



- Indexes
- constraints/relationships
- Object search
- Full text search
- Join operations
- Virtual views
- SQL-like requests
- Cursors

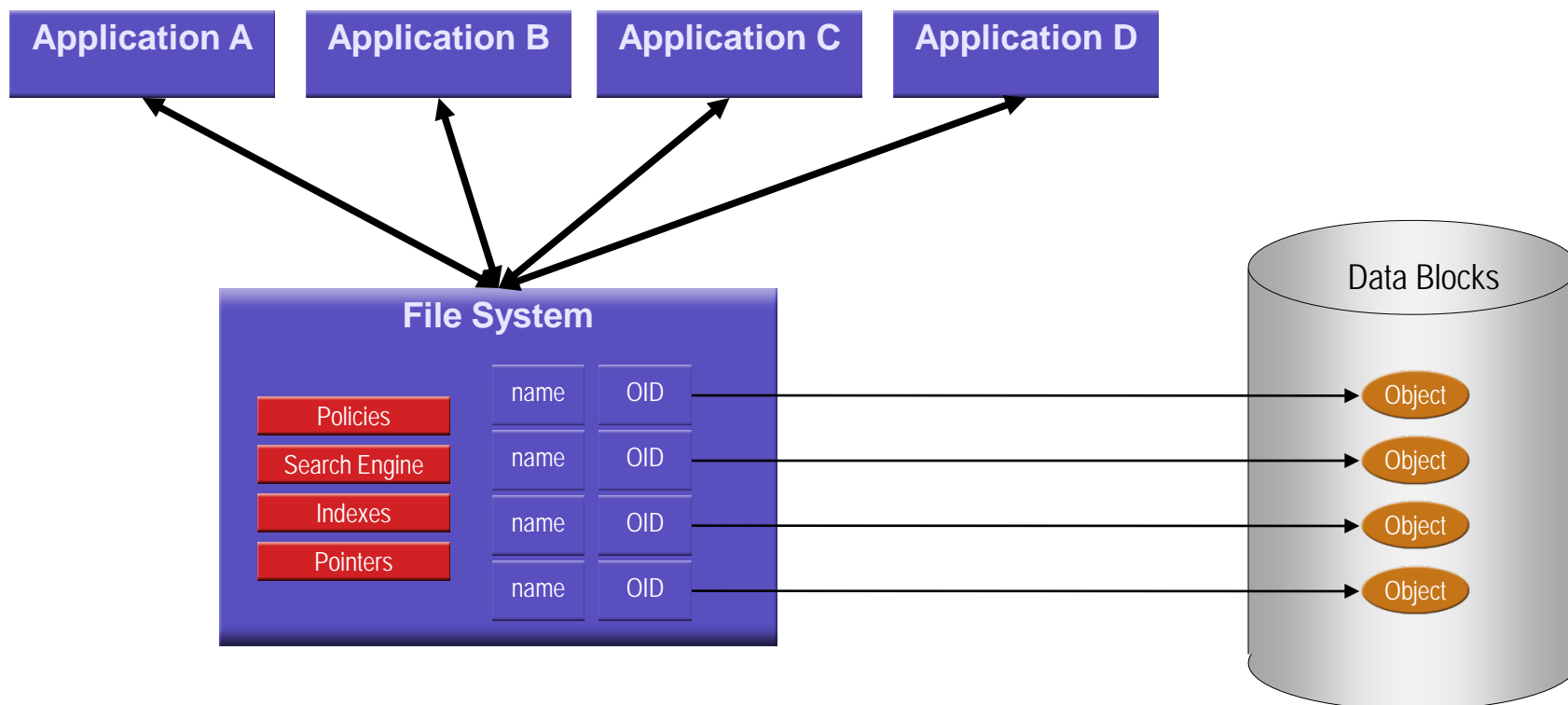
Application Content Repositories

- Combination of application, database, data services and pointers into external file system
- Application specific



File-Based Content Repositories

- Combination of files system, database and data services
- Application agnostic



Data Serving Hierarchy

3 Levels of Abstraction

- Application may interface with the storage subsystem in anyone of three layers:
 - ◆ **Block** – highest performance and very little meta data
 - ◆ **File** – high performance and some meta data
 - ◆ **Object** – medium performance and *rich* meta data

