

# **HPC Cluster Parallel File Systems & Storage Architecture**

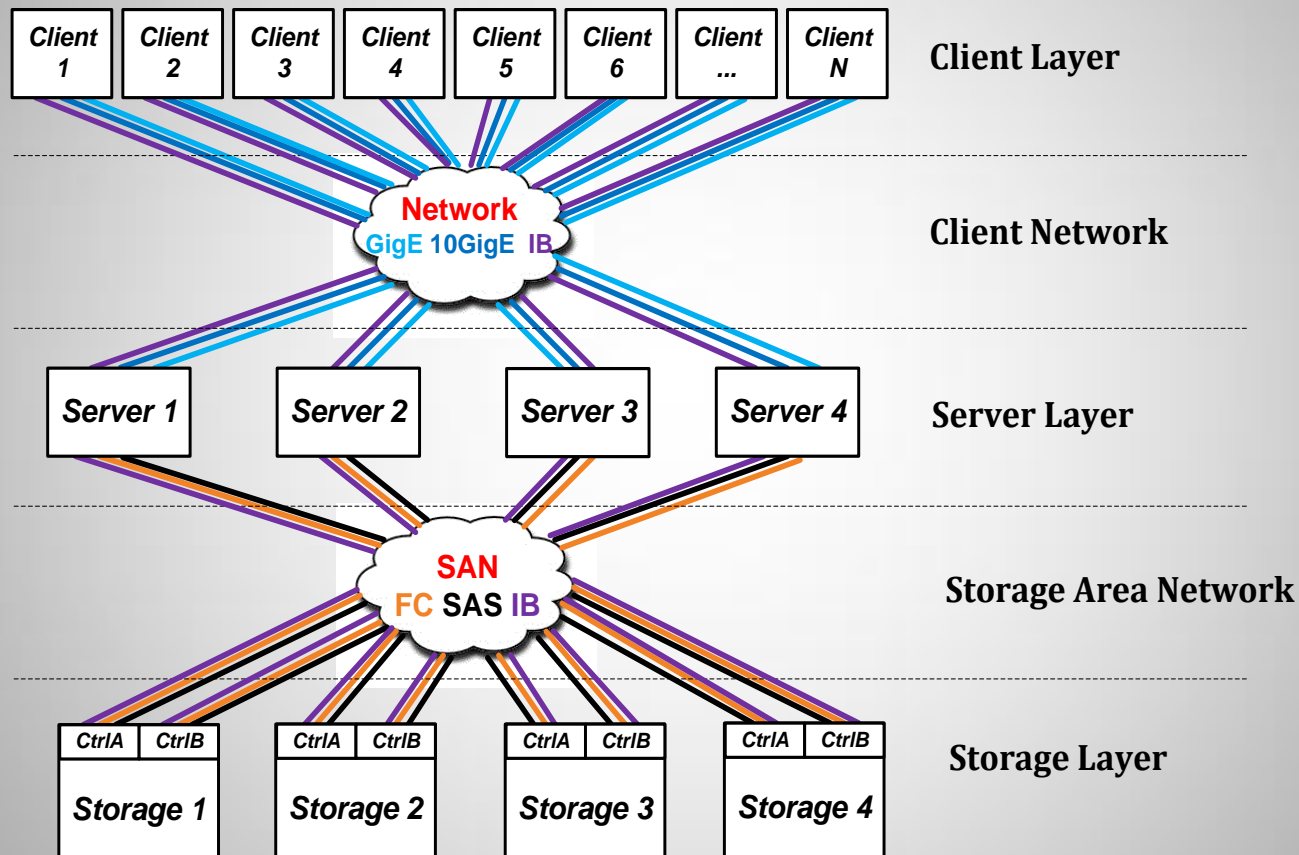
## **Lustre & GPFS**

**Ercan Kamber, PhD**

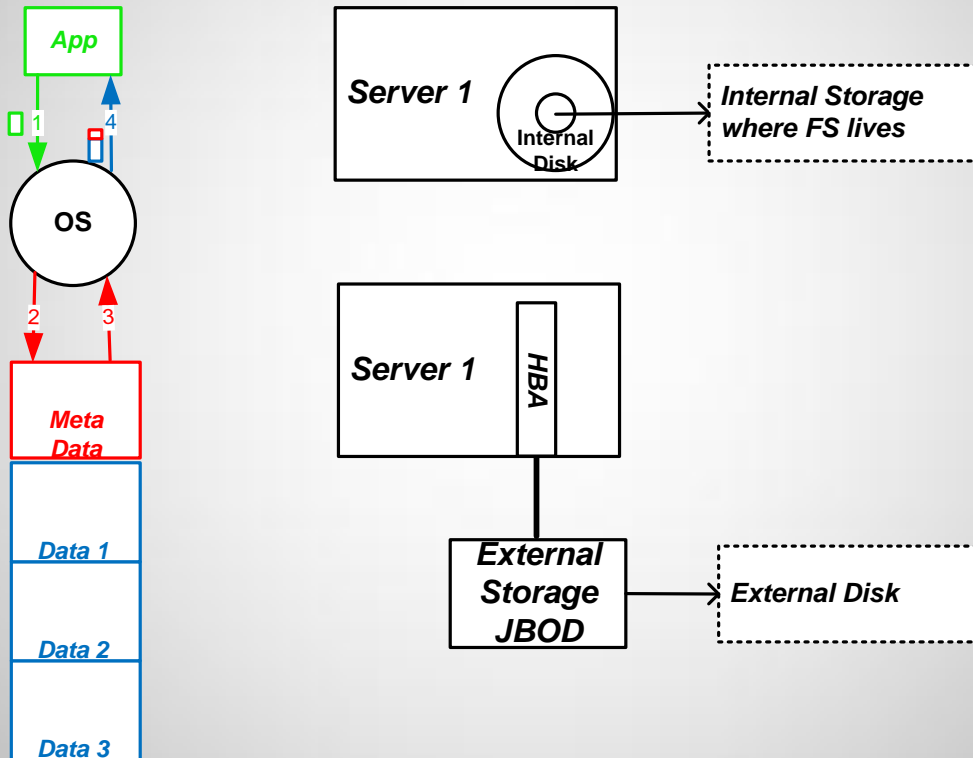
**VP of Engineering**

**RAID Inc.**

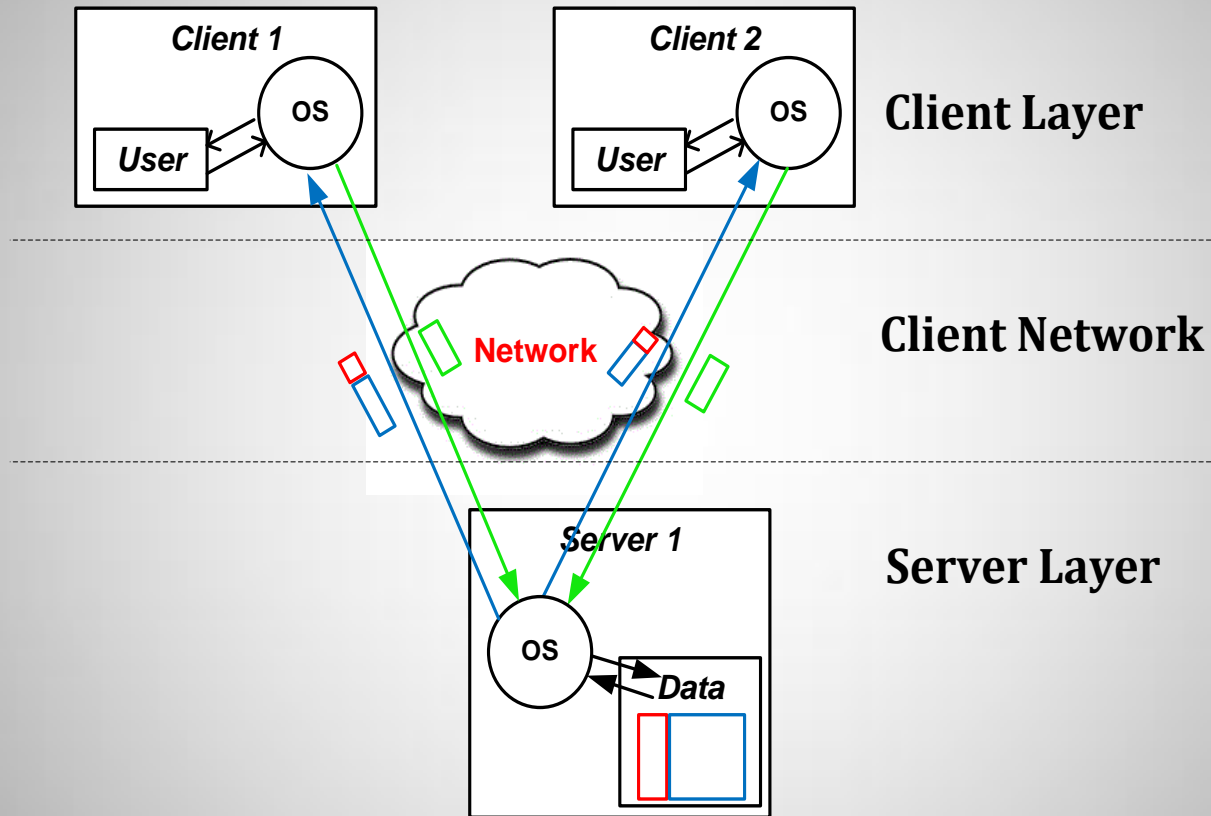
