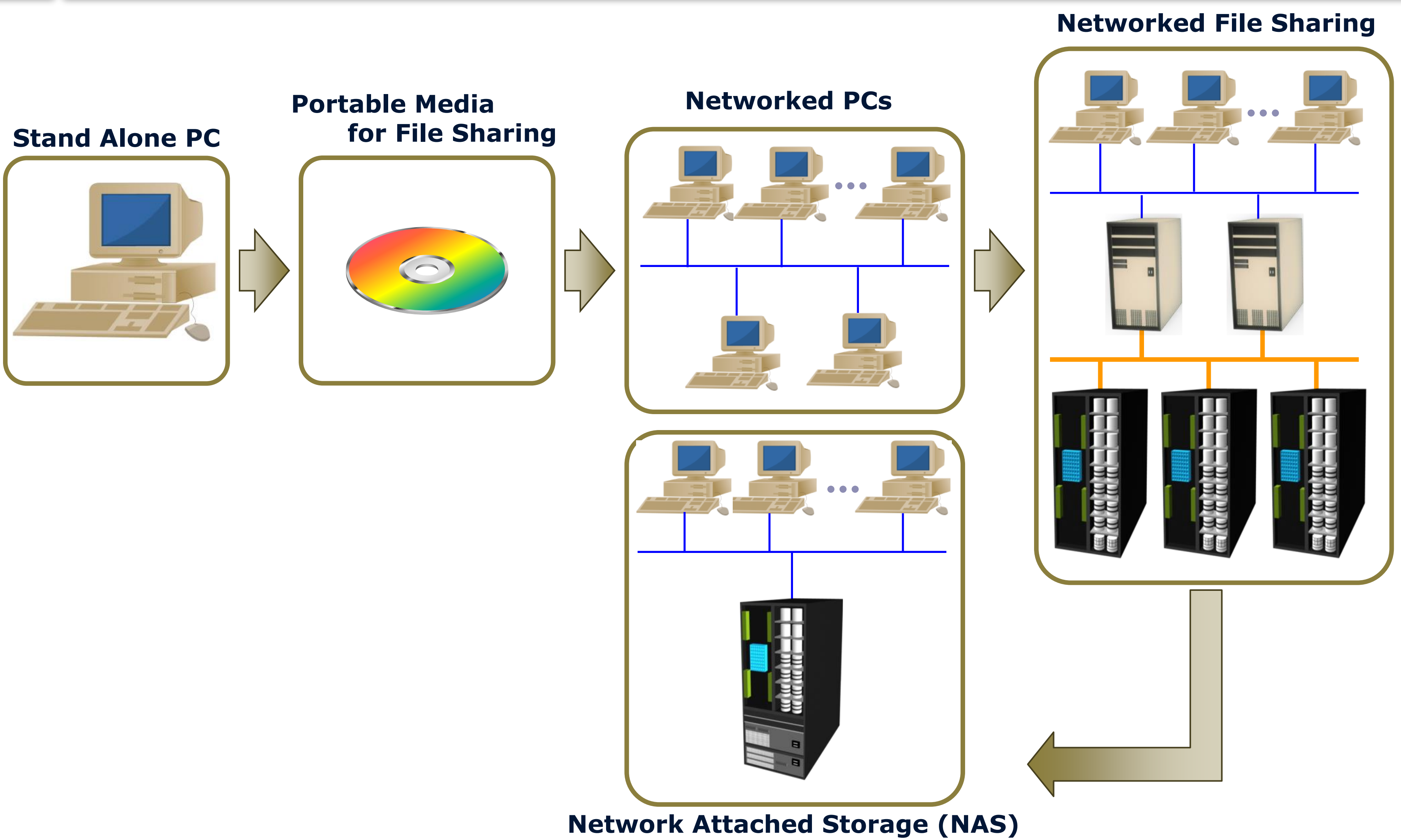
STORAGE SYSTEM ARCHITECTURES

Network-Attached Storage

File Sharing Environment

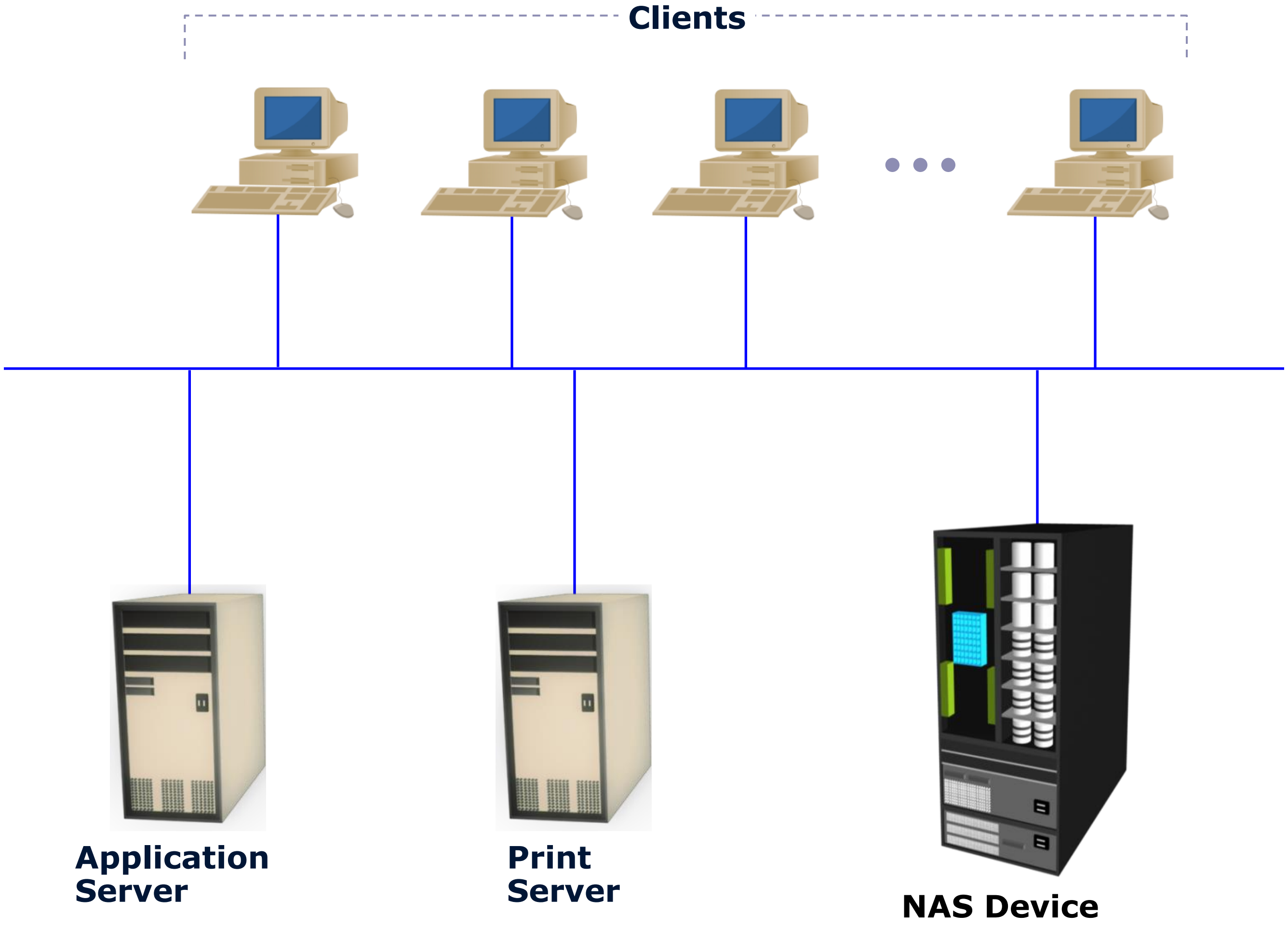
- File system is structured way of storing and organizing data files
- File Sharing
  - Storing and accessing data files over network
  - FS must be mounted in order to access files
- Traditional *client/server model,* implemented with file-sharing protocols for remote file sharing
  - Example: FTP, DNS
- DFS is another examples of file sharing implementation

