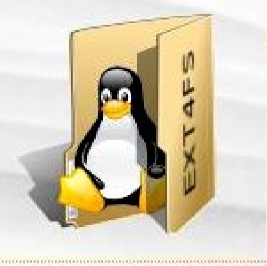
SOFTWARE UNIVERSITY



4 – Linux File Systems

Marian Marinov
CEO of 1H Ltd.
mm@1h.com

Stoyan Stoyanov
System Administrator
sto [at] softuni.bg

Agenda

- File System Architecture
- Virtual File System Layer
- Directory Structure
- Mount operations

Agenda

- File system types
  - Local
  - Log-structured - NAND
  - Pseudo
  - Network
  - Cluster
  - Distributed

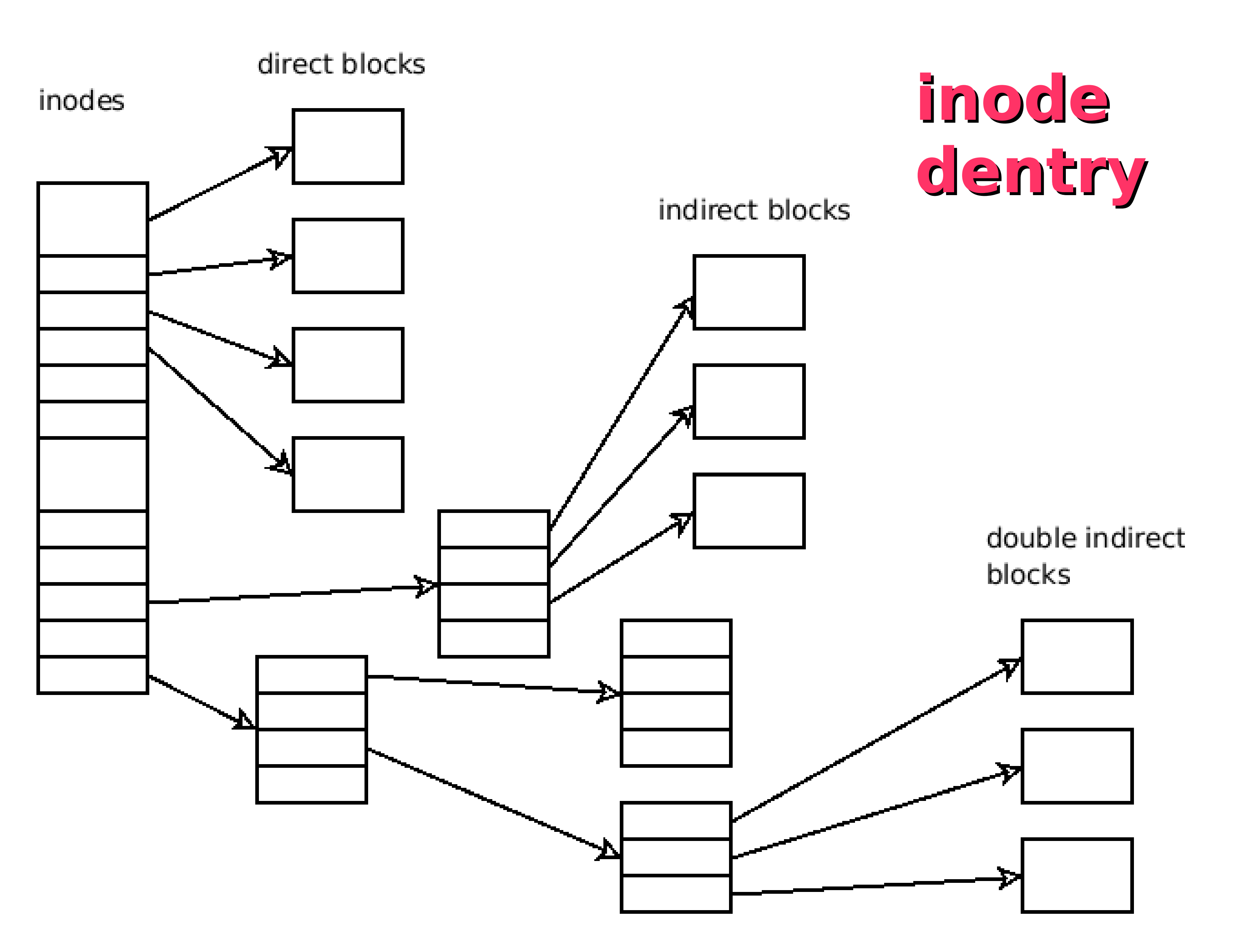
File System Architecture

There is NO silver bullet!

File System Architecture

There is NO DIRECTORY!

