MODULE – 5

FIBRE CHANNEL STORAGE AREA NETWORK (FC SAN)

Module 5: Fibre Channel Storage Area Network (FC SAN)

Upon completion of this module, you should be able to:

- Describe FC SAN and its components
- Describe FC architecture
- Describe FC SAN topologies and zoning
- Describe virtualization in SAN environment

Module 5: Fibre Channel Storage Area Network (FC SAN)

Lesson 1: Overview of FC SAN

During this lesson the following topics are covered:

- Evolution of FC SAN
- Components of FC SAN
- FC interconnectivity options
- FC port types

Business Needs and Technology Challenges

- An effective information management solution must provide:
  - Just-in-time information to business users □**availability**
  - Flexible and resilient (**strong**) storage infrastructure
- Information management challenges in direct-attached storage (DAS) environment:
  - Explosive growth of information storage that remains isolated and underutilized
  - Proliferation (**Increasing**) of new servers and applications
  - Complexity in sharing storage resources across multiple servers
  - High cost of managing information
- Storage area network (SAN) addresses these challenges

What is a SAN?

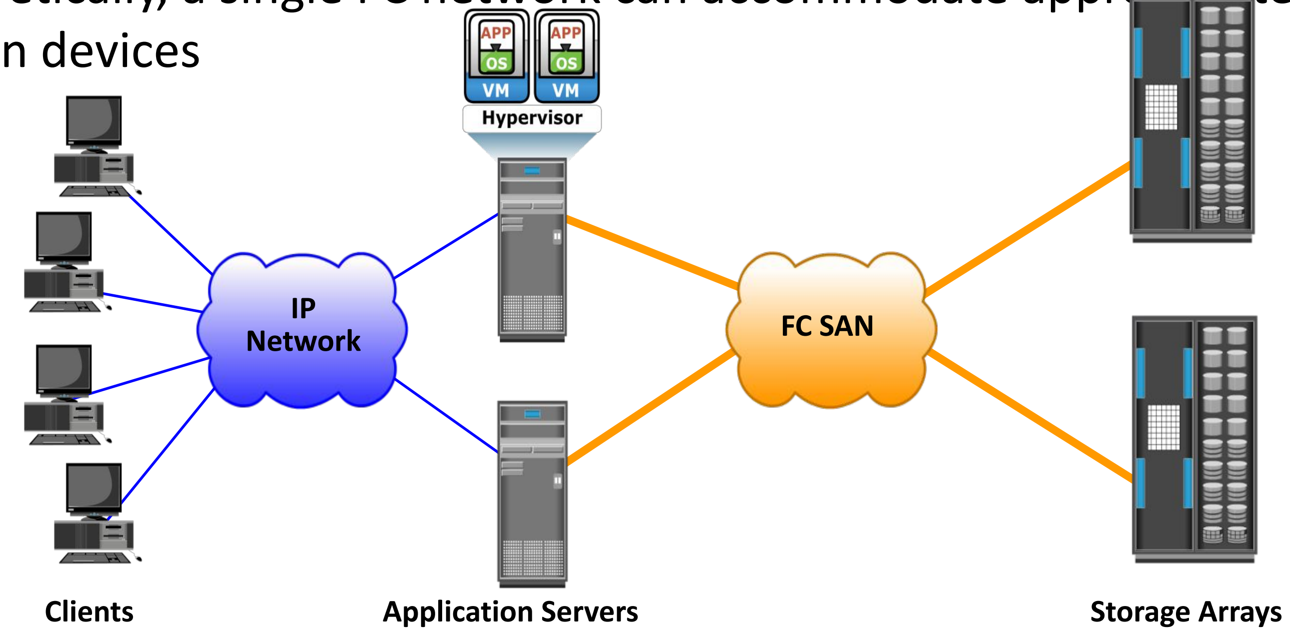
SAN

It is a high-speed, dedicated network of servers and shared storage devices.

- Centralizes storage and management
- Enables sharing of storage resources across multiple servers
- Meets increasing storage demands efficiently with better economies of scale
- Common SAN deployments are:
  - **Fibre Channel (FC) SAN: uses FC protocol for communication**
  - IP SAN: uses IP-based protocols for communication

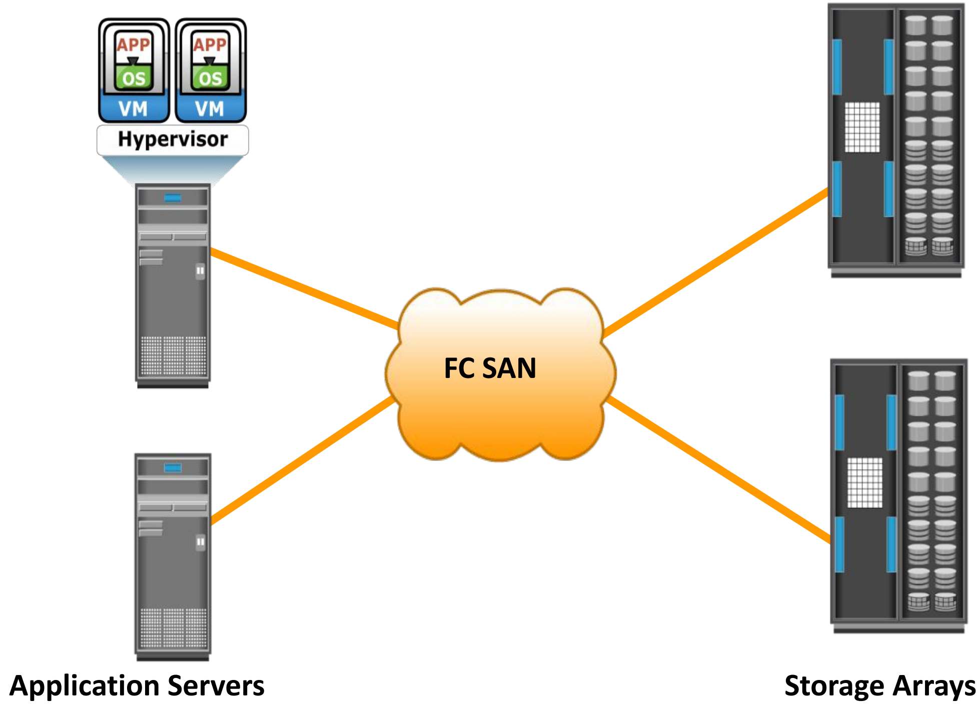
Understanding Fibre Channel

- High-speed network technology that uses:
  - Optical fiber cables (for front end connectivity)
  - Serial copper cables (for back end connectivity)
- Latest FC implementation supports speed up to 16 Gb/s
- Highly scalable
  - Theoretically, a single FC network can accommodate approximately 15 million devices



SAN

- SAN carries data between servers (or *hosts*) and storage devices through Fibre Channel network.
- Enables **storage consolidation (merge)**
  4 Storage management becomes **centralized** and **less complex**, which further **reduces** the **cost** of managing information
- Enables organizations to **connect** geographically dispersed servers and storage.



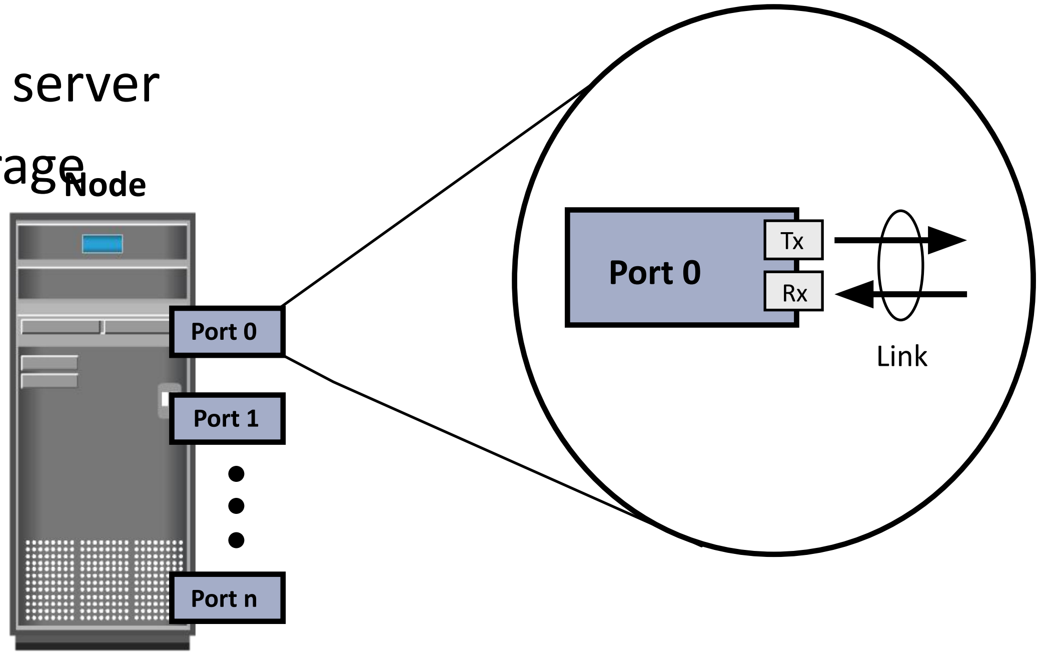
- Enables storage to be **shared** across multiple servers.
  - Improves the **utilization** of storage resources compared to direct-attached storage architecture
  - Reduces the **total** amount of storage an organization needs to purchase and manage.

Components of FC SAN

- Node (server and storage) ports
- Cables
- Connectors
- Interconnecting devices such as FC switches and hubs
- SAN management software

Components of FC SAN: (1) Node Ports

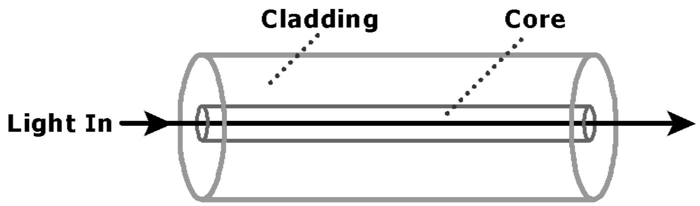
- In FC network, the end devices, such as hosts, storage arrays, and tape libraries, are all referred to as *nodes*.
- Each node requires one or more ports to provide a physical interface for communicating with other nodes.
- Exist on
  - Host bus adapter (HBA) in server
  - Front-end adapters in storage
- Each port has a transmit (Tx) link and a receive (Rx) link



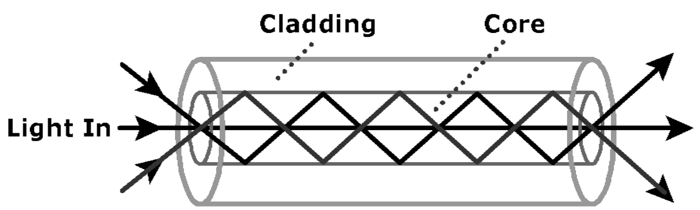
Components of FC SAN: (2) Cables

- SAN implementation uses
  - Copper cables for short distance (back-end)
  - Optical fiber cables for long distance
- Two types of optical cables: single-mode and multimode

Single-mode fiber (SMF)	Multimode fiber (MMF)
Carries single beam of light	Can carry multiple beams of light simultaneously
Distance up to 10km	Used for short distance (Modal dispersion = *collision* - weakens signal strength after certain distance)



Single-mode Fiber



Multimode Fiber

Components of FC SAN: (3) Connectors

- Attached at the end of a cable
- Enable swift (**rapid**) connection and disconnection of the cable to and from a port
- Commonly used connectors for fiber optic cables are:
  - Standard Connector (SC)
    - Duplex connectors
  - Lucent Connector (LC)
    - Duplex connectors
  - Straight Tip (ST)
    - Patch panel connectors
    - Simplex connectors

Standard Connector

Lucent Connector

Straight Tip Connector

Components of FC SAN: (4) Interconnecting Devices

- Commonly used interconnecting devices in FC SAN are:

Hubs

- **Physically** connect nodes in a logical loop or a physical star topology
- Provide **limited** connectivity and scalability
- All nodes must **share** the loop because data travels through all the connection points.
- Because of the availability of low-cost and high-performance switches, hubs are no longer used in FC SANs

Switches

- More intelligent than hubs and directly route data from one physical port to another
- Switches are available with fixed port count or modular design
- Nodes **do not share** the bandwidth.
- Instead, each node has a dedicated communication path

Directors

- High-end switches with a **higher port count** and better fault tolerance capabilities.
- Always modular, and its port count can be increased by inserting additional 'line cards' or 'blades'
- High-end switches and directors contain redundant components

Components of FC SAN: (5) SAN Management Software

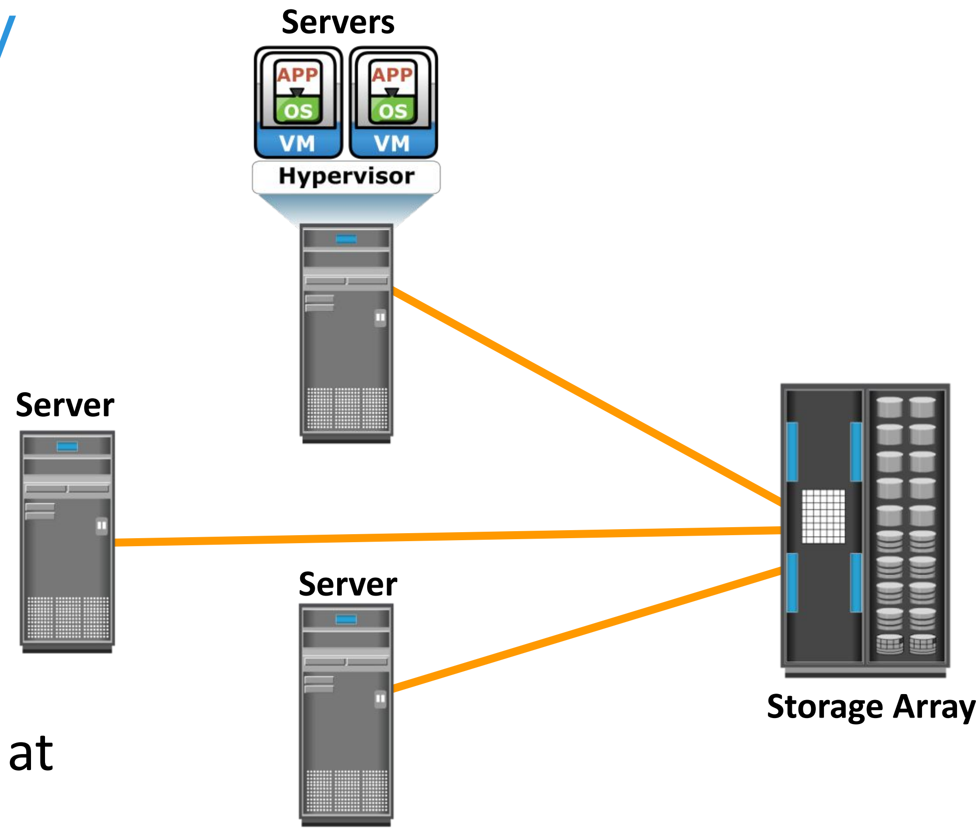
- A suite of tools used in a SAN to manage interfaces between host and storage arrays
- Provides integrated management of SAN environment
- Enables web-based management using GUI or CLI
- Key management functions:
  - Mapping of storage devices, switches and servers
  - Monitoring and generating alerts for discovered devices
  - Logical partitioning of SAN (zoning)
  - Management of SAN components (HBAs, storage components and interconnecting devices)

FC Interconnectivity Options

- Point-to-Point
- Fibre Channel Arbitrated Loop (FC-AL)
- Fibre Channel Switched Fabric (FC-SW)

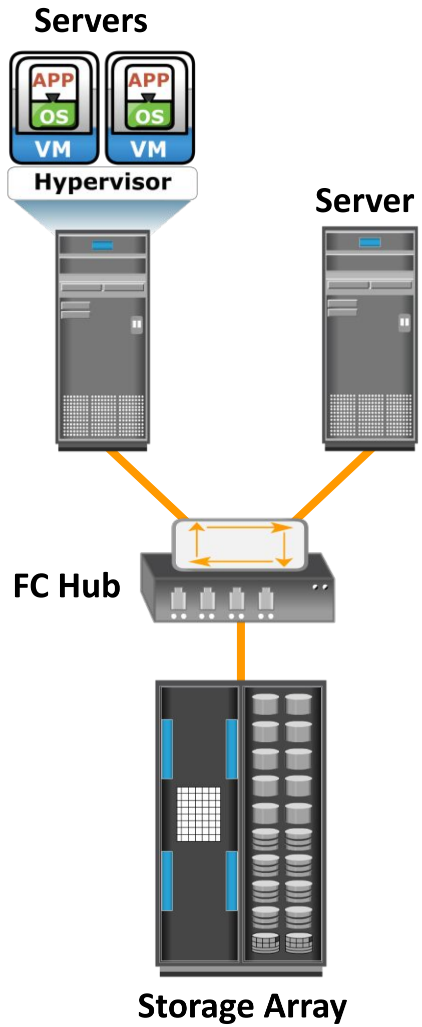
Point-to-Point Connectivity

- Two devices are **connected directly** to each other
- Enables direct connection between nodes
- Offers limited connectivity and scalability
  - 4 Only two devices can communicate with each other at a given time
  - 4 Cannot be scaled to accommodate a large number of nodes
- Used in DAS environment