File Sharing Technology Evolution

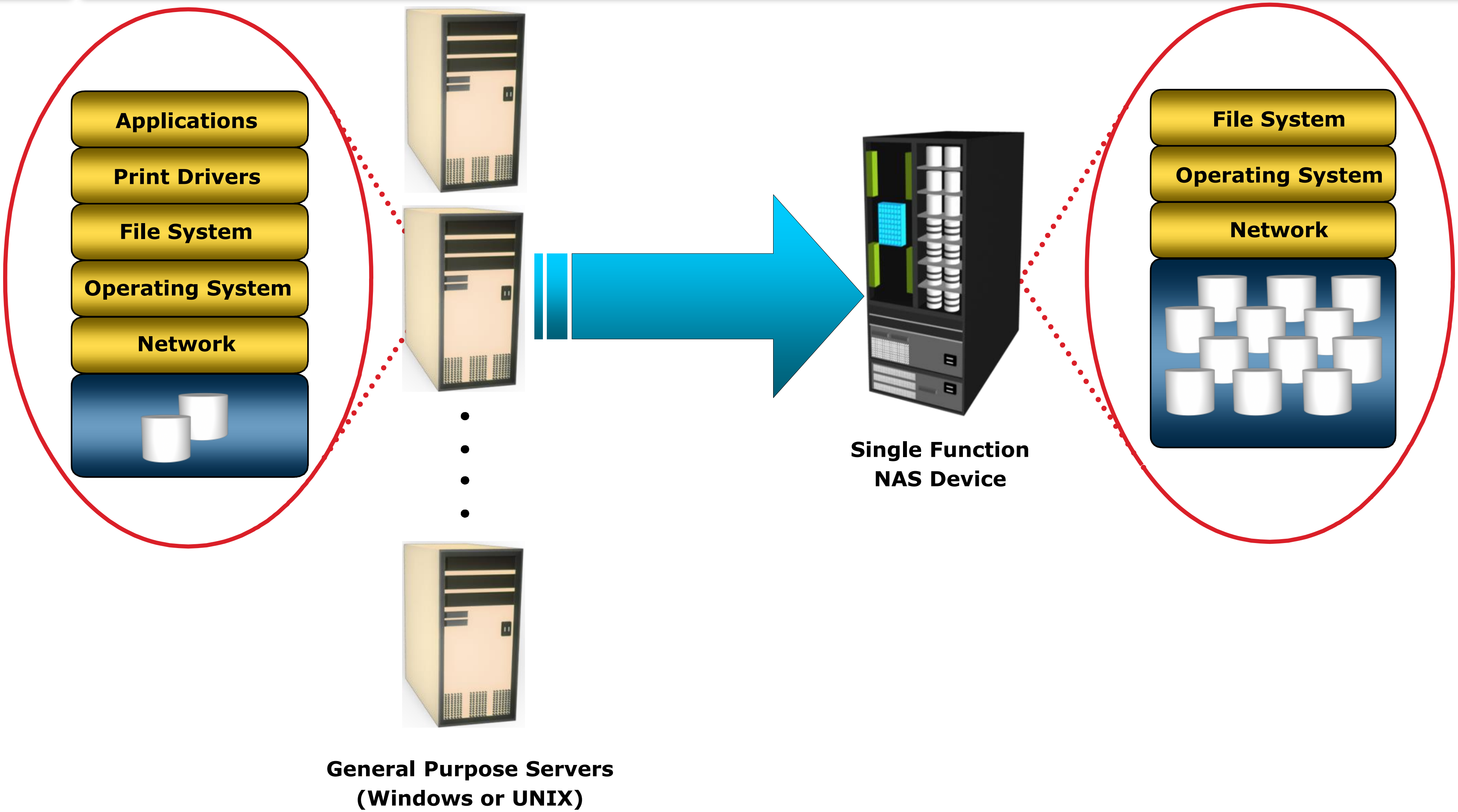


What is NAS ?

- NAS is shared storage on a network infrastructure



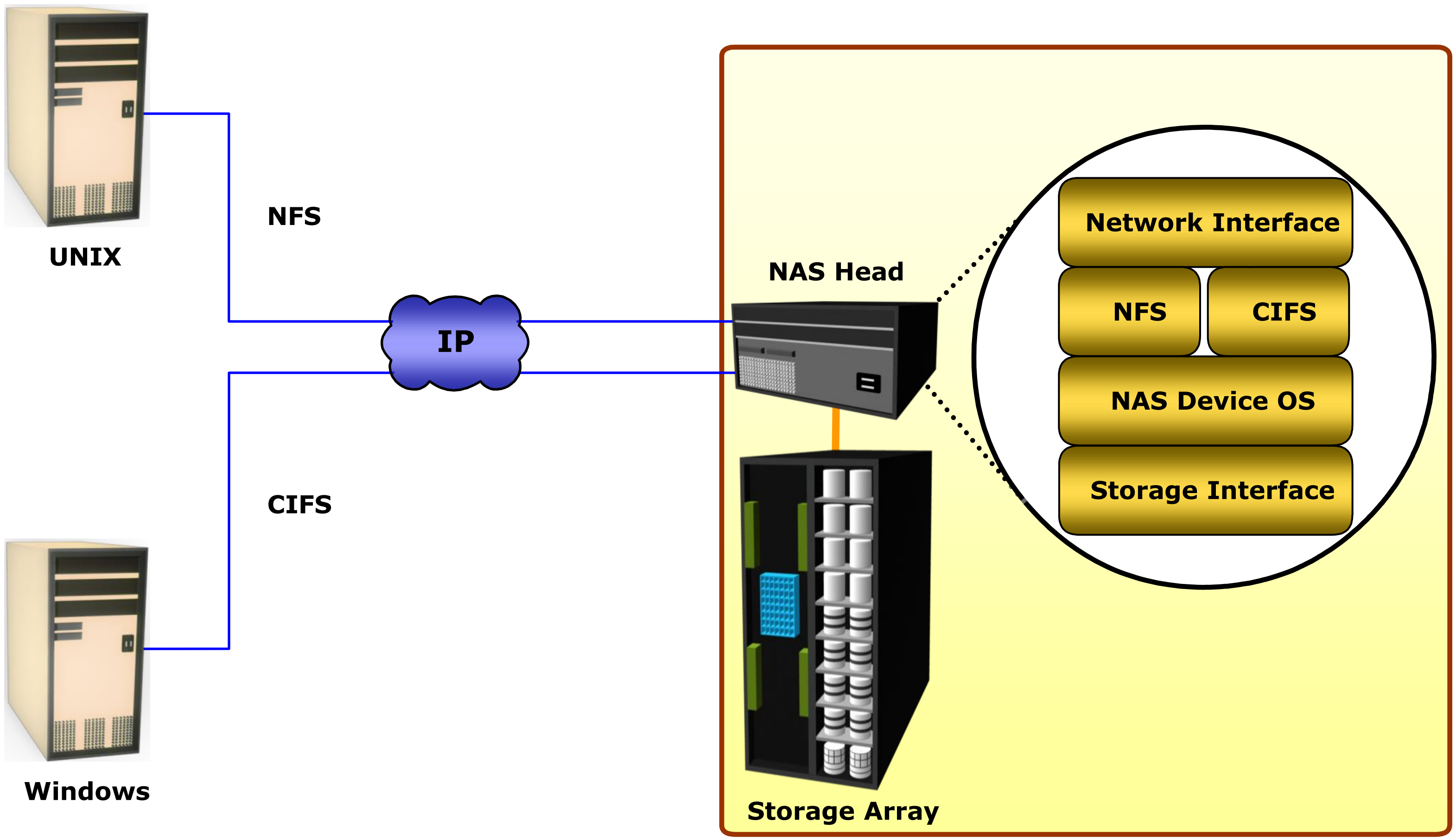
General Purpose Servers vs. NAS Devices



Benefits of NAS

- Support comprehensive access to information
- Improves efficiency
- Improved flexibility
- Centralizes storage
- Simplifies management
- Scalability
- High availability – through native clustering
- Provides security integration to environment (user authentication and authorization)

Components of NAS



NAS File Sharing Protocols

- Two common NAS file sharing protocols are:
  - CIFS – Common Internet File System protocol
    - Traditional Microsoft environment file sharing protocol, based upon the Server Message Block protocol
  - NFS – Network File System protocol
    - Traditional UNIX environment file sharing protocol