File System Architecture

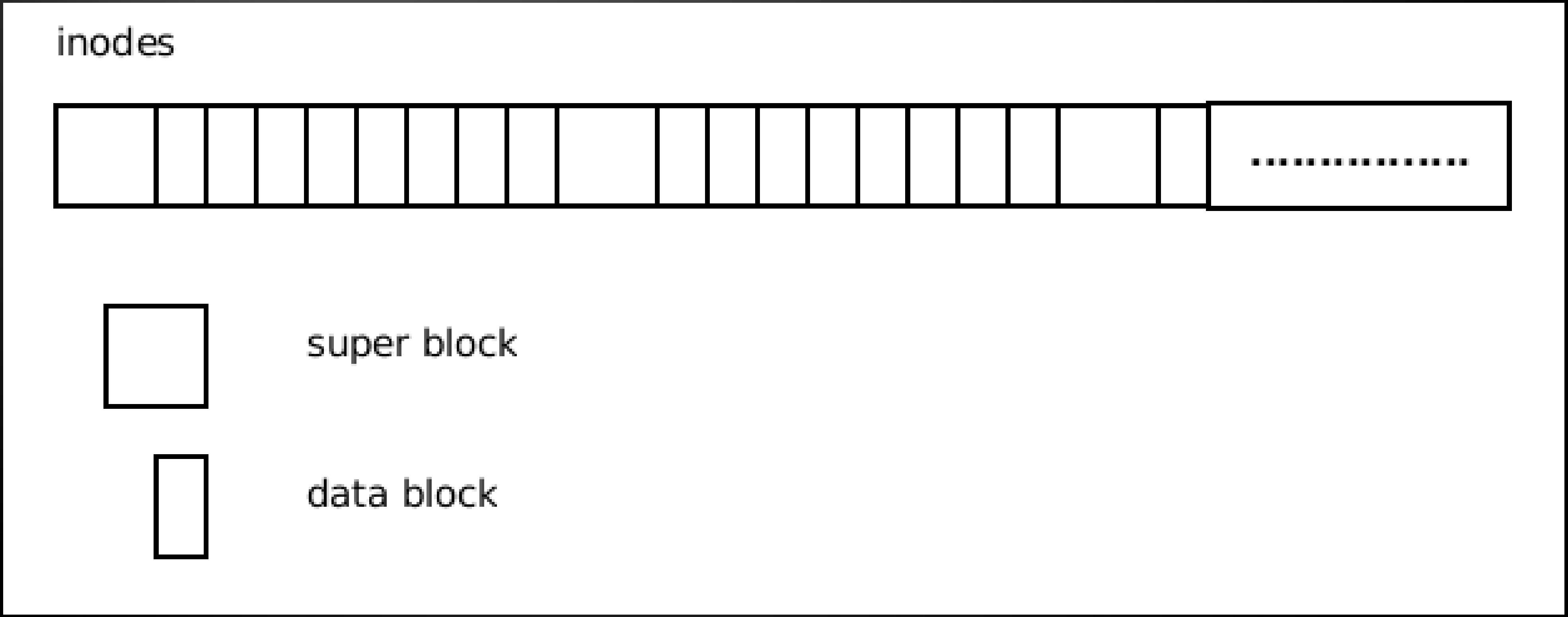


inode

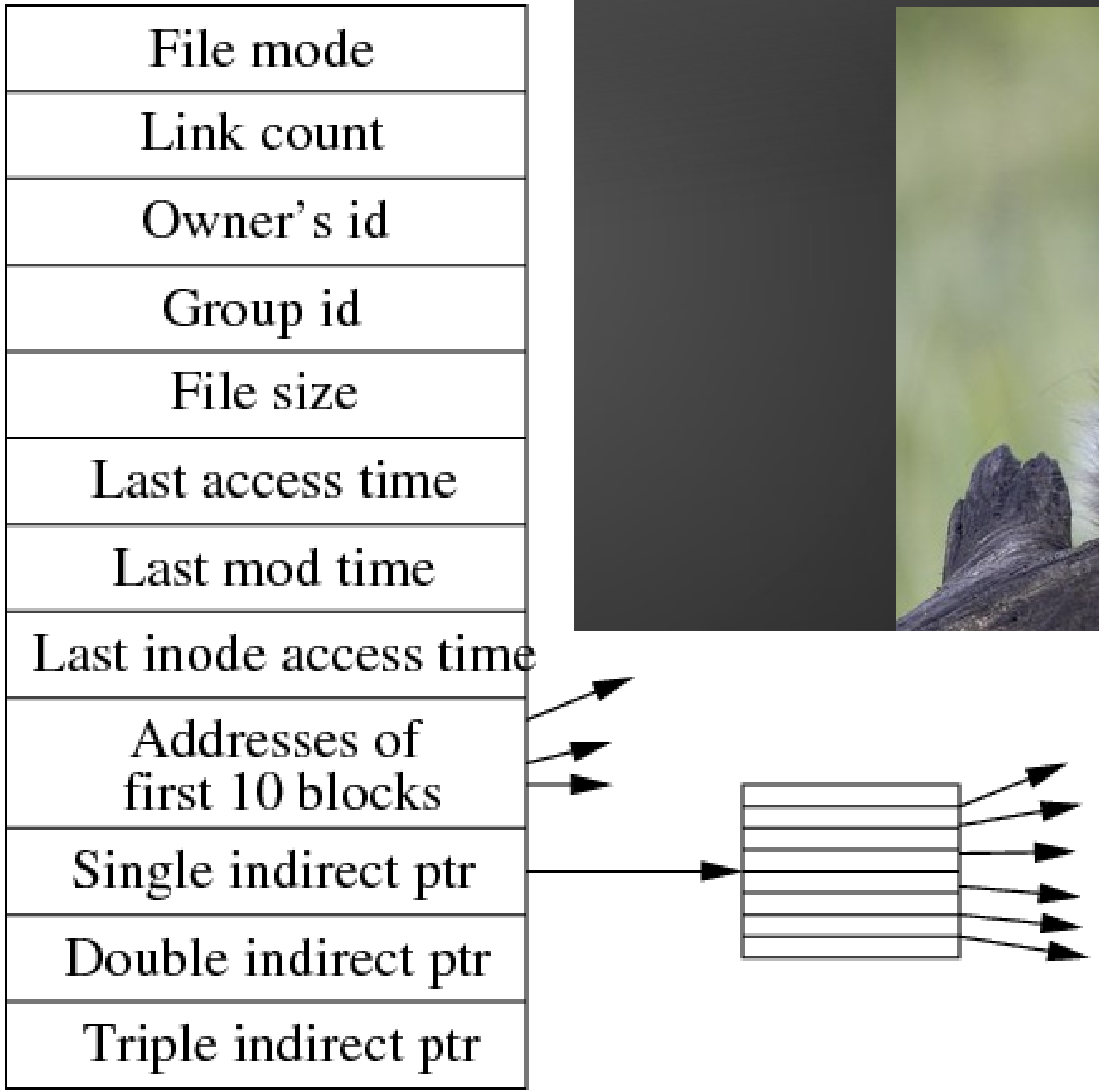
dentry

File System Architecture

Super Blocks



File System Architecture



File mode

Link count

Owner's id

Group id

File size

Last access time

Last mod time

Last inode access time

Addresses of first 10 blocks

Single indirect ptr

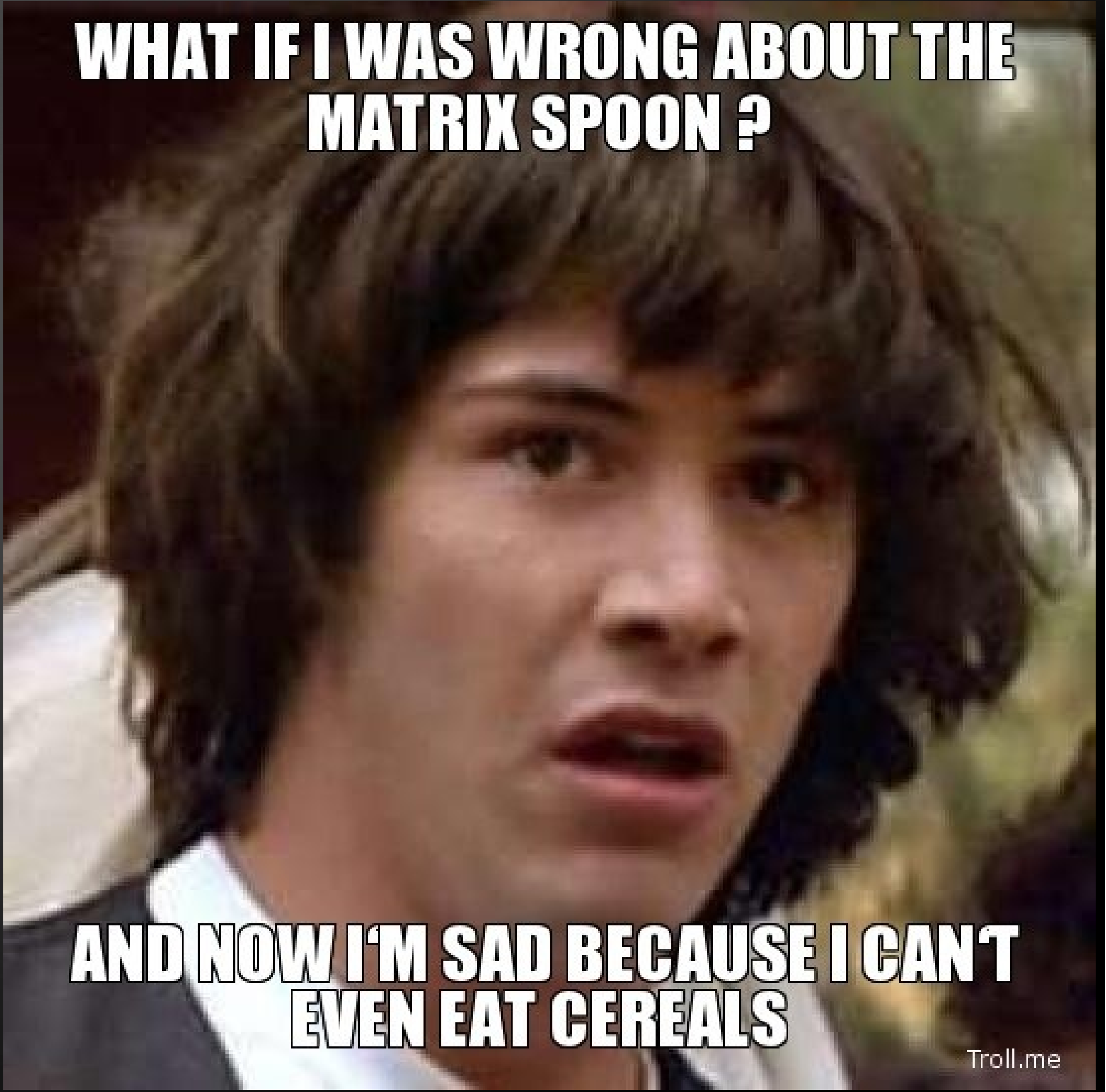
Double indirect ptr

Triple indirect ptr

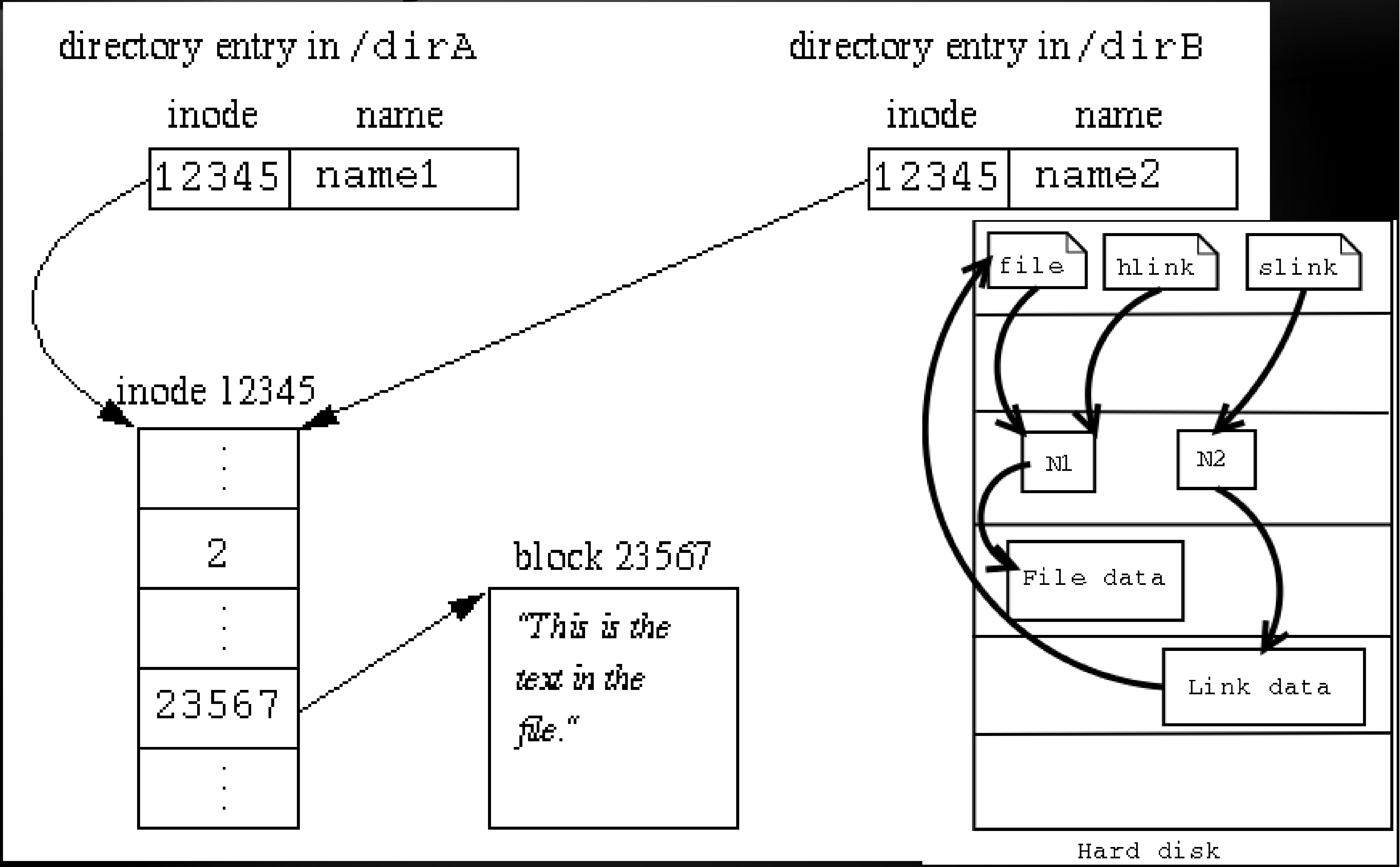
SOFTWARE UNIVERSITY

File System Architecture

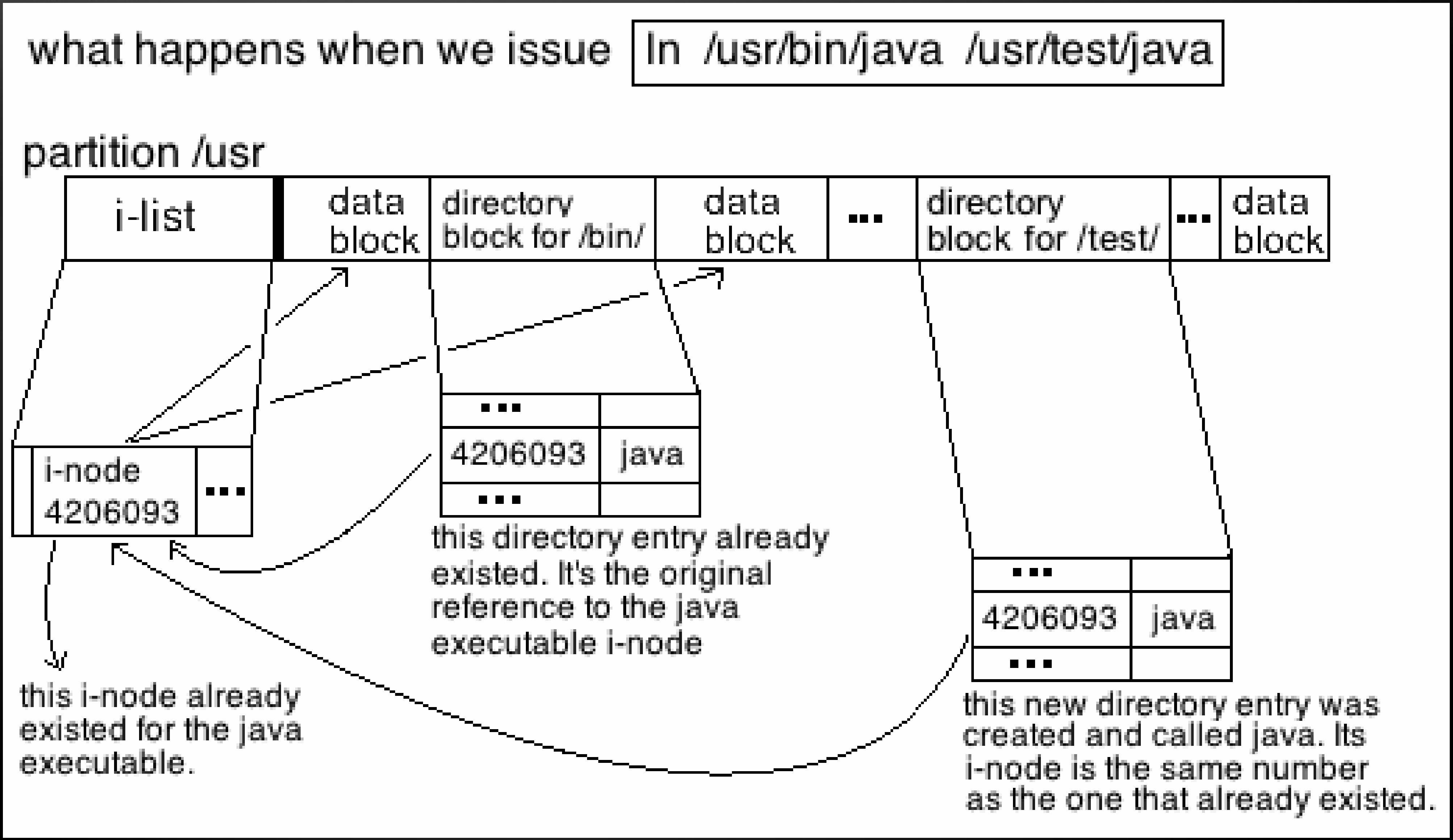
Soft/symlinks
Hardlinks


WHAT IF I WAS WRONG ABOUT THE MATRIX SPOON ?
AND NOW I'M SAD BECAUSE I CAN'T EVEN EAT CEREALS
Troll.me

File System Architecture



File System Architecture



what happens when we issue `ln /usr/bin/java /usr/test/java`

this i-node already existed for the java executable.

this directory entry already existed. It's the original reference to the java executable i-node

this new directory entry was created and called java. Its i-node is the same number as the one that already existed.

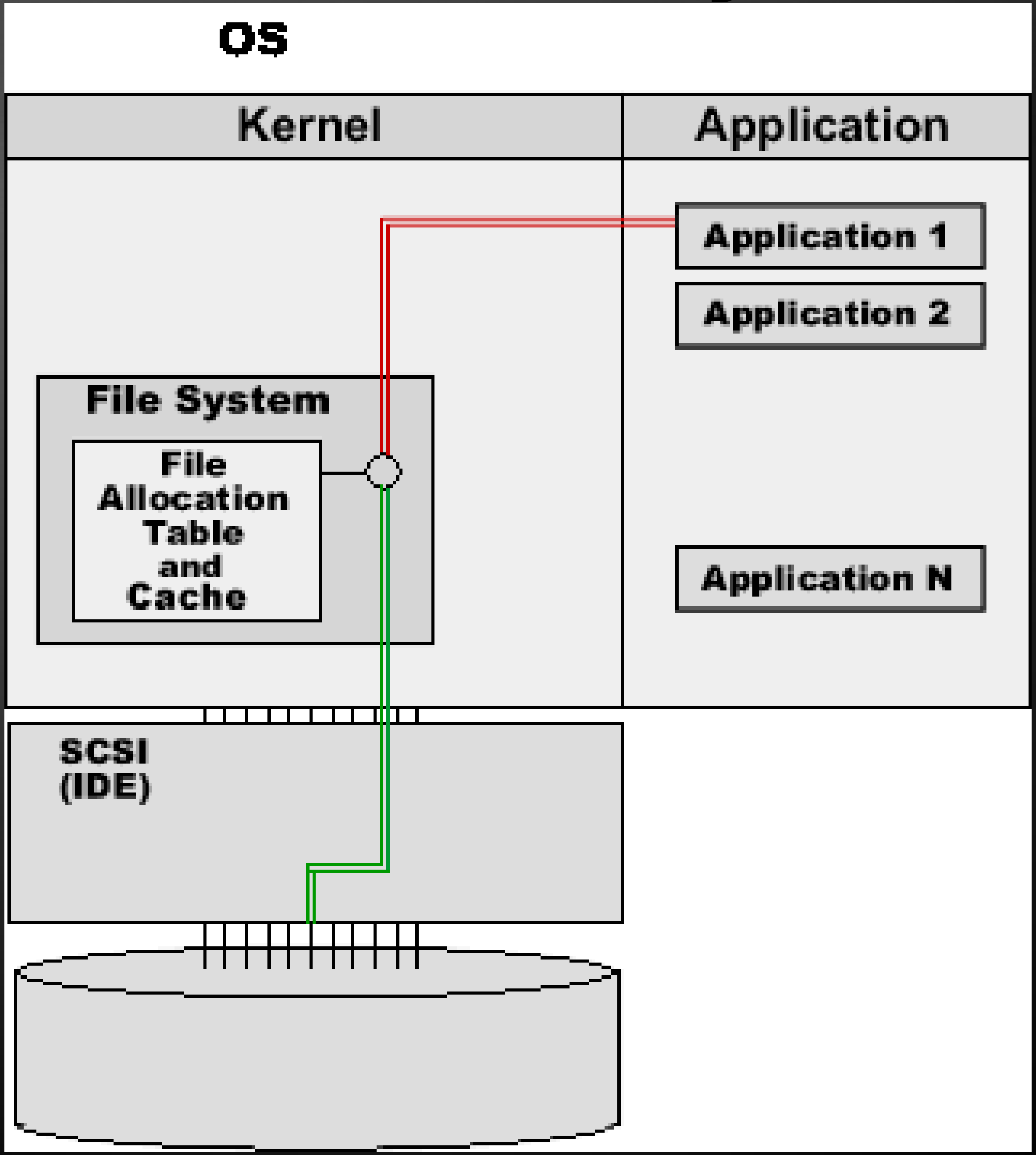
ln source destination

ln -s source destination

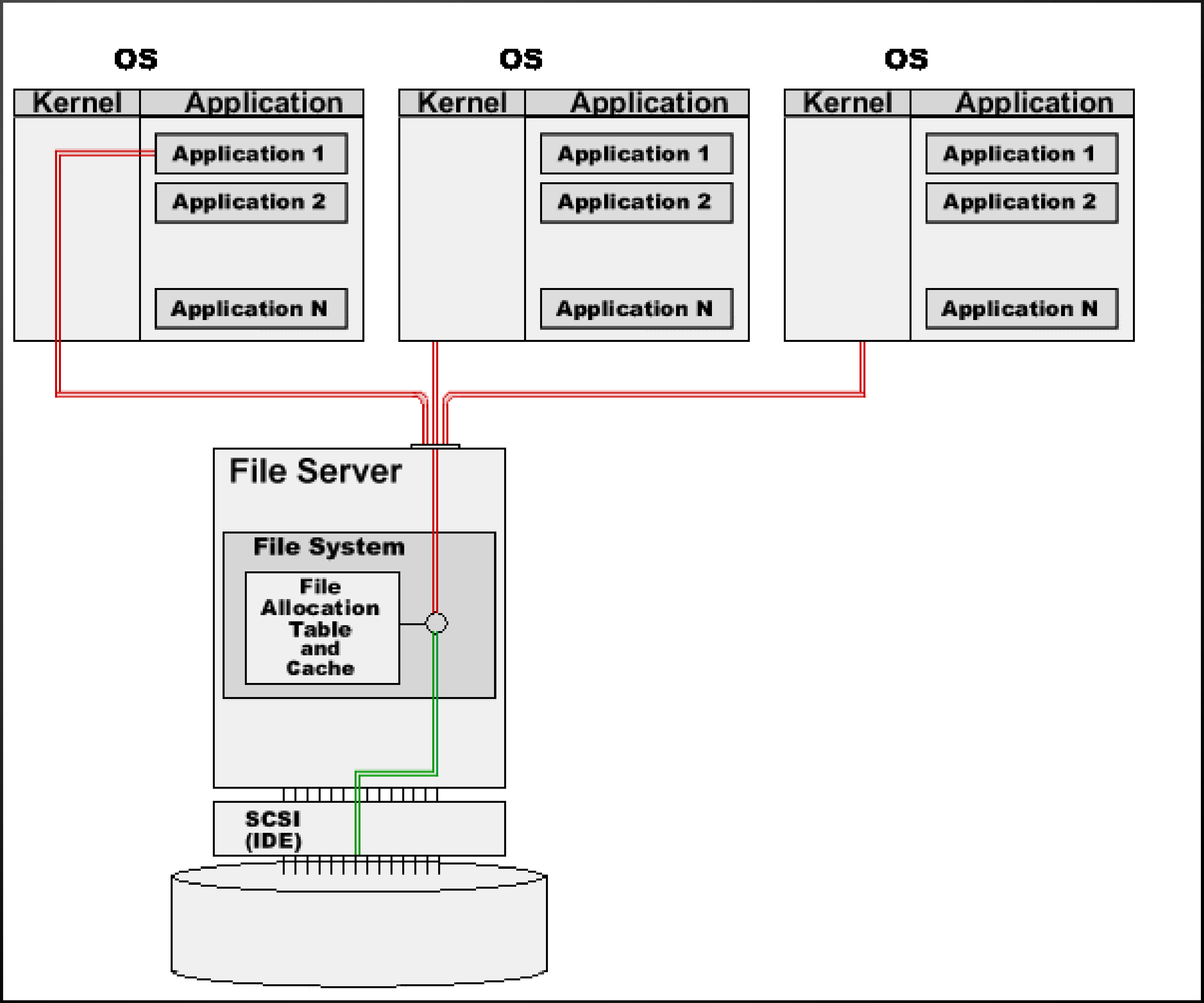
File System Architecture

DATA is never erased… it gets OVERWRITTEN!