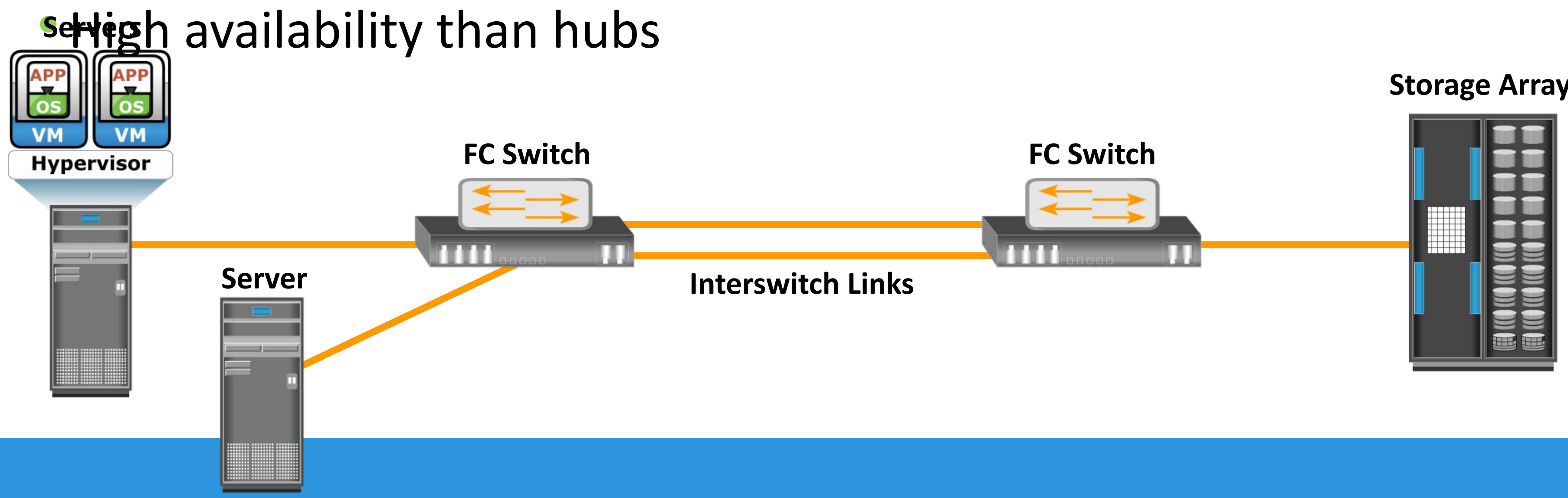
Fibre Channel Arbitrated Loop (FC-AL) Connectivity

- Provides **shared loop** to attached nodes
  - Each node **contends** with other nodes to perform I/O operations
  - Nodes must **arbitrate** to gain control
  - At any given time, only **one device** can perform I/O operations on the loop
- Implemented using ring or star topology
- Limitations of FC-AL
  - Only **one device** can perform I/O operation at a time
  - Supports up to 126 nodes
  - Addition or removal of a node causes momentary (**temporary**) pause in loop traffic



Fibre Channel Switched Fabric (FC-SW) Connectivity

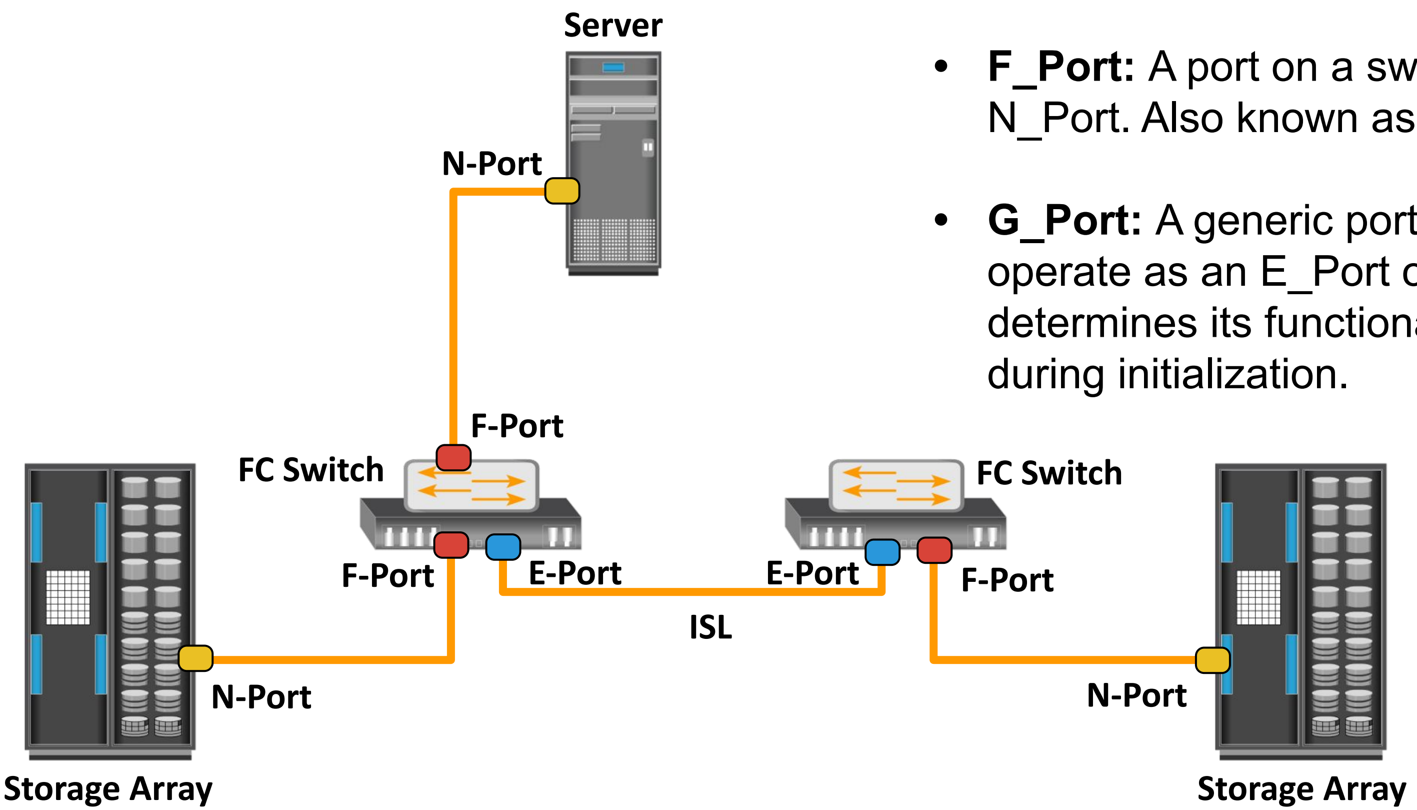
- Creates a logical space (called fabric) in which all nodes communicate with one another **using switches**
  - Interswitch links (ISLs) enable switches to be connected together
- Provides dedicated path between nodes
- Addition/removal of node does not affect traffic of other nodes
- High availability than hubs



Switched Fabric Ports

Ports in a switched fabric can be one of the following types:
N-Port, F-Port, E-Port, G-Port

- **N_Port:** An end point in the fabric. Also known as the *node port*. Typically, it is a host port (HBA) or a storage array port connected to a switch in a switched fabric.
- **E_Port: F**orms the connection between two FC switches. Also known as the *expansion port*.
- **F_Port:** A port on a switch that connects an N_Port. Also known as a *fabric port*.
- **G_Port:** A generic port on a switch that can operate as an E_Port or an F_Port and determines its functionality automatically during initialization.



Module 5: Fibre Channel Storage Area Network (FC SAN)

Lesson 2: Fibre Channel (FC) Architecture

During this lesson the following topics are covered:

- FC protocol stack
- FC addressing
- WWN addressing
- Structure and organization of FC data
- Fabric services
- Fabric login types

FC Architecture Overview

- Provides benefits of both channel and network technologies
  - Provides high performance with low protocol overheads
    - Due to the static nature of channels & high level h/w and s/w integration
  - Provides high scalability with long distance capability
    - Uses share bandwidth for communication
- FC SAN uses **FibreChannel Protocol (FCP)** that provides both channel speed for data transfer with low protocol overhead and scalability of network technology

FC Architecture Overview

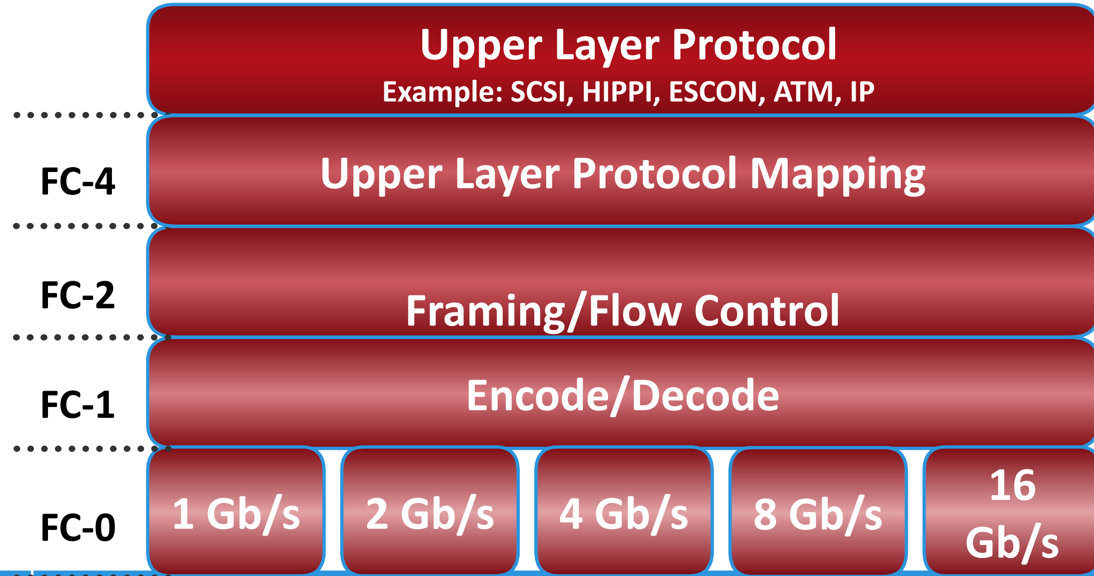
- FCP forms the fundamental construct of the FC SAN infrastructure:
  - Provides a serial data transfer interface that operates over copper wire and optical fiber
  - Implements SCSI over FC network
  - Storage devices, attached to SAN, appear as local storage devices to host operating system
- Advantages of FCP:
  - Sustained transmission bandwidth over long distances
  - Support large number of addressable devices over a network
  - Support speeds up to 16 Gbps

Fibre Channel Protocol Stack

High Performance Parallel Interface (HIPPI) Framing Protocol

Enterprise Storage Connectivity (ESCON)

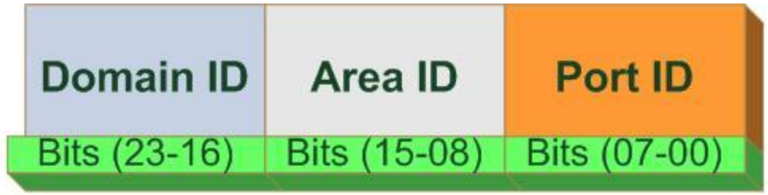
Asynchronous Transfer Mode (ATM),



FC Layer	Function	Features Specified by FC Layer
FC-4	Mapping interface	Mapping upper layer protocol (e.g. SCSI) to lower FC layers
FC-3	Common services	Common services (Not implemented)
FC-2	Routing, flow control	Frame structure, FC addressing, flow control
FC-1	Encode/decode	8b/10b or 64b/66b encoding, bit and frame synchronization
FC-0	Physical layer	Media, cables, connector

FC Addressing in Switched Fabric

- FC Address is assigned to nodes during fabric login
  - Used for communication between nodes within FC SAN
- Address format



- Domain ID is a unique number provided to each switch in the fabric
  - 8-bits field - 239 addresses are available for domain ID
  - Because some addresses are deemed special and reserved for fabric management services
  - E.g: FFFFFC is reserved for the name server
- Area ID: to identify group of switch ports used for connecting nodes
- Port ID: to identify port within the group
- Maximum possible number of node ports in a switched fabric:
  - 239 domains X 256 areas X 256 ports = 15,663,104

World Wide Name (WWN)

- Each device in the FC environment is assigned a 64-bit unique identifier
- Static to node ports on an FC network
    4 Similar to Media Access Control (MAC) address used in IP networking
- World Wide Node Name (WWNN) and World Wide Port Name (WWPN) are used: to uniquely identify nodes and ports respectively

World Wide Name - Array															
5	0	0	6	0	1	6	0	0	0	6	0	0	1	B	2
010	000	000	011	000	000	011	000	000	000	011	000	000	000	101	001
1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0
Format Type	Company ID 24 bits						Port	Model Seed 32 bits							

World Wide Name - HBA															
1	0	0	0	0	0	0	0	c	9	2	0	d	c	4	0
Format Type	Reserved 12 bits			Company ID 24 bits						Company Specific 24 bits					

Structure and Organization of FC Data

- FC data is organized as Exchange, Sequence, and Frame

Exchange	Sequence	Frame

• Enables two N_Ports to identify and manage a set of information units
  • Information unit: upper layer protocol-specific information that is sent to another port to perform certain operation
  • Each information unit maps to a sequence
• Includes one or more sequences
	• Contiguous set of frames that correspond to an information unit	• Fundamental unit of data transfer
		• Each frame consists of five parts: start of frame (SOF), frame header, data field, cyclic redundancy check (CRC), and enf of frame (EOF)

SOF	Frame Header	Data Field	CRC	EOF
4 Bytes	24 Bytes	0 - 2112 Bytes	4 Bytes	4 Bytes

FC Frame

Fabric Services

- FC switches provide fabric services as defined in FC standards

Fabric Login Server	Name Server	Fabric Controller	Management Server
• Used during the initial part of the node's fabric login process • Located at pre-defined address of FFFFFE	• Responsible for name registration and management of node ports • Located at pre-defined address FFFFFC	• Responsible for managing and distributing Registered State Change Notifications (RSCNs) to attached node ports • Responsible for distributing SW-RSCNs to every other switch • SW-RSCNs keep the name server up-to-date on all switches • Located at pre-defined address FFFFFD	• Enables FC SAN management using fabric management software • Located at pre-defined address FFFFFA

Login Types in Switched Fabric

- Fabric login (FLOGI)
  - Occurs between an N_Port and an F_Port
  - Node sends a FLOGI frame with WWN to Fabric Login Server on switch
  - Node obtains FC address from switch
  - Immediately after FLOGI, N_Port registers with Name Server on switch, indicating its WWN, port type, assigned FC address, etc.
  - N_Port queries name server about all other logged in ports
- Port login (PLOGI)
  - Occurs between two N_Ports to establish a session
  - Exchange service parameters relevant to the session
- Process login (PRLI)
  - Occurs between two N_Ports to exchange ULP related parameters

MODULE – 6

IP SAN

Module 6: IP SAN

Upon completion of this module, you should be able to:

- Describe IP SAN protocols, components, and topology

Module 6: IP SAN and FCoE

Lesson 1: IP SAN

During this lesson the following topics are covered:

- Drivers for IP SAN
- IP SAN Protocols: iSCSI and FCIP
- Components, topologies, and protocol stack for iSCSI and FCIP

Drivers for IP SAN

- Traditional SAN enables the transfer of block I/O over FibreChannel and provides **high performance** and **scalability**
  - These advantages of FC SAN come with the **additional cost** of buying FC components, such as FC HBA and switches.
- Organizations typically have an **existing** Internet Protocol (**IP**)-based infrastructure, which could be **leveraged** (use to maximum advantage) for storage networking.
  - Advancements in technology have enabled IP to be used for transporting block I/O over the IP network.

Drivers for IP SAN (contd.)