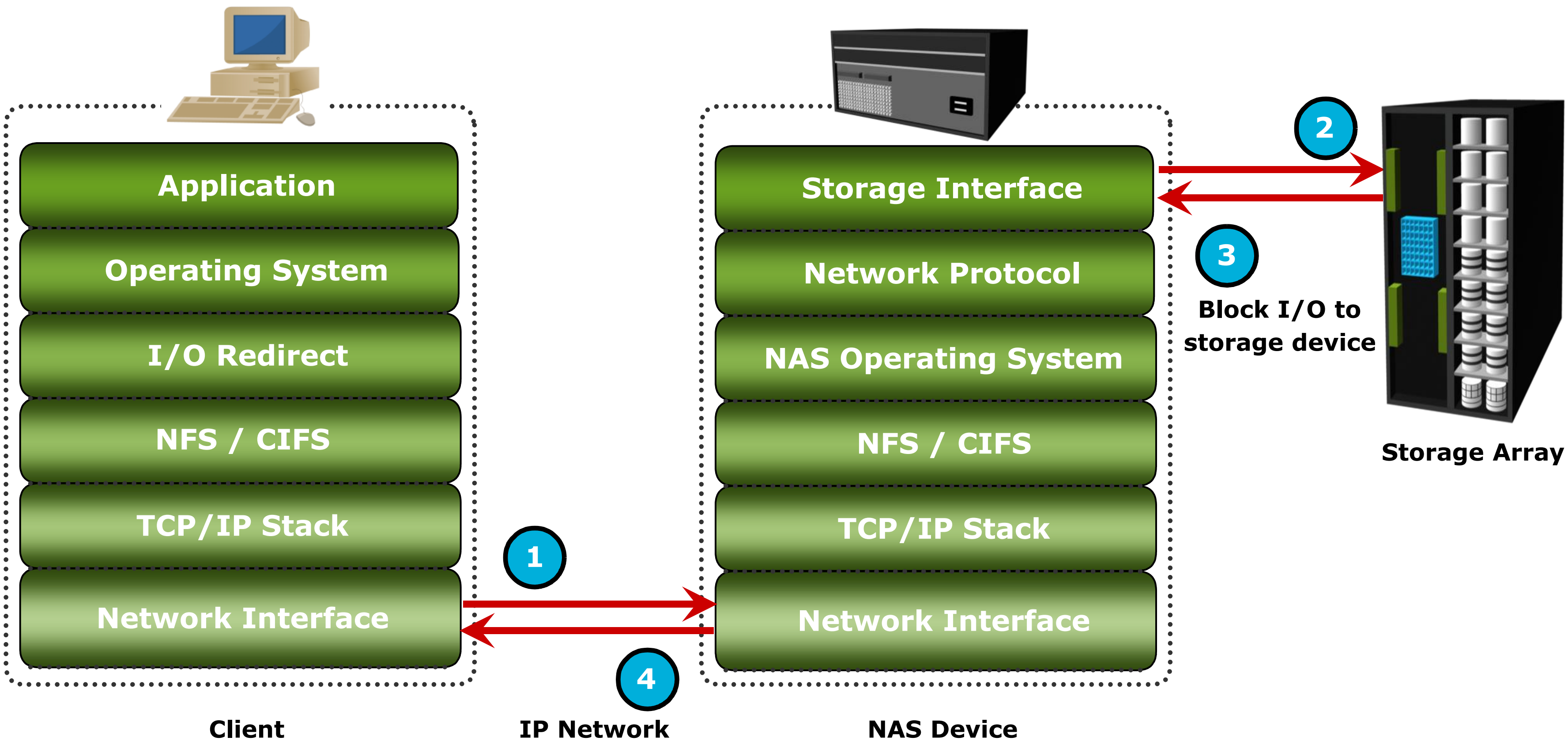
Network File System (NFS)

- Client/server application
- Uses RPC mechanisms over TCP protocol
- Mount points grant access to remote hierarchical file structures for local file system structures
- Access to the mount can be controlled by permissions

NAS File Sharing - CIFS

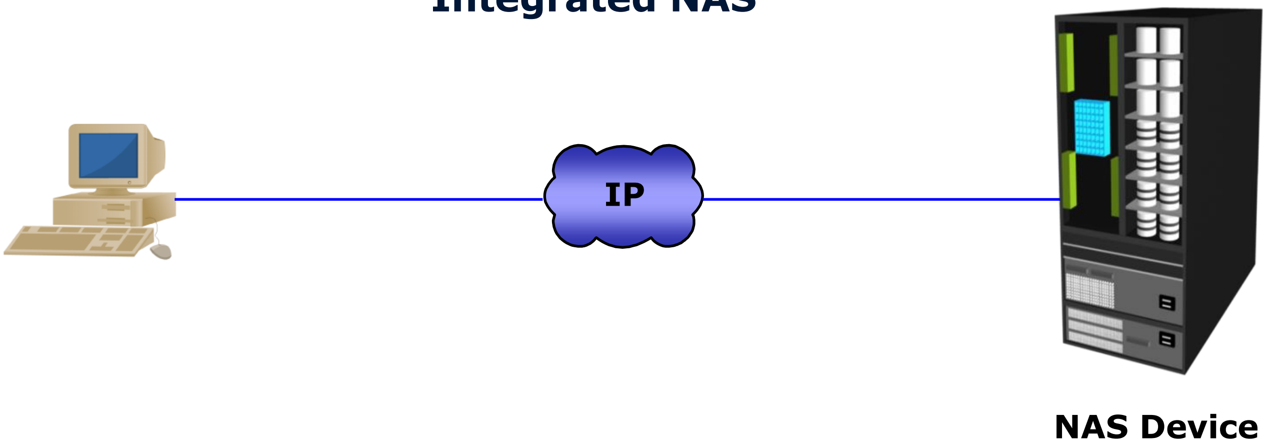
- Common Internet File System
  - Developed by Microsoft in 1996
  - An enhanced version of the Server Message Block (SMB) protocol
  - Stateful Protocol
    - Can automatically restore connections and reopen files that were open prior to interruption
  - Operates at the Application/Presentation layer of the OSI model
  - Most commonly used with Microsoft operating systems, but is platform-independent
  - CIFS runs over TCP/IP and uses DNS (Domain Naming Service) for name resolution

NAS I/O


Application
Operating System
I/O Redirect
NFS / CIFS
TCP/IP Stack
Network Interface
Client
1
IP Network
4
Storage Interface
Network Protocol
NAS Operating System
NFS / CIFS
TCP/IP Stack
Network Interface
NAS Device
2
3
Block I/O to storage device
Storage Array