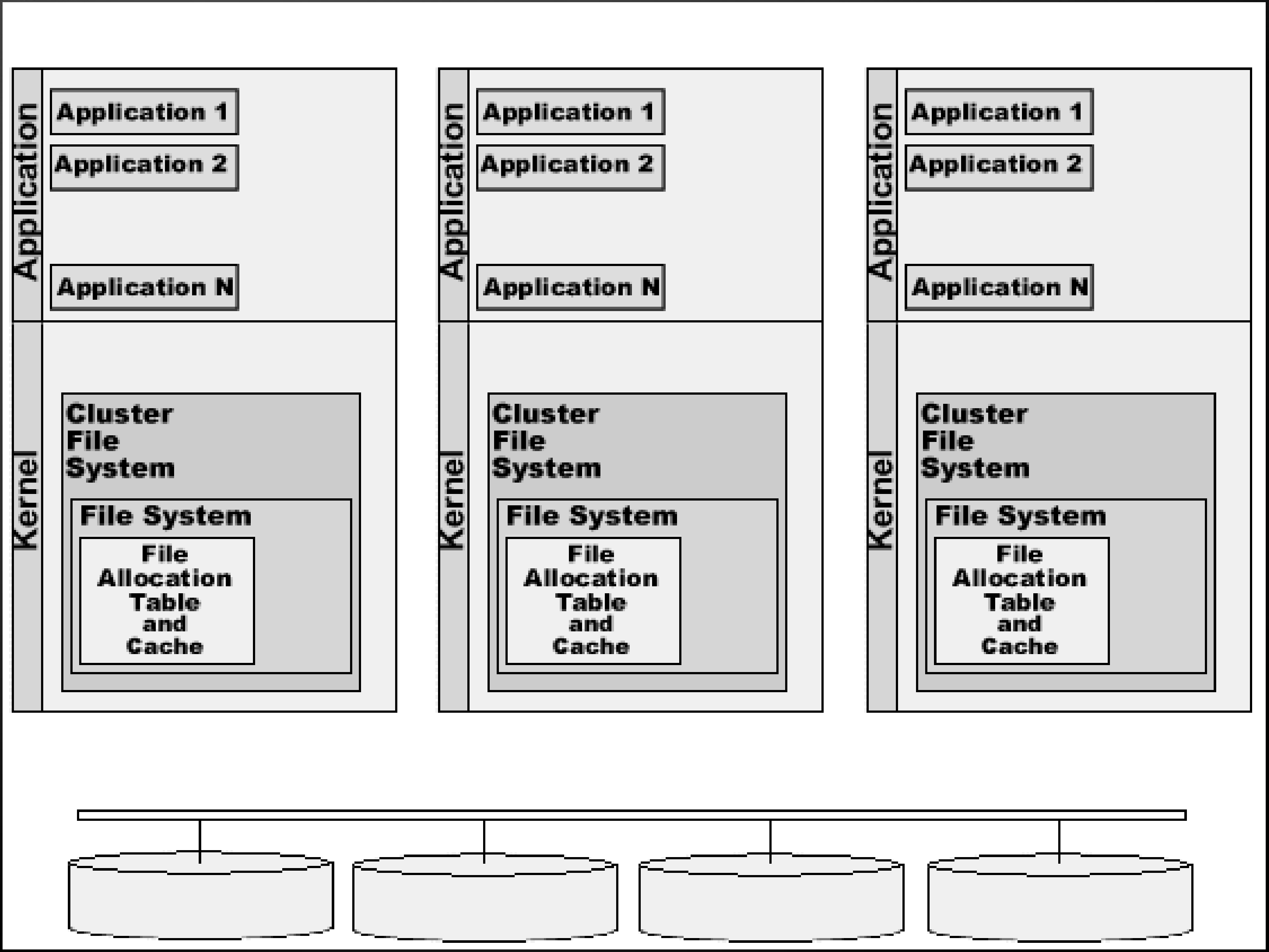
Local File System



Network File System



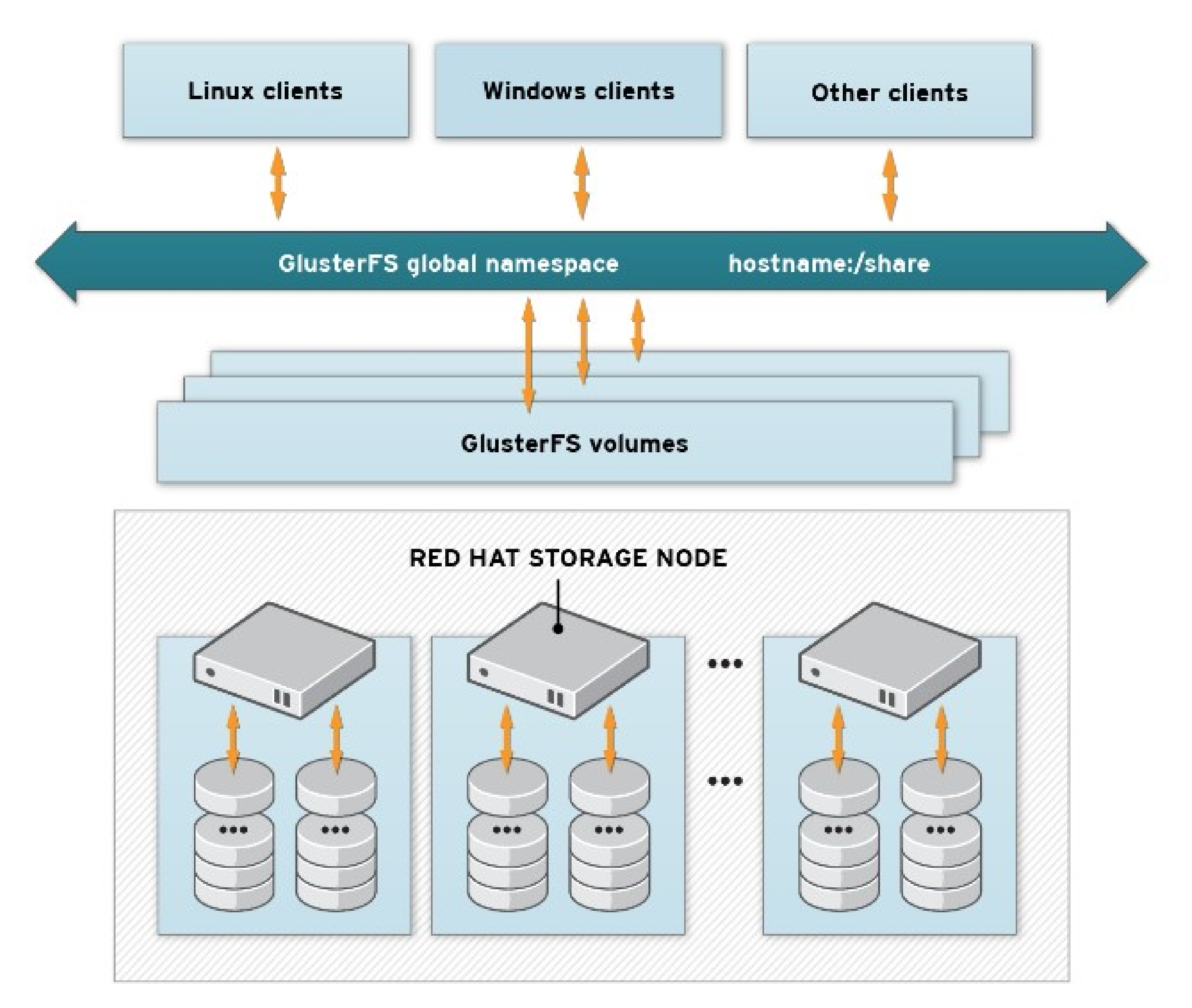
Cluster File System



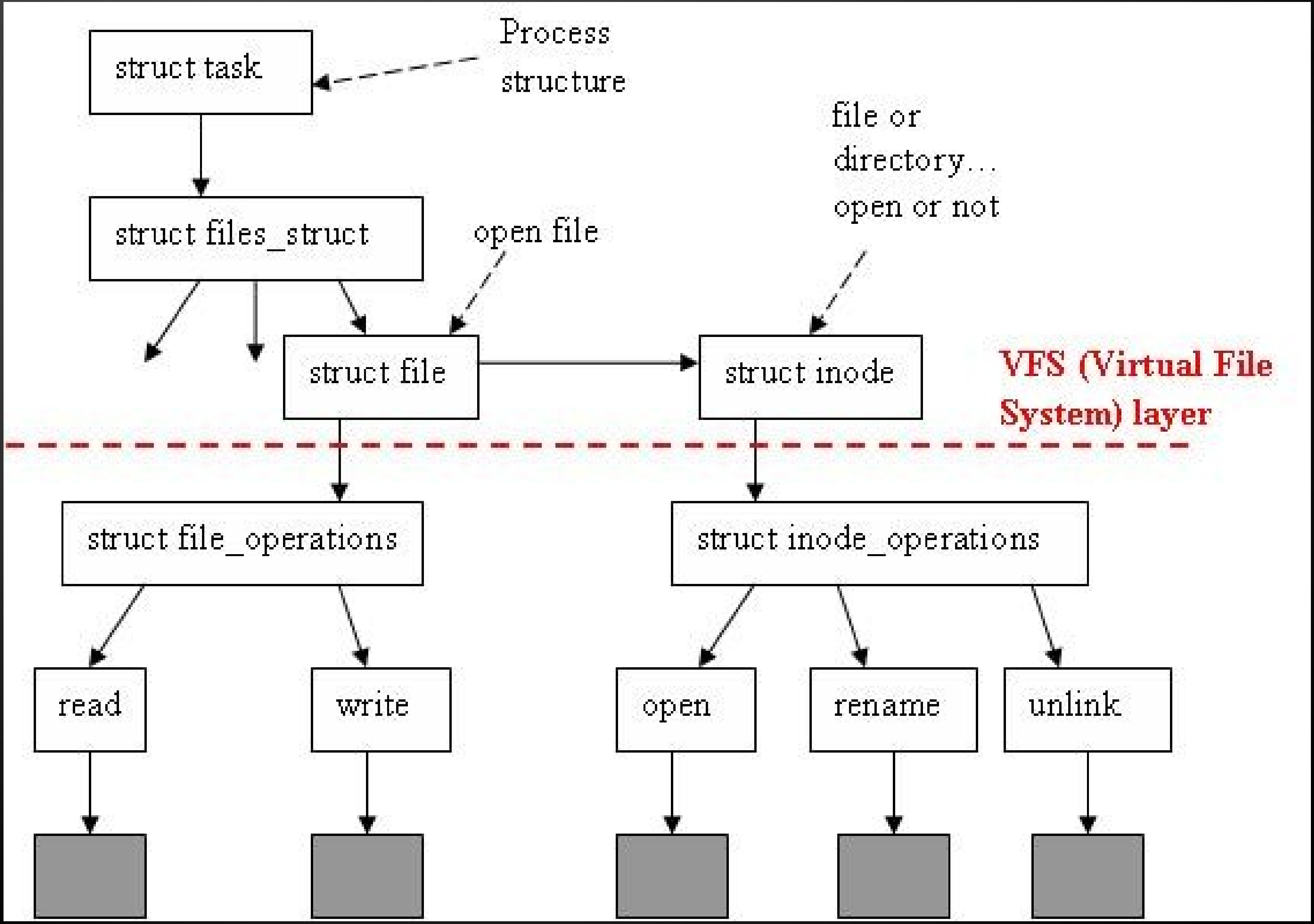
Cluster File System

“One DLM to rule them all, one DLM to mind them, one DLM to sync them all, and in the cluster, bind them”

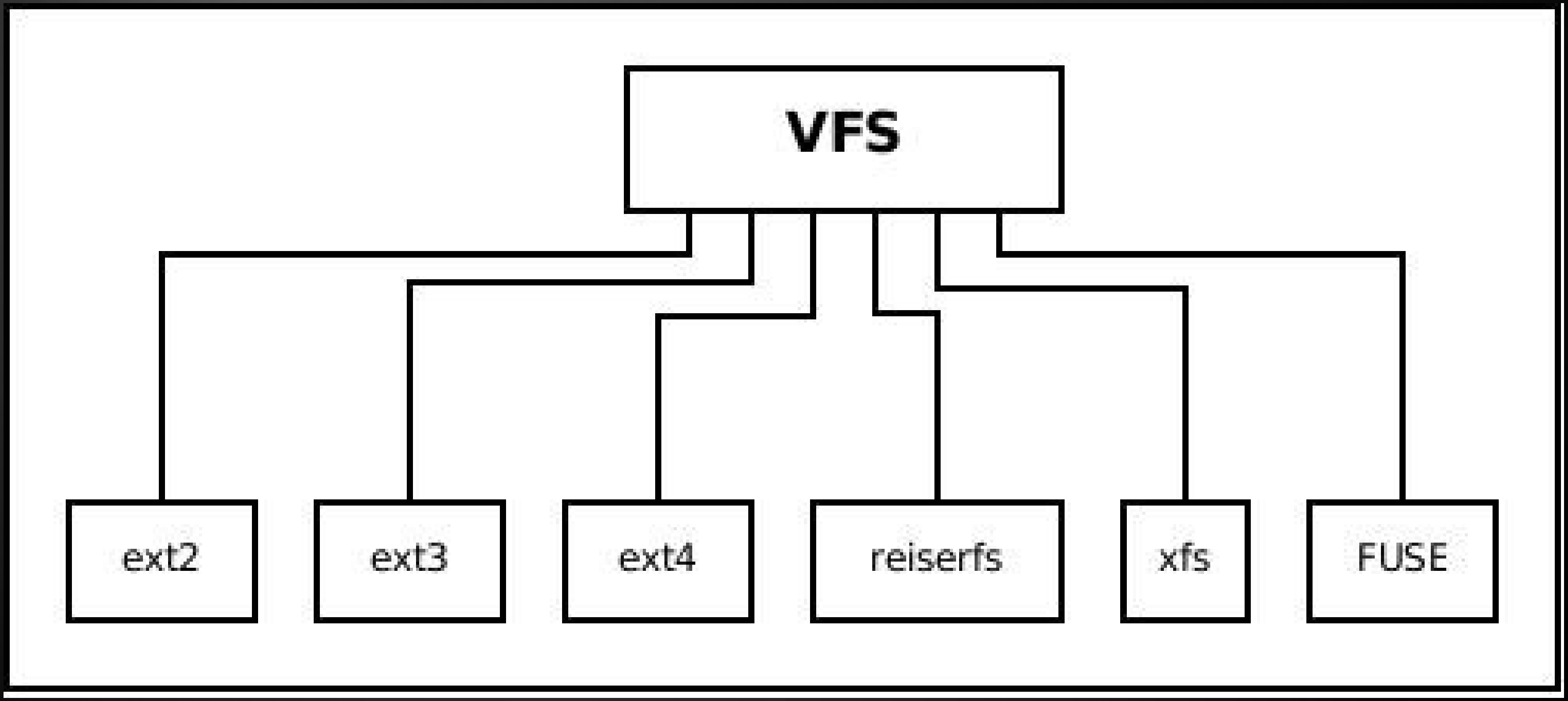
Distributed File System



Virtual File System Layer

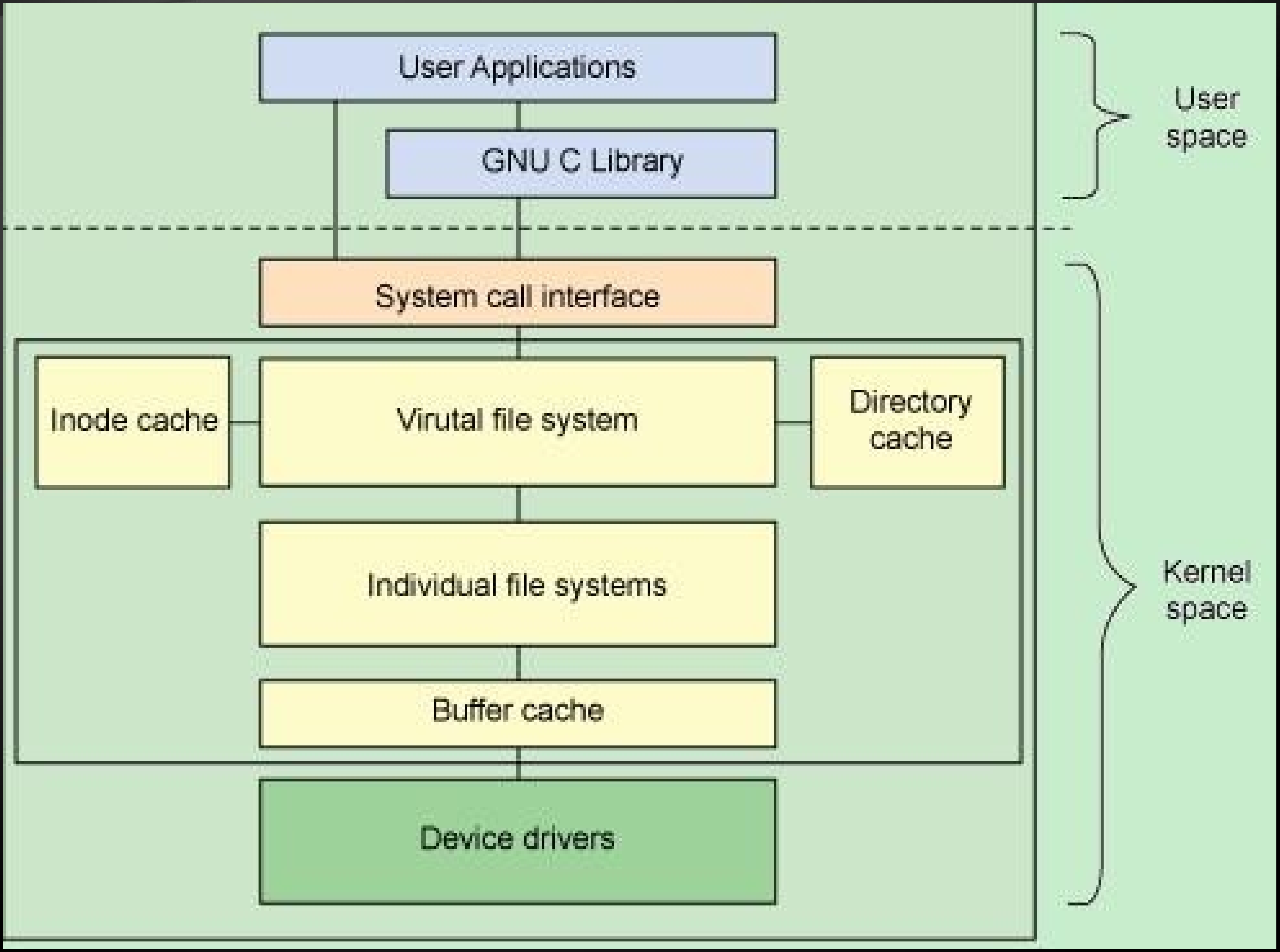


Virtual File System Layer

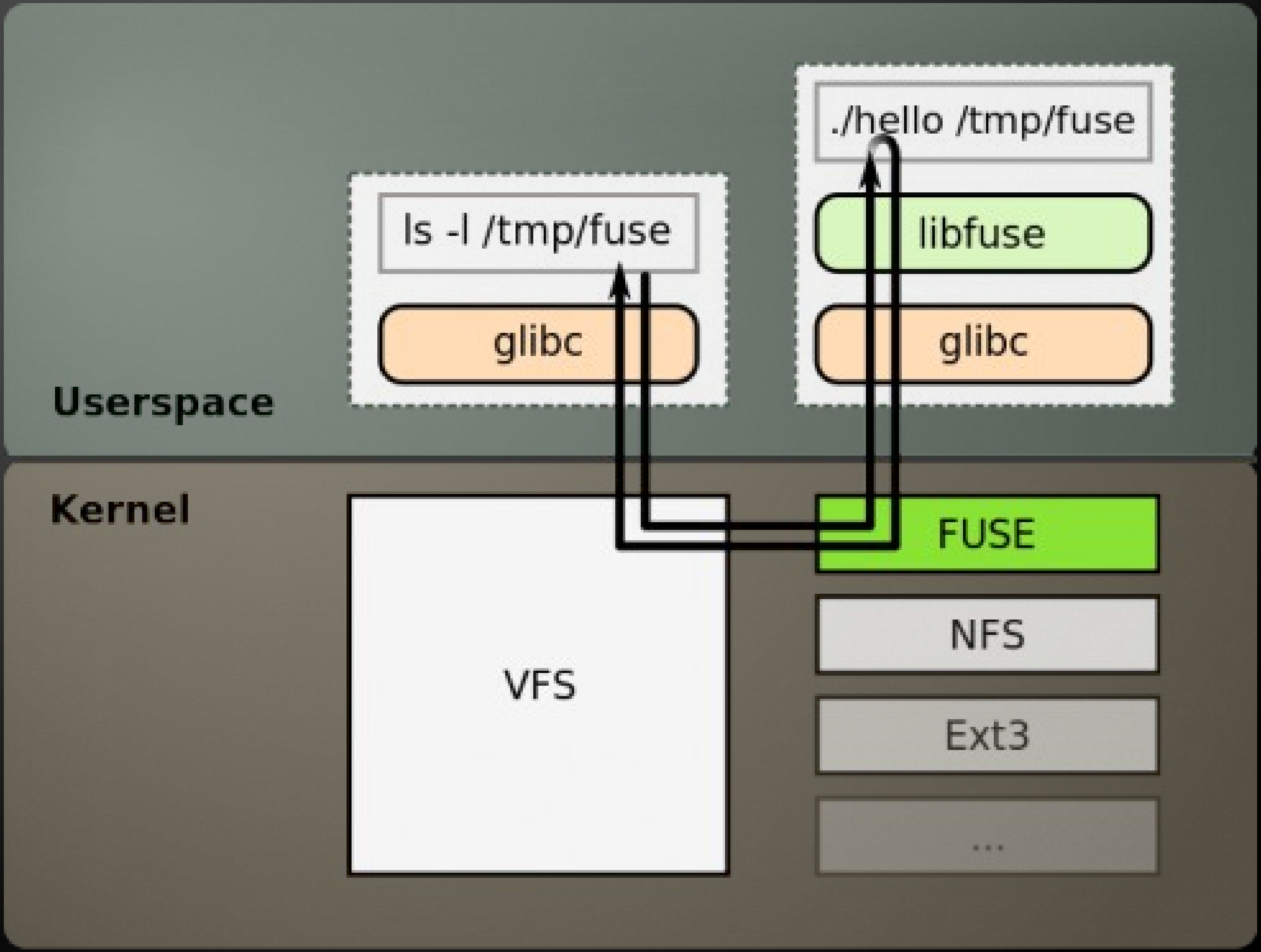


Introduced April 1992

VFS Caches


User Applications
GNU C Library
User space
System call interface
Inode cache
Virutal file system
Directory cache
Individual file systems
Buffer cache
Device drivers
Kernel space

FUSE


./hello /tmp/fuse
ls -l /tmp/fuse
libfuse
glibc
glibc
Userspace
Kernel
FUSE
NFS
VFS
Ext3
...

Mounting

- Attaching a device into the directory tree
- Mount point – a destination directory where a device is mounted
- Creates an entry in the kernel for each mounted device/dir
- /proc/mounts
  /etc/fstab, /etc/mtab

Mounting - CMD

- **cat /proc/partitions**
- **cat /proc/mounts**
- **mount**
- **umount**
- **/etc/fstab**

```
/dev/sdb2       /        ext4   defaults,noatime,nodiratime     0 0
/dev/sdb1       /boot    ext2   defaults,noatime,nodiratime     0 0
proc            /proc    proc   defaults                        0 0
tmpfs           /dev/shm   tmpfs   defaults                     0 0
/home/hackman   /fedora/home/hackman   none   rw,bind,auto   0 0
//10.2.0.11/share      /storage/beast        cifs
user=hackman,password=p1r@tk3,uid=1000,gid=1000,noauto 0 0
```

Ext/2/3/4

- **First Linux FS – MinixFS**
- **And there we go....**
  - **Ext - April 1992, Linux 0.96c**
  - **Ext2 – January 1993**
  - **Ext3 – November 2001**
  - **Ext4 – October 2006**

MinixFS

- Max. partition size – 64MB
- Max. file name size – 14 chars
- Ownership – uid, gid
- Permissions – user, group, others

Ext

- Max. partition size – 2GB
- Max. file name size – 255 chars
- No support for time stamps
  - Access
  - Inode modification
  - Data modification

Ext2

- **Max. partition size – 32TB**
- **Max. file name size – 255 chars**
- **Max. file size – 2TB**
- **Max. Number of files - 10^{18}**
- **FS Perms**
- **Time stamps**

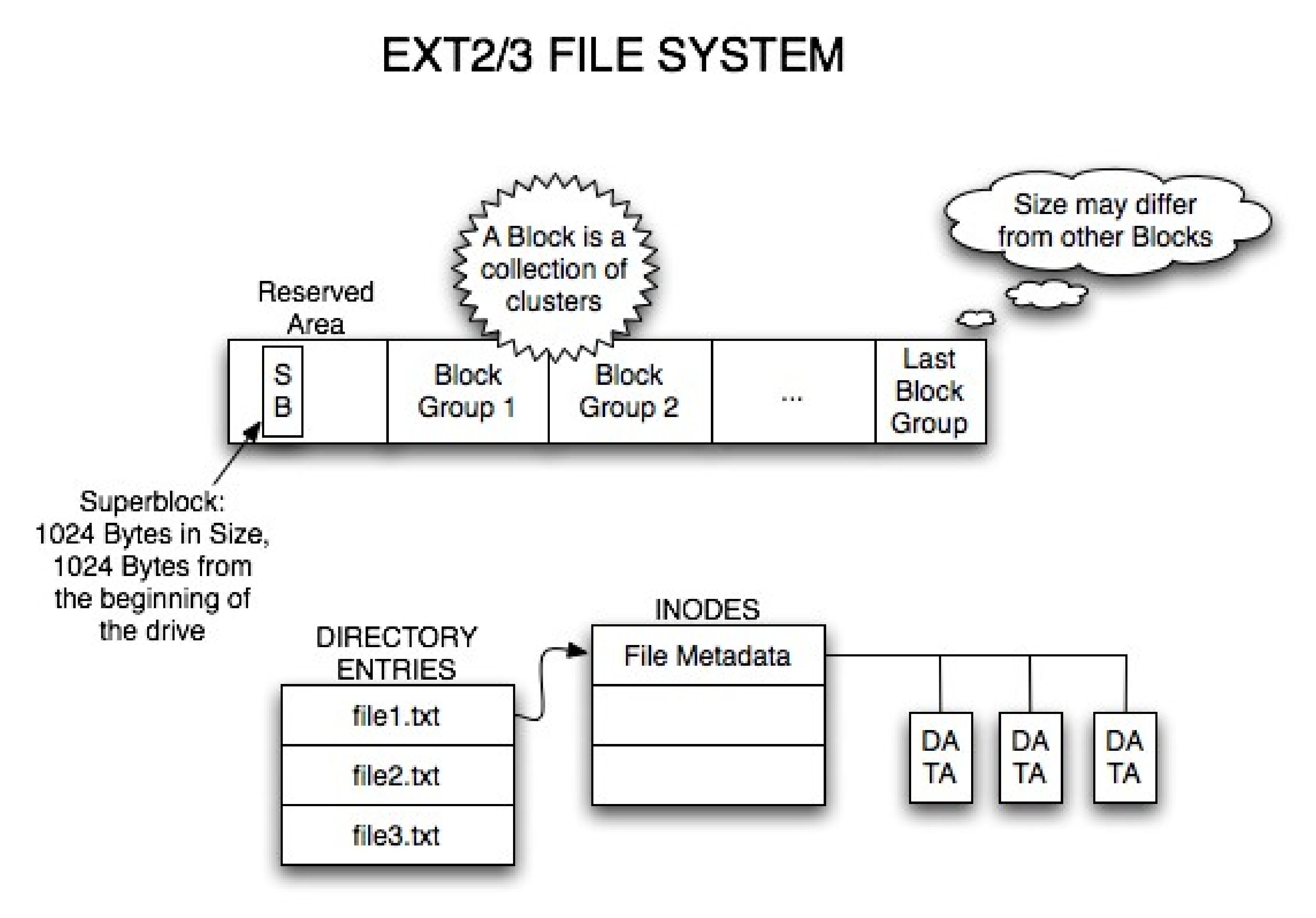
Ext3

- Max. partition size – 32TB
- Max. file name size – 255 chars
- Max. file size – 2TB
- Max. Number of files - 10^{18}
- Sub directory limit - 32,000
- Time stamps
- Has Jurnal