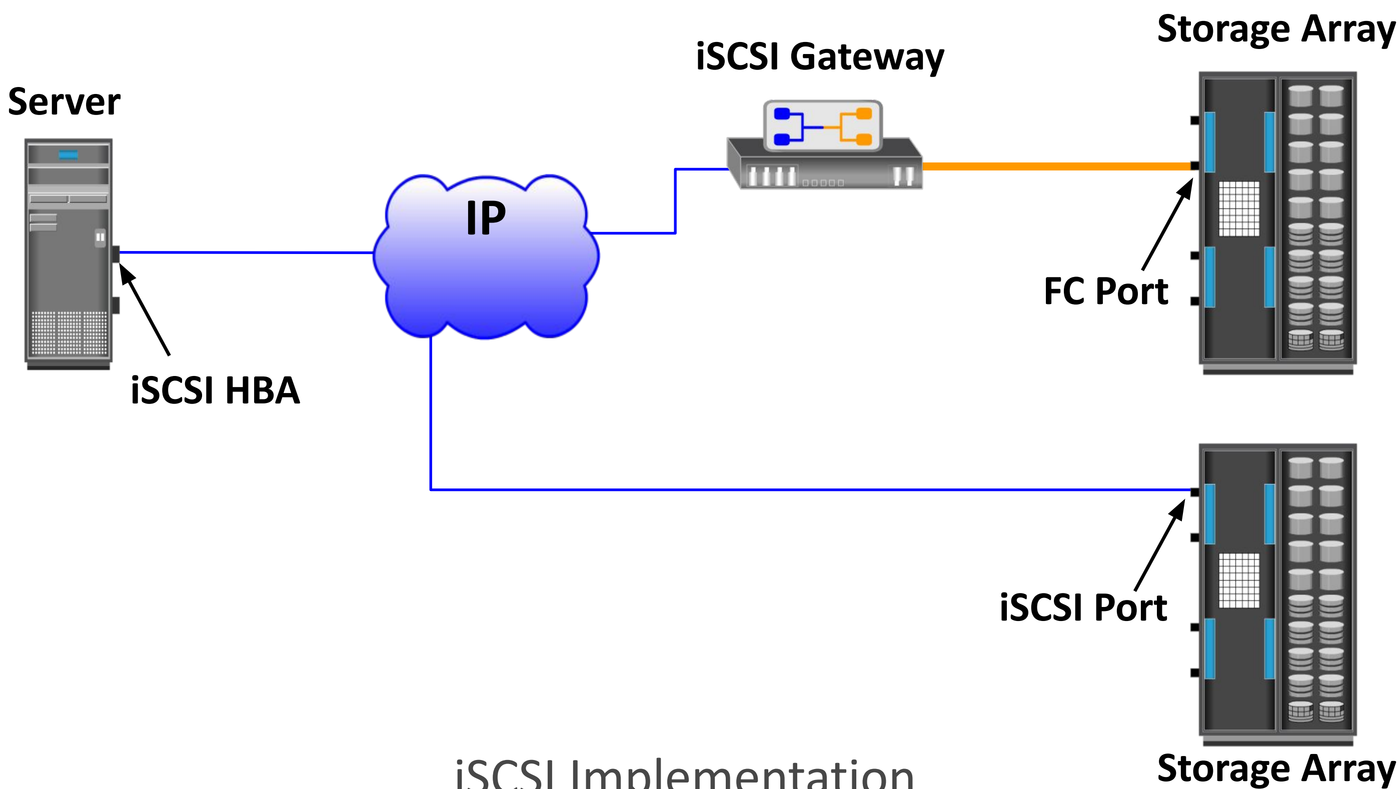
- IP SAN transports block-level data over IP network
- IP is being positioned as a storage networking option because:
  - Existing network infrastructure can be leveraged
  - Reduced cost compared to investing in new FC SAN hardware and software
  - Many long-distance disaster recovery solutions already leverage IP-based network
  - Many robust and mature security options are available for IP network

IP SAN Protocol: iSCSI

- IP based protocol that is used to connect host and storage
  - A **transport layer protocol** that describes how Small Computer System Interface **(SCSI) packets** should be transported over a **TCP/IP** network.
- Carries block-level data over IP-based network
- Widely adopted for **connecting servers** to storage because it is relatively inexpensive and easy to implement

iSCSI Implementation

iSCSI **encapsulates** SCSI **commands** and data into an IP packet and transports them using TCP/IP



iSCSI Implementation

Components of iSCSI

iSCSI initiator

- Hardware or software residing on a computer that handles communications with an iSCSI array (storage devices)
  - Software-based iSCSI initiator, such as Microsoft's free iSCSI initiator - uses the existing network card (NIC)
  - Hardware-based iSCSI initiator – such as iSCSI HBA (Host Bus Adapter)
    - **Does not consume CPU** resources to handle iSCSI commands

iSCSI target

- The iSCSI-enabled device to which you will be storing data
- Storage array with iSCSI port
- iSCSI gateway – enables communication with FC storage array
- Common deployment scenarios for an iSCSI target include: Storage array, disk drives

IP network

- Interconnected Ethernet switches and/or routers

iSCSI Host Connectivity Options

Standard NIC with software iSCSI initiator

- Simple, least expensive
- NIC provides network interface
- Software initiator provides iSCSI functionality
- Requires host CPU cycles for **iSCSI** and **TCP/IP** processing
  - Encapsulation of SCSI into IP packets & decapsulation are carried out by **host CPU**
  - Additional **overhead** on the host CPU
  - Heavy I/O load -> host CPU become **bottleneck**

TCP Offload Engine (TOE) NIC with software iSCSI initiator

- Moves TCP processing load off the host CPU onto the NIC card
- Software initiator provides iSCSI functionality
- Requires host CPU cycles for **iSCSI** processing

iSCSI HBA

- Offloads both iSCSI and TCP/IP processing from host CPU
- Simplest option for boot from SAN via iSCSI

iSCSI Topologies

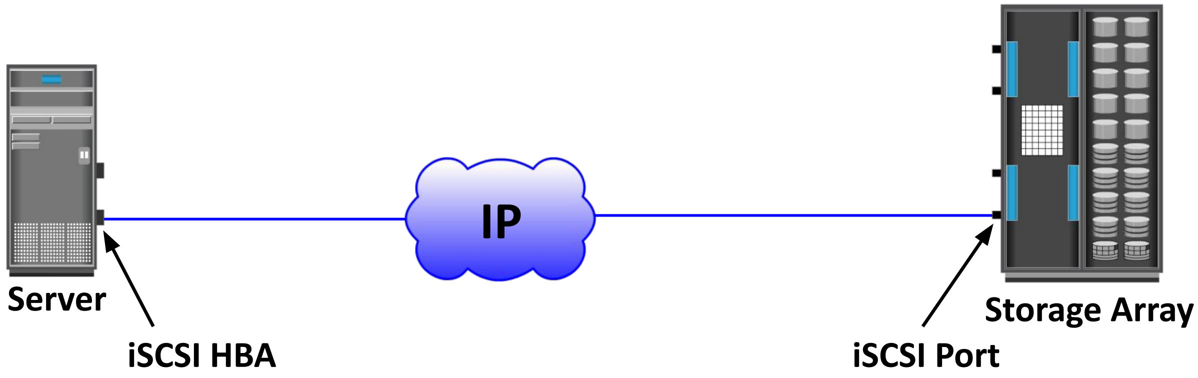
(1) Native iSCSI Connectivity

(2) Bridged iSCSI Connectivity

(3) Combining FCP and Native iSCSI Connectivity

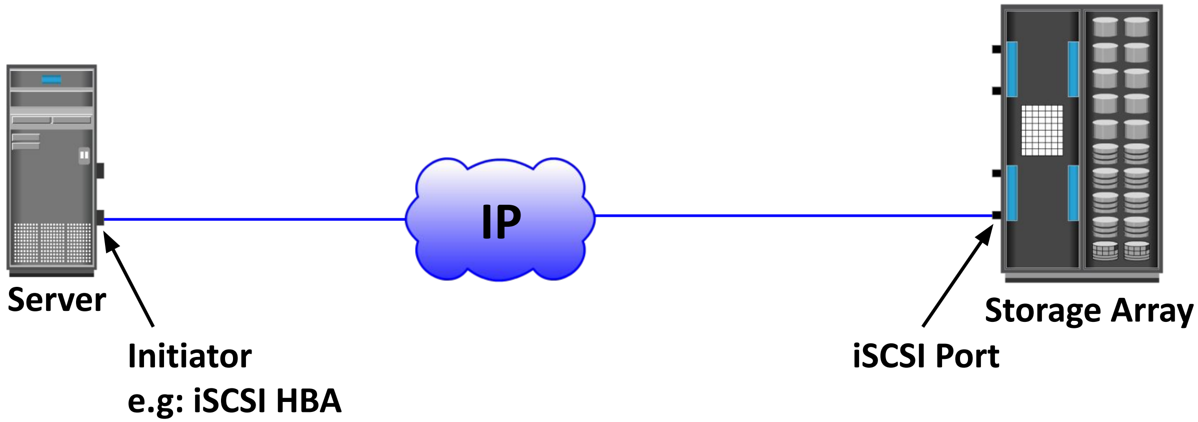
iSCSI Topologies: (1) Native iSCSI

- iSCSI initiators are either **directly attached to storage array** or **connected through IP network**
  - No FC component
- Storage array has iSCSI port
- Each iSCSI port on the array is configured with an IP address and port number.



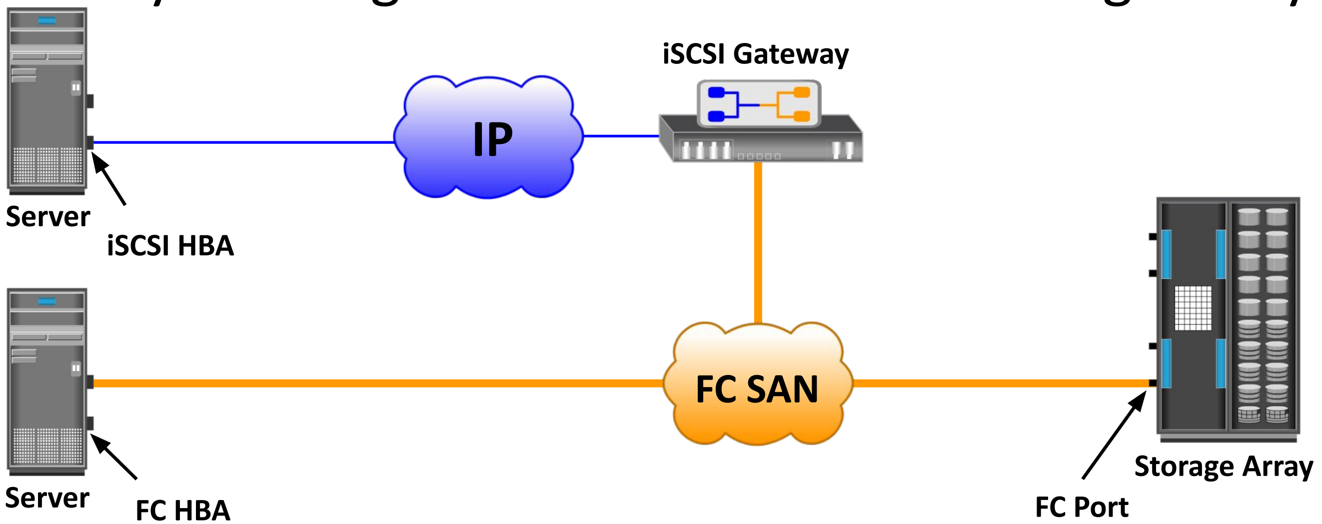
iSCSI Topologies: (1) Native iSCSI (contd.)

- The array has one or more **iSCSI ports** configured with an IP address and connected to a standard Ethernet switch.
- After an **initiator** is logged on to the network, it can access the available LUNs on the storage array.
- A single array port can service multiple hosts or initiators as long as the array port can handle the amount of storage traffic that the hosts generate.



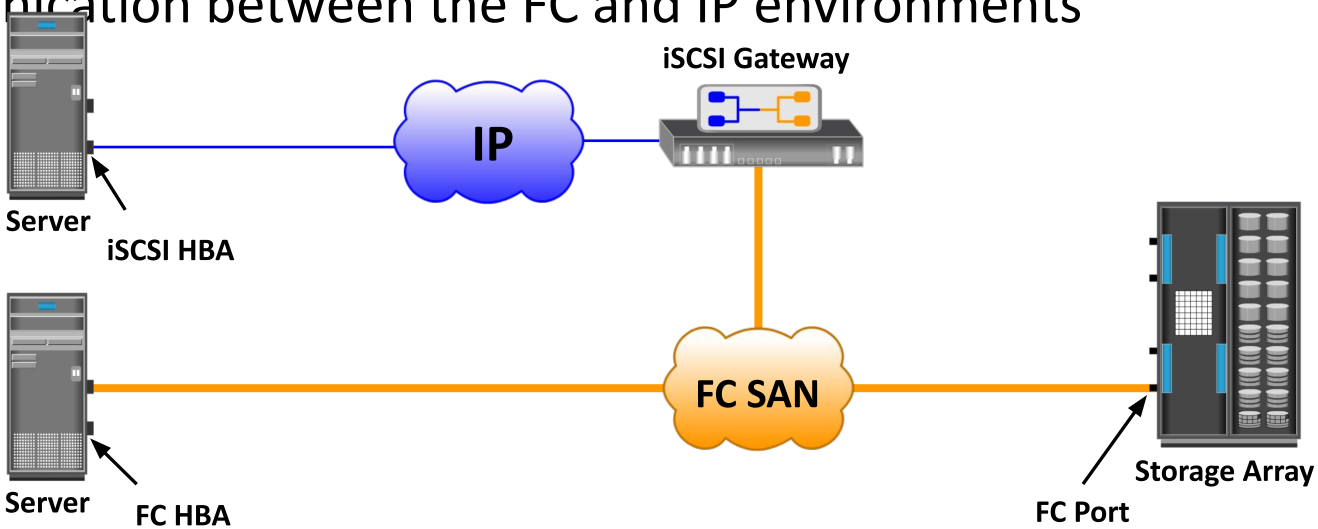
iSCSI Topologies: (2) Bridged iSCSI

- iSCSI gateway is used to enable communication between iSCSI host and FC storage
- iSCSI gateway works as **bridge** between FC and IP network
  - Converts IP packets to FC frames and vice versa
- iSCSI initiator is configured with gateway's IP address as its target
- iSCSI gateway is configured as FC initiator to storage array



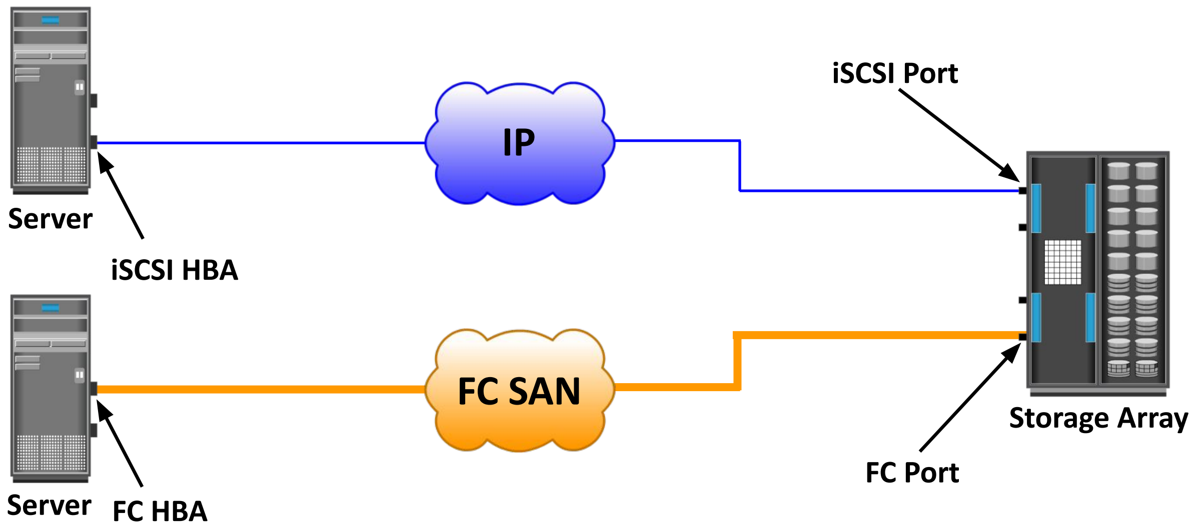
iSCSI Topologies: (2) Bridged iSCSI (contd.)

- In the figure illustrates an iSCSI host connectivity to an FC storage array
- In this case, the array **does not have** any iSCSI ports
- An external device, called a gateway or a multiprotocol router, must be used to **facilitate the communication** between the iSCSI host and FC storage
- Bridge devices contain **both FC and Ethernet ports** to facilitate the communication between the FC and IP environments



iSCSI Topologies: (3) Combining FC and Native iSCSI Connectivity

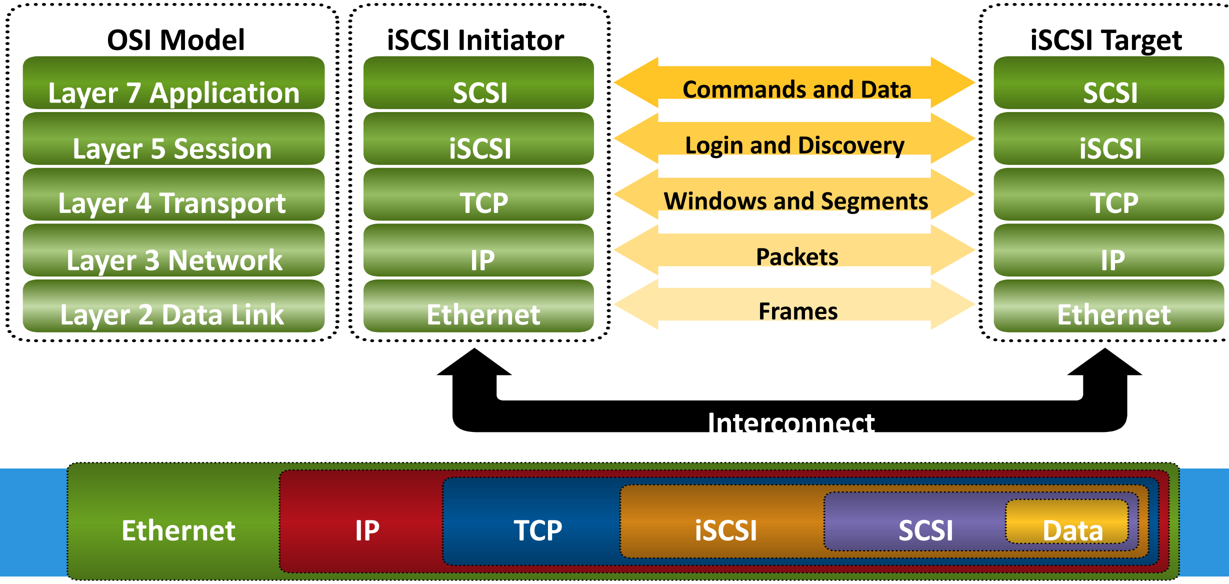
- Array provides both FC and iSCSI ports
  - Enable iSCSI and FC connectivity in the same environment
  - No bridge devices needed



iSCSI Protocol Stack

- SCSI is the command protocol that works at the **application layer** of the Open System Interconnection (OSI) model
- Initiators and targets use **SCSI commands** and responses to talk to each other
  - Commands, data, and status messages are **encapsulated** into TCP/IP and transmitted across the network between the initiators and targets
- iSCSI is the **session-layer protocol** that initiates a reliable session between devices that recognize SCSI commands and TCP/IP
  - Responsible for **handling** login, authentication, target discovery, and session management