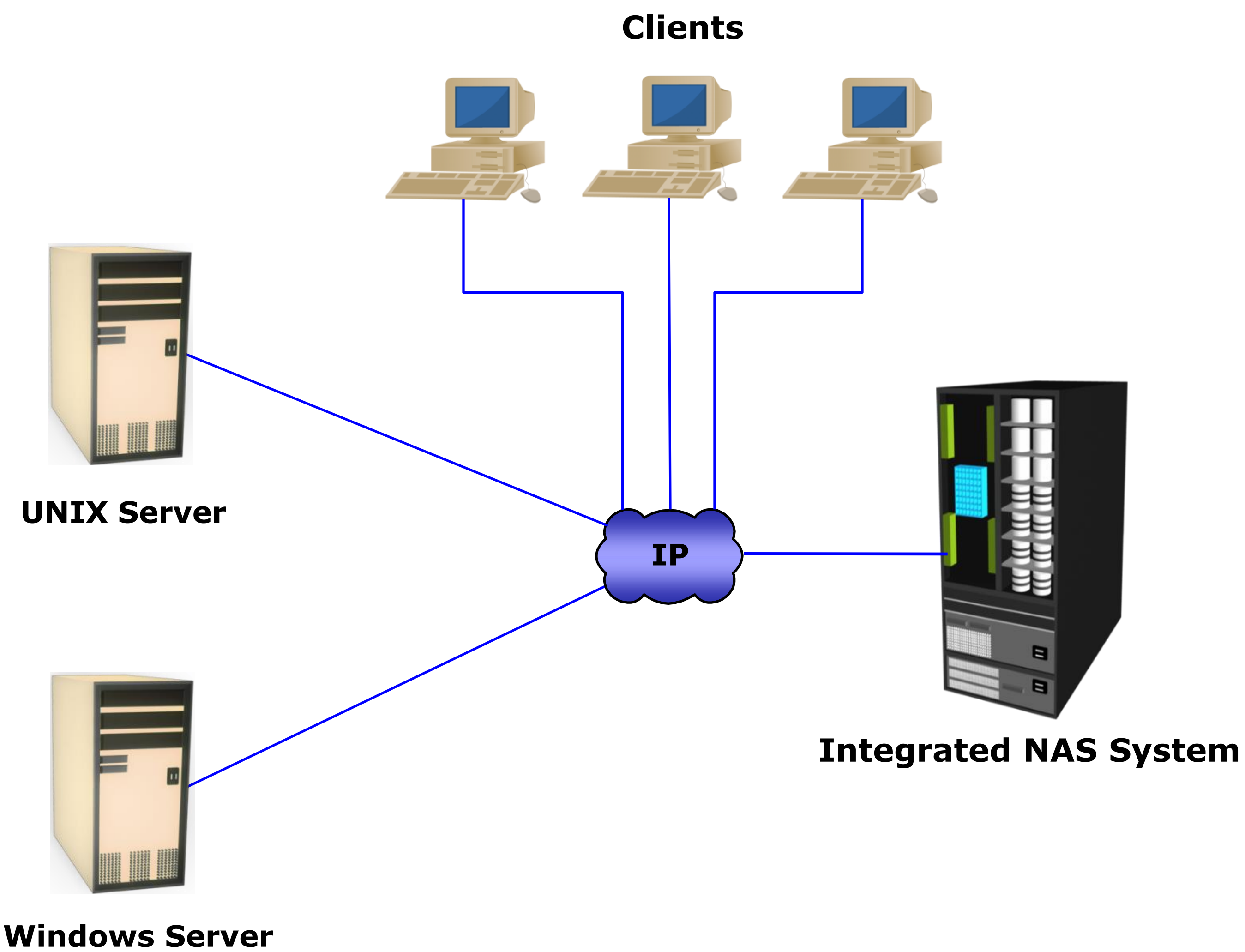
NAS Implementations

Integrated NAS

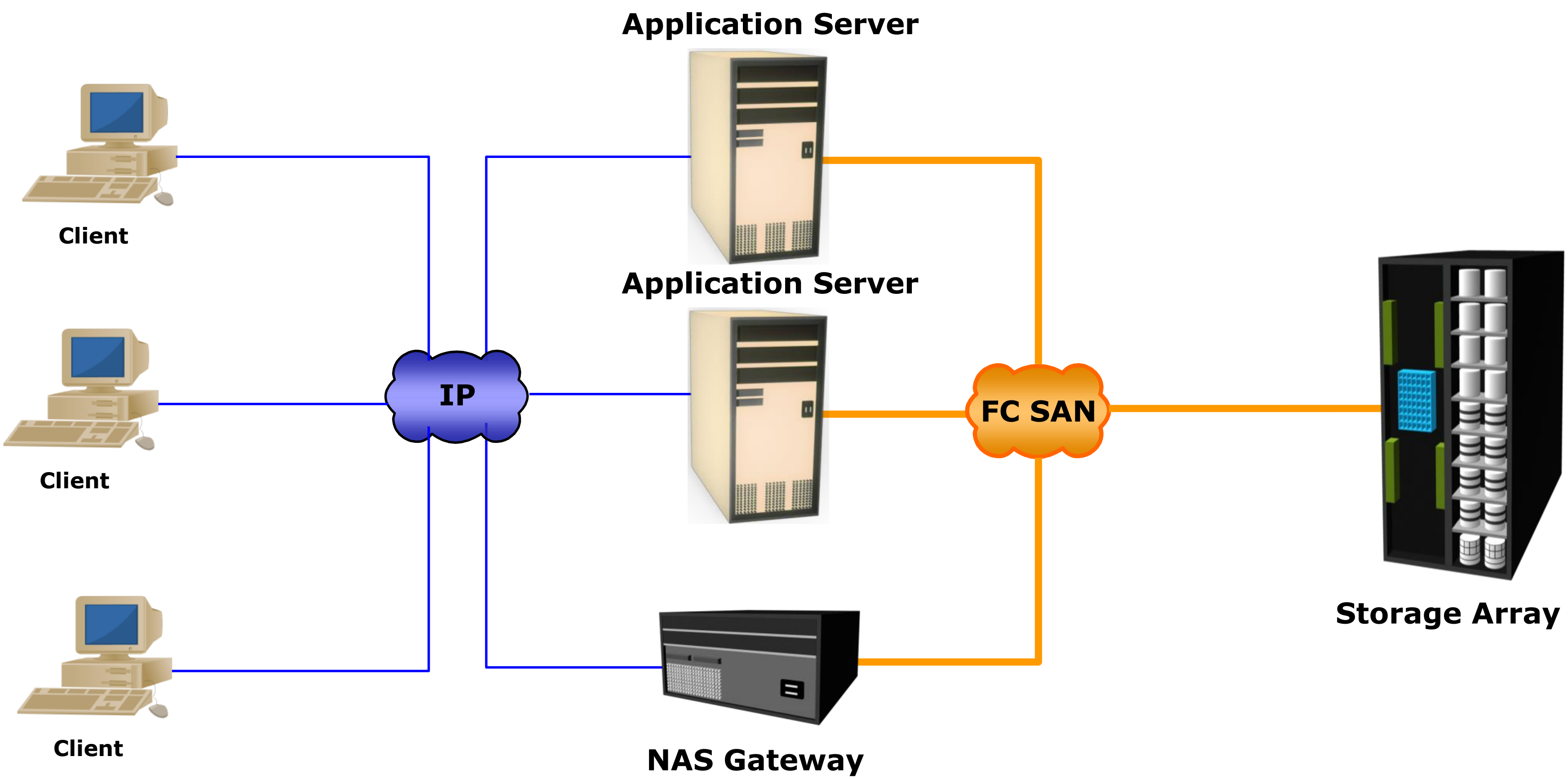


NAS Gateway

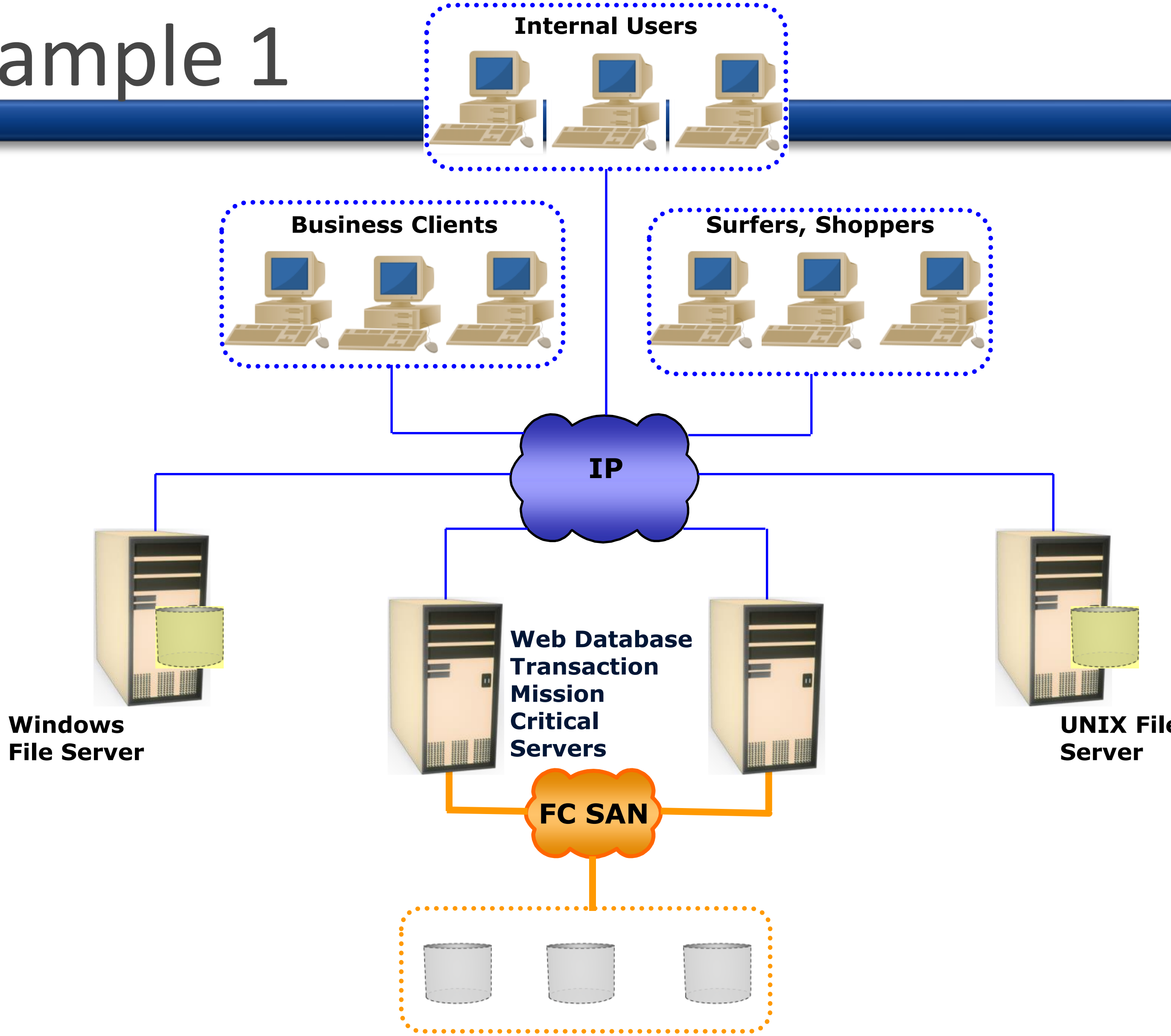