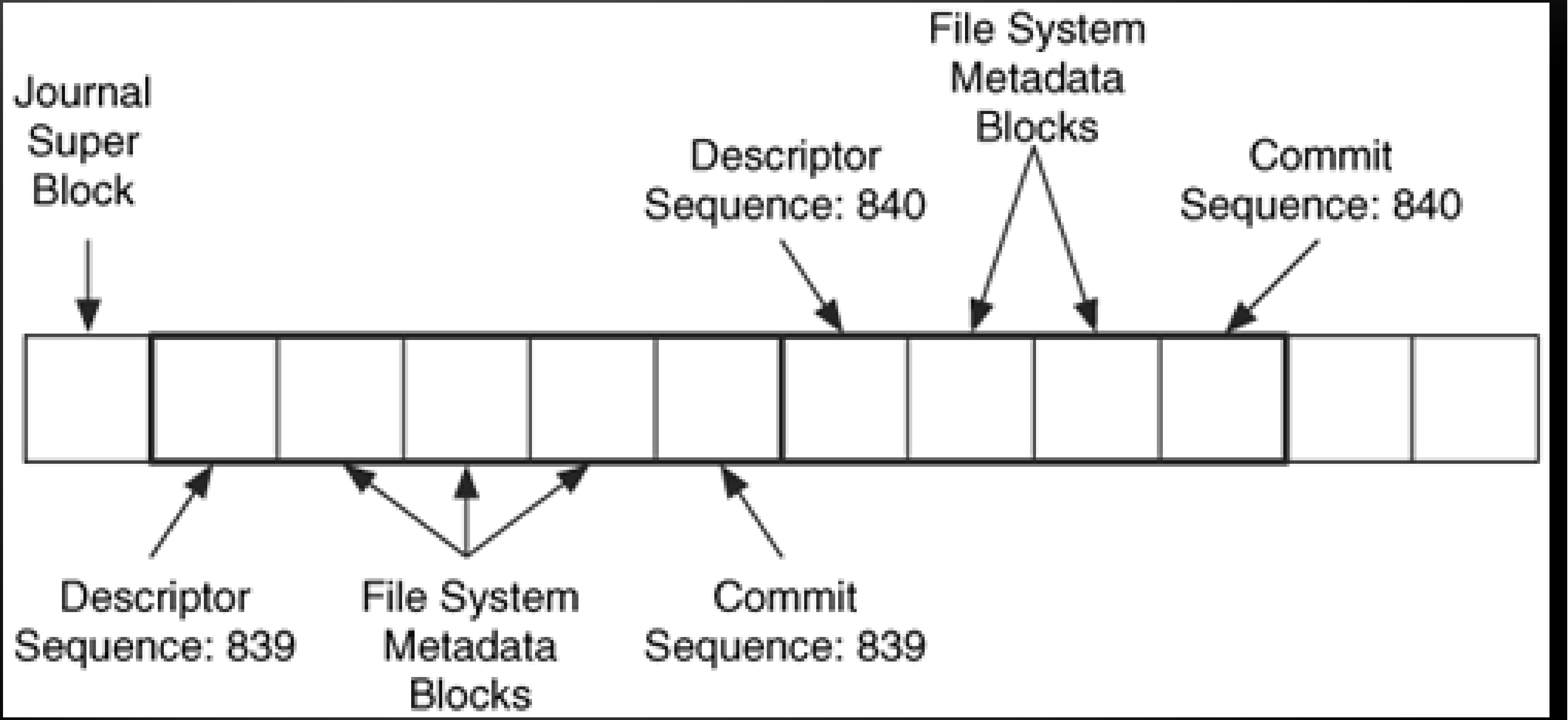
Ext4

- Max. partition size – 1EB
- Max. file name size – 255 chars
- Max. file size – 2TB
- Max. Number of files - 10^{18}
- Sub directory limit - 64,000
- File space pre-allocation
- File space delayed allocation

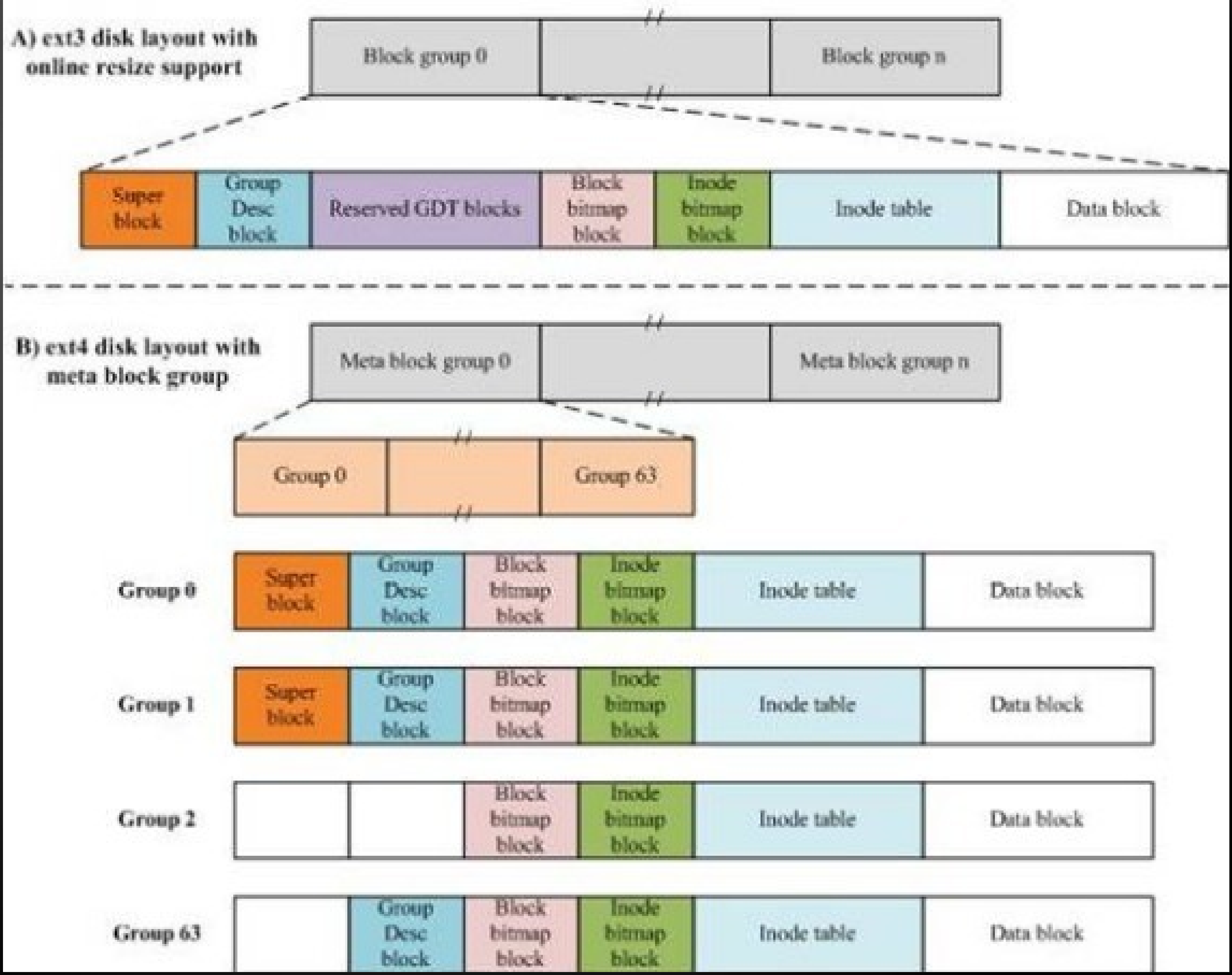
Ext/2/3


EXT2/3 FILE SYSTEM
A Block is a collection of clusters
Size may differ from other Blocks
Reserved Area
S B
Block Group 1
Block Group 2
...
Last Block Group
Superblock: 1024 Bytes in Size, 1024 Bytes from the beginning of the drive
INODES
File Metadata
DIRECTORY ENTRIES
file1.txt
file2.txt
file3.txt
DA TA
DA TA
DA TA

Ext3


File System
Metadata
Blocks
Journal
Super
Block
Descriptor
Sequence: 840
Commit
Sequence: 840
Descriptor
Sequence: 839
File System
Metadata
Blocks
Commit
Sequence: 839

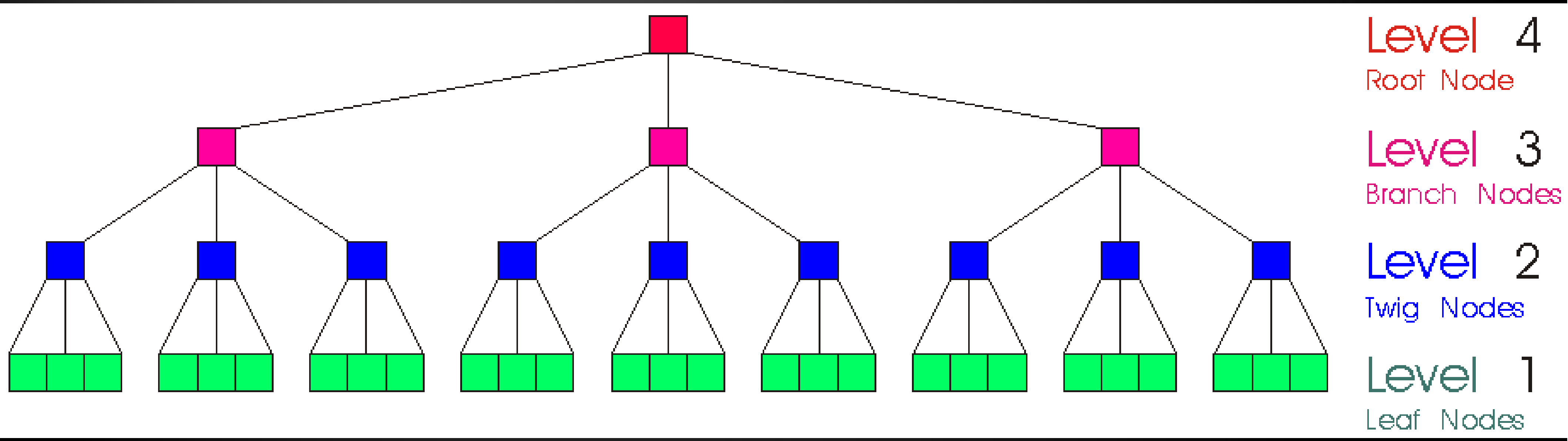
Ext4


A) ext3 disk layout with online resize support
Block group 0
Block group n
Super block
Group Desc block
Reserved GDT blocks
Block bitmap block
Inode bitmap block
Inode table
Data block
B) ext4 disk layout with meta block group
Meta block group 0
Meta block group n
Group 0
Group 63
Group 0
Super block
Group Desc block
Block bitmap block
Inode bitmap block
Inode table
Data block
Group 1
Super block
Group Desc block
Block bitmap block
Inode bitmap block
Inode table
Data block
Group 2
Block bitmap block
Inode bitmap block
Inode table
Data block
Group 63
Group Desc block
Block bitmap block
Inode bitmap block
Inode table
Data block

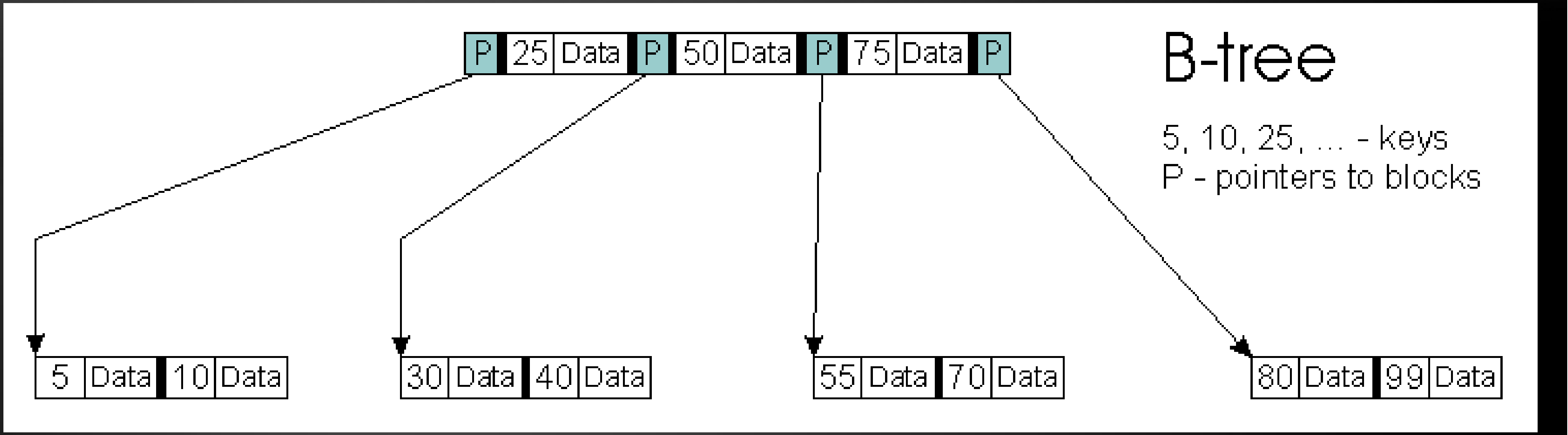
ReiserFS

- Introduced 2001
- Metadata-only journaling
- Online resizing (growth only)
- Tail packing, a scheme to reduce internal fragmentation.
- Max. file size - 1EB
- Max. number of files - 2^{32}

ReiserFS

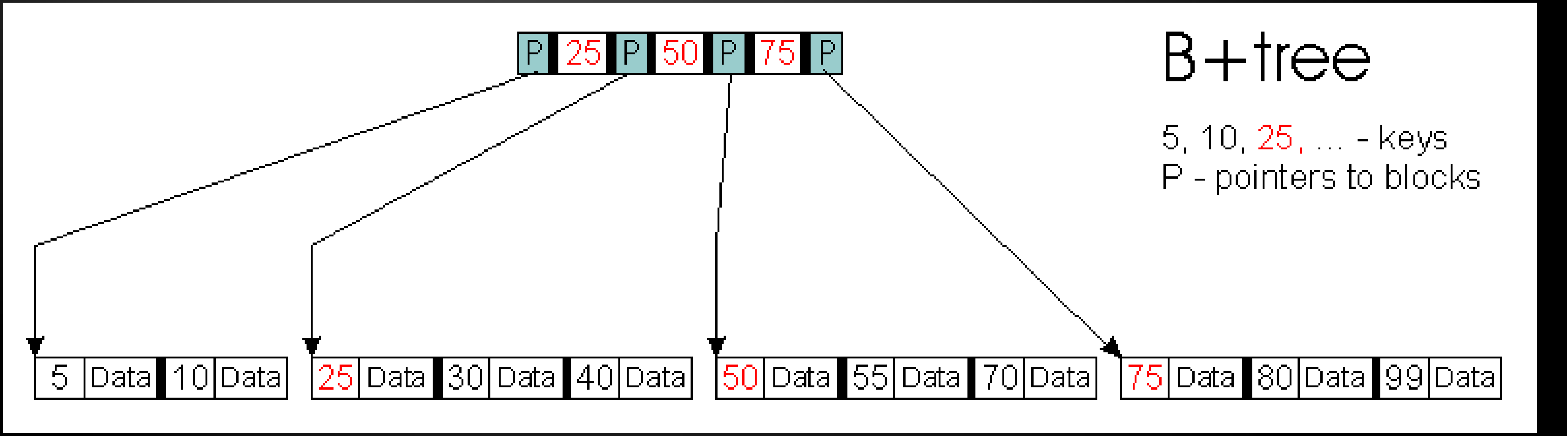

Level 4
Root Node
Level 3
Branch Nodes
Level 2
Twig Nodes
Level 1
Leaf Nodes

ReiserFS



B-tree

5, 10, 25, … - keys
P - pointers to blocks



B+tree

5, 10, 25, … - keys
P - pointers to blocks

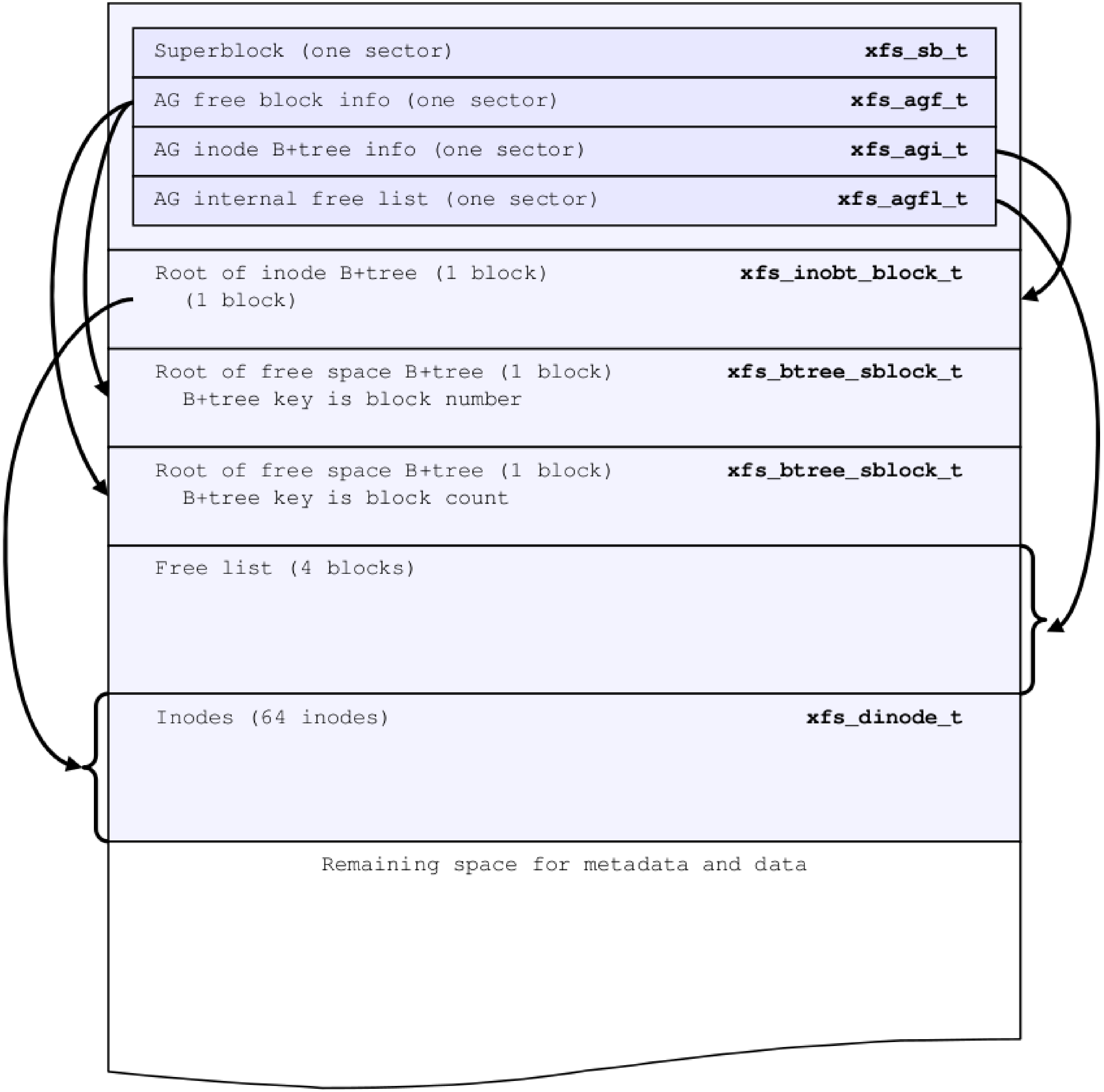
XFS

- Introduced 2001
- Max file size 8 EB
- Max volume size 16 EB
- Online resize(growth only)
- Online defragmentation