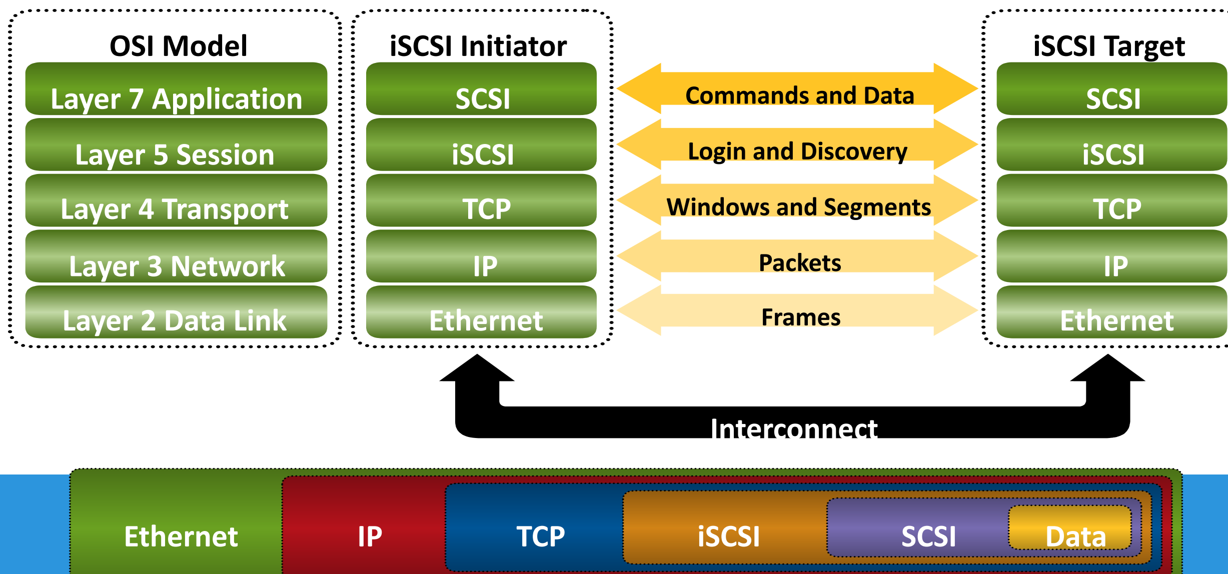
Figure: iSCSI protocol layers and the encapsulation order of the SCSI commands for their delivery through a physical carrier



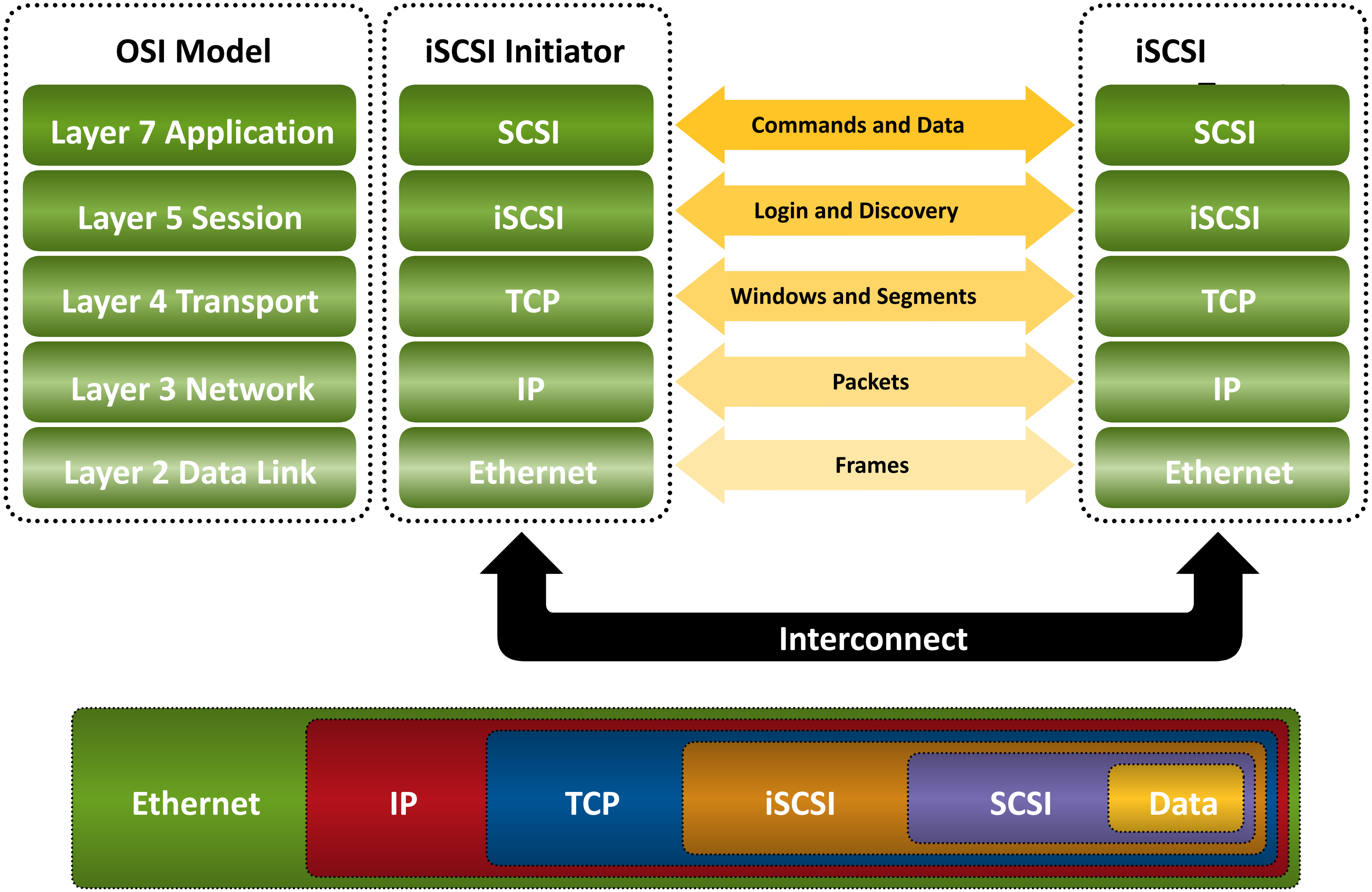
iSCSI Protocol Stack (contd.)

- TCP is used with iSCSI at the **transport layer** to provide reliable transmission
  - TCP controls message flow, windowing, error recovery, and retransmission.
  - Rely upon the **network layer** of the OSI model to provide global addressing and connectivity.
- The Layer 2 protocols at the **data link layer** of this model enable node-to-node communication through a physical network.

Figure: iSCSI protocol layers and the encapsulation order of the SCSI commands for their delivery through a physical carrier



iSCSI Protocol Stack (contd.)



iSCSI Discovery

- For iSCSI communication, initiator must discover **location** and **name** of **target** on a network
- iSCSI discovery takes place in two ways:

SendTargets discovery	Internet Storage Name Service (iSNS)
• Initiator is **manually configured** with the target's network portal • Initiator issues SendTargets command; target responds with required parameters	• Initiators and targets **register** themselves with iSNS server • Initiator can query iSNS server for a list of available targets

iSCSI Name

- iSCSI name is a **unique iSCSI identifier** that is used to identify initiators and targets within an iSCSI network
- Two common types of iSCSI names are:
  - IQN: iSCSI Qualified Name
    - To use IQN, the company must own a registered domain name
    - A date is included in the name to avoid potential conflicts caused by the transfer of domain names
    - iqn.2008-02.com.example:optional_string
      - Example: iqn.1992-05.com.emc:apm000339013630000-10
    - The optional_string can be the name of the host, serial number, asset number or any other storage identifier
  - EUI: Extended Unique Identifier
    - Use the WWN (World Wide Name)
    - eui.0300732A32598D26

MODULE – 7

NETWORK-ATTACHED STORAGE (NAS)

Module 7: Network-Attached Storage (NAS)

Upon completion of this module, you should be able to:

- Describe NAS, its benefits, and components
- Discuss NAS file-sharing protocols
- Describe different NAS implementations
- Describe file-level virtualization

Module 7: Network-Attached Storage (NAS)

Lesson 1: NAS Components and Benefits

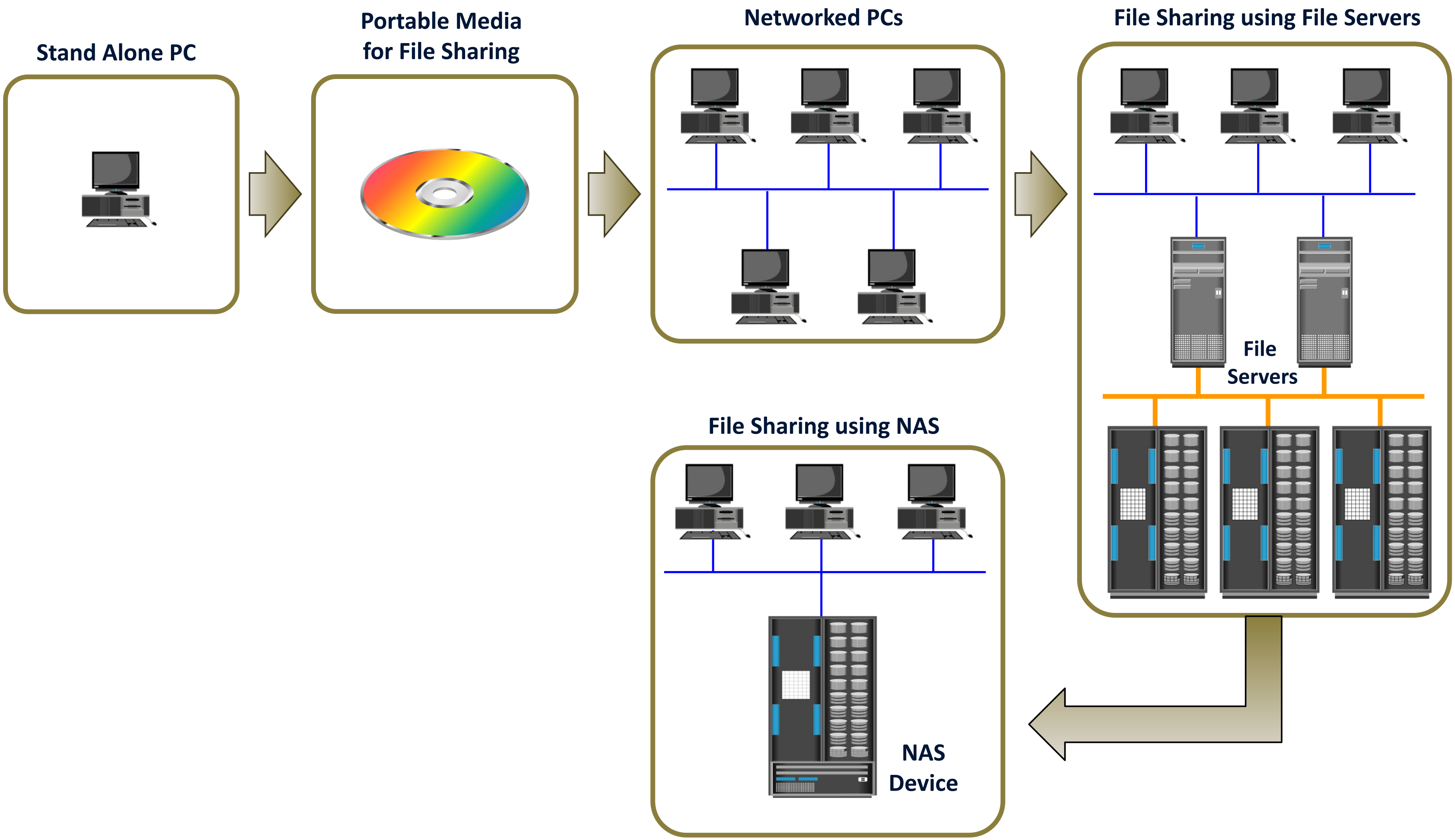
During this lesson the following topics are covered:

- File sharing technology evolution
- Benefits of NAS
- NAS components
- NAS file sharing protocols
- NAS I/O operations

File Sharing Environment

- File sharing:
  - Traditional method: copy files to portable media and deliver to other users
    - Not suitable for large no of users – different locations
  - Network-based file sharing: Storing and accessing data files over network
    - Enables users to share files with other users
    - Creator or owner of a file determines the type of access to be given to other users
    - Ensures data integrity when multiple users access a shared file at the same time
  - Examples of file sharing methods:
    - File Transfer Protocol (FTP), Distributed File System (DFS), Network File System (NFS) and Common Internet File System (CIFS), Peer-to-Peer (P2P)

File Sharing Technology Evolution

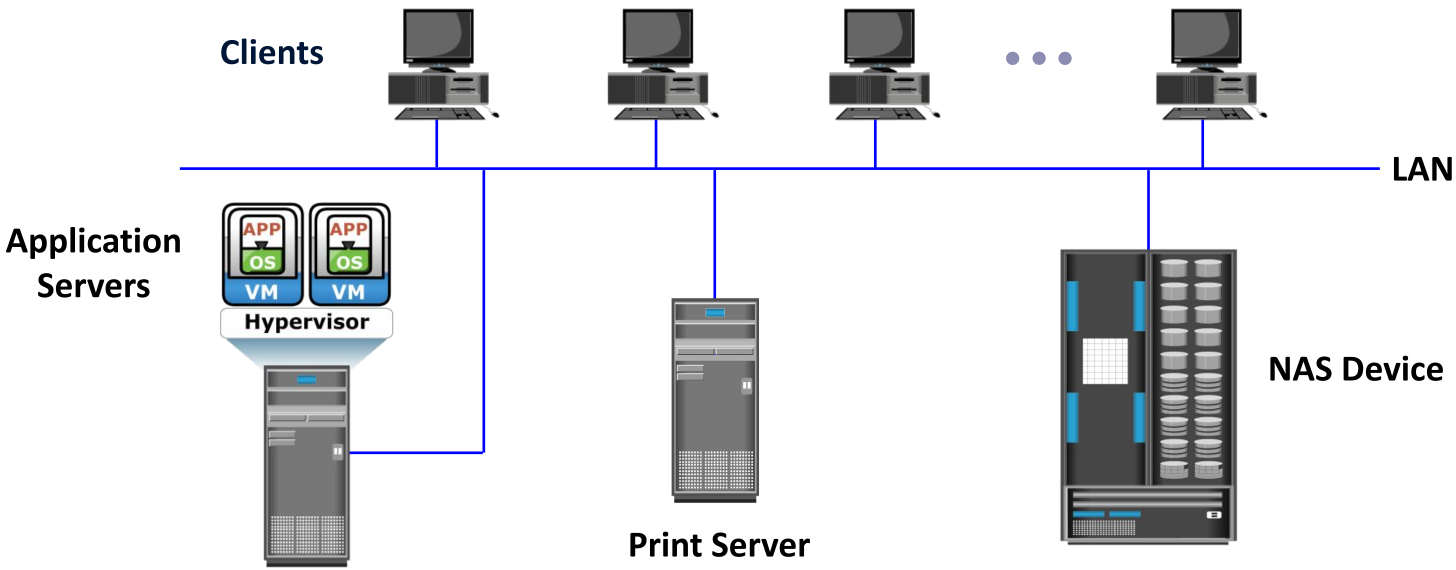


What is Network-Attached Storage (NAS)?

NAS

It is an IP-based, dedicated, high-performance file sharing and storage device.