Integrated NAS Connectivity



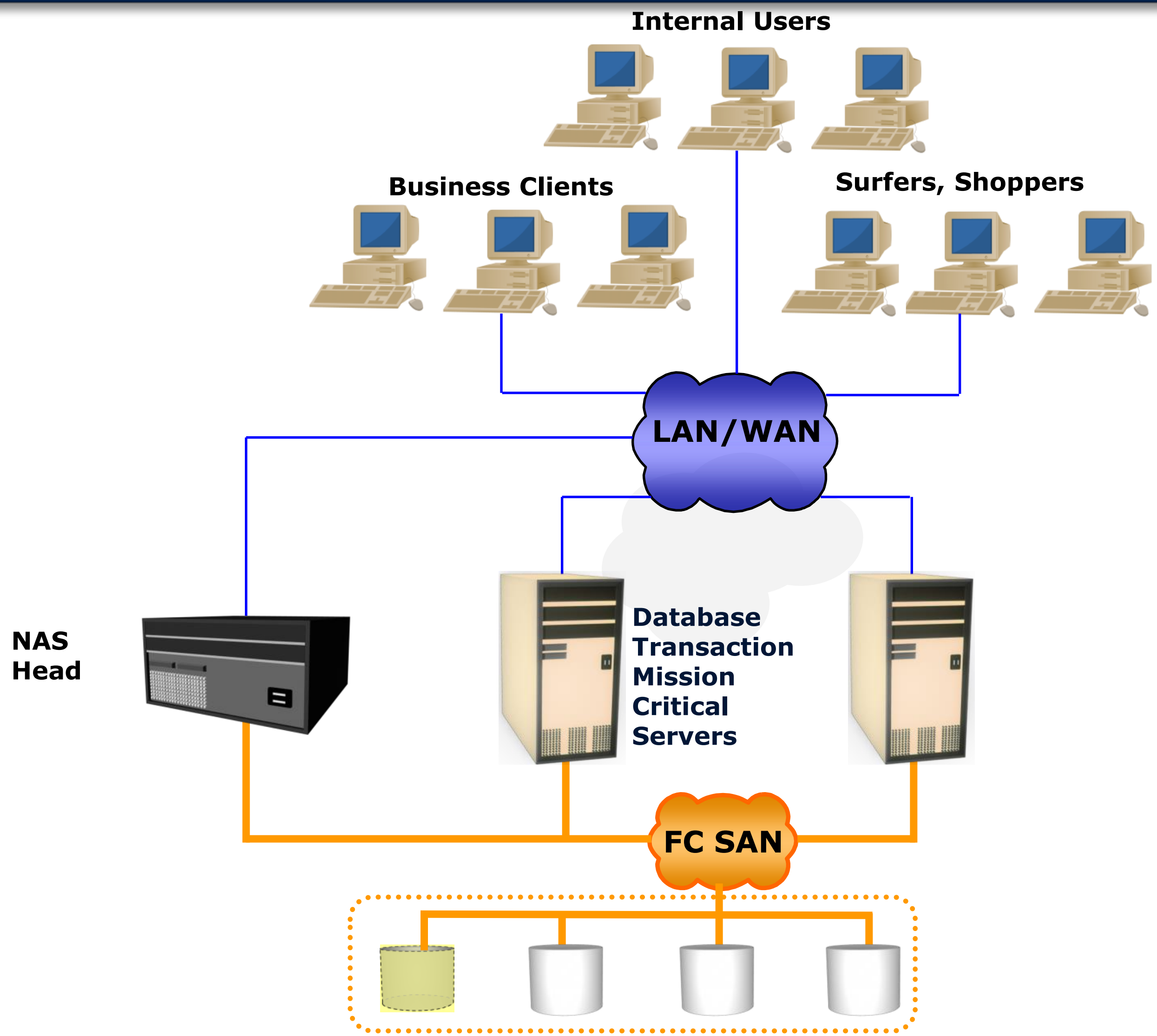
Gateway NAS Connectivity



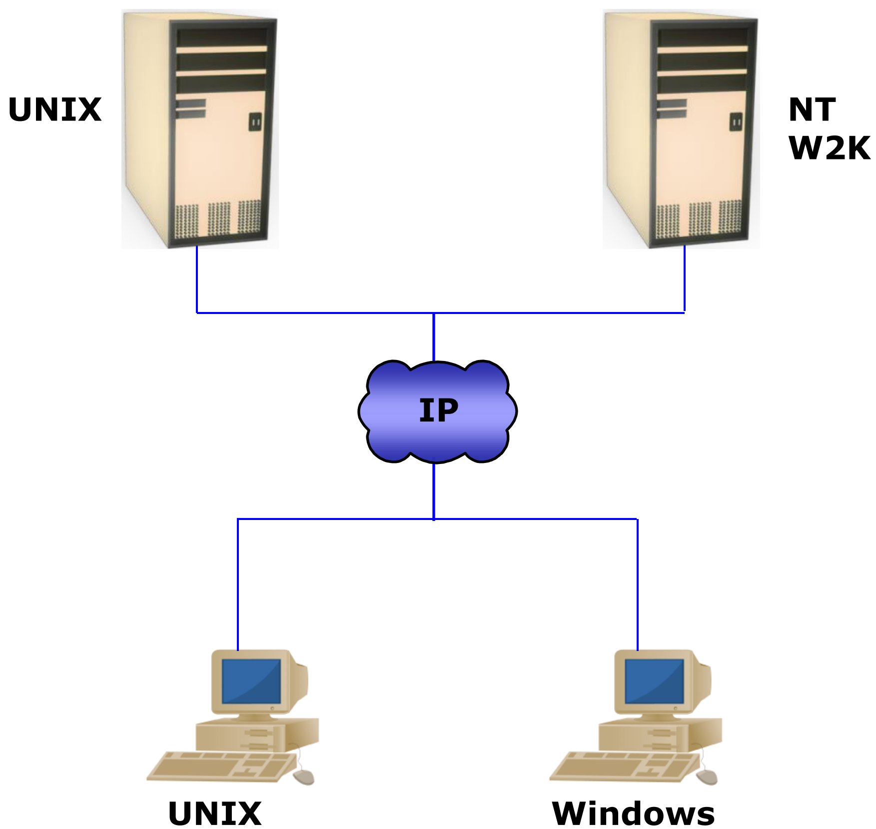
Traditional File Server Environment – Example 1