# A General View of Cluster File System



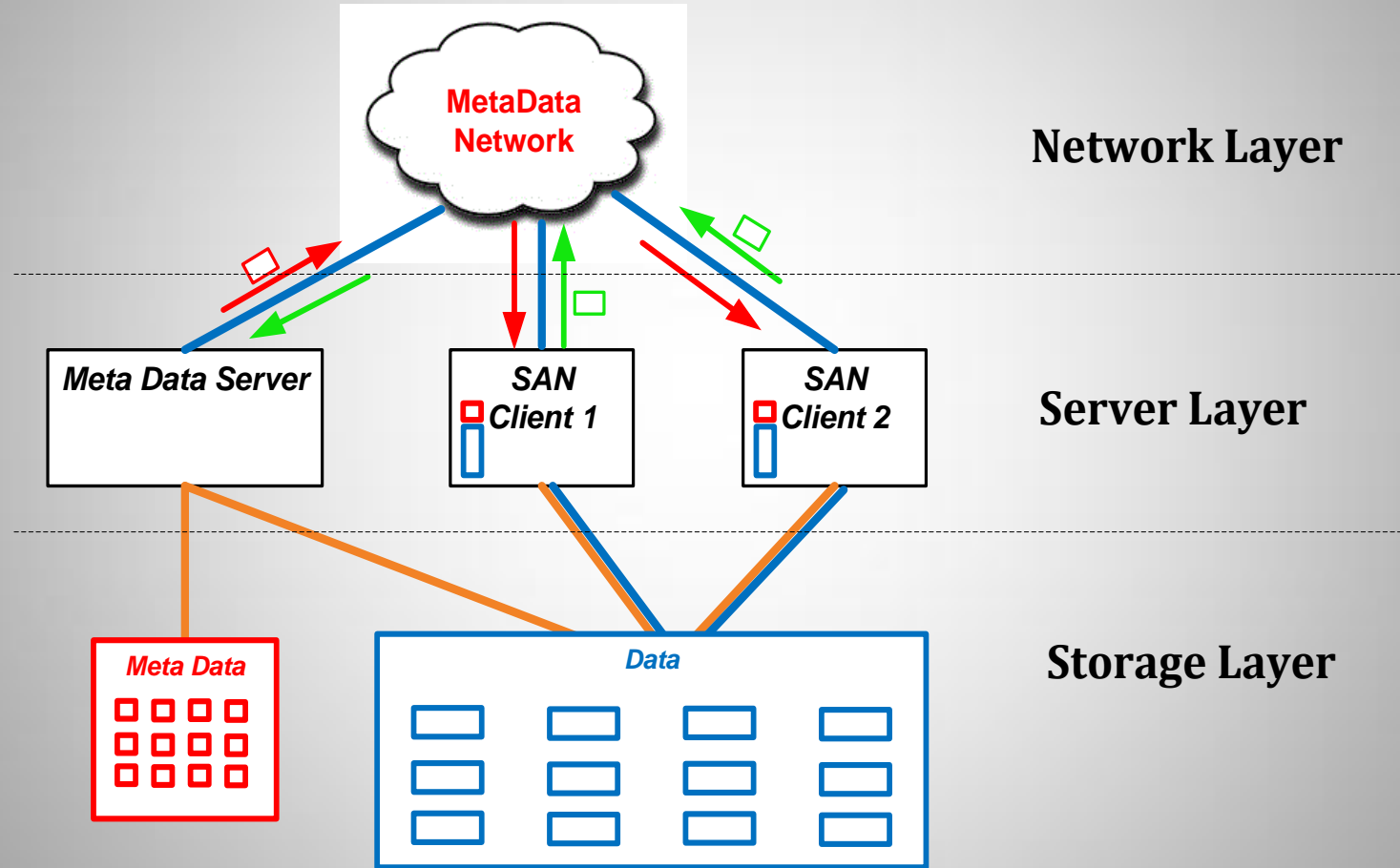
# Local File System



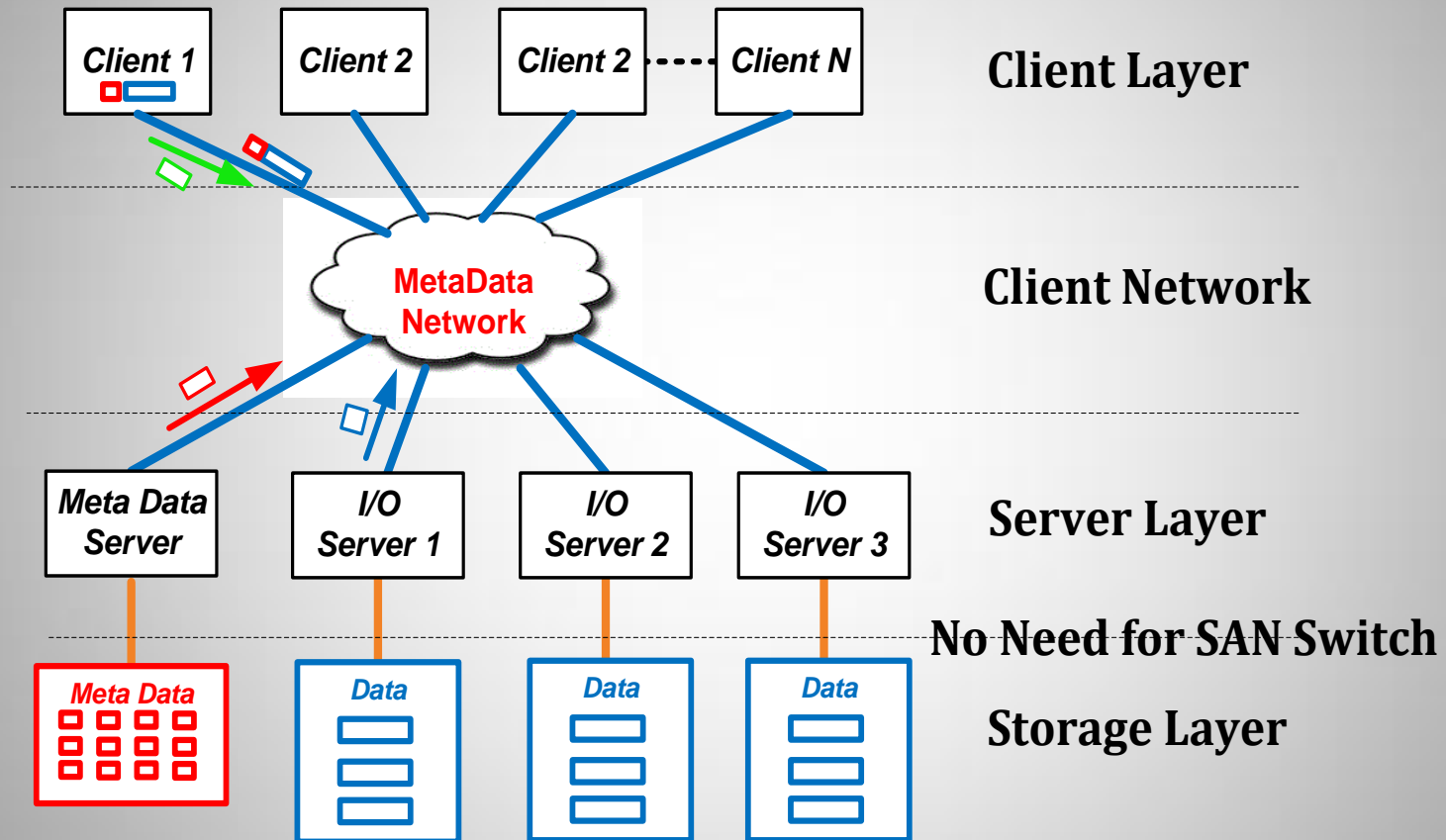
# Client-Server and FS : NFS



# Shared Disk File System



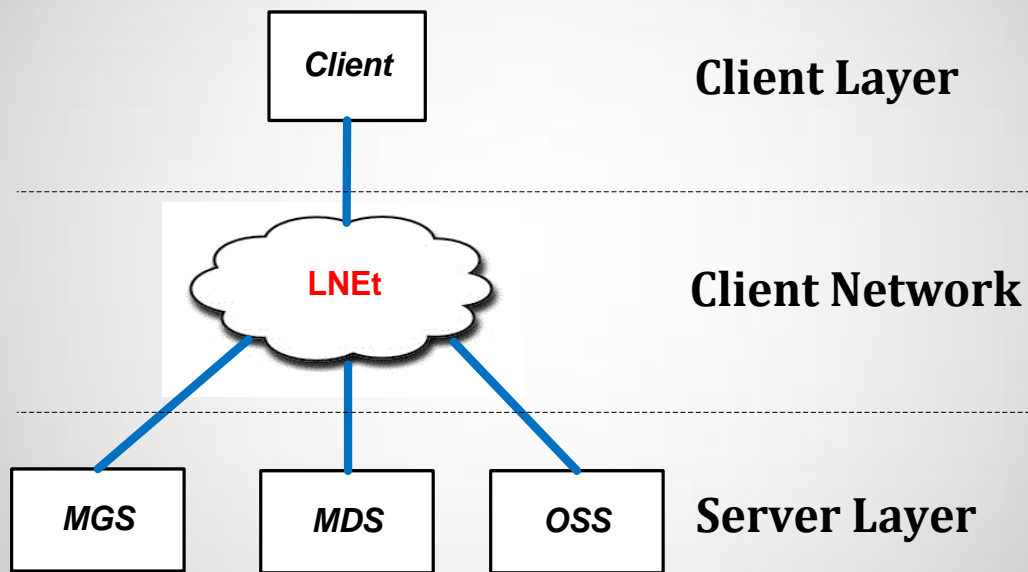
# Clustered File System : Lustre



# Lustre History

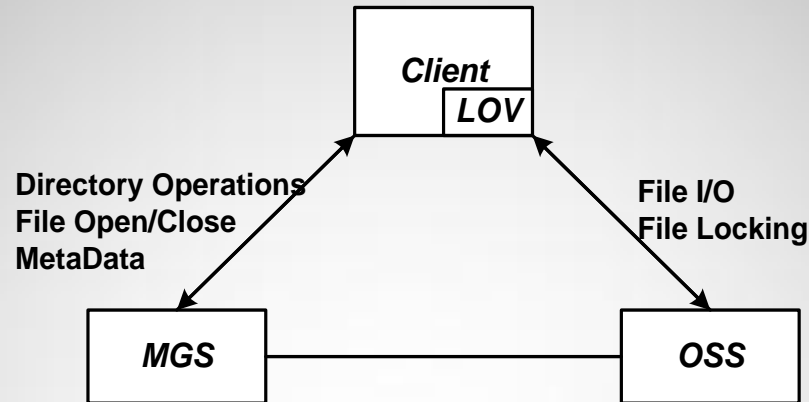
- Designed for HPC Community
- Used by everyone with large-scale needs
  - US National Labs
  - 60 out of Top 100 supercomputers
  - Also seismographic and rendering/animation
- Early use – Beowulf clusters, MPI jobs
- Very large shared file access
  - Processing science data
  - Rendering animation