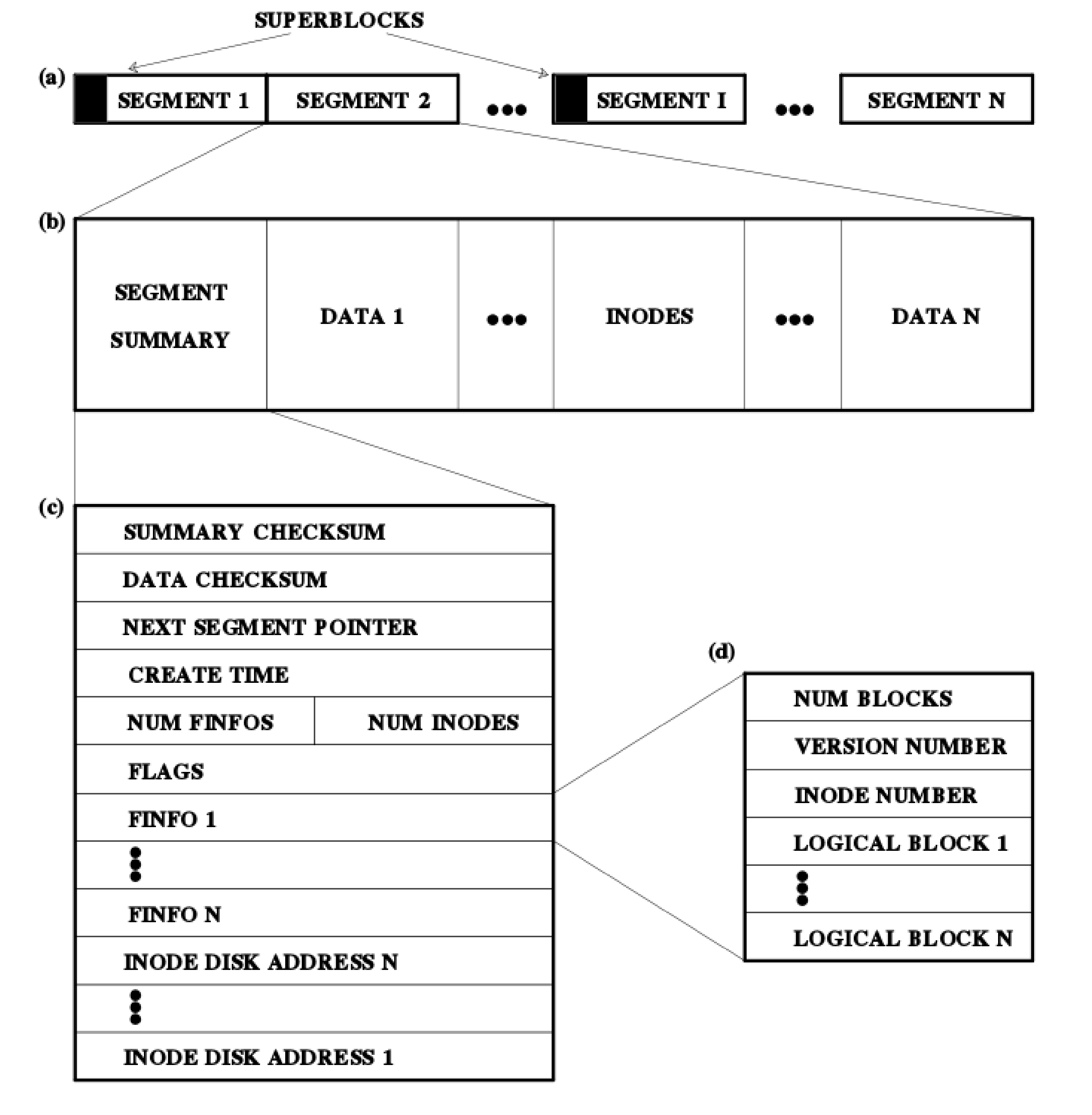
XFS

Equally sized chunks

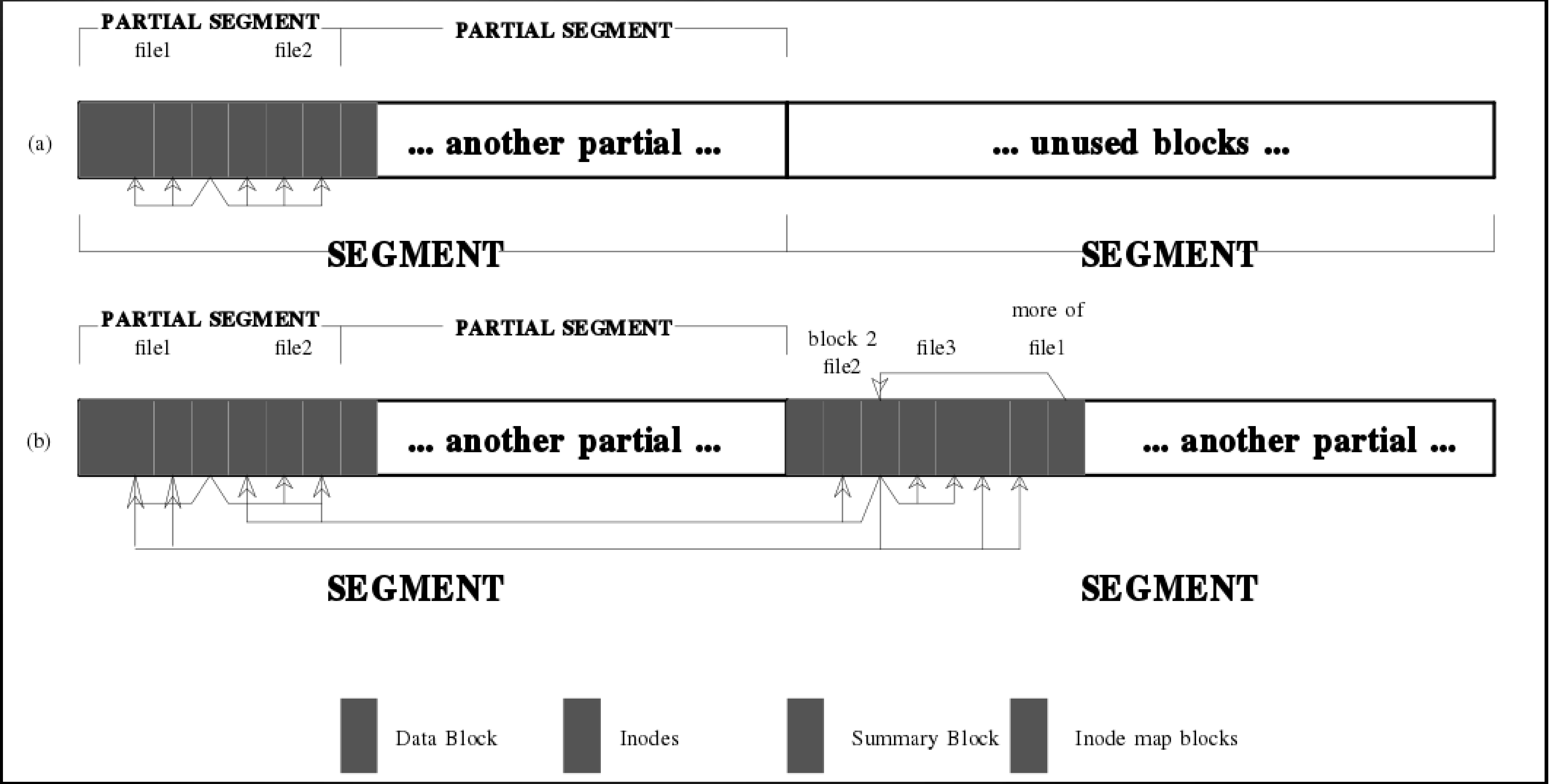
Allocation groups – AG


Superblock (one sector)
xfs_sb_t
AG free block info (one sector)
xfs_agf_t
AG inode B+tree info (one sector)
xfs_agi_t
AG internal free list (one sector)
xfs_agfl_t
Root of inode B+tree (1 block)
(1 block)
xfs_inobt_block_t
Root of free space B+tree (1 block)
B+tree key is block number
xfs_btree_sblock_t
Root of free space B+tree (1 block)
B+tree key is block count
xfs_btree_sblock_t
Free list (4 blocks)
Inodes (64 inodes)
xfs_dinode_t
Remaining space for metadata and data

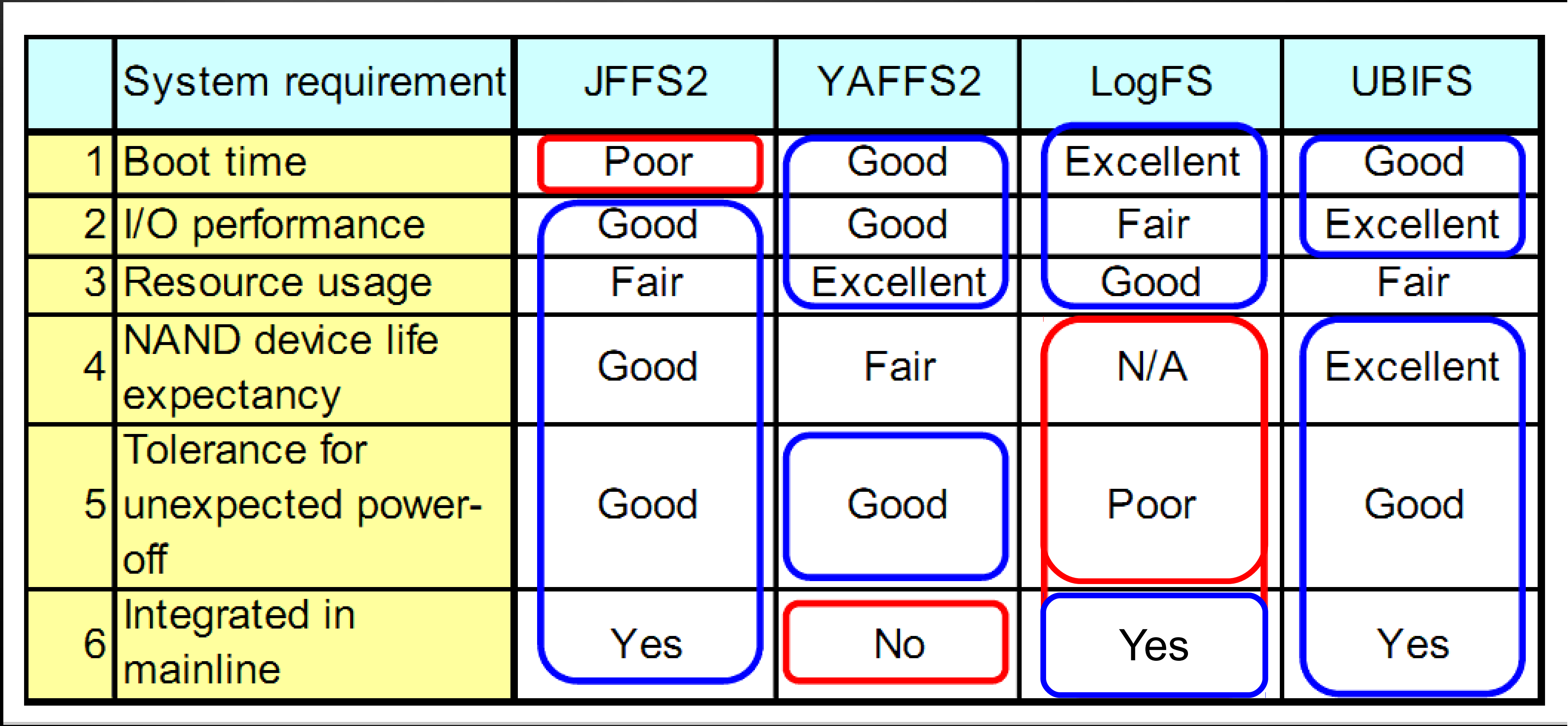
Log-structured File Systems Architecture



Log-structured File Systems Architecture



Log-structured NAND File Systems



	System requirement	JFFS2	YAFFS2	LogFS	UBIFS
1	Boot time	Poor	Good	Excellent	Good
2	I/O performance	Good	Good	Fair	Excellent
3	Resource usage	Fair	Excellent	Good	Fair
4	NAND device life expectancy	Good	Fair	N/A	Excellent
5	Tolerance for unexpected power-off	Good	Good	Poor	Good
6	Integrated in mainline	Yes	No	Yes	Yes

NILFS2
F2FS

Pseudo File Systems

- procfs
- sysfs
- debugfs
- configfs
- tmpfs
- others

Pseudo File Systems

- **debugfs is designed to provide Kernel Devs with simple way to push data into User space**
- **configfs is for creating, managing and destroying kernel objects from user-space**
- **sysfs is for viewing and manipulating objects from user-space which are created and destroyed by kernel space**

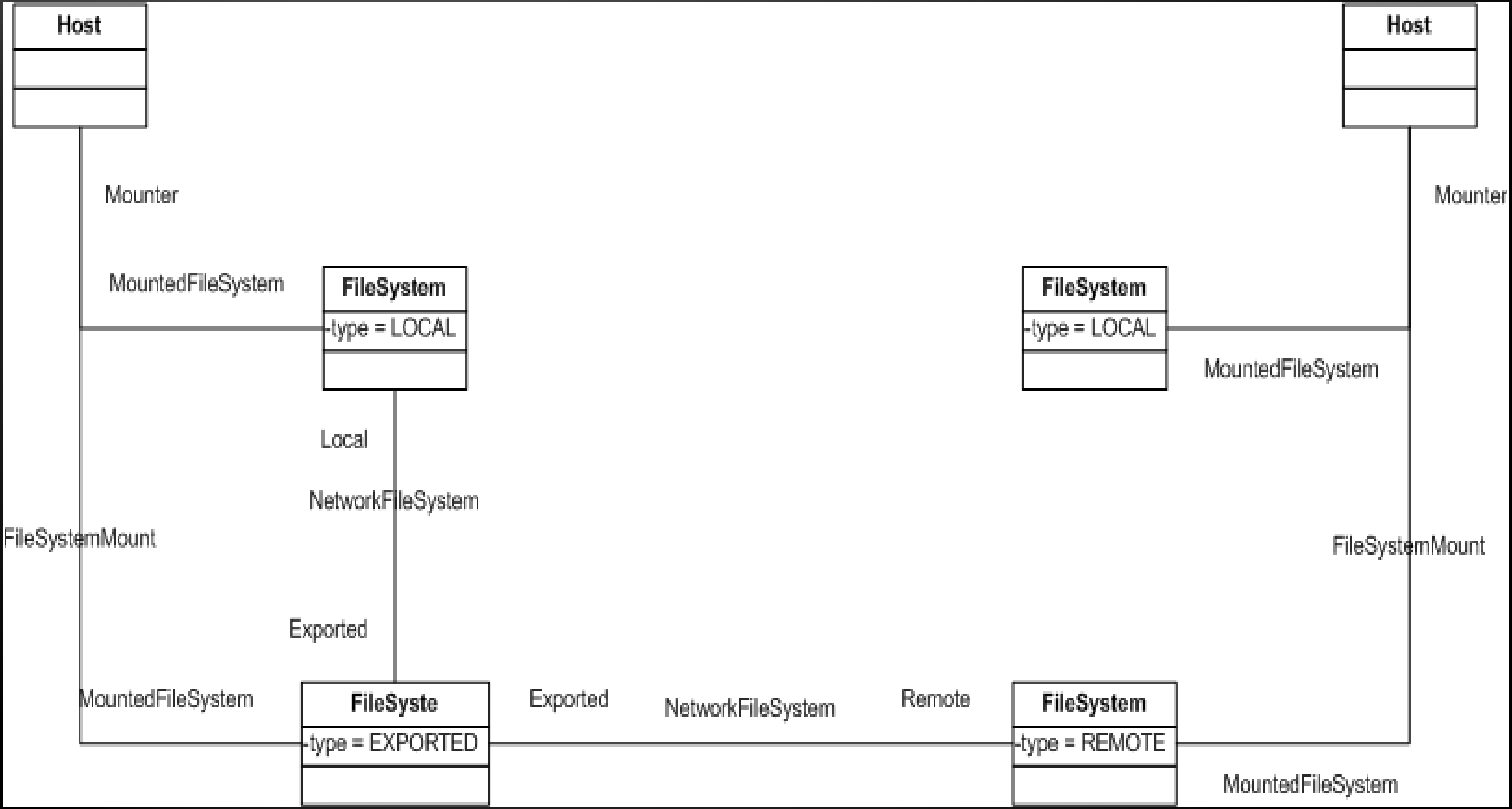
Pseudo File Systems

- **procfs is the first FS to provide easy access to kernel-space from user-space**
- **tmpfs is a very fast in-memory file system**