Storage Consolidation with NAS

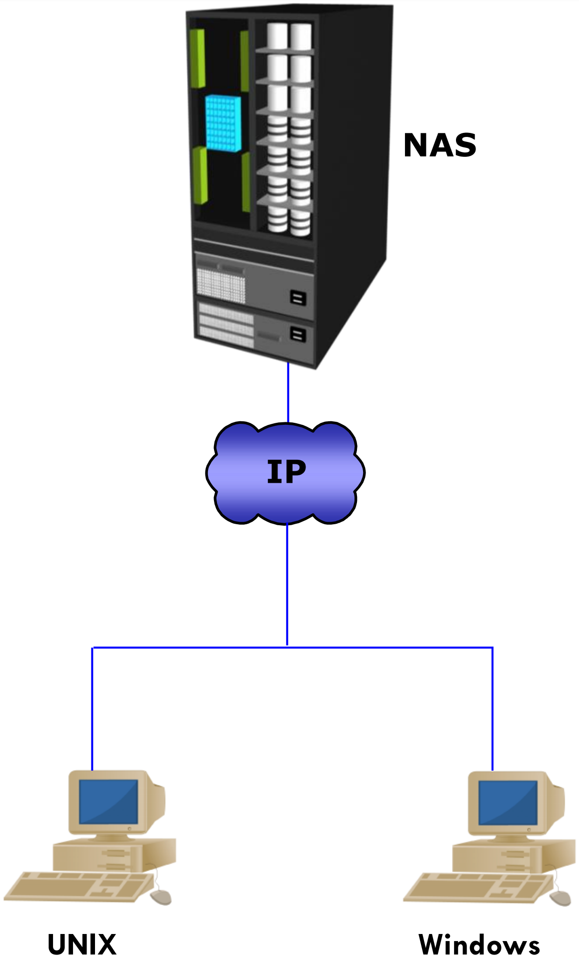


Traditional File Server Environment – Example 2



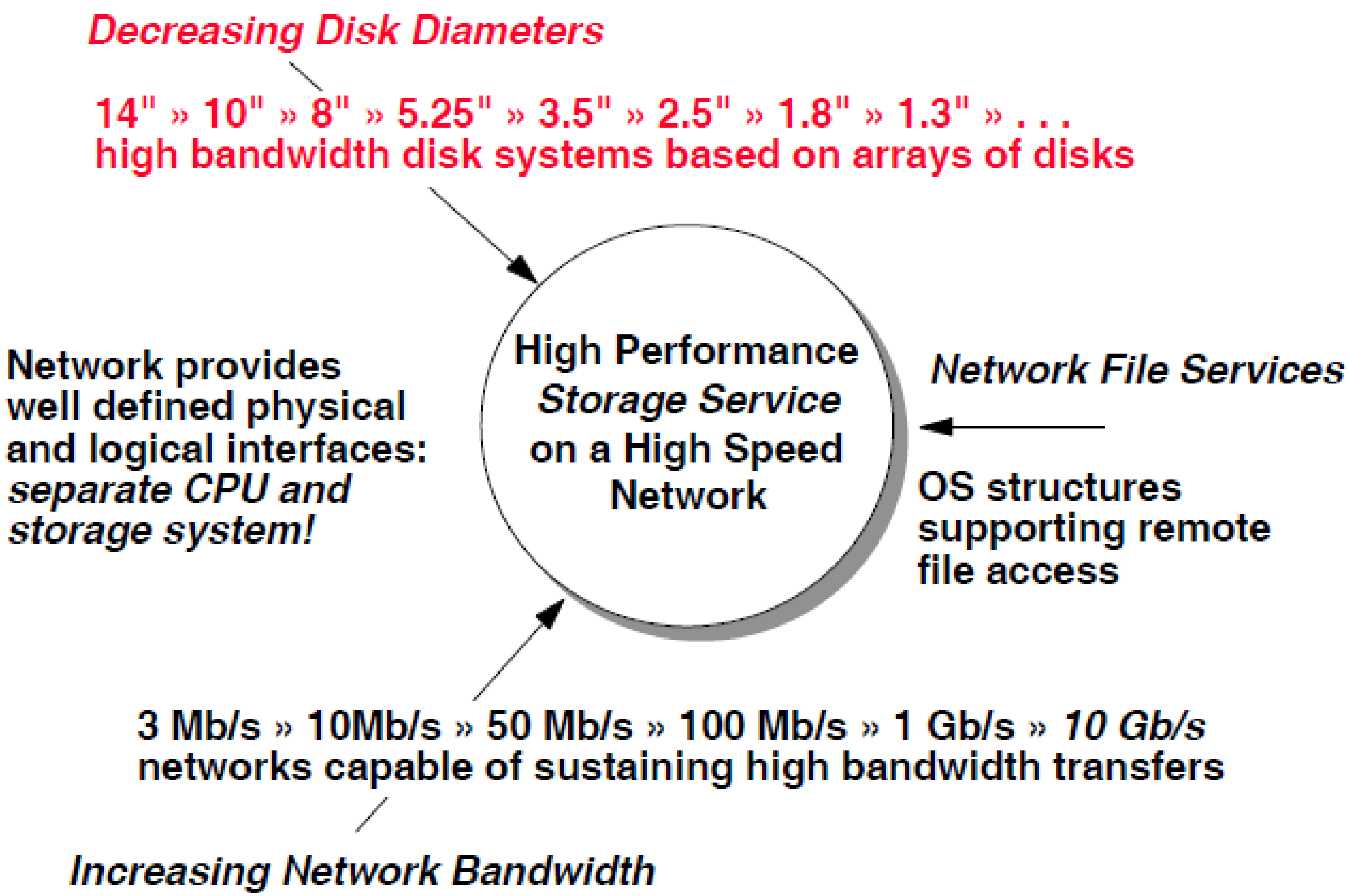
General purpose OS serving files via FTP, CIFS, NFS, HTTP. . .