# Lustre Components



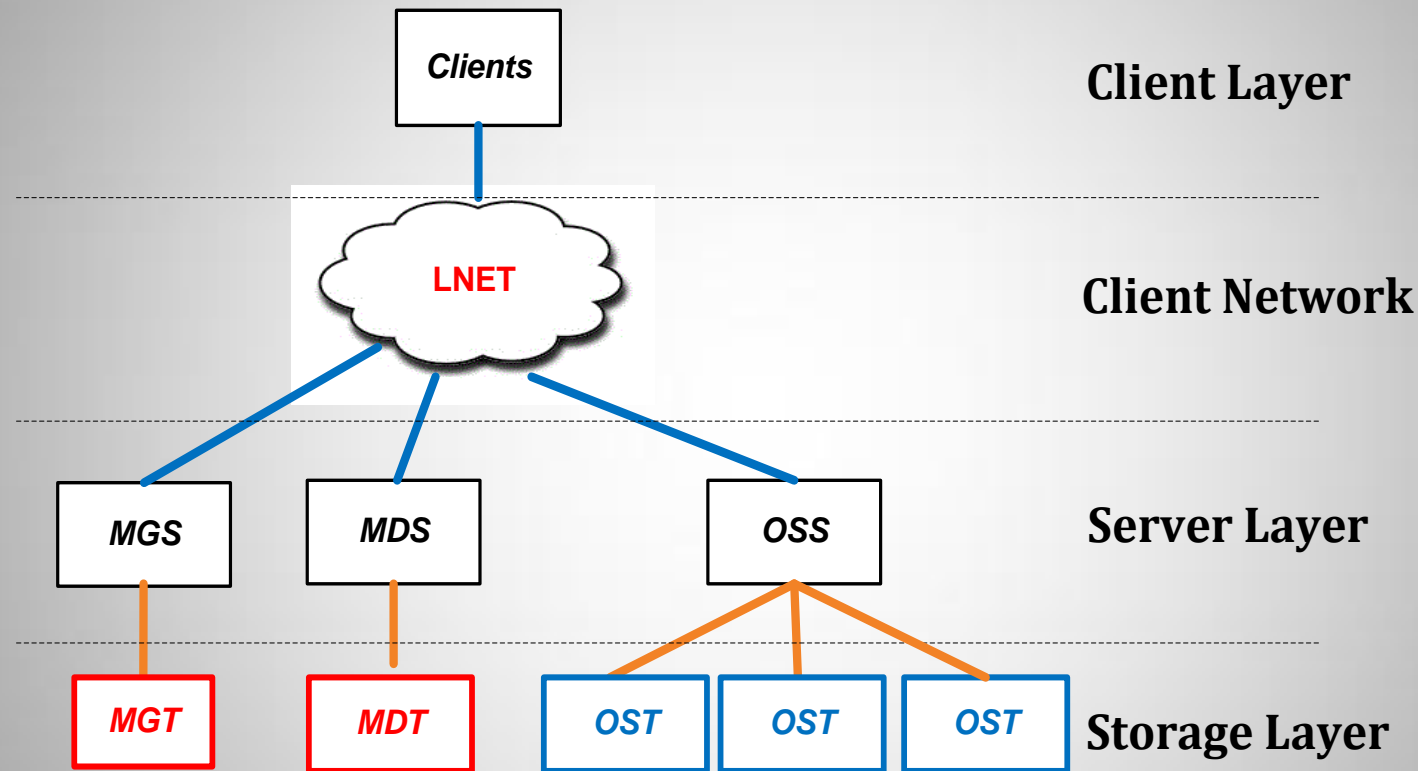


# Lustre FS structure



- Client accesses filesystem via Linux VFS, standard IO calls
- VFS talks to Lustre Logical Object Volume (LOV)
- LOV sends request to the proper server
  - Metadata actions to MDS
  - Block IO to OSS
- All transactions handed to LNET for transmission

# Lustre Storage Components



# Lustre Storage Considerations

- No data Protection
- No replication
- Can protect nodes with shared storage
- Consider LVM for additional backup of critical data
- Storage health is entirely on storage unit

# Lustre install in a nutshell

- MDS and MGS
- `mkfs.lustre --fsname test --mgs --mdt /dev/sda`
- OSS
- `mkfs.lustre --ost --fsname test --mgsnode = mgs_nid /dev/sda`
- Clients
- `Mount -t lustre mgs@tcp0:/test /mnt/test`

# Lustre Features

## Posix compliant

- POSIX specification is single node, no clusters
- Atomic transactions
- Lustre passes full POSIX test suite when run from a single client

## Security

- Today: roughly equal to NFS
- Privileged port used for network
- Tomorrow: Kerberos

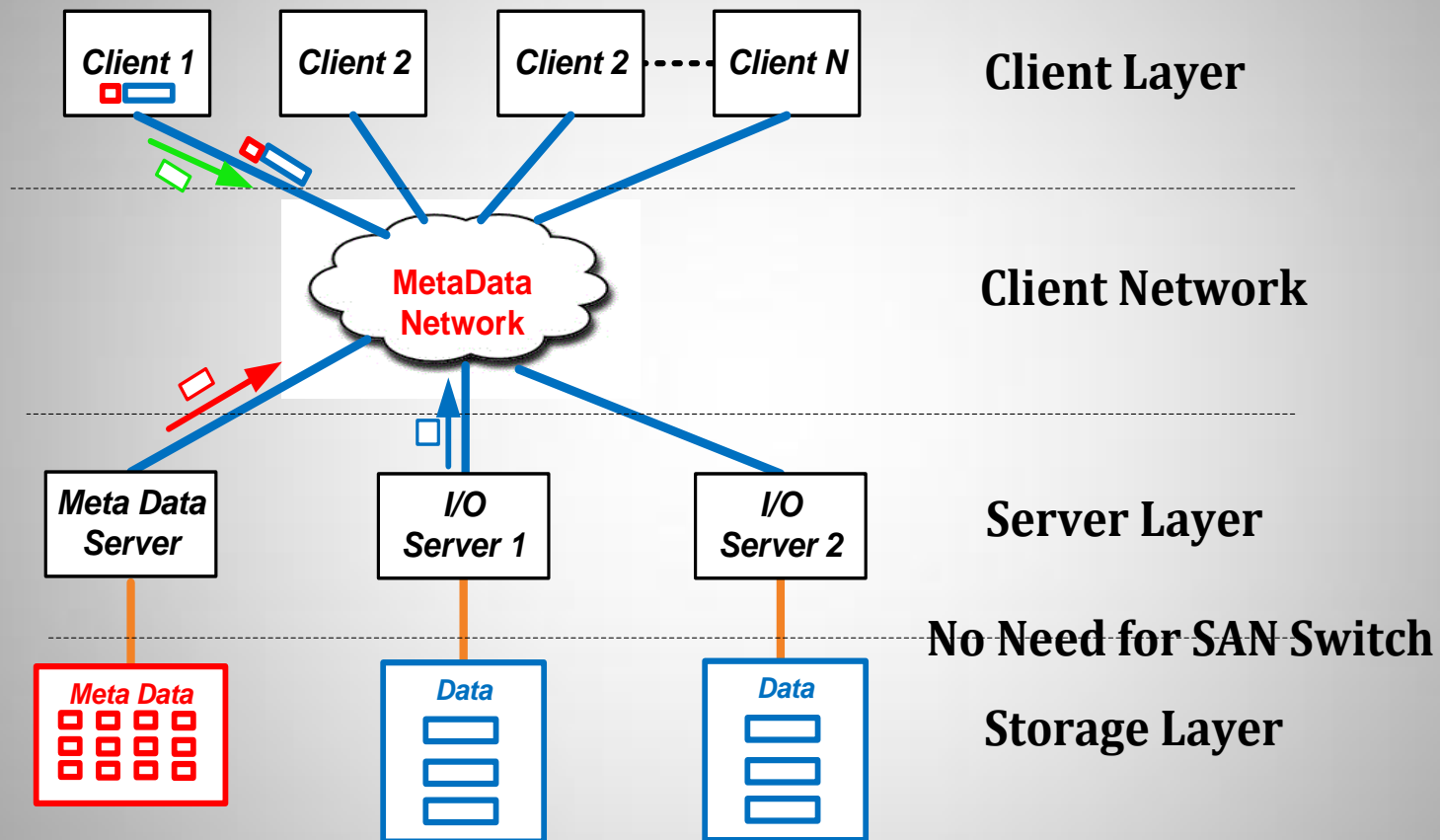
# Lustre Utilities

- Lfs – a long list of functions
  - Lfs find – faster find for lustre, can search by OST
  - Lfs df – free space
  - Lfs get|setstripe – striping policy
  - Lfs check servers - health
- Lctl – configuration/health
  - Lctl dl – device list
  - Lctl network – start|stop|test network
  - Lctl list\_nids – list LNET NIDs