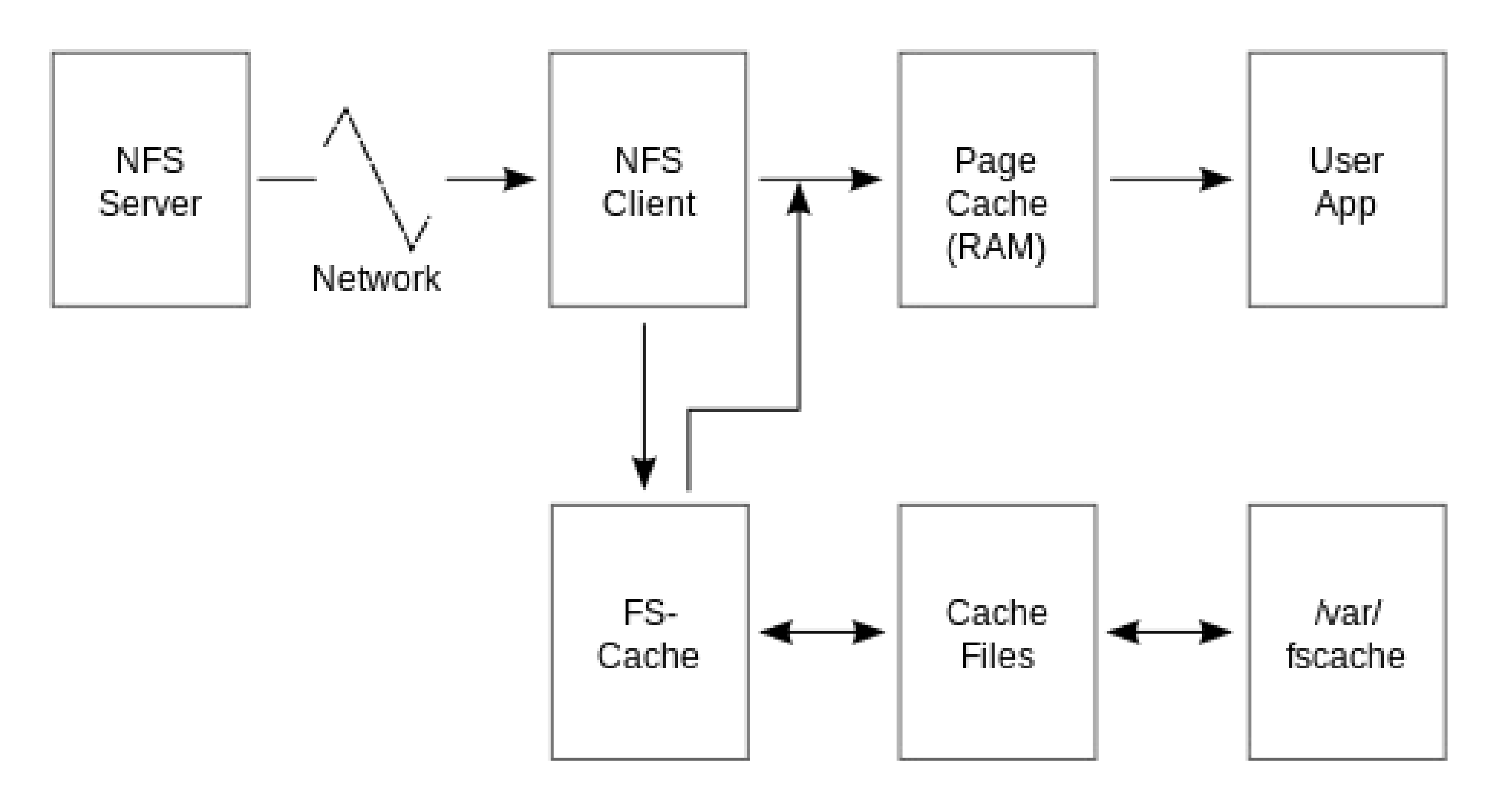
Network File Systems

- Network File System – NFS v3/v4
- Common Internet File System - CIFS

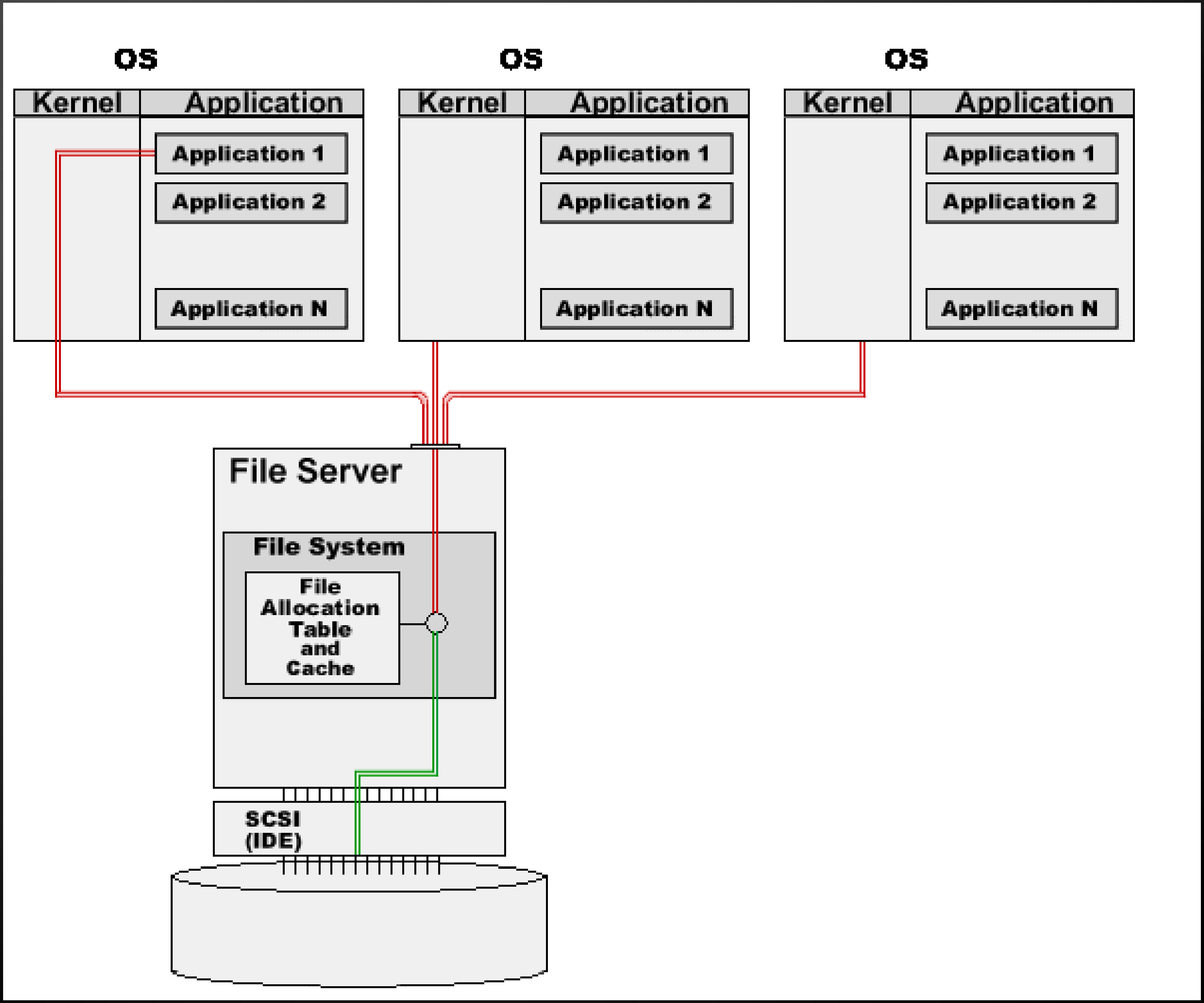
Network File Systems



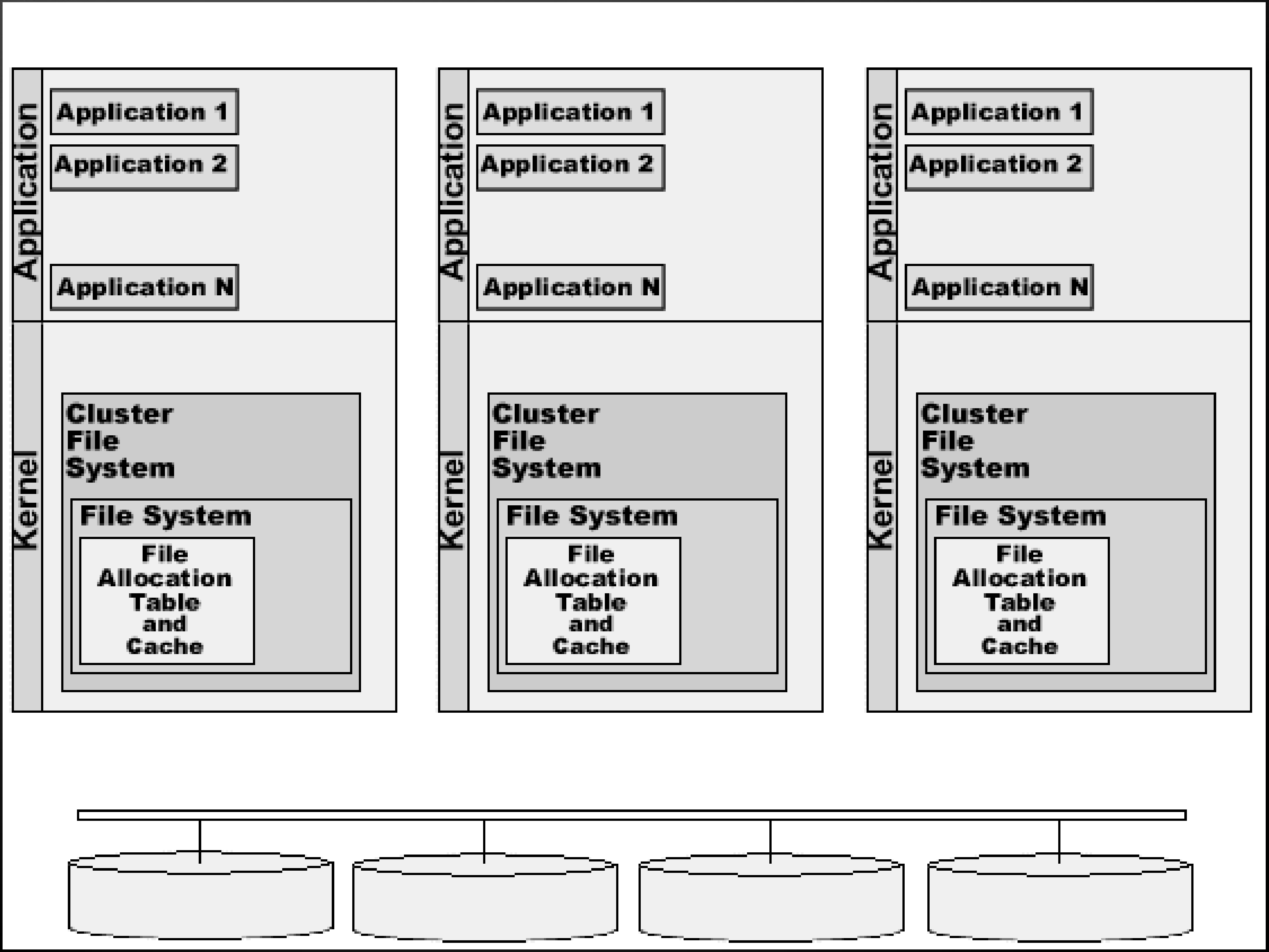
Network File Systems



Network File System



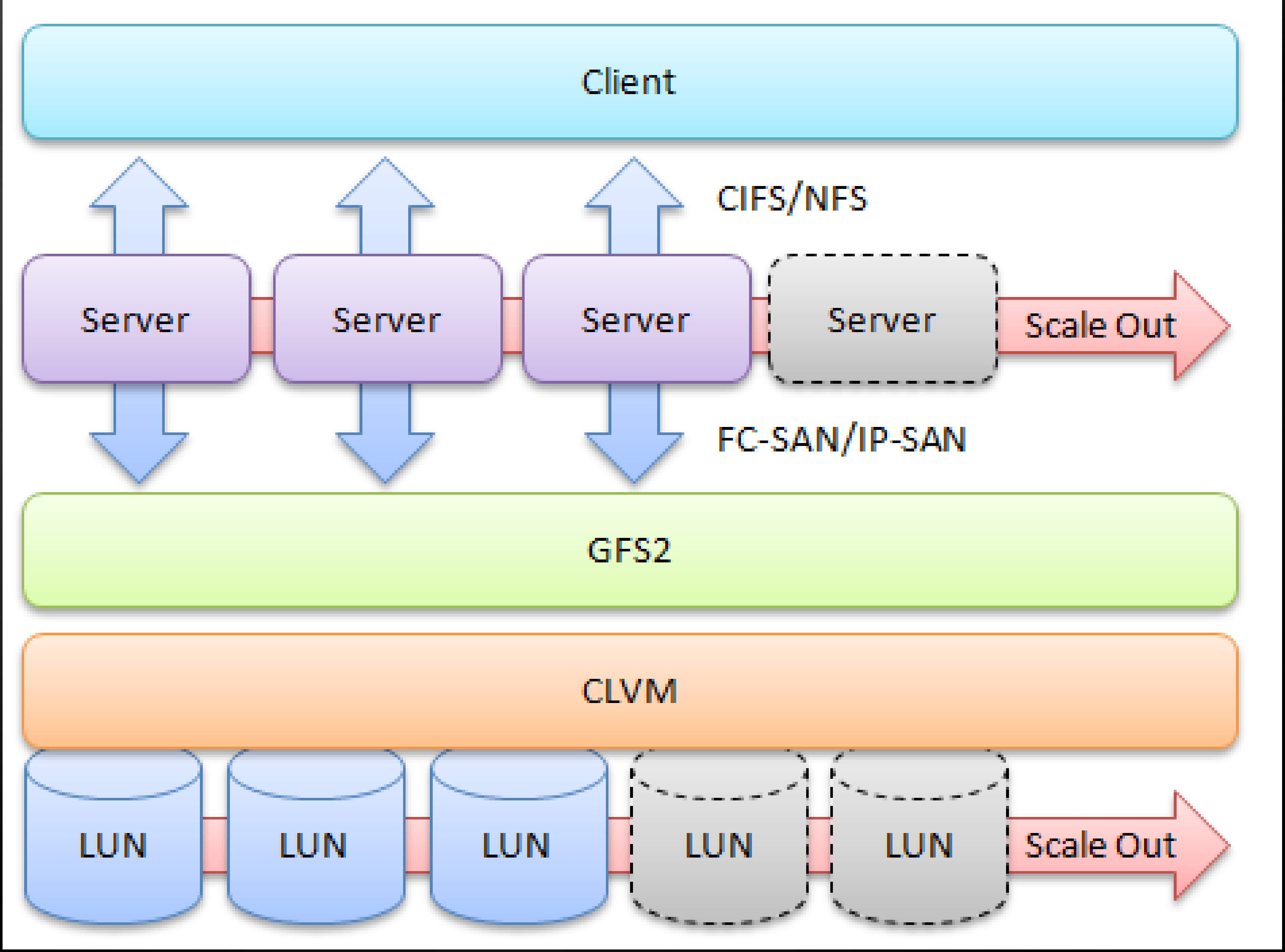
Cluster File System



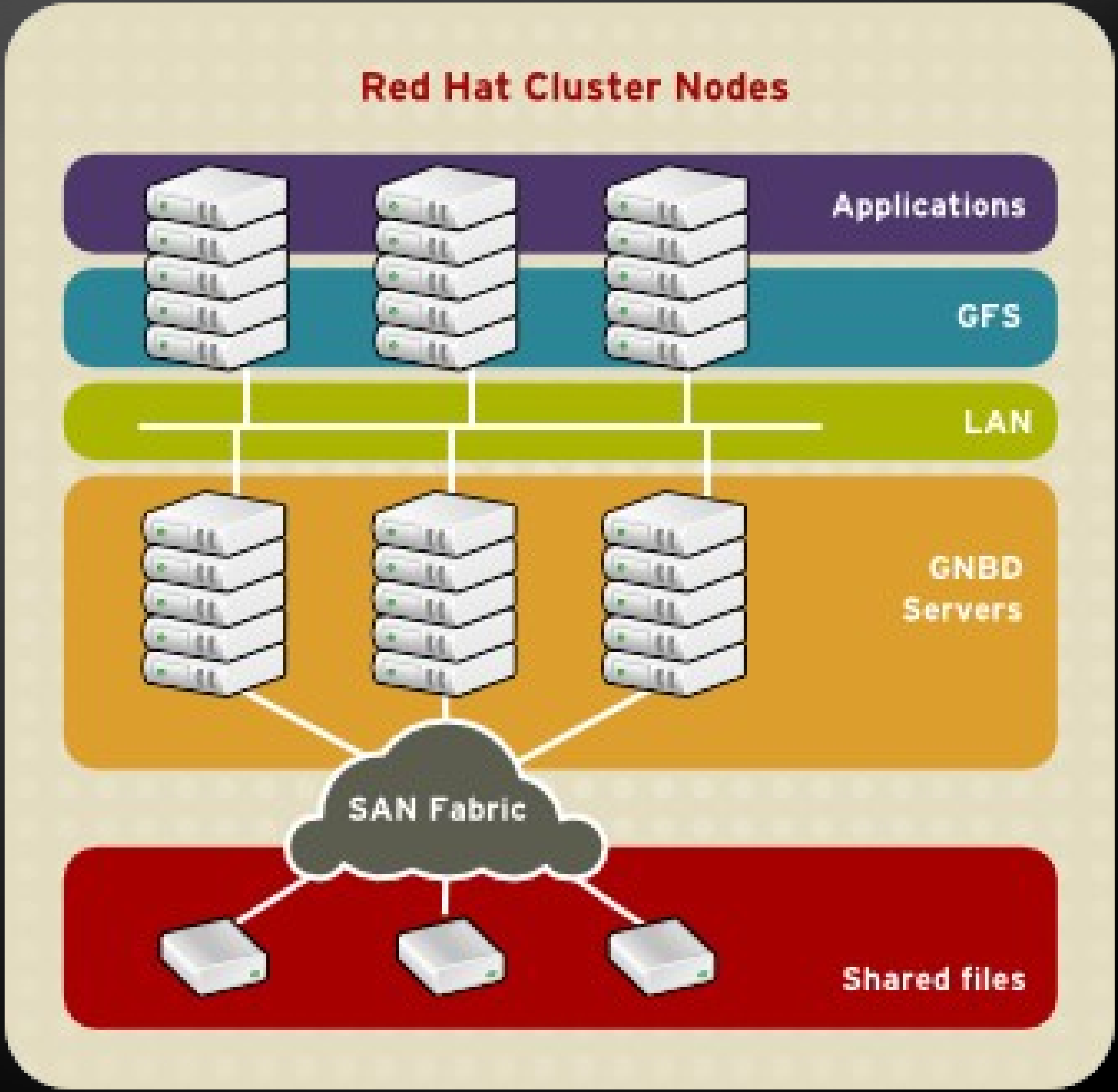
Cluster File Systems

- GFS, GFS2
- OCFS2

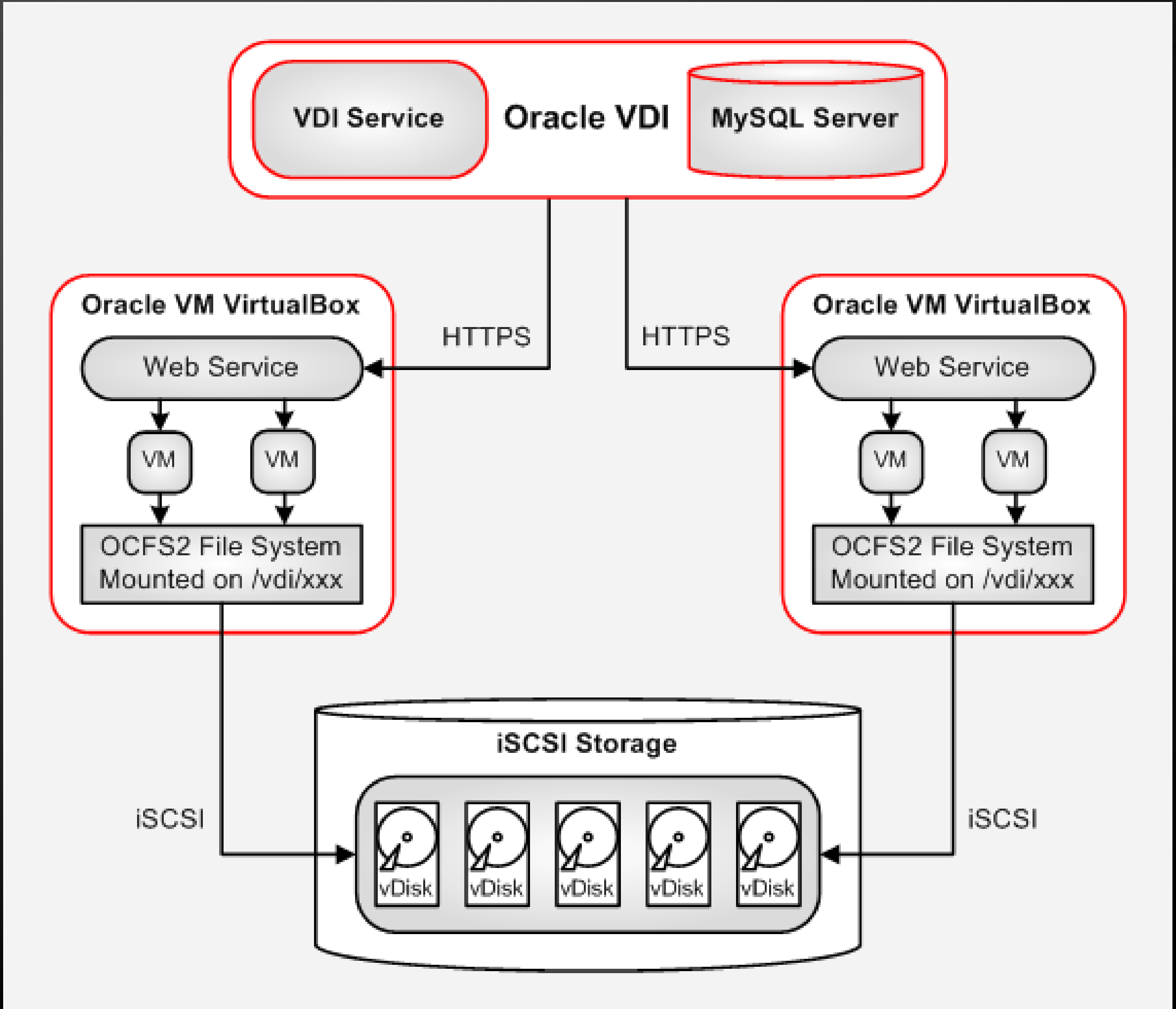
Cluster File Systems - GFS2



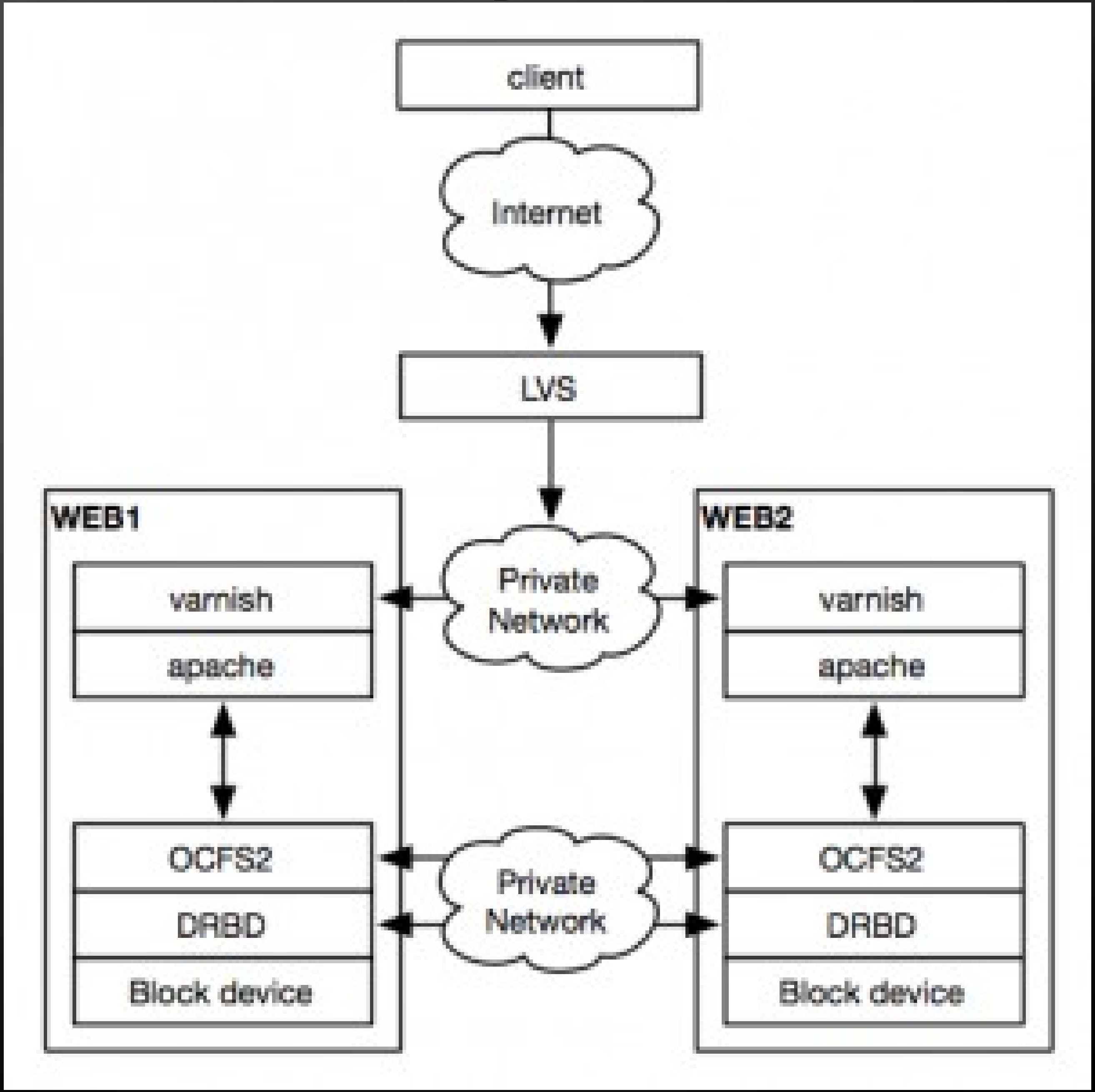