Server Consolidation with NAS

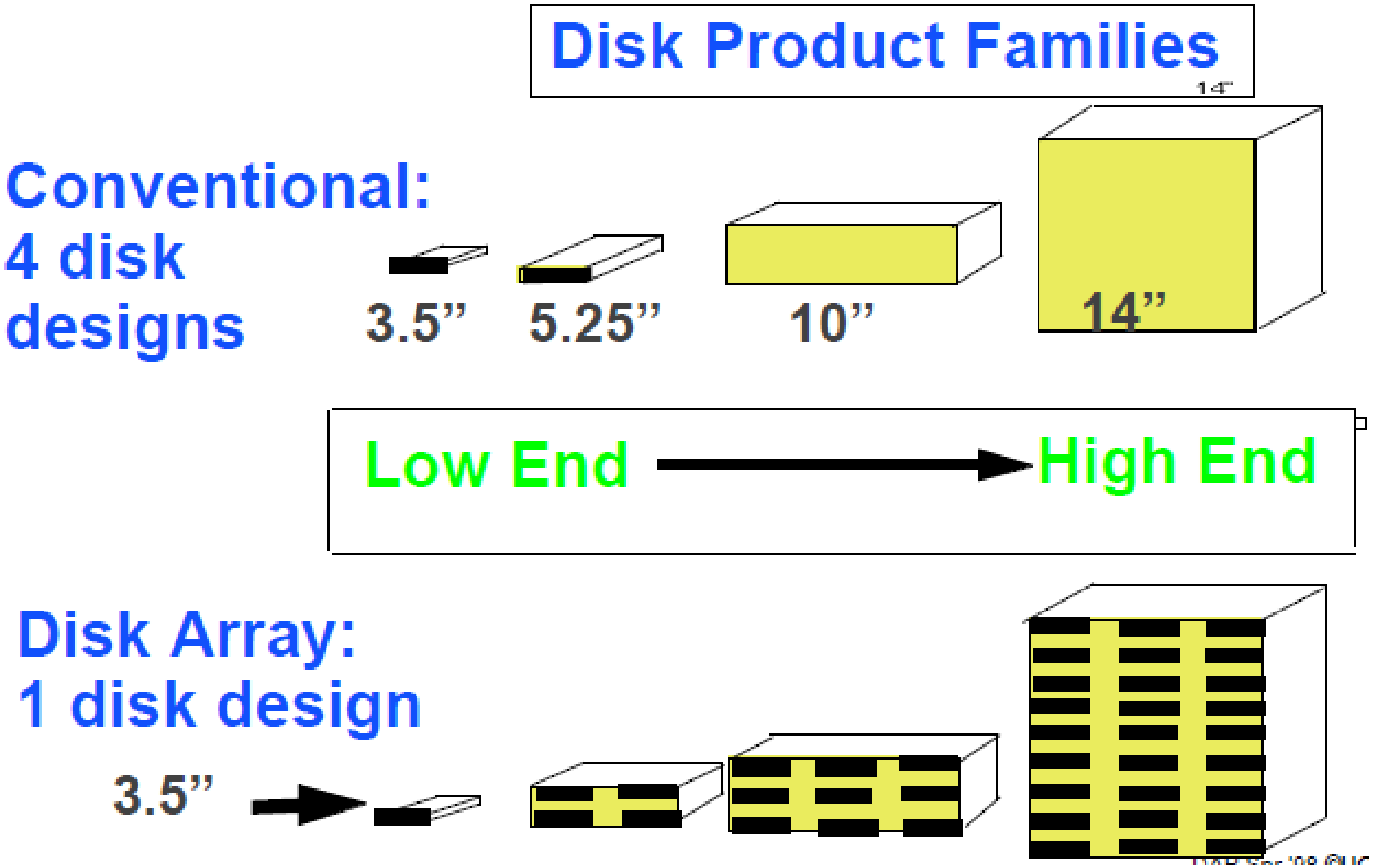


General purpose OS serving files via FTP, CIFS, NFS, HTTP. . .

Network Attached Storage



Manufacturing Advantages of Disk Arrays



Replace Small # of Large Disks with Large # of Small Disks!

	IBM 3390 (K)	IBM 3.5" 0061	*x70*
Data Capacity	20 GBytes	320 MBytes	*23 GBytes*
Volume	97 cu. ft.	0.1 cu. ft.	*11 cu. ft.*
Power	3 KW	11 W	*1 KW*
Data Rate	15 MB/s	1.5 MB/s	*120 MB/s*
I/O Rate	600 I/Os/s	55 I/Os/s	*3900 IOs/s*
MTTF	250 KHrs	50 KHrs	*??? Hrs*
Cost	$250K	$2K	*$150K*

Disk Arrays have potential for
- large data and I/O rates
- high MB per cu. ft., high MB per KW
- reliability?

Array Reliability

- Reliability of N disks = Reliability of 1 Disk ÷ N
  - 50,000 Hours ÷ 70 disks = 700 hours
  - Disk system MTTF: Drops from 6 years to 1 month!
- Arrays (without redundancy) too unreliable to be useful!

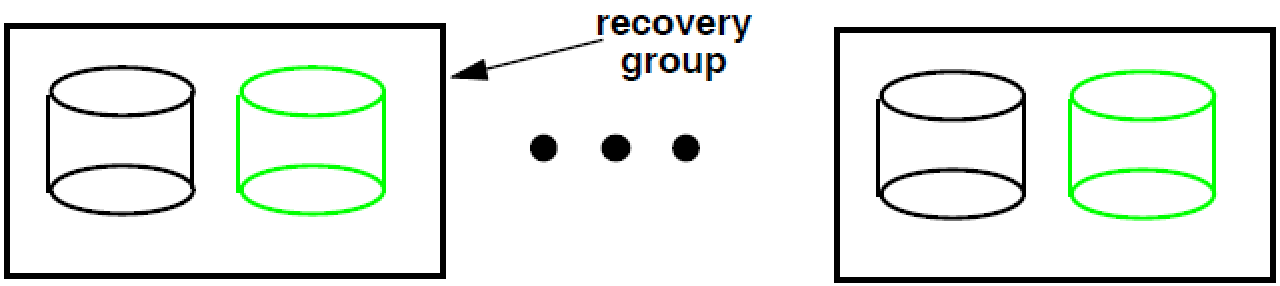
Redundant Arrays of Disks

- Files are "striped" across multiple spindles
- Redundancy yields high data availability
  - Disks will fail
  - Contents reconstructed from data redundantly stored in the array
    - Capacity penalty to store it
    - Bandwidth penalty to update

Techniques:

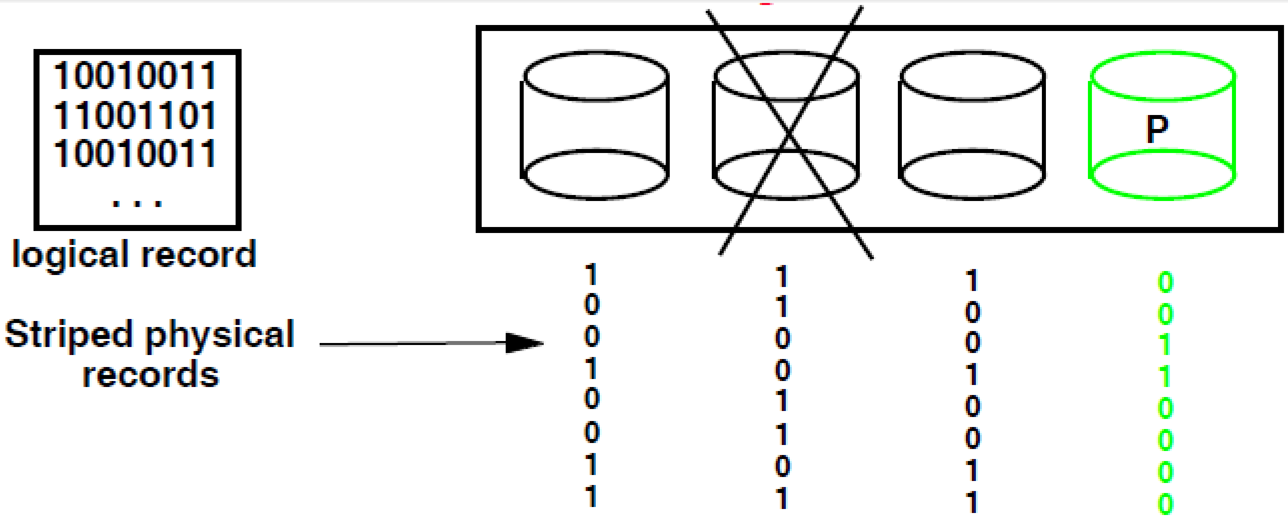
- Mirroring/Shadowing (high capacity cost)
- Horizontal Hamming Codes (overkill)
- Parity & Reed-Solomon Codes
- Failure Prediction (no capacity overhead!)
  VaxSimPlus — Technique is controversial