# Lustre Advanced Topics

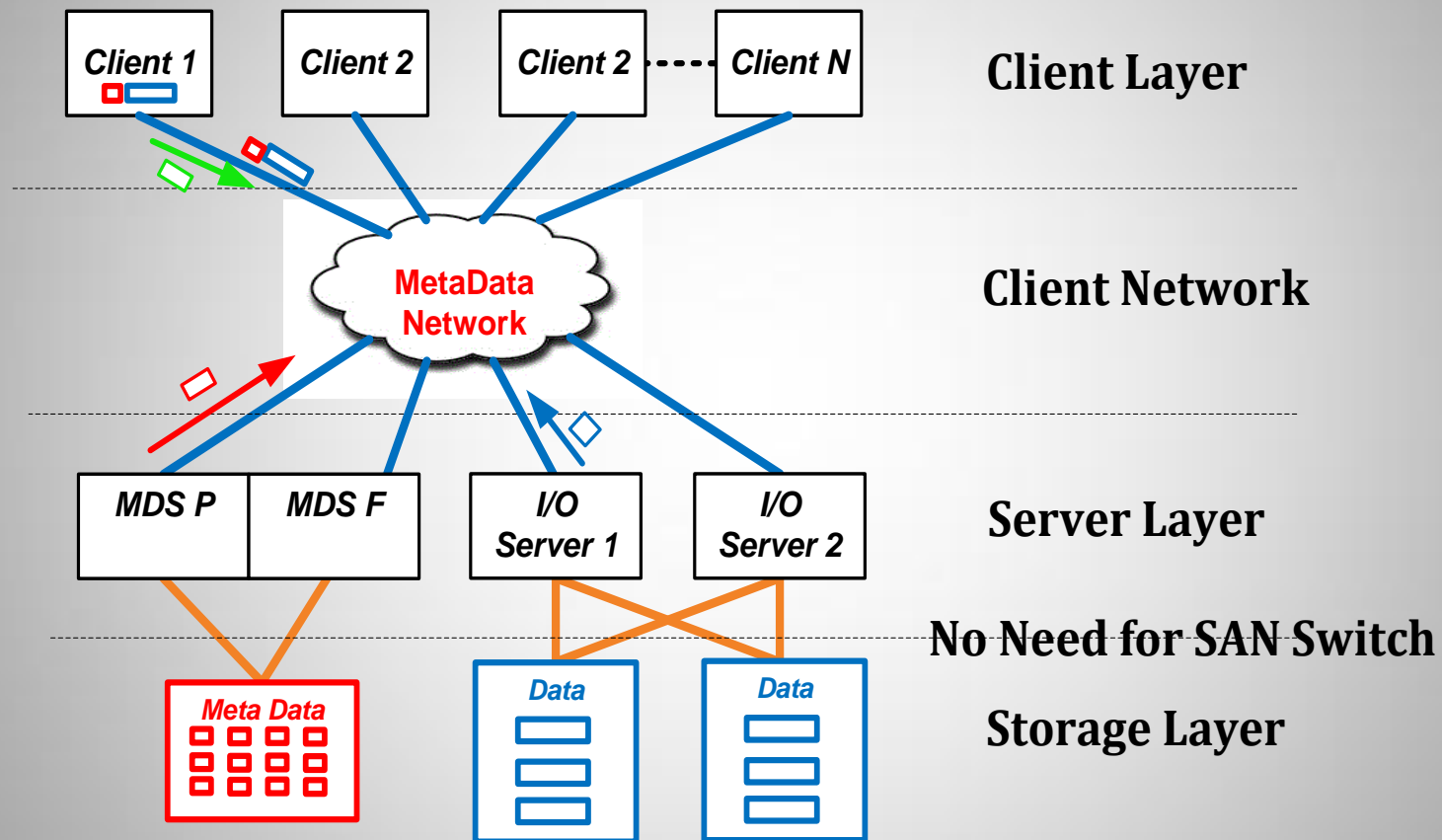
- Network routing
- Pools
- Failover

# CFS: Simple Lustre

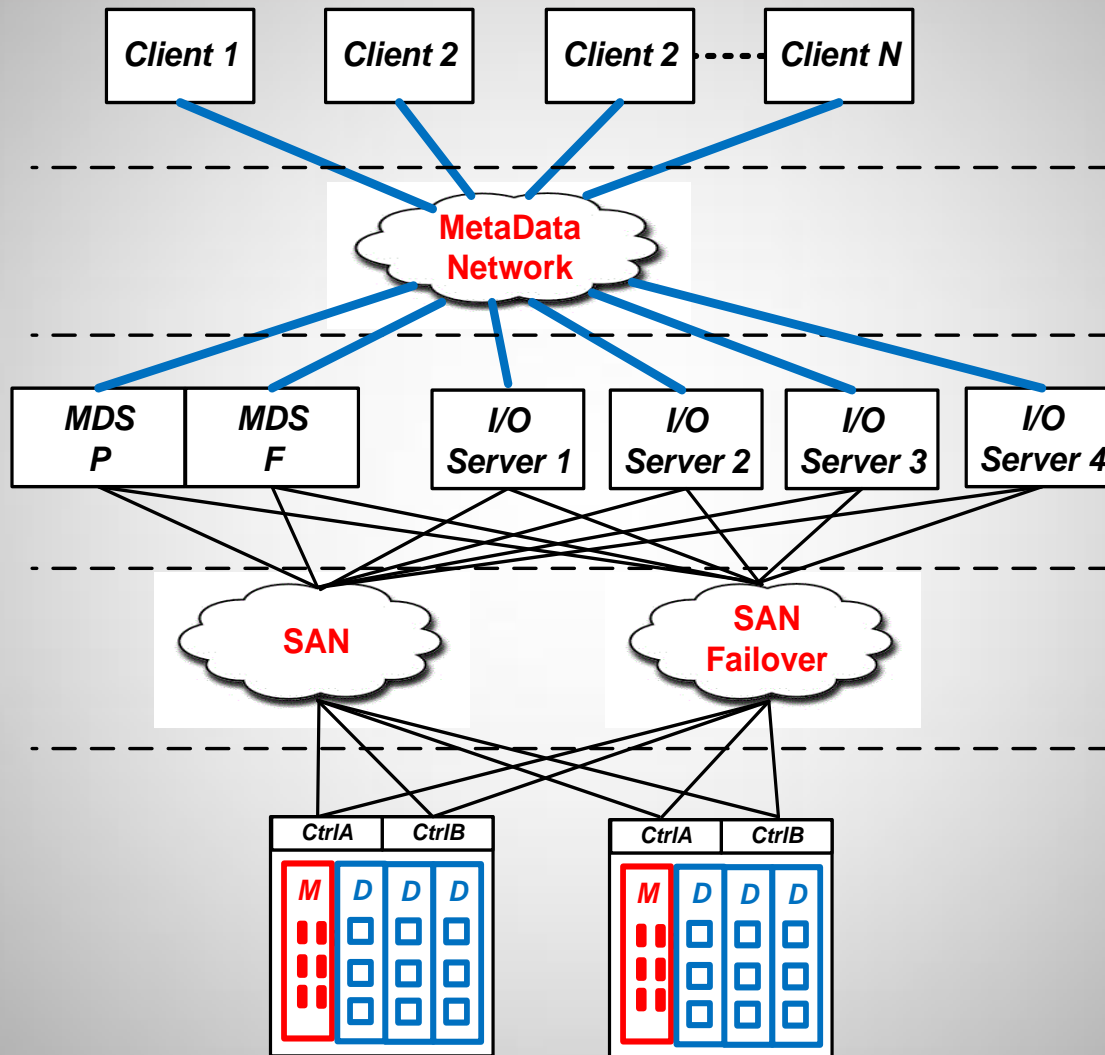