Cluster File Systems - GFS2



Cluster File Systems - OCFS2



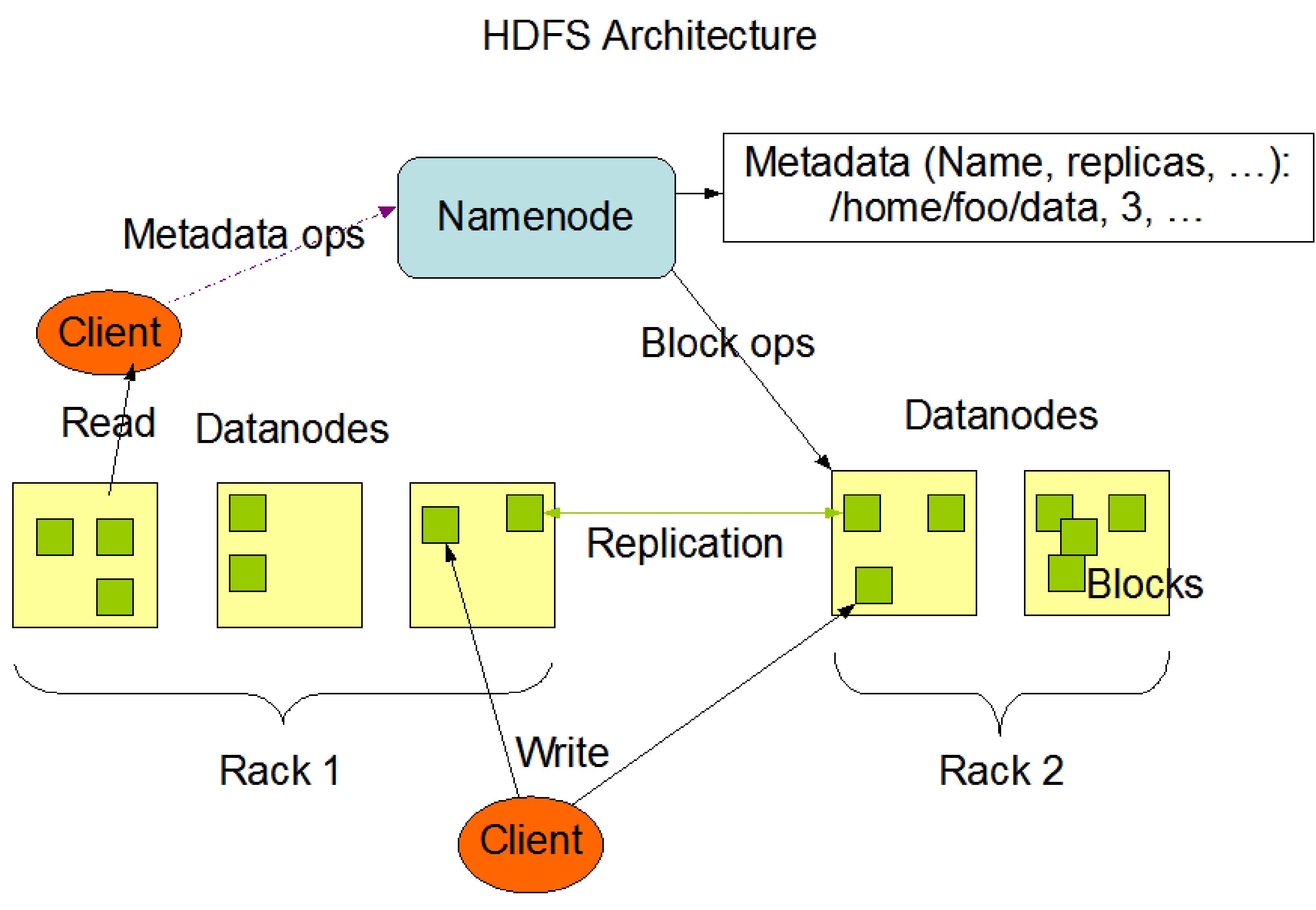
Cluster File Systems - OCFS2



Distributed File Systems

- Hadoop
- Lustre
- GlusterFS
- GFarm
- FhgFS
- PohmelFS
- Ceph
- PVFS2

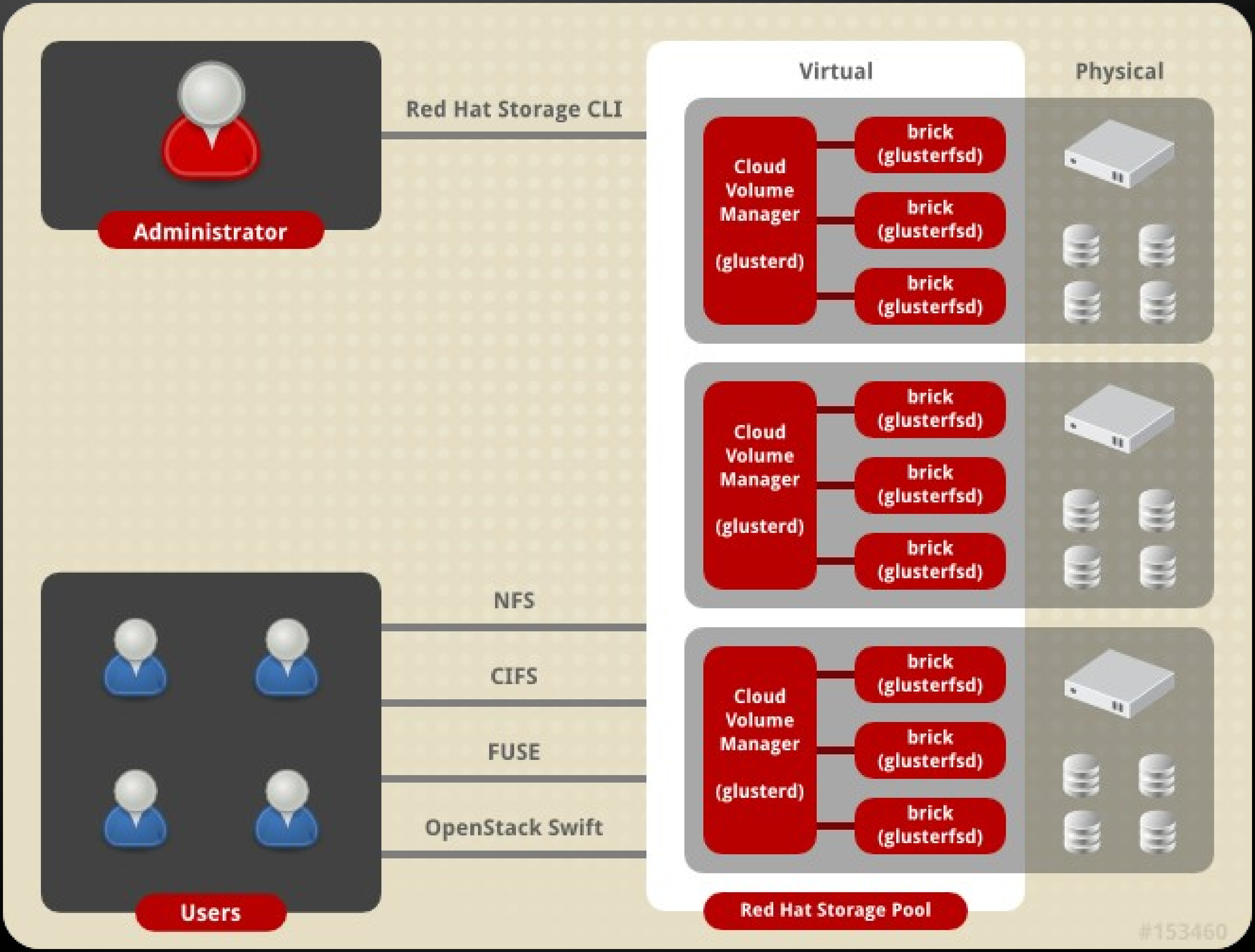
Hadoop


HDFS Architecture
Metadata ops
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client

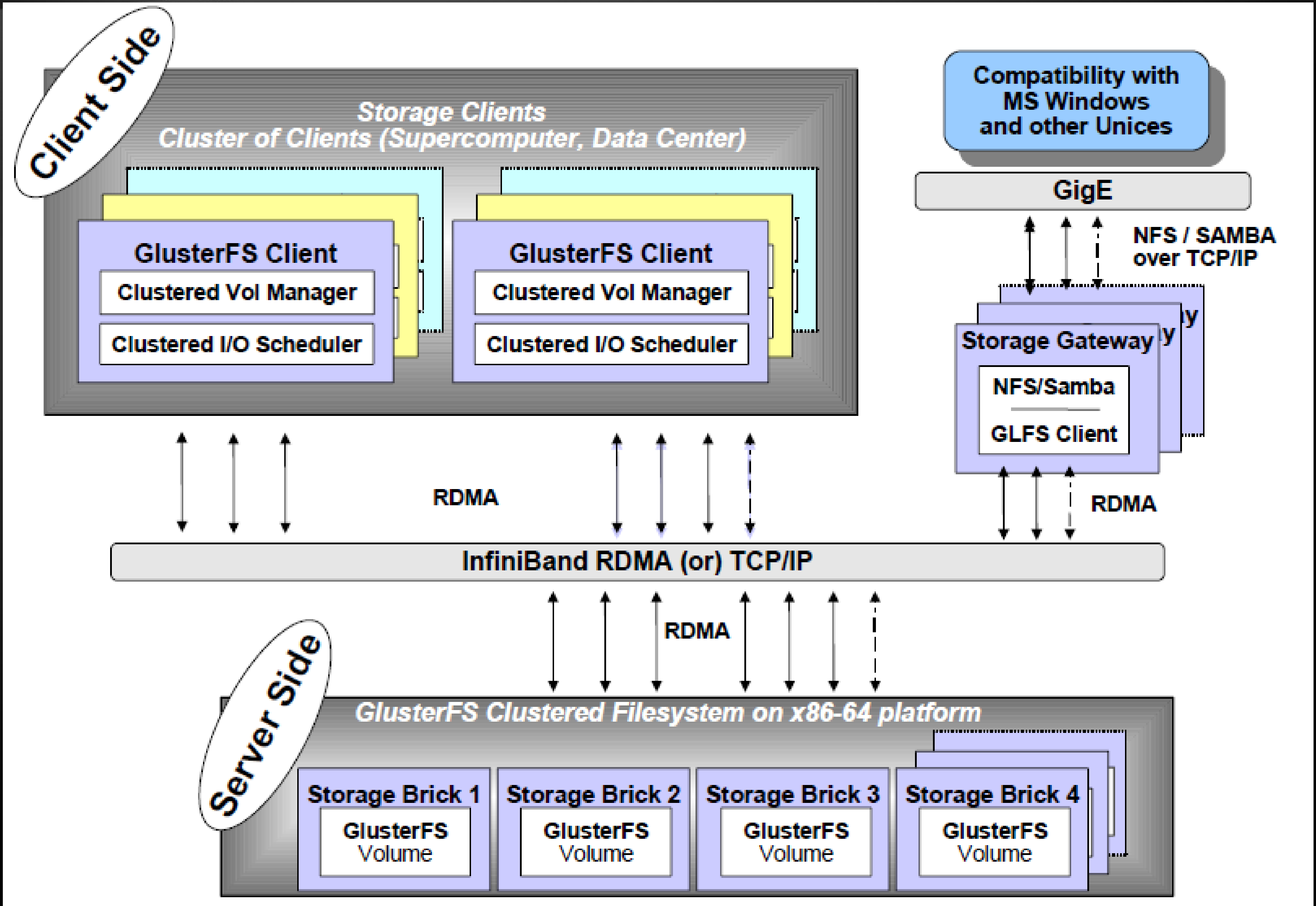
Hadoop

- **Large block FS – 64MB**
- **Write mostly FS**
- **Writes smaller then one block wait**
- **Adding/removing nodes requires restart of the cluster**