Redundant Arrays of Disks RAID 1: Disk Mirroring/Shadowing



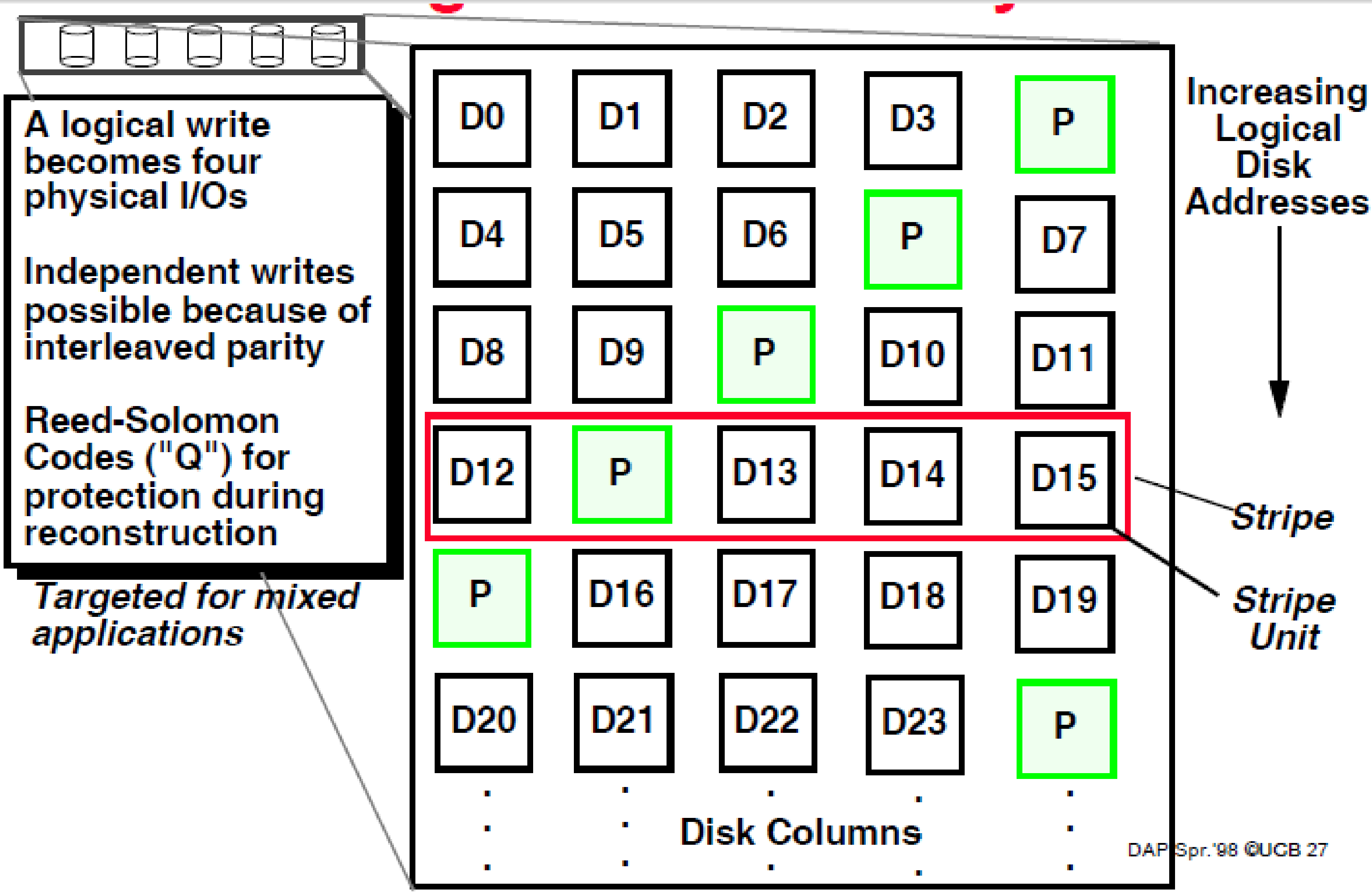
- Each disk is fully duplicated onto its "shadow“
  - Very high availability can be achieved
- Bandwidth sacrifice on write:
  - Logical write = two physical writes
- Reads may be optimized
- Most expensive solution: 100% capacity overhead
  - *Targeted for high I/O rate , high availability environments*

Redundant Arrays of Disks
RAID 3: Parity Disk



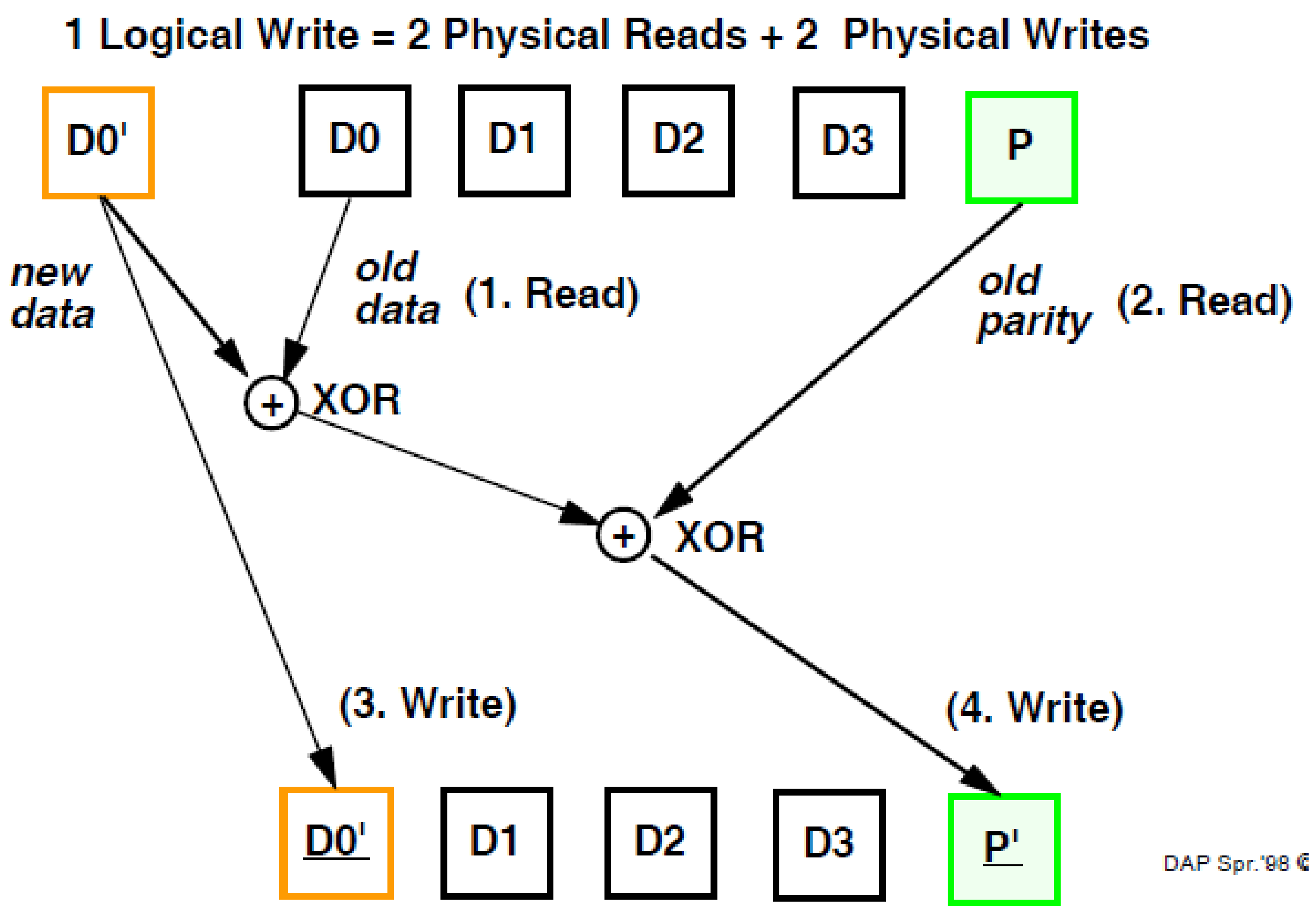
- Parity computed across recovery group to protect against hard disk failures
  - 33% capacity cost for parity in this configuration
  - wider arrays reduce capacity costs, decrease expected availability, increase reconstruction time
- Arms logically synchronized, spindles rotationally synchronized
  - logically a single high capacity, high transfer rate disk
- *Targeted for high bandwidth applications: Scientific, Image Processing*

Redundant Arrays of Disks RAID 5+: High I/O Rate Parity



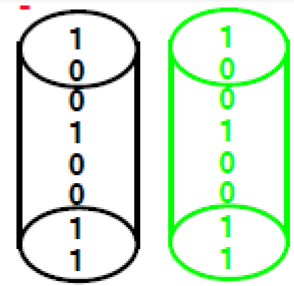
Problems of Disk Arrays: Small Writes

RAID-5: Small Write Algorithm

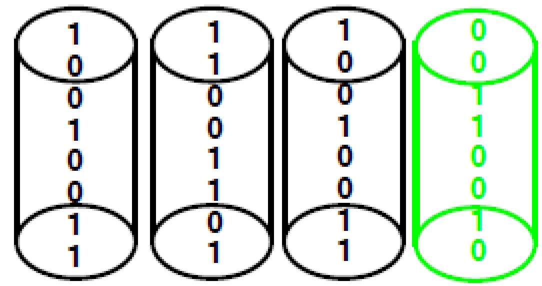


RAID

- *Disk Mirroring, Shadowing (RAID 1)*
  - Each disk is fully duplicated onto its "shadow"
  - Logical write = two physical writes
  - 100% capacity overhead



- *Parity Data Bandwidth Array (RAID 3)*
  - Parity computed horizontally
  - Logically a single high data bw disk



- *High I/O Rate Parity Array (RAID 5)*
  - Interleaved parity blocks
  - Independent reads and writes
  - Logical write = 2 reads + 2 writes
  - Parity + Reed-Solomon codes