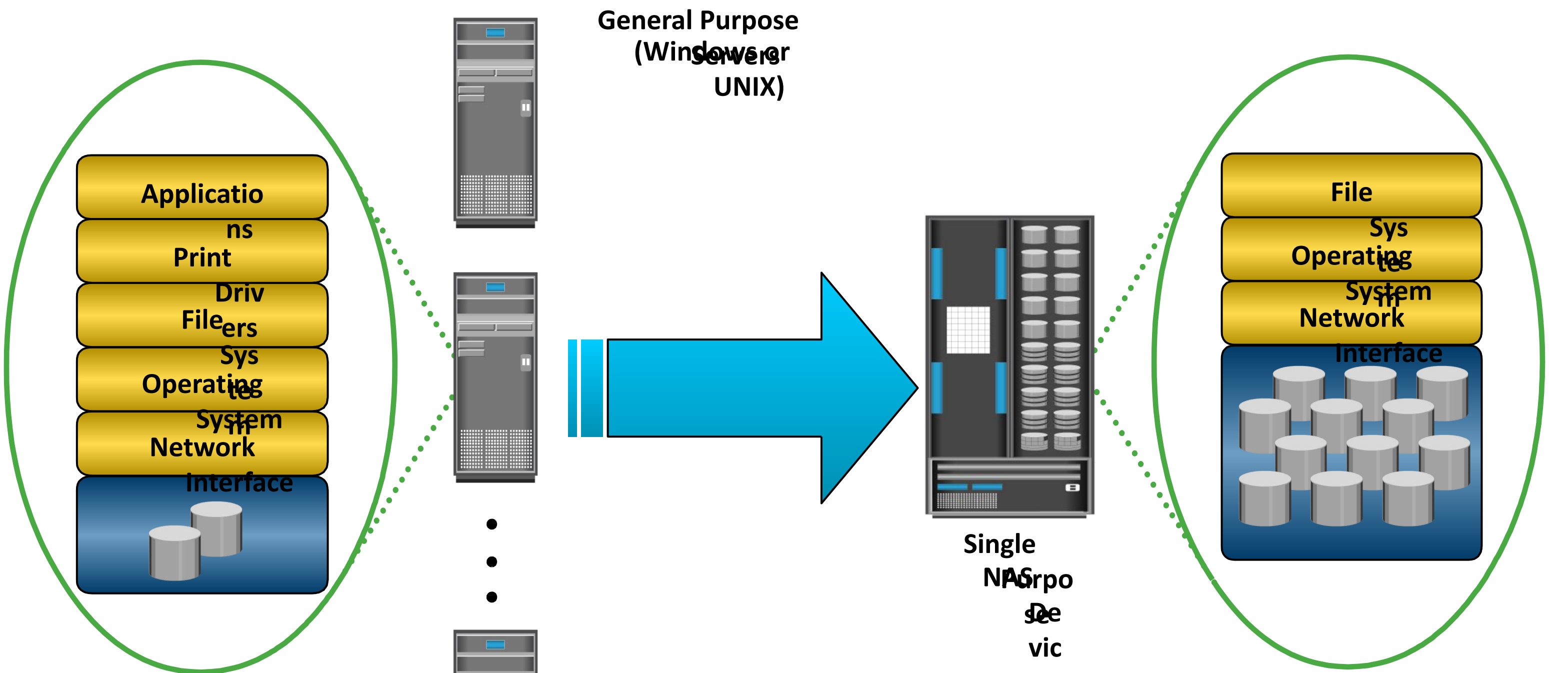
- Enables NAS clients to **share files** over **IP network**
- Uses specialized **operating system** that is **optimized for file I/O**
- Enables both UNIX and Windows users to **share data**



NAS Device

- A dedicated, high-performance, high-speed, single-purpose file serving and storage system.
  - Serves a **mix of clients** and **servers** over an IP network
  - Support **multiple interfaces** and **network**
- Uses its own operating system and integrated hardware/software components to meet specific file service needs.
  - The operating system is **optimized for I/O**
  - Performs file I/O **better** than a general purpose server
    - Able to serve **more clients** than traditional file servers

General Purpose Servers Vs. NAS Devices



- General-purpose server can be used **to host any application**, as it runs a generic operating system.

- A NAS device is **dedicated** for file-serving
- Uses **real-time OS** dedicated for file-serving purpose

Benefits of NAS

1 Comprehensive access to information

Enables efficient file sharing and supports many-to-one and one-to-many configurations.

2 Improved efficiency

Eliminates bottlenecks because NAS uses an operating system specialized for file serving.

3 Improved flexibility

- Platform independent
- Flexible to serve requests from different types of clients.

4 Centralized storage

Centralizes data storage to minimize data duplication on client workstations, simplify data management, and ensures greater data protection.

5 Simplified management

Provides a centralized console that makes it possible to manage file systems efficiently

Benefits of NAS (contd.)

6 Scalability

Scales well with different utilization

7 High availability

- Replication and recovery options
- Use redundant components maximum connectivity
- Clustering technology for failover

8 Security

Authentication, authorization, and file locking in conjunction with industry-standard security

9 Low cost

Uses commonly available and inexpensive Ethernet components

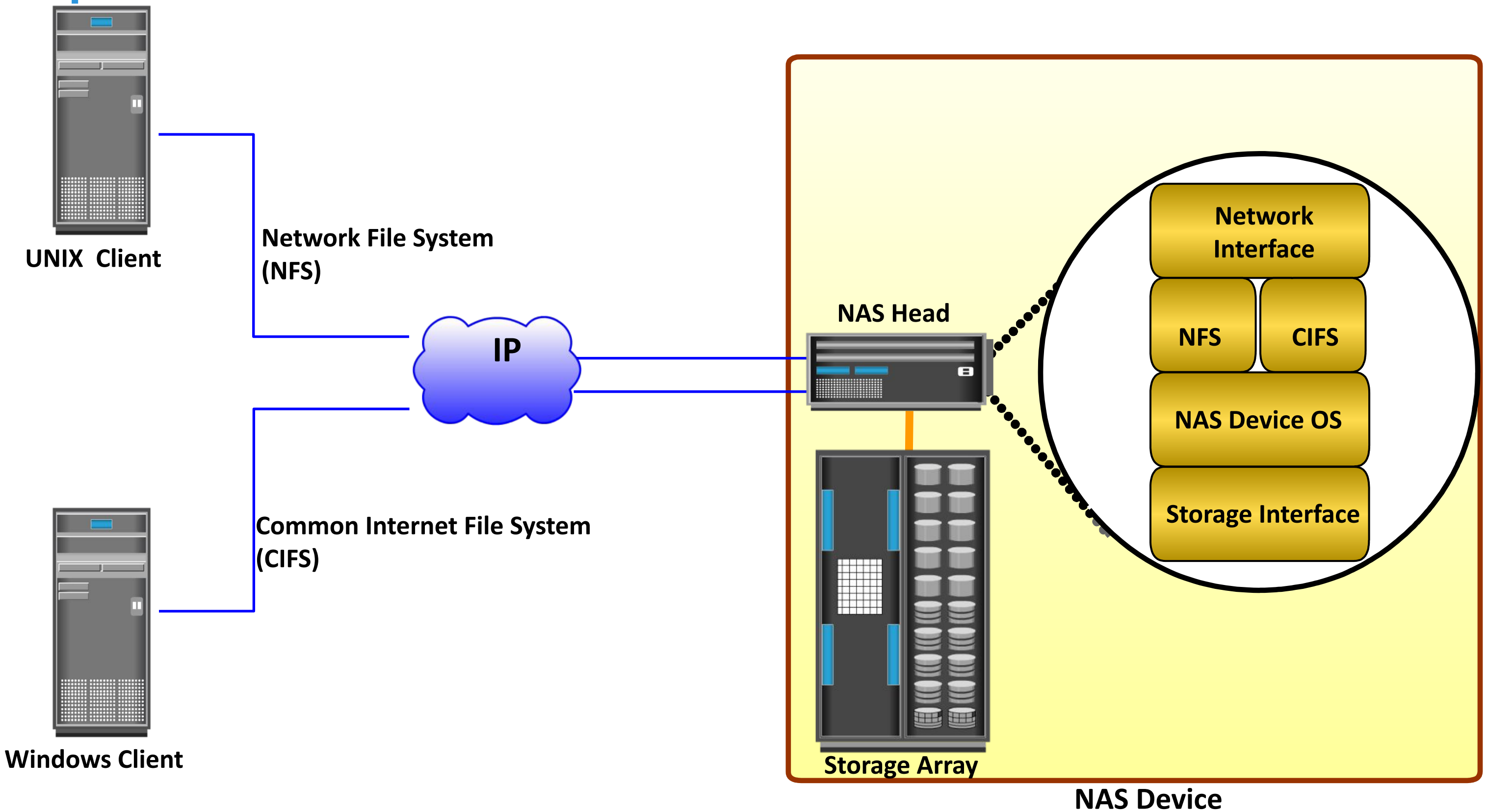
10 Ease of deployment

Client configuration is minimal because clients have required NAS connection built in

SAN vs. NAS

	SAN	NAS
File Access	Provides block-level access (provides **only storage**)	Provides file-level access (provides both **storage** and **file system**)
Operating System View	Appears as a **drive** that can be managed by a volume management utility (formatted, mounted, etc.), similar to a DAS	Appears as a **network drive**

Components of NAS



1. NAS head (CPU and Memory)
2. One or more NICs - provide connectivity to the network
3. An optimized operating system for managing NAS functionality
4. NFS and CIFS protocols for file sharing
5. Industry-standard storage protocols to connect and manage physical disk resources, such as ATA, SCSI, or FC

NAS File Sharing Protocols

- Two common NAS file sharing protocols are:

Common Internet File System (CIFS)

- Traditional Microsoft environment file sharing protocol, based upon the Server Message Block protocol

Network File System (NFS)

- Traditional UNIX environment file sharing protocol

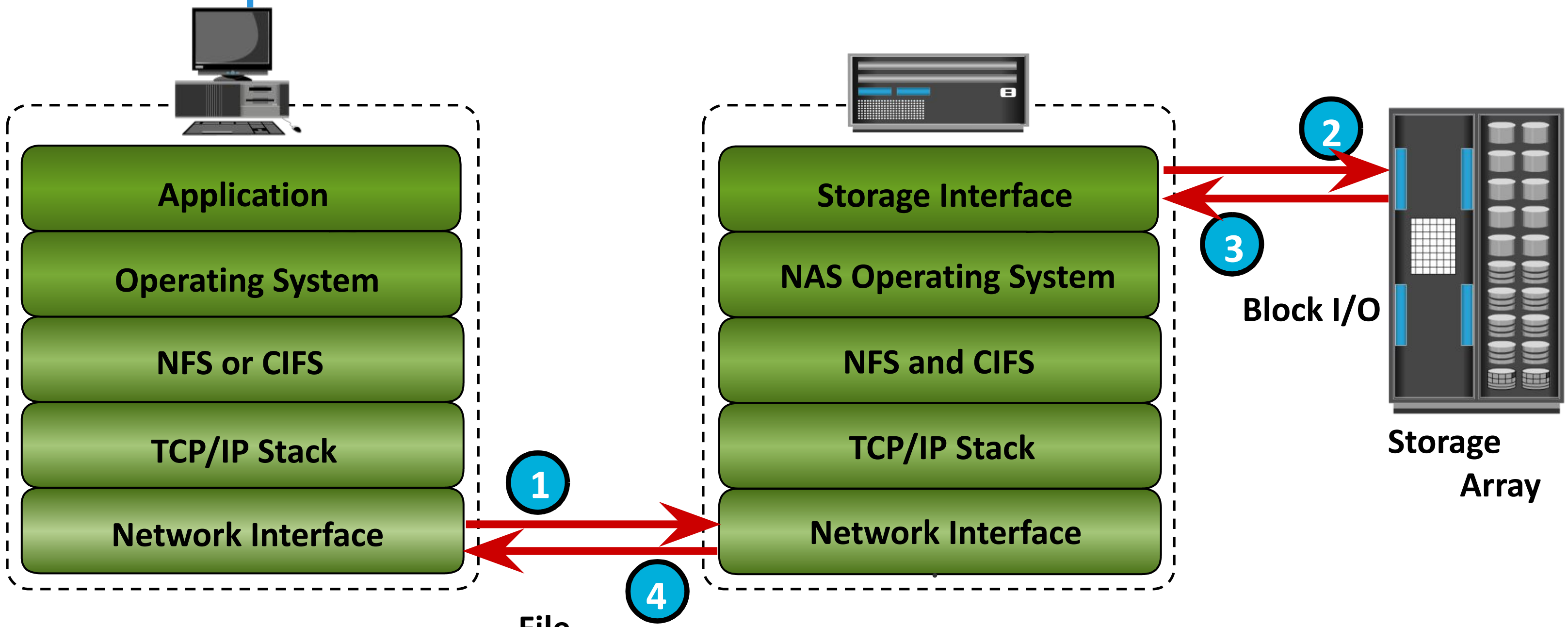
Common Internet File System (CIFS)

- Client-server application protocol that enables clients to access files that are on a server over TCP/IP
  - An open variation of the Server Message Block (SMB) protocol
- Stateful Protocol
  - Maintains connection information regarding every connected client
  - If a network failure or CIPS server failure occurs, client receives a disconnection notification
  - 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 (available to Unix/Linux through Samba)

Network File System (NFS)

- Client-server application protocol that enables clients to access files that are on a server
- Uses Remote Procedure Call (RPC) mechanism to provide access to remote file system
  - Searching files and directories
  - Opening, reading, writing to, and closing a file
  - Changing file attributes
  - Modifying file links and directories
- Currently, 3 versions of NFS are in use:
  - NFS v2 is stateless and uses UDP as transport layer protocol
  - NFS v3 is stateless and uses UDP or optionally TCP as transport layer protocol
  - NFS v4 is stateful and uses TCP as transport layer protocol

NAS I/O Operation



1. The requestor packages an **I/O request** into TCP/IP and forwards it through the network stack. The NAS device **receives** this request from the **network**.
2. The NAS device **converts** the I/O request into an **appropriate** physical storage **request**, which is a block-level I/O, and then performs the operation against the physical storage pool.
3. When the data is **returned** from the physical storage pool, the NAS device **processes** and **repackages** the data into an appropriate file protocol response.
4. The NAS device packages this **response** into TCP/IP again and **forwards** it to the client through the network.

Module 7: Network-Attached Storage (NAS)

Lesson 2: NAS Implementation and File-level Virtualization

During this lesson the following topics are covered:

- NAS implementations
- NAS use cases
- File-level virtualization

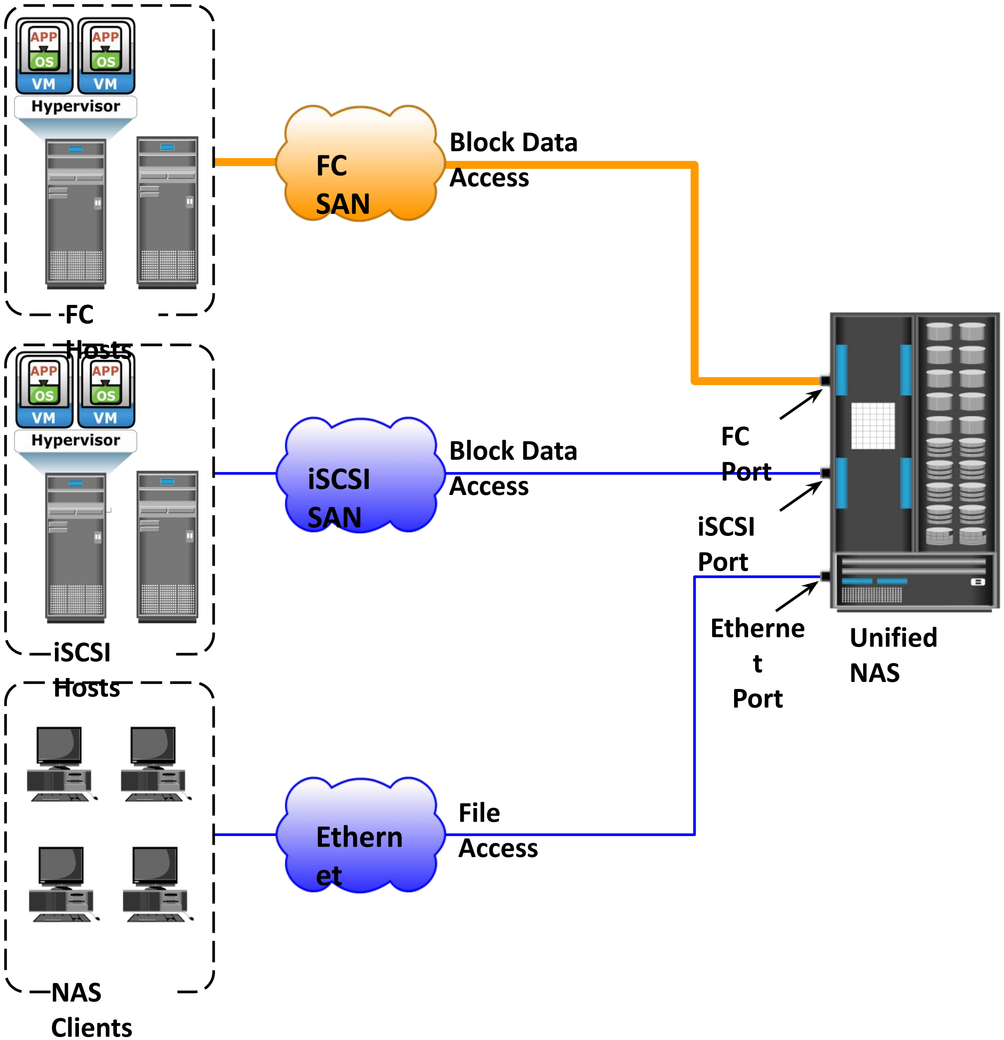
NAS Implementation

- 3 types of NAS implementations:

Unified NAS	Gateway NAS	Scale-out NAS
• Has **all** of its components and storage system in a **single** enclosure or frame	• NAS **head shares** its storage with SAN environment.	• Ideal for enterprise data centers • Consolidating both virtualized and non-virtualized file storage into **one storage pool** with a single point of management