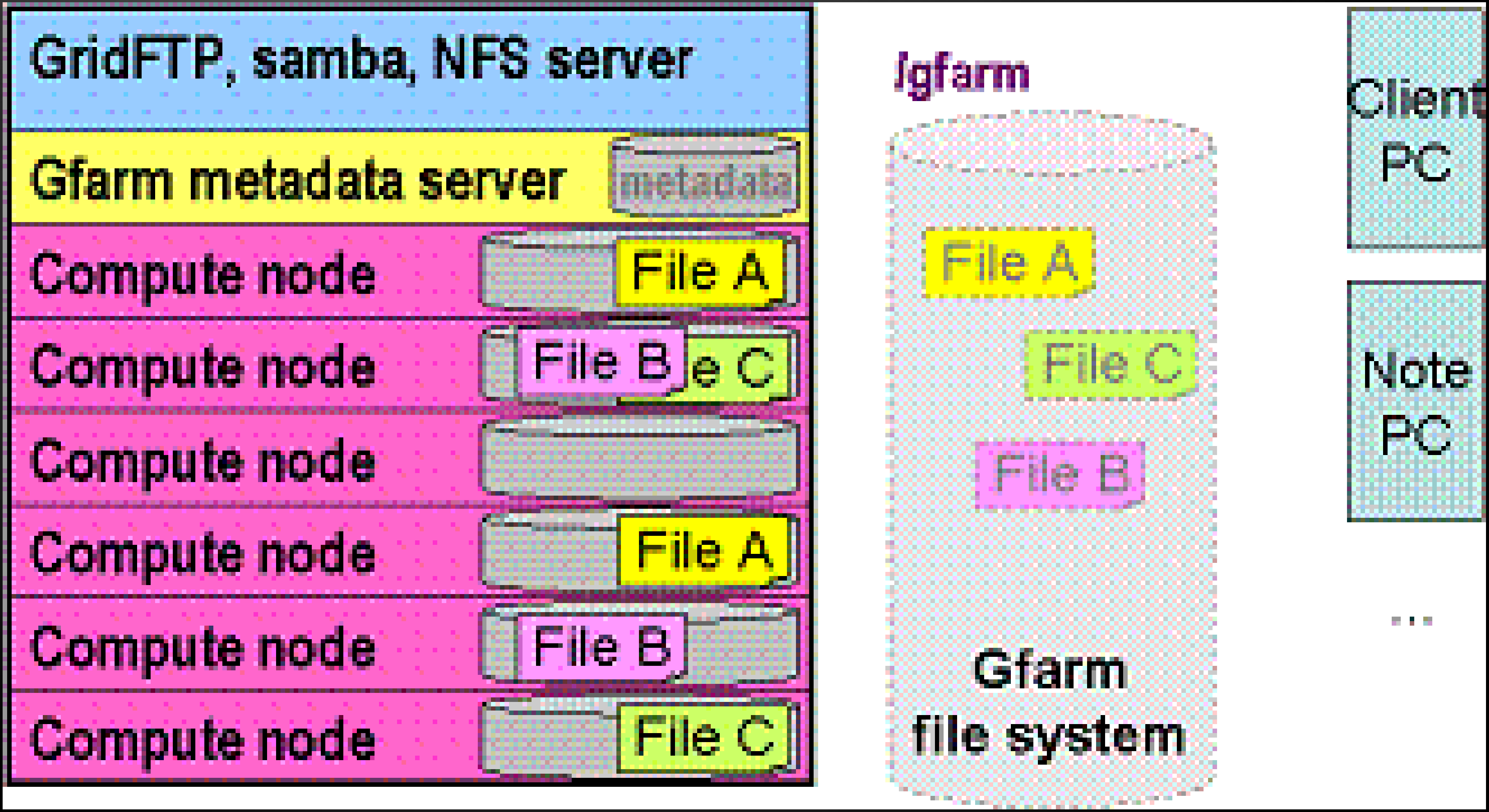
GlusterFS


Administrator
Red Hat Storage CLI
Virtual
Physical
Cloud Volume Manager (glusterd)
brick (glusterfsd)
brick (glusterfsd)
brick (glusterfsd)
Cloud Volume Manager (glusterd)
brick (glusterfsd)
brick (glusterfsd)
brick (glusterfsd)
Cloud Volume Manager (glusterd)
brick (glusterfsd)
brick (glusterfsd)
brick (glusterfsd)
Users
NFS
CIFS
FUSE
OpenStack Swift
Red Hat Storage Pool
#153460

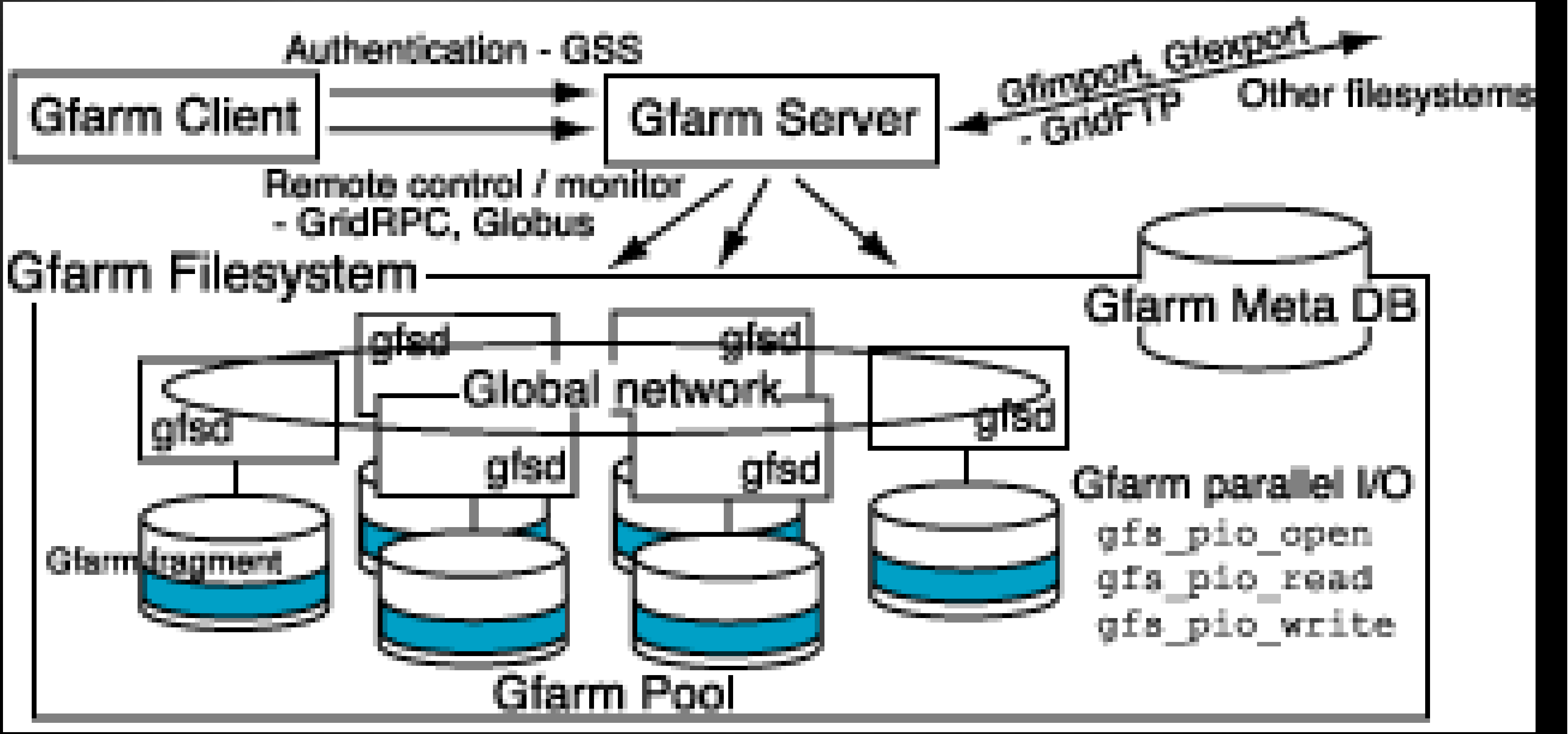
GlusterFS


Client Side
Storage Clients
Cluster of Clients (Supercomputer, Data Center)
GlusterFS Client
Clustered Vol Manager
Clustered I/O Scheduler
GlusterFS Client
Clustered Vol Manager
Clustered I/O Scheduler
Compatibility with MS Windows and other Unices
GigE
NFS / SAMBA over TCP/IP
Storage Gateway
NFS/Samba
GLFS Client
RDMA
RDMA
InfiniBand RDMA (or) TCP/IP
RDMA
Server Side
GlusterFS Clustered Filesystem on x86-64 platform
Storage Brick 1
GlusterFS Volume
Storage Brick 2
GlusterFS Volume
Storage Brick 3
GlusterFS Volume
Storage Brick 4
GlusterFS Volume

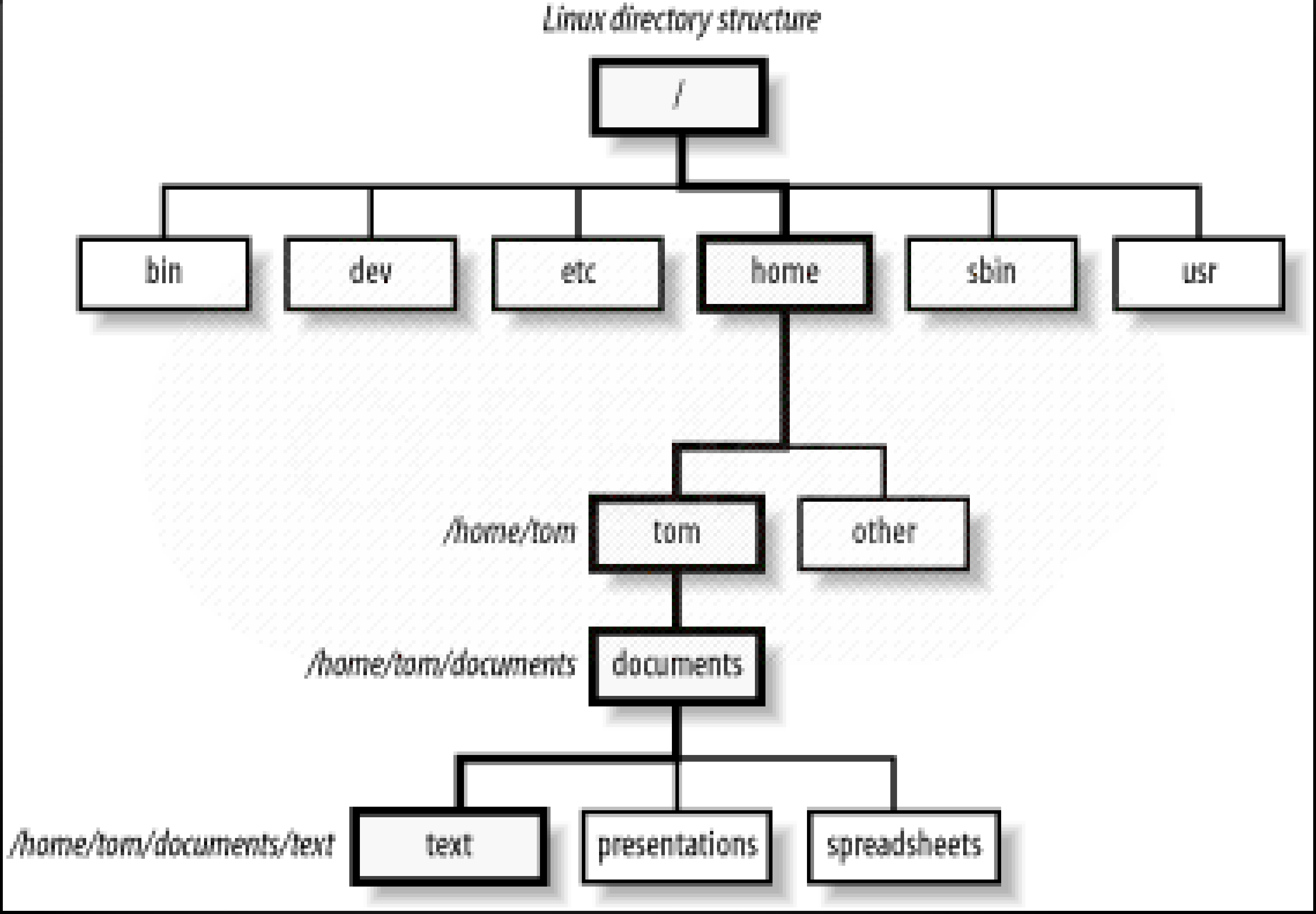
GFarm


GridFTP, samba, NFS server
Gfarm metadata server
metadata
Compute node
File A
Compute node
File B
e C
Compute node
Compute node
File A
Compute node
File B
Compute node
File C
/gfarm
File A
File C
File B
Gfarm
file system
Client
PC
Note
PC
...

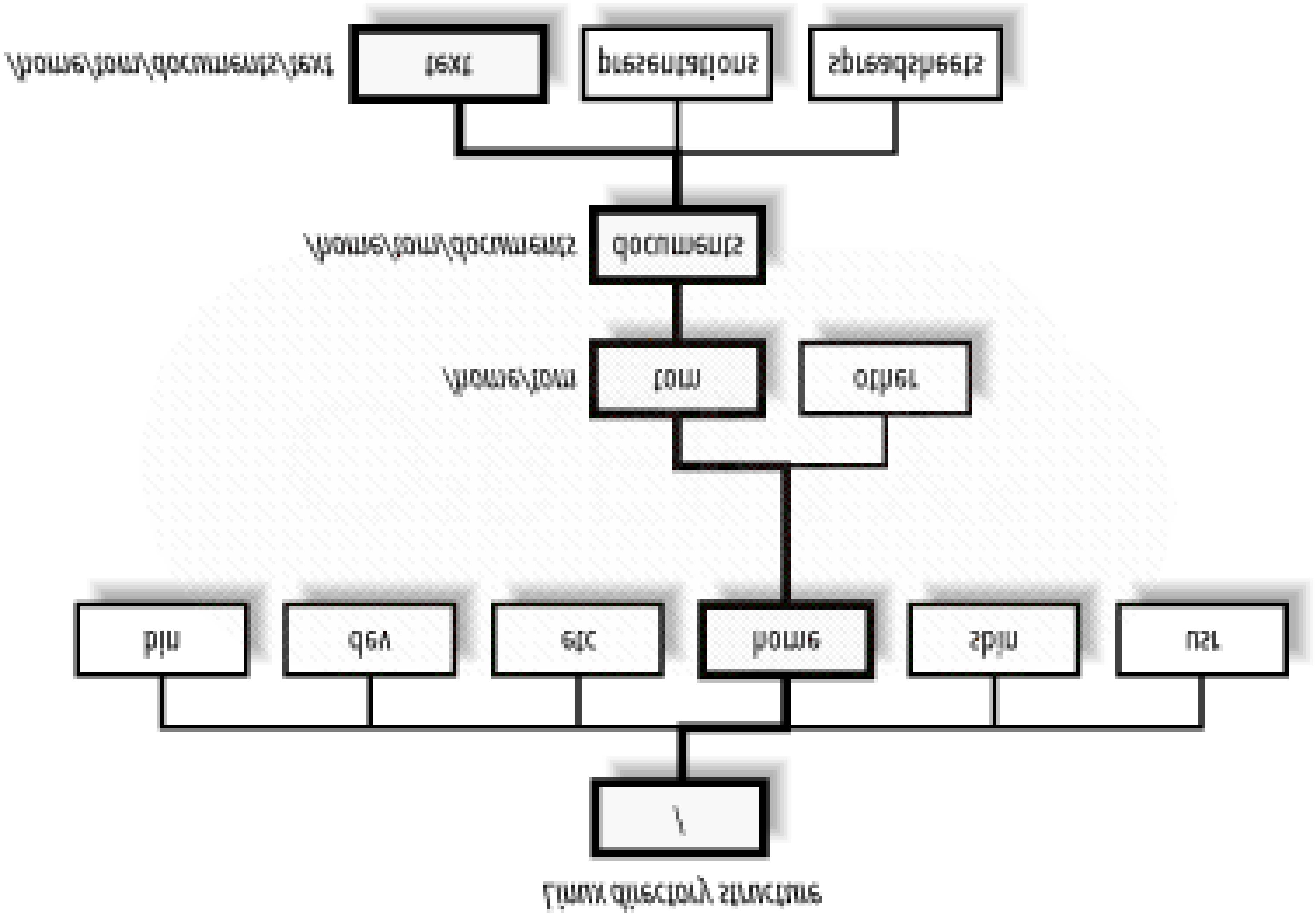
GFarm

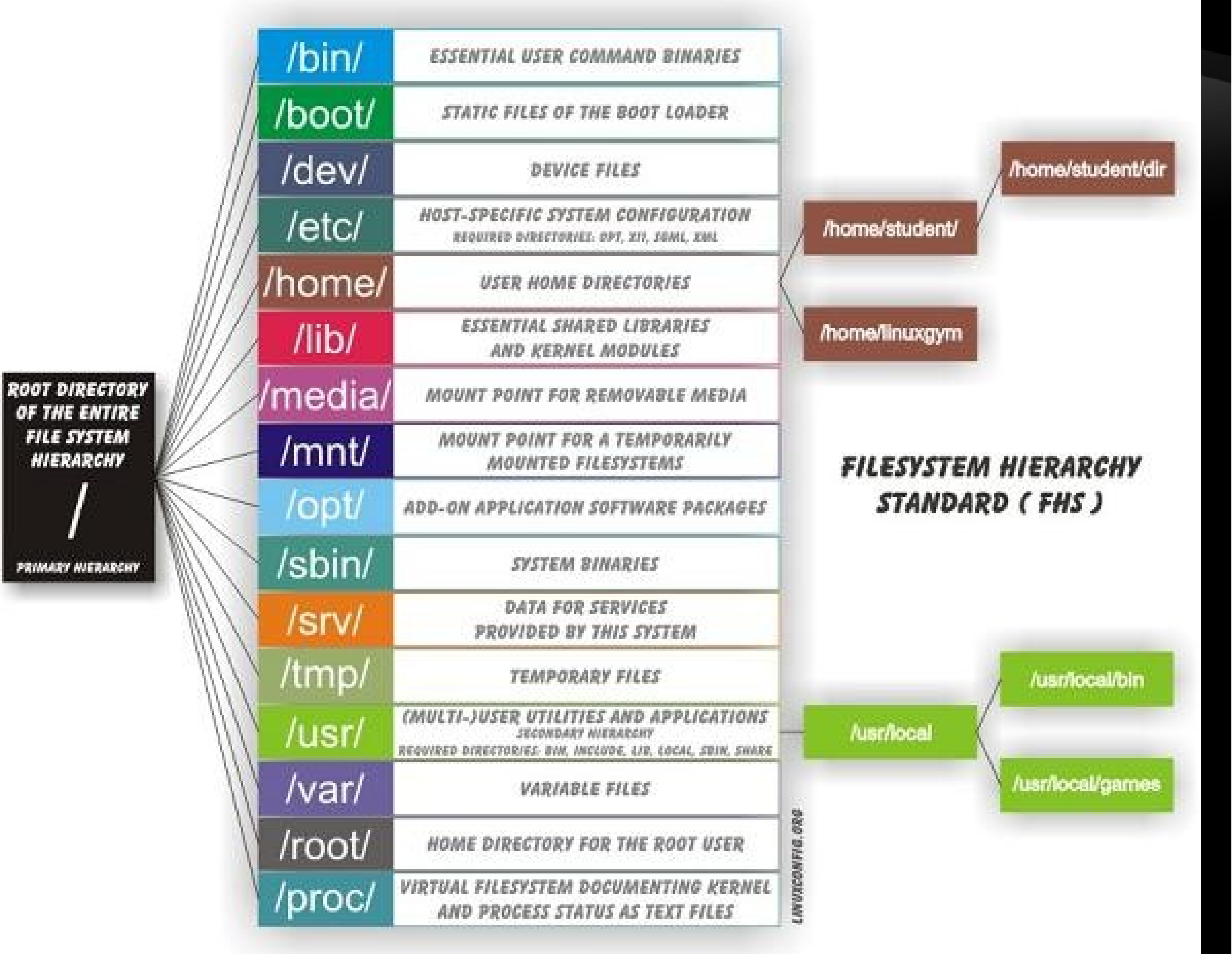

Authentication - GSS
Gfarm Client
Gfarm Server
Gfimport, Gfexport
- GridFTP
Other filesystems
Remote control / monitor
- GridRPC, Globus
Gfarm Filesystem
Gfarm Meta DB
gfsd
gfsd
Global network
gfsd
gfsd
gfsd
gfsd
Gfarm parallel I/O
gfs_pio_open
gfs_pio_read
gfs_pio_write
Gfarm fragment
Gfarm Pool

Linux Directory Structure



Linux Directory Structure





ROOT DIRECTORY OF THE ENTIRE FILE SYSTEM HIERARCHY

/

PRIMARY HIERARCHY

/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

/home/student/

/home/student/dir

/home/linuxgym

/usr/local

/usr/local/bin

/usr/local/games

FILESYSTEM HIERARCHY STANDARD (FHS)

LINUXCONFIG.ORG

Links

Wikipedia - Comparison of file systems

Ext2 and OCFS2 on-disk layout

Ext2 on-disk layout

XFS on-disk structure

ReiserFS on-disk structure

RFSTool for Windows

XFS Scalability

BtrFS on-disk structure

NILFS2 the new kid on the block

Usenix paper on Log-Structured File Systems

Linux File Systems

Questions?

http://softuni.bg/trainings/fasttracks/details/1060

THAT TIME AGAIN?

Yes, then we'll have a beer