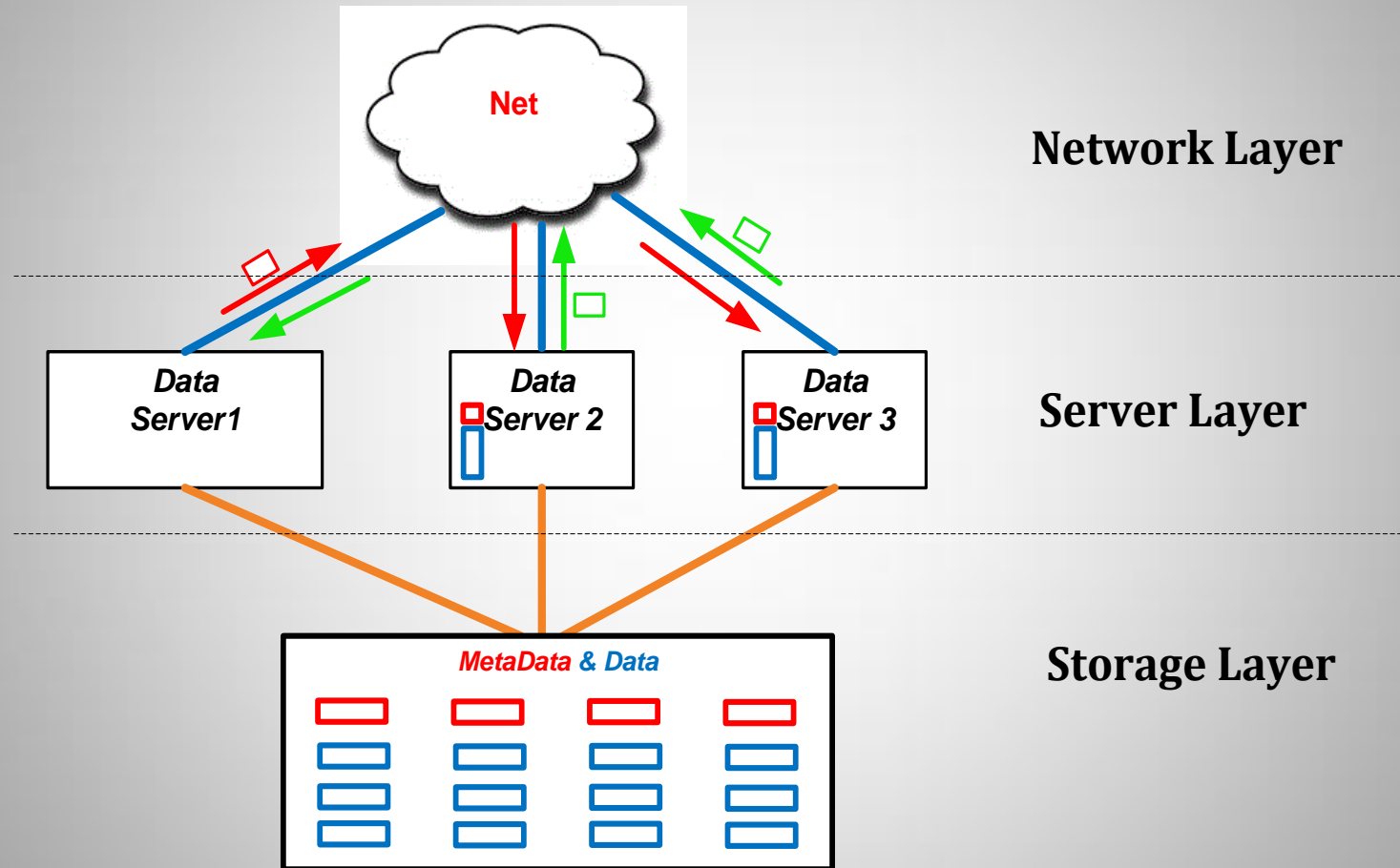
# CFS: Simple HA Lustre



# Clustered File System : More HA Lustre



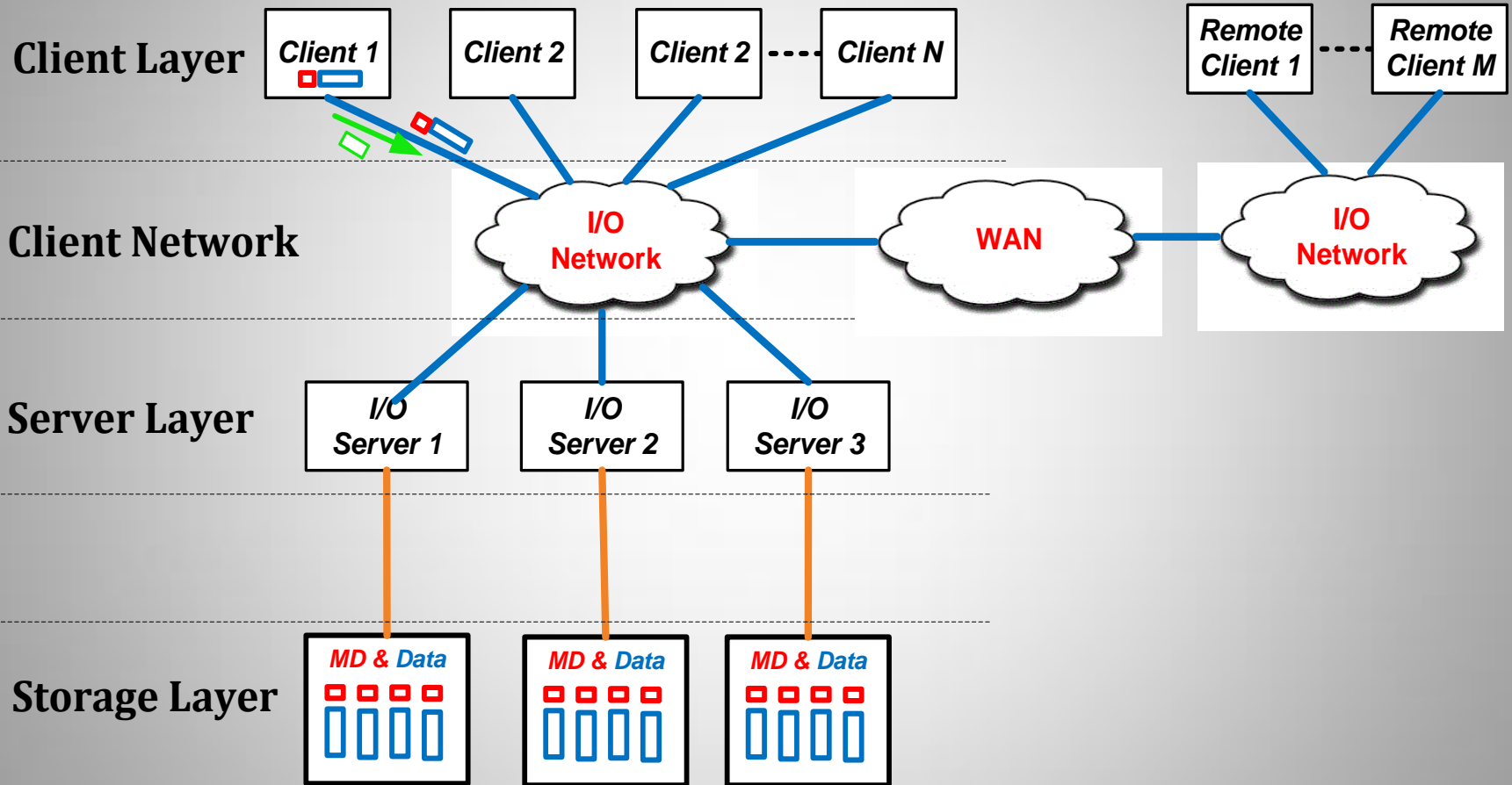
# Shared Disk File System: Where GPFS is born