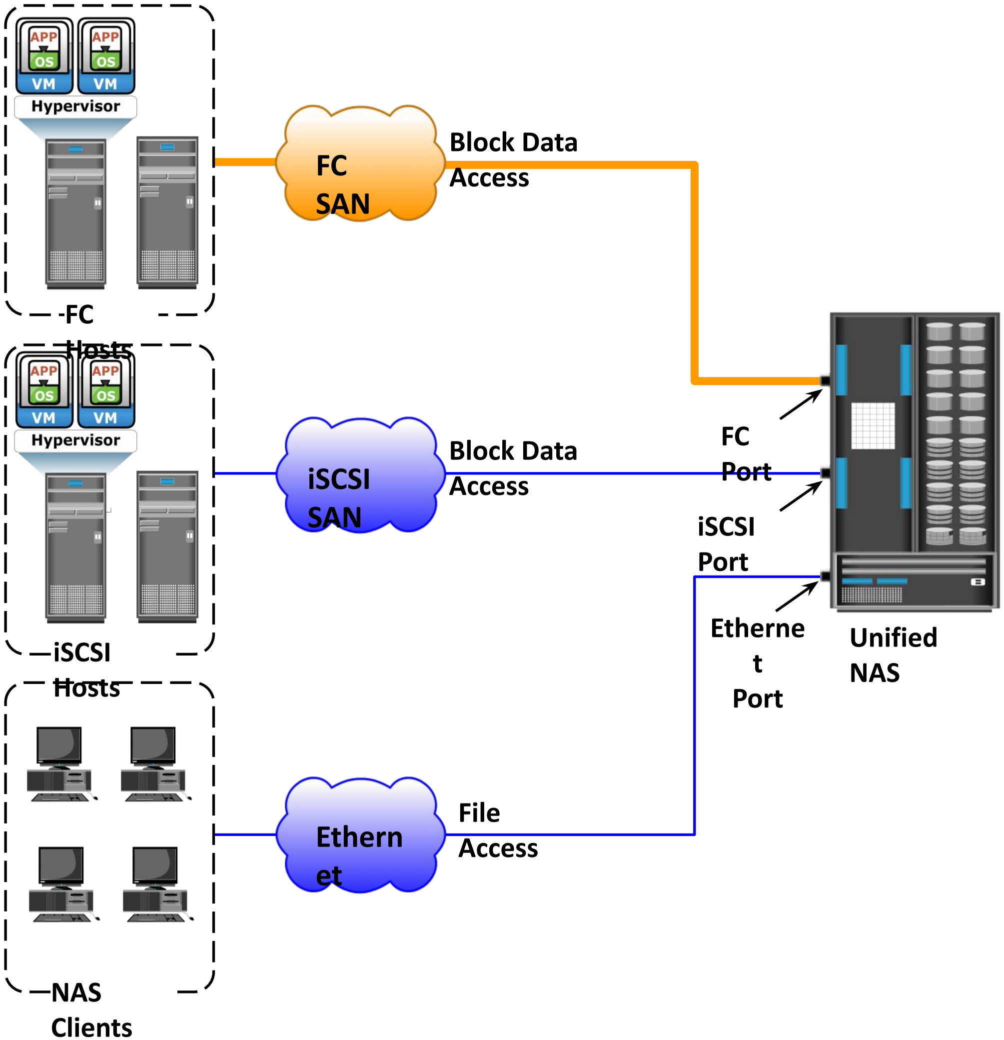
NAS Implementation – Unified NAS

- Consolidates **NAS-based** (file-level) and **SAN-based** (block-level) access on a **single** storage platform
- Supports both CIFS and NFS protocols for file access and iSCSI and FC protocols for block level access
- Provides **unified** management for both NAS head and storage



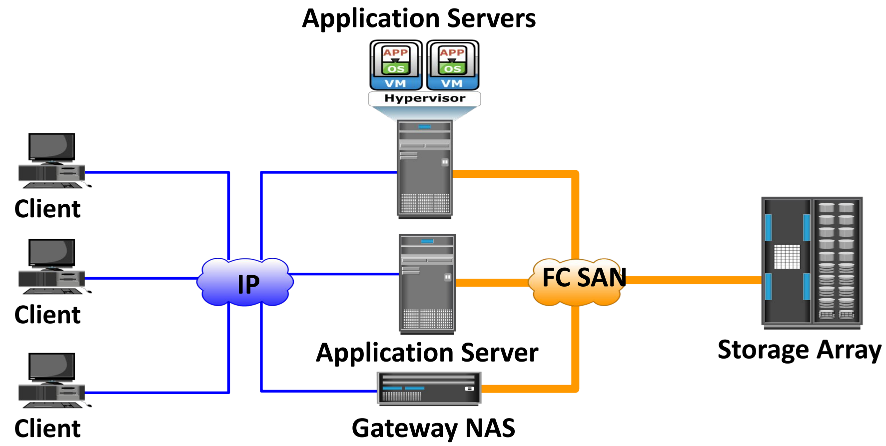
NAS Implementation – Unified NAS Connectivity

- NAS head has **front-end** Ethernet ports which connect to **IP**.
  - Front-end ports: provide connectivity to clients
  - Back-end ports: provide connectivity to storage controllers
- iSCSI and FC ports on a **storage controller**
  - Enable hosts to access the storage directly or through a storage network at the block level.



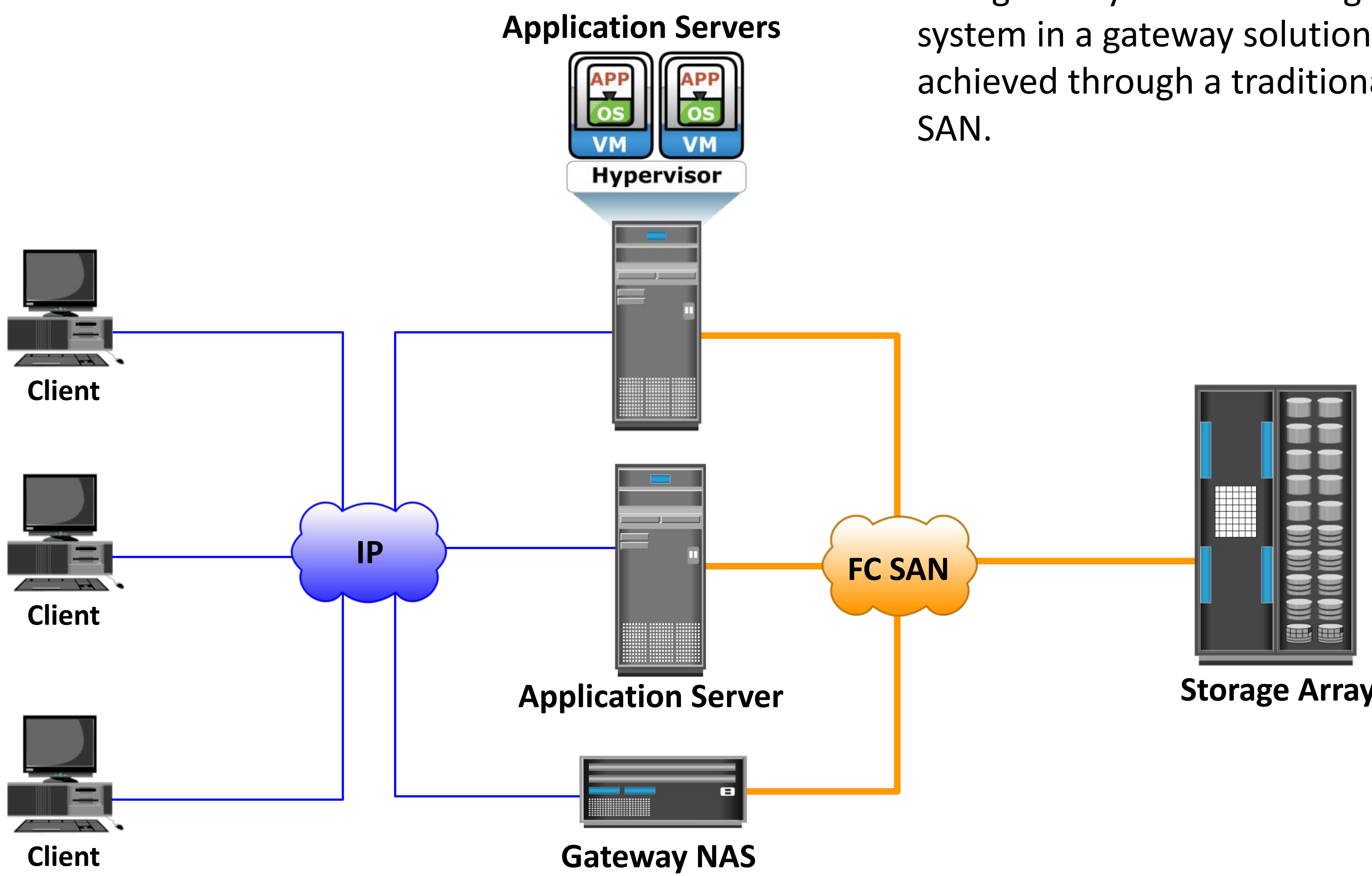
NAS Implementation – Gateway NAS

- Uses external and independently-managed storage
  - NAS heads access SAN-attached or direct-attached storage arrays
- NAS heads **share storage** with other application servers that perform block I/O
- Requires **separate management** of NAS head and storage
- The gateway NAS is the **most scalable** because NAS heads and storage arrays can be **independently scaled up when required**.
- Gateway NAS enables **high utilization** of storage capacity by sharing it with SAN environment.



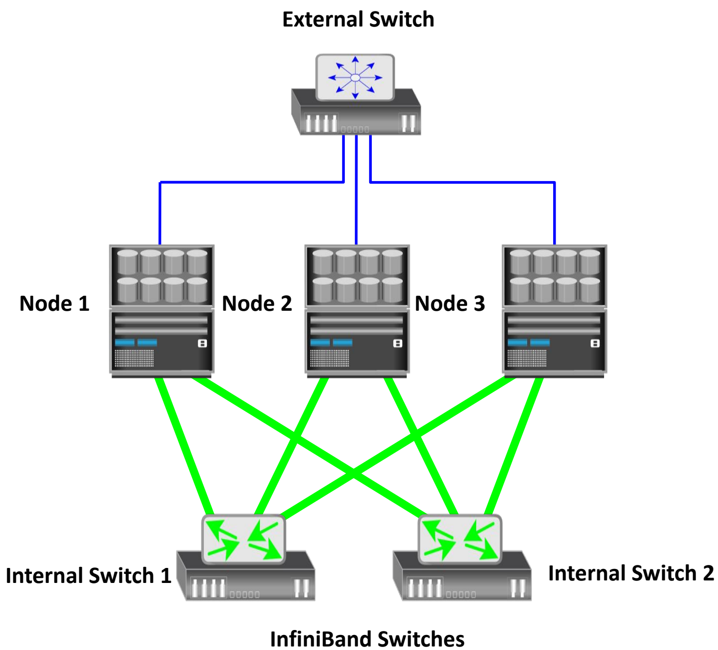
Gateway NAS Connectivity

- Communication between the NAS gateway and the storage system in a gateway solution is achieved through a traditional FC SAN.



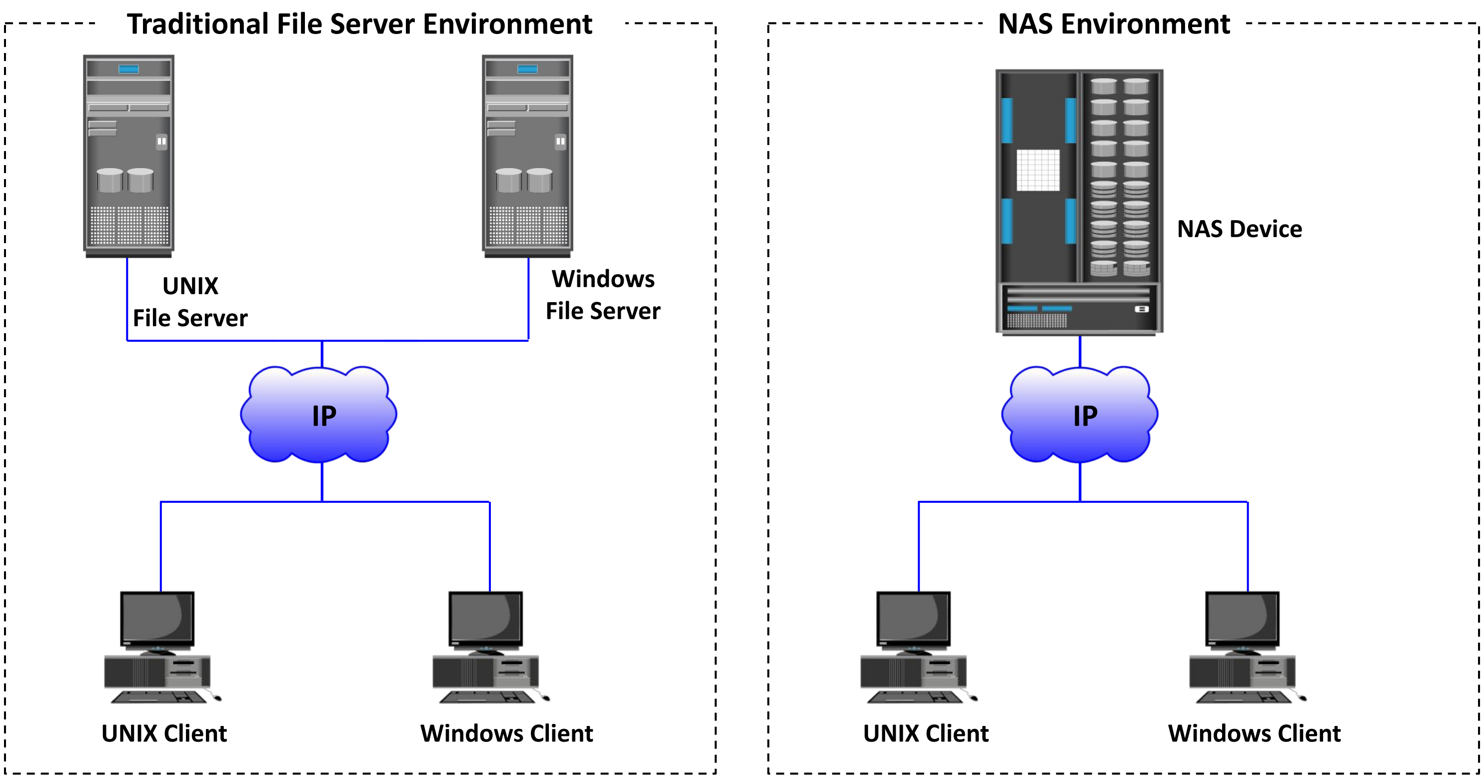
NAS Implementation – Scale-out NAS

- Pools **multiple nodes** together in a **cluster** that works as a single NAS device
  - Pool is managed centrally
- Scales performance and/or capacity with addition of nodes to the pool non-disruptively
- Creates a **single file system** that runs on **all nodes** in the cluster
  - Clients, connected to any node, can access entire file system
  - File system grows dynamically as nodes are added
- **Stripes** data across all nodes in a pool along with **mirror** or **parity** protection



InfiniBand is a networking technology that provides a low-latency, high-bandwidth communication link between hosts and peripherals.

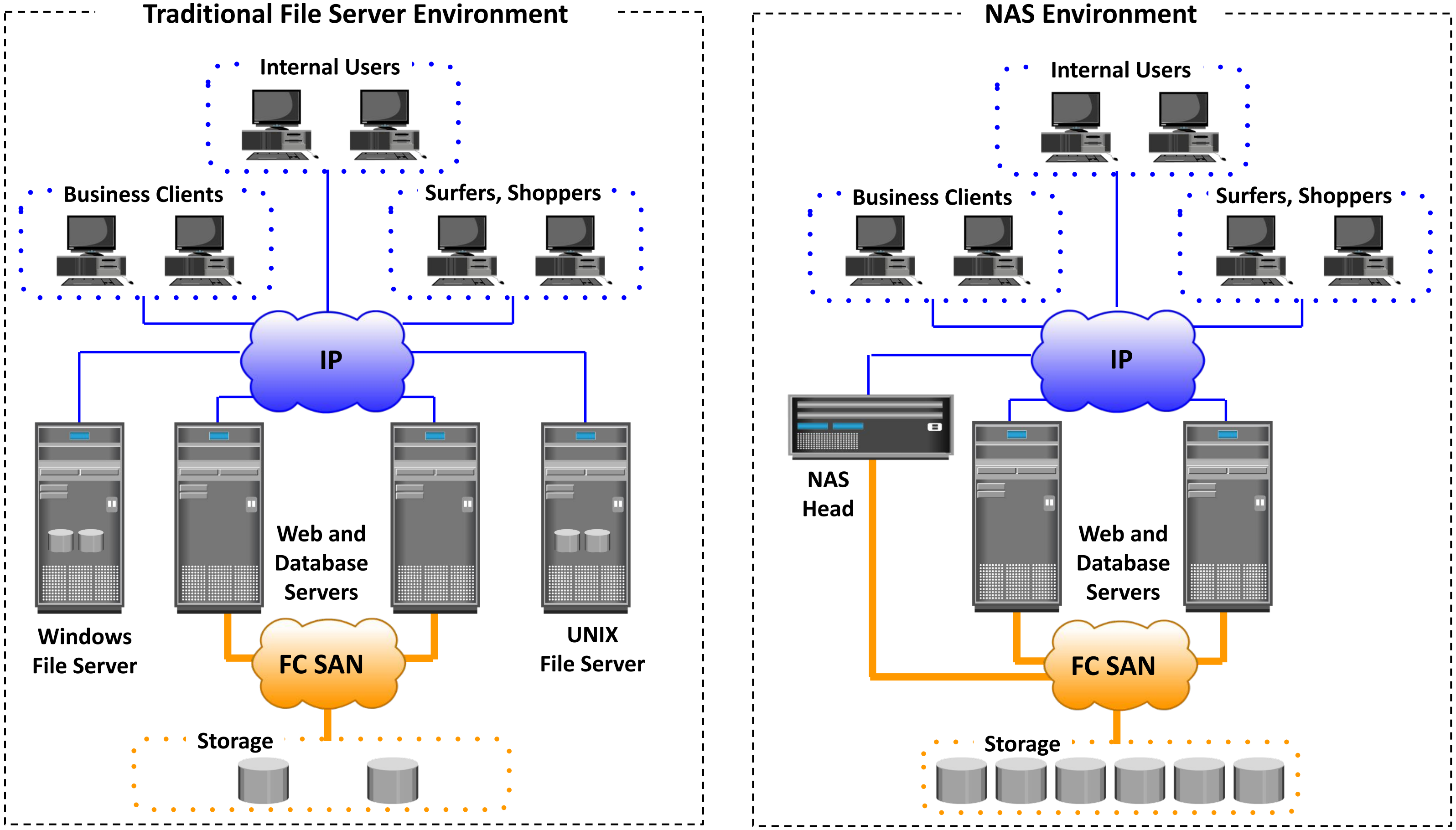
NAS Use Case 1 – Server Consolidation with NAS



Traditionally, network file system for UNIX and Microsoft Windows are housed on separate servers. This requires maintenance of both the environments

Using NAS, both Windows and UNIX file structures can be housed together in a single system, while still maintaining their integrity

NAS Use Case 2 – Storage Consolidation with NAS



Storage resources in a traditional file server environment can be consolidated using NAS

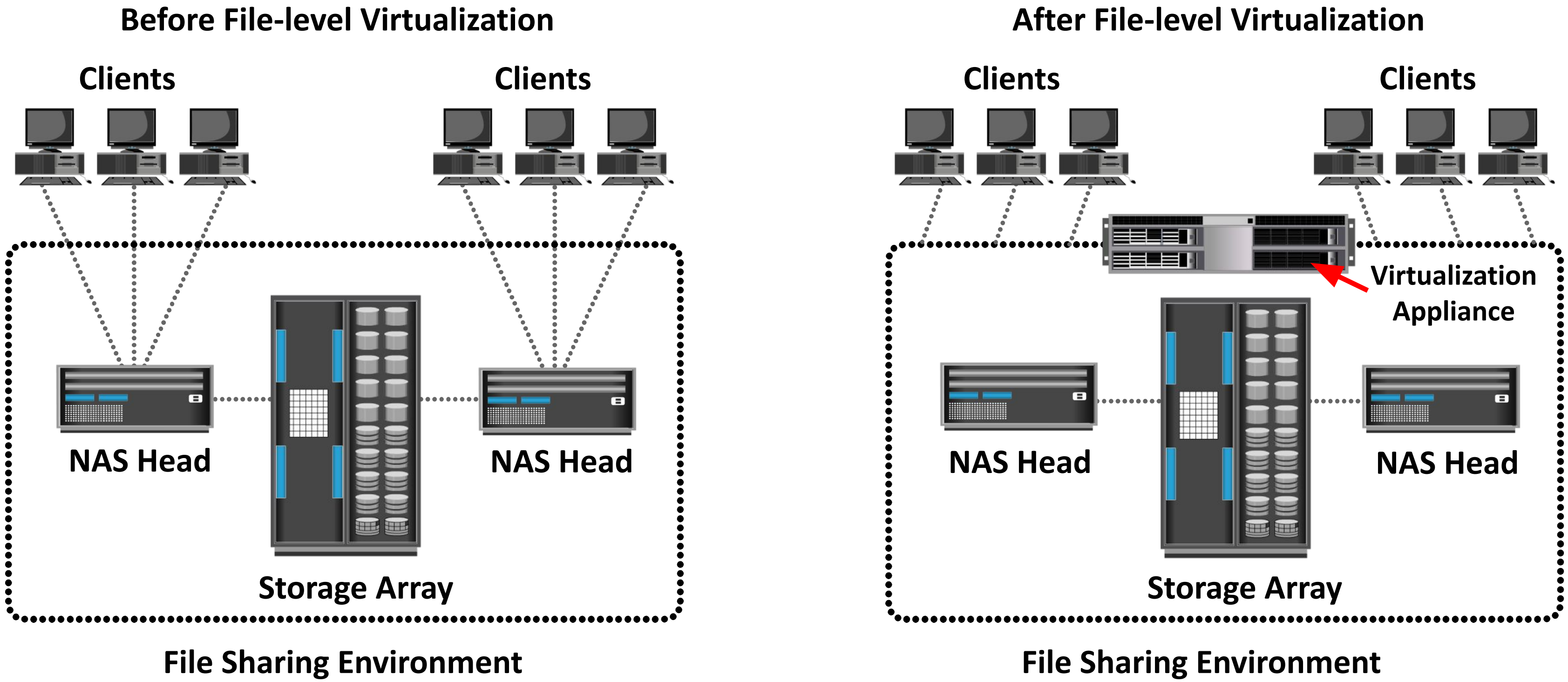
File-level Virtualization

- Implemented in NAS or the file server environment
- **Eliminates dependency** between data accessed at the file-level and the location where the files are physically stored
- Enables users to use a **logical path**, rather than a physical path, to access files
- Uses global namespace that **maps logical path** of file resources to their physical path

File-level Virtualization (contd.)

- Provides **non-disruptive file mobility** across file servers or NAS devices
  - Provides user or application independence from the location where the files are stored.
- File-level virtualization **facilitates the movement** of files across the online file servers or NAS devices.
  - While the files are being moved, clients can access their files non-disruptively.
  - Clients can also read their files from the old location and write them back to the new location without realizing that the physical location has changed.

Comparison: Before and After File-level Virtualization



- Dependency between client access and file location
- Underutilized storage resources
- Downtime is caused by data migrations

- Break dependencies between client access and file location
- Storage utilization is optimized
- Non-disruptive migrations

MODULE – 8

OBJECT-BASED AND UNIFIED STORAGE

Module 8: Object-based and Unified Storage

Upon completion of this module, you should be able to:

- Describe the object-based storage model
- List the key components of object-based storage
- Describe the storage and retrieval process in object-based storage
- Describe content-addressed storage
- List the key components of unified storage
- Describe the process of data access from unified storage

Module 8: Object-based and Unified Storage

Lesson 1: Object-based Storage

During this lesson the following topics are covered:

- Comparison of hierarchical file system and flat address space
- Object-based storage model
- Key components of object-based storage
- Storage and retrieval process in object-based storage devices
- Content-addressed storage

Drivers for Object-based Storage

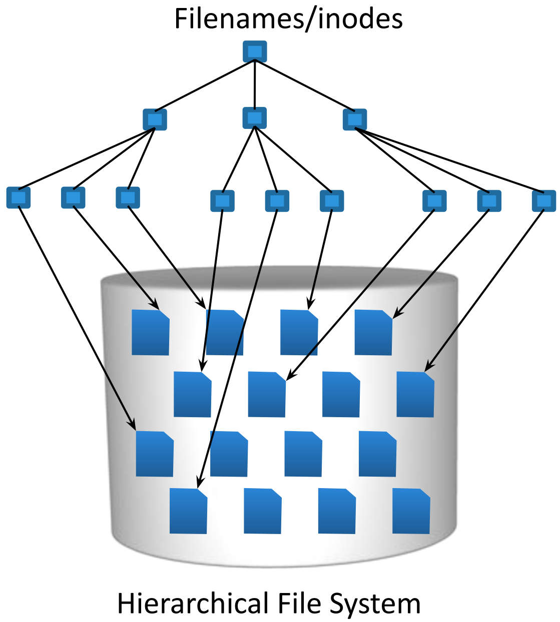
- More than 90% of the data being generated is unstructured
- Traditional solutions are inefficient to handle the growth
  - High overhead on NAS due to managing large number of permissions and nested directories
- These challenges demanded a smarter approach to manage unstructured data based on its content rather than metadata about its name, location, etc.

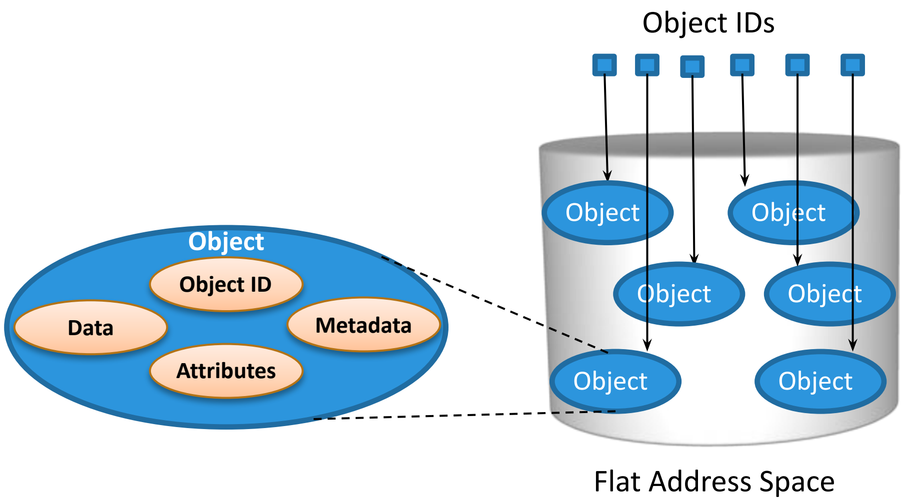
*Object-based storage is a way to store file data in the form of objects on **flat address space** based on its **content** and **attributes** rather than the name and location*

Object-based Storage Device (OSD)

- A device that **organizes** and **stores unstructured data**, such as movies, office documents, and graphics, as objects
- Object-based storage provides a scalable, self-managed, protected, and shared storage option.

Hierarchical File System Vs. Flat Address Space





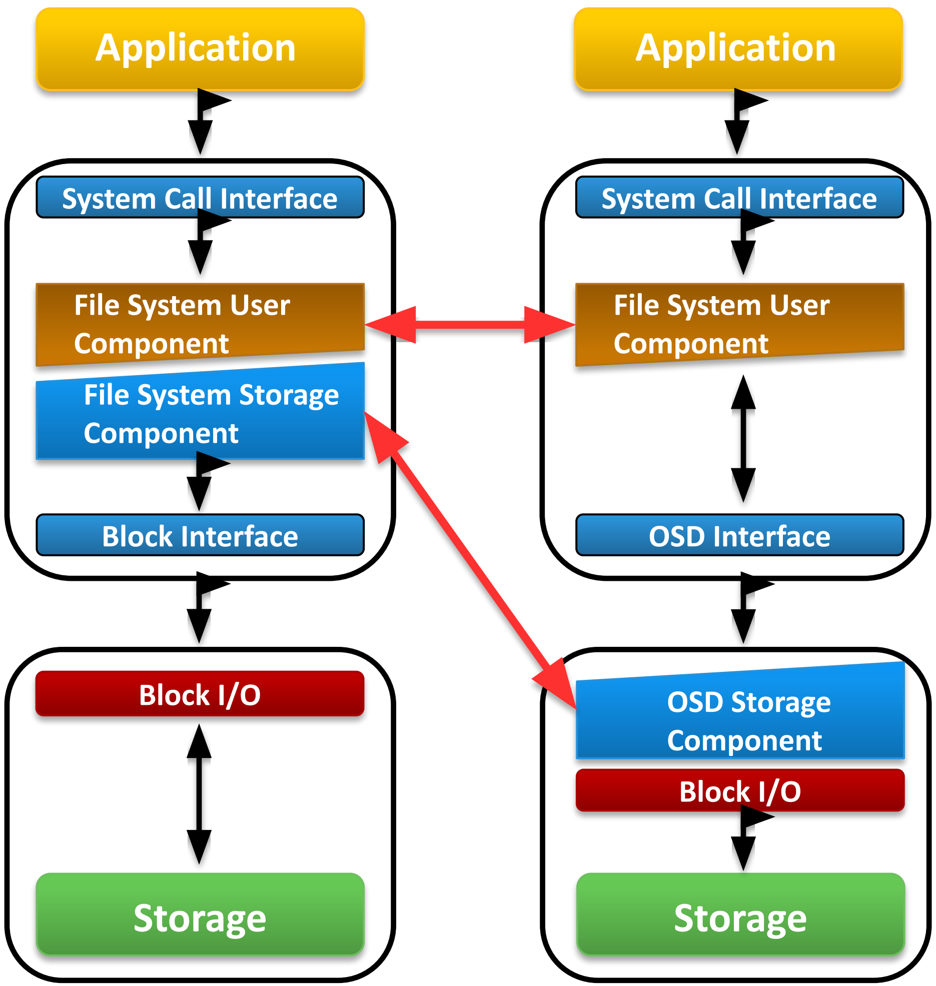
- Hierarchical file system organizes data in the form of files and directories

- Object-based storage devices store the data in the form of objects
  - It uses **flat address space** that enables storage of large number of objects
  - An object contains user data, related metadata, and other attributes
  - Each object has a unique object ID, generated using specialized algorithm

Traditional Vs. Object-based Storage Model

An I/O in the traditional block access method passes through **various layers** in the I/O path.

The I/O generated by an application **passes through** the file system, the channel, or network and reaches the disk drive.



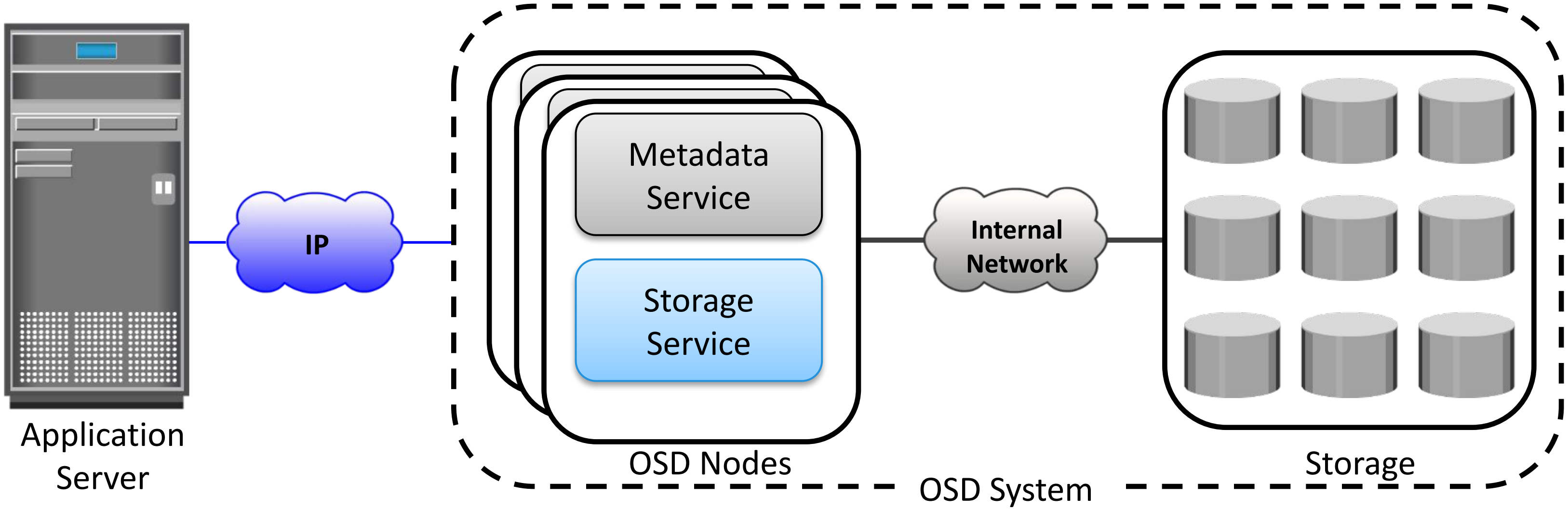
Block-Level Access

Object-Level Access

When an application accesses data stored in OSD, the request is sent to the file system **user component**.