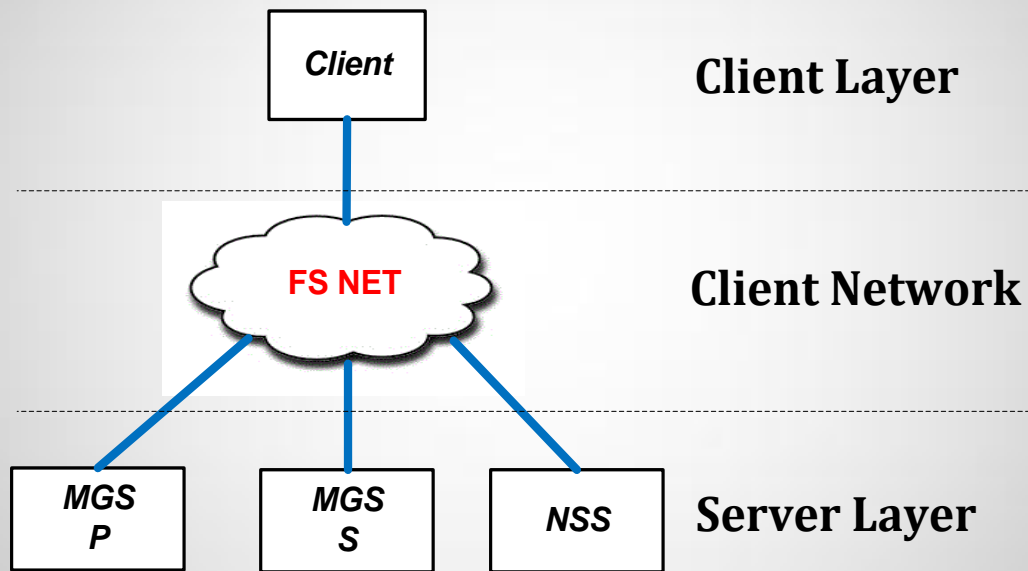
# GPFS History

- Video Streaming
- Tiger Shark -Real Time -Streaming -Read Perf -Wide Stripe
- GPFS 2.3
- HPC
- Large Scale Clusters 1000's of nodes
- GPFS 3.1
- ILM - Storage Pools - Filesets - Policy Engine
- Ease of Administration
- Multiple- networks
- Distributed Token Management
- GPFS 3.2
- Faster Failover
- Multiple NSD servers
- GPFS General File Serving

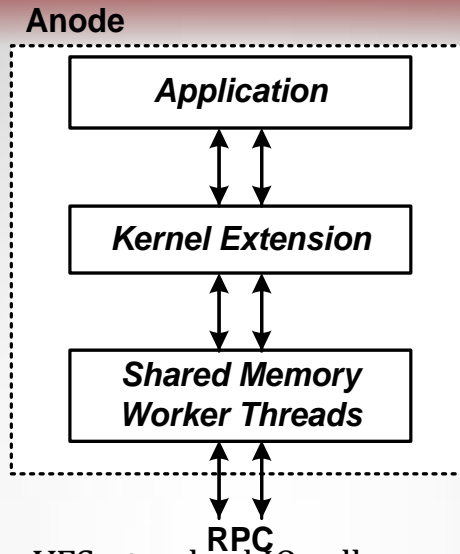
# CFS: Where GPFS is now.



# GPFS Components

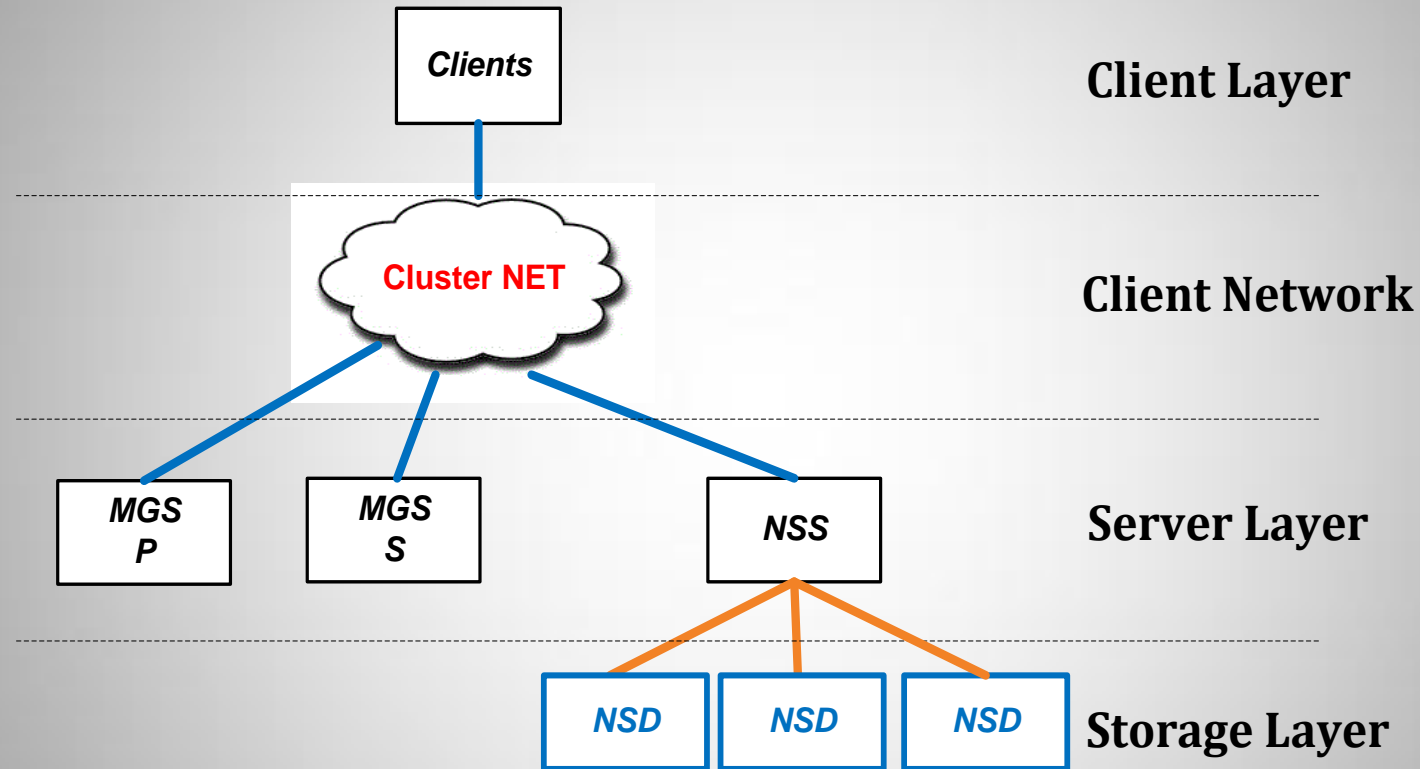


# GPFS FS structure



- Client accesses filesystem via Linux VFS, standard IO calls
- VFS talks to Kernel Extension Layer
- Kernel Extension Layer sends request to the proper server
  - Block IO to NSS
- All transactions handed by RPCs.

# GPFS Storage Components





# GPFS Storage Considerations

- Replication, yes.
- No protection other than replication
- Can use block devices as well as dm-0 etc.
- Storage health is entirely on storage unit
- Can integrate with HSM especially Tivoli.

# GPFS install in a nutshell

- Creating a cluster

`mmcrcluster`

- Creating network shared disks (NSD)

`mmcrnsd`

- Creating a file system

`mmcrfs`

- Mounting a file system

`mmmout`

# GPFS Features

## Posix compliant

- POSIX specification is single node, no clusters
- Atomic transactions

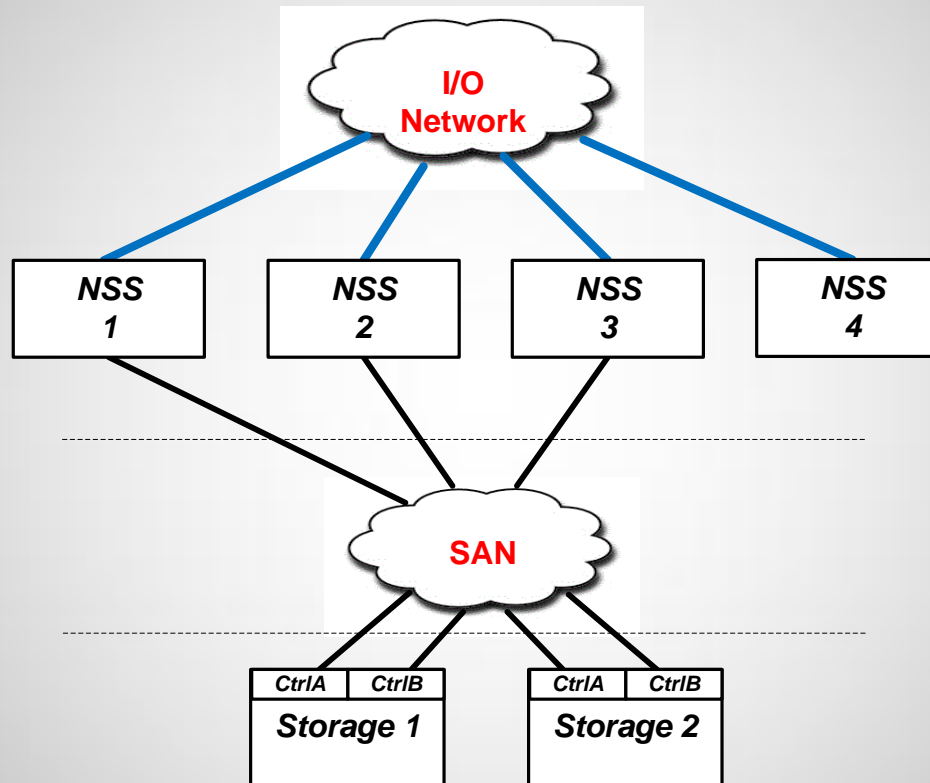
## Security

- Passwordless ssh may be a problem in some sites.
- Openssl for remote cluster communications

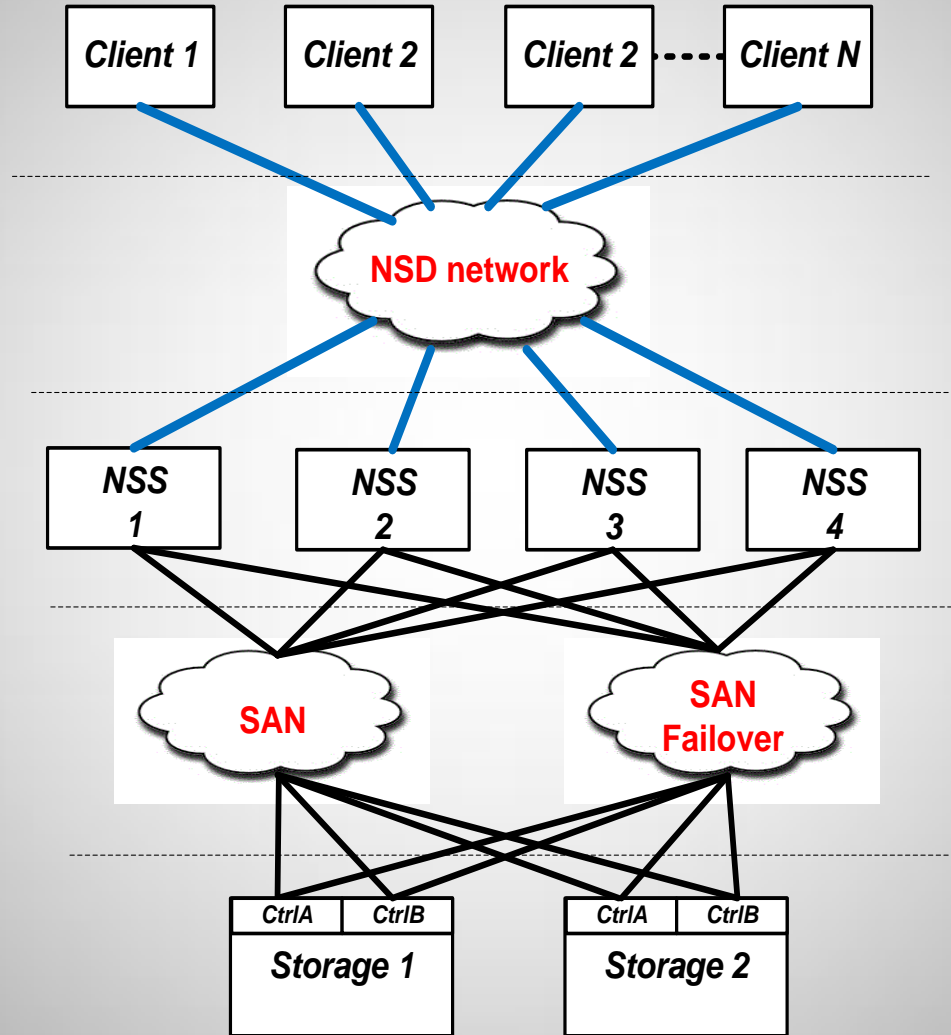
# GPFS Advanced Topics

- Pools
- Failover
- Snapshots
- HSM integration
- Remote clusters

# CFS: GPFS as Shared Disk



# CFS: GPFS as Clustered FS and HA



# CFS: GPFS more HA