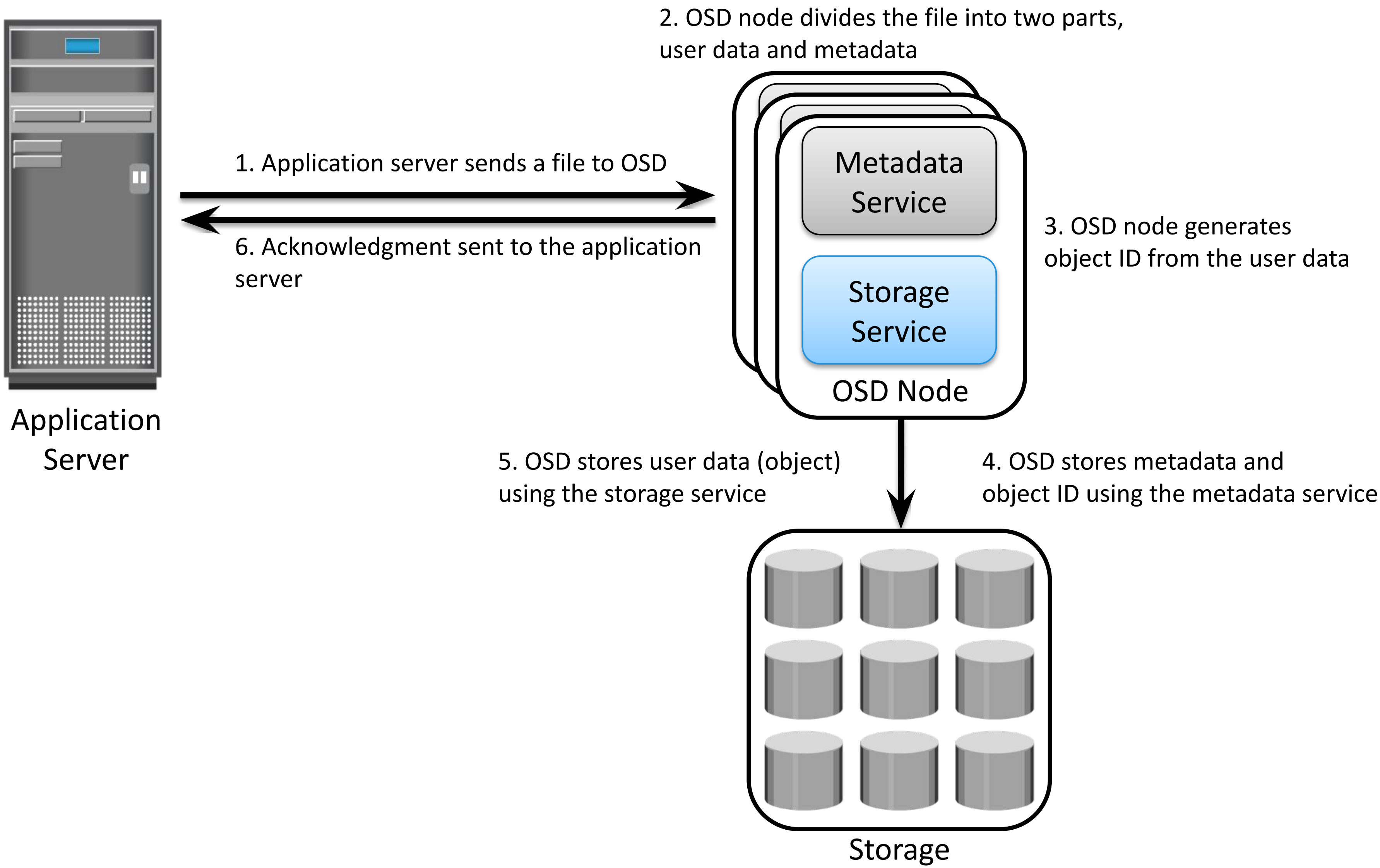
The file system user component communicates to the **OSD interface.**

Key Components of Object-based Storage Device

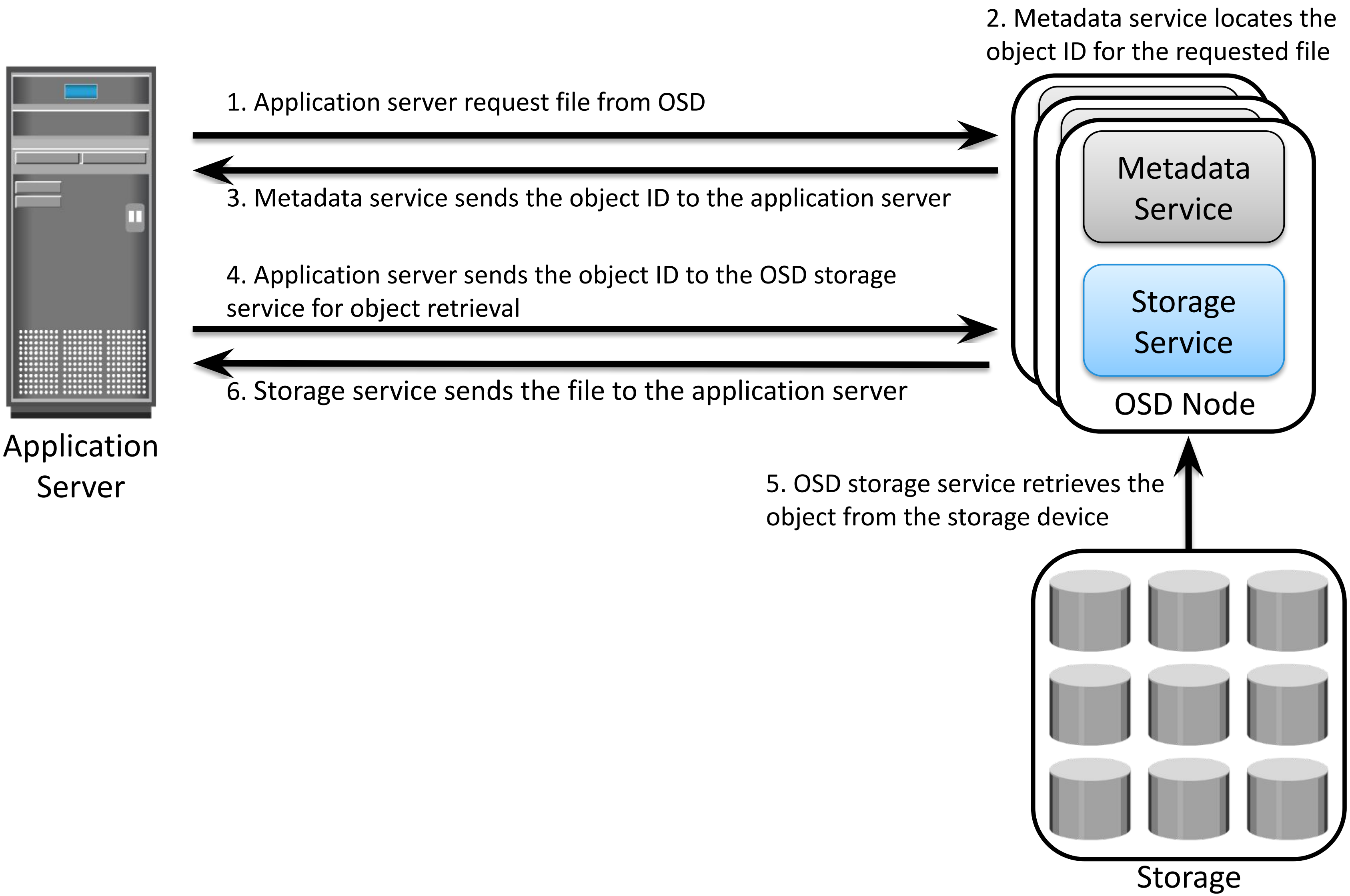


- OSD system typically comprises 3 key components:
  - OSD nodes: A server that runs the OSD operating environment and provides services to store, retrieve, and manage data in the system
  - Internal network: Provides node-to-node connectivity and node-to-storage connectivity
  - Storage: OSD typically uses low-cost and high-density disk drives to store the objects. As more capacity is required, more disk drives can be added to the system.

Process of Storing Object in OSD



Process of Retrieving Object from OSD

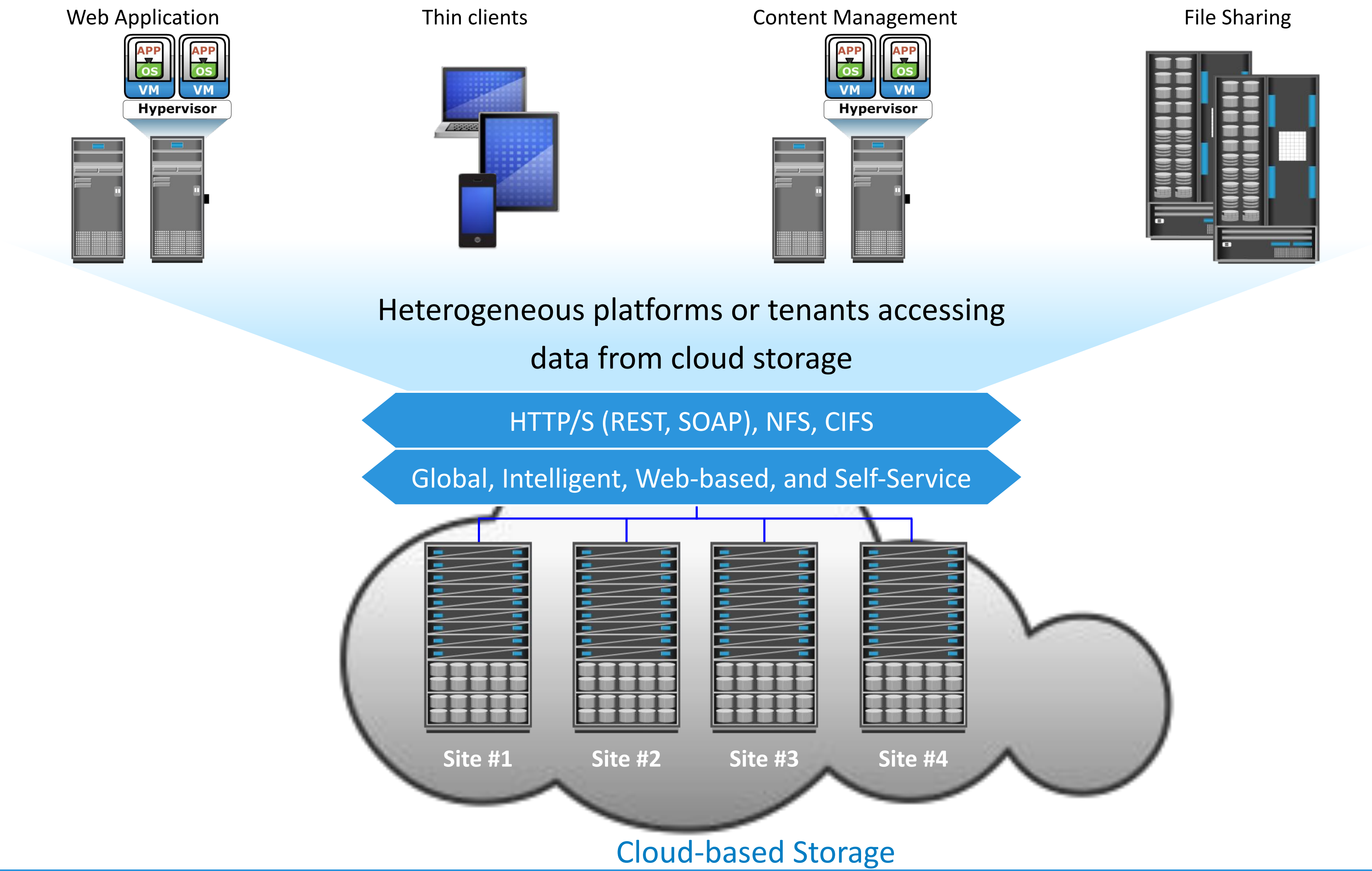


Key Benefits of Object-based Storage

Benefits	Description
Security and reliability	• Unique object ID generated by specialized algorithms ensures data **integrity** and content **authenticity** • Request authentication is performed at storage device
Platform independence	• Because objects are abstract containers of data, it enables sharing of objects across **heterogeneous platforms** • This capability makes object-based storage suitable for cloud computing environment
Scalability	• Both OSD nodes and storage can be **independently** scaled
Manageability	• Have inherent intelligence to **manage** objects • Have self-healing capability – **replicate** objects • Policy based management capability enables OSD to handle **routine** jobs automatically

Use Case 1: Cloud-based Storage

Representational state transfer (REST)
Simple object access protocol(SOAP)



Use Case 2: Content Address Storage (CAS)

- Traditional archival solutions – CD, DVD-ROM do not provide scalability and performance
- OSD stores data as objects, associates them with a unique object ID, and ensures high data integrity
- OSD provides scalability and data protection, practical option for long term data archiving for fixed content
- CAS – a special type of object-based storage device built for **storing fixed content**
  - Each object is assigned a globally unique identifier, known as content address (CA)
    - CA is derived from the binary representation of the data
  - CAS device can be accessed via the CAS API running on the application server

Key Features of CAS

1 Content authenticity and integrity

It assures the genuineness of stored content and assures that the stored content has not been altered

2 Location independence

The location from which the data is accesses is transparent to the application

3 Single instance storage (SiS)

The unique signature is used to guarantee the storage of only a single instance of an object

4 Retention enforcement

Protecting and retaining data objects is a core requirement of an archive system

The retention policies are enforced until the policies expire

5 Data protection

All fixed content is stored in CAS once and is backed up with a protection scheme. This ensures that the content stored on the CAS system is available in the event of disk or node failure

Key Features of CAS (contd.)

6 Fast record retrieval

Stores all objects on disks – faster access compared to tapes and optical discs

7 Load balancing

Distributes data objects on multiple nodes to provide maximum throughput, availability, and capacity utilization

8 Scalability

Adding more nodes to the cluster without any interruption to data access and with minimum administrative overhead.

9 Self diagnosis and repair

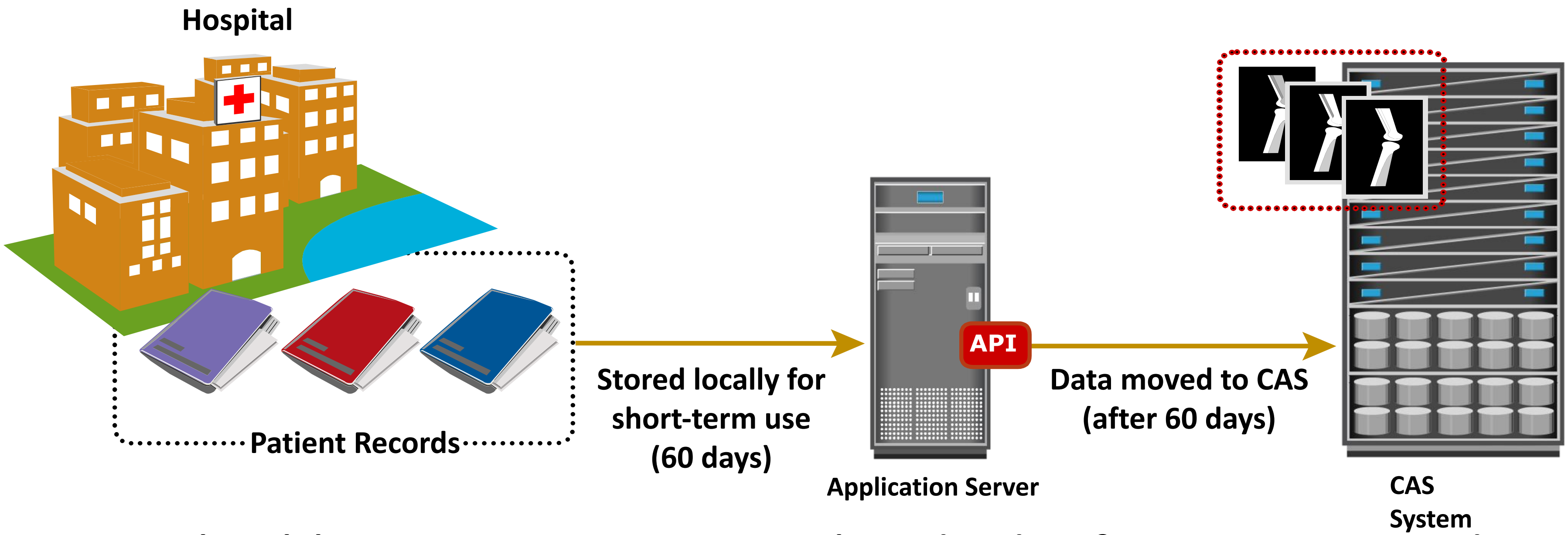
Automatically detects and repairs corrupted objects and alert the administrator of any potential problem.

10 Audit trail and event notification

Enable documentation of management activity and any access and disposition of data

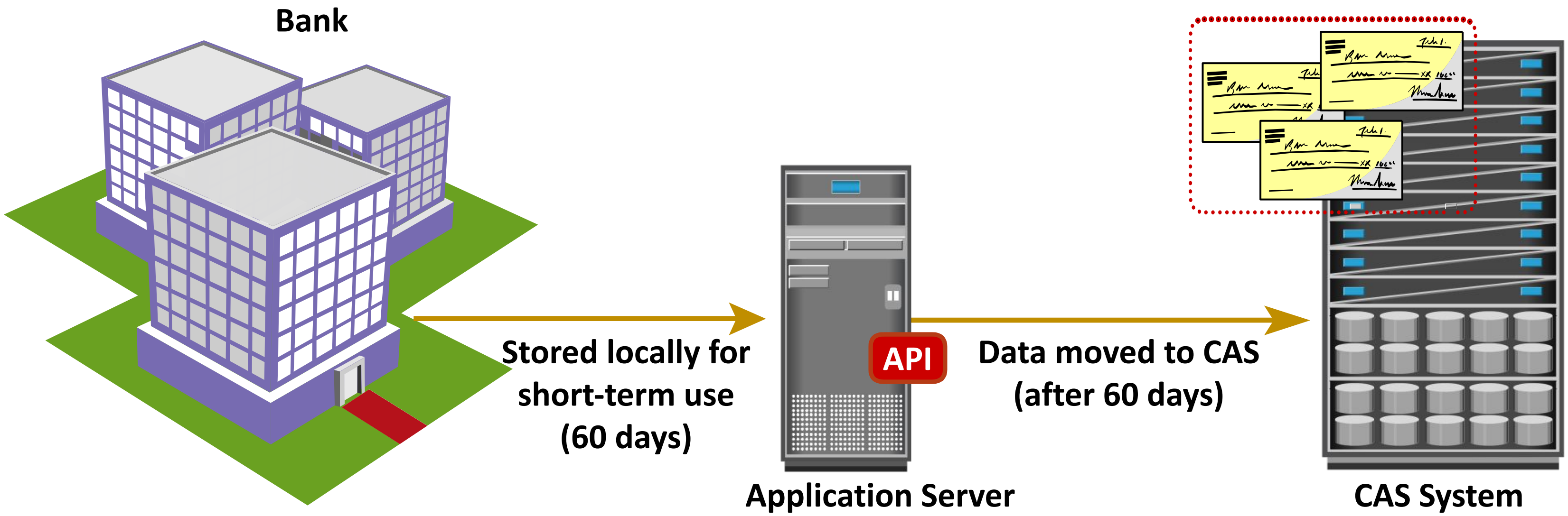
Provides on-demand reporting and event notification

Use Case 1: Healthcare Solution



- Large healthcare centers examine hundreds of patients every day and generate large volumes of medical records.
- Each X-ray image size range from about 15MB to over 1GB
- Patient records are stored online for a specific period of time
- E.g.: Kept in the original format for 60 days, beyond 60 days patient records are archived to CAS

Use Case 2: Financial Solution



- In a typical banking scenario, images of checks, each approximately 25 KB in size, are created and sent to archive services over an IP network
- Check imaging service provider might process around 90 million check images per month
- E.g.: Checks are stored online for a period of 60 days, beyond 60 days data is archived to CAS