CpE654 / NIS654
Design and Analysis of Network Systems

Network Architecture

Introduction
What is architecture?

It is the art and science of designing and constructing the Network by establishing relationships between network components

Introduction - Network Architecture

Network architecture is the process of developing a high-level, end-to-end structure for the network.

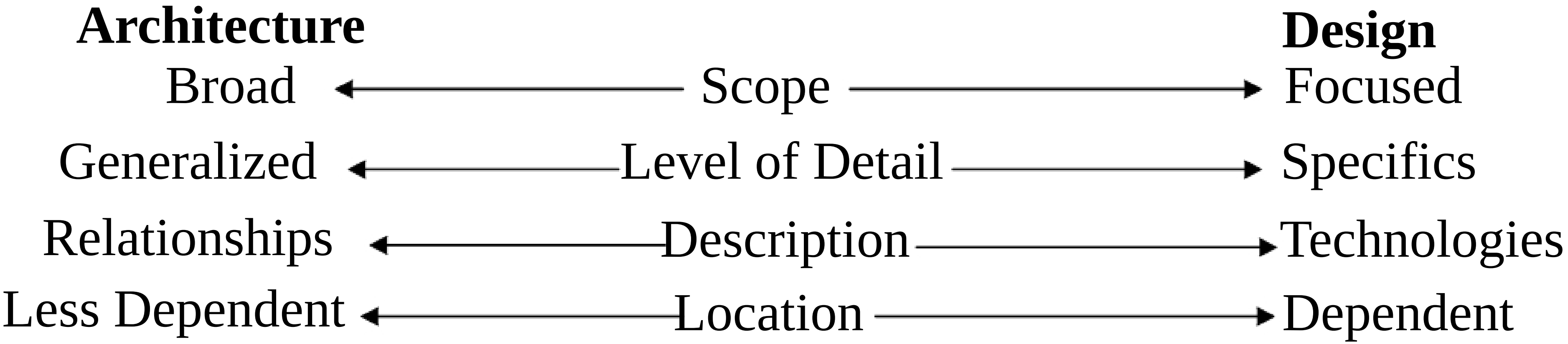
Major Functional Components of Architecture include:

- Addressing and routing
- Performance
- Security
- Network management
- Others (Database, servers, etc.)

Network architecture defines the relationships within and between these major functional architectures of the network

Major functional architectural components are dictated by technologies, mechanisms, and knowhow available at the time.

Differences Between Architecture and Design



Architecture **Design**

Broad ← Scope → Focused

Generalized ← Level of Detail → Specifics

Relationships ← Description → Technologies

Less Dependent ← Location → Dependent

In Summary:

- broad vs. focused.
- high level view vs. details, specificity
- Broad relationships between components vs. specifies technologies, protocols, products.
- Design: exact locations, products at these locations in Design

Functions and Component Architectures

Four major component architectures, or functional component architectures are:

- Addressing/routing
- Network Management
- Performance
- Security

Each functional component architecture represents a major capability of that network.

There ore interdependencies we need to consider between functional architectures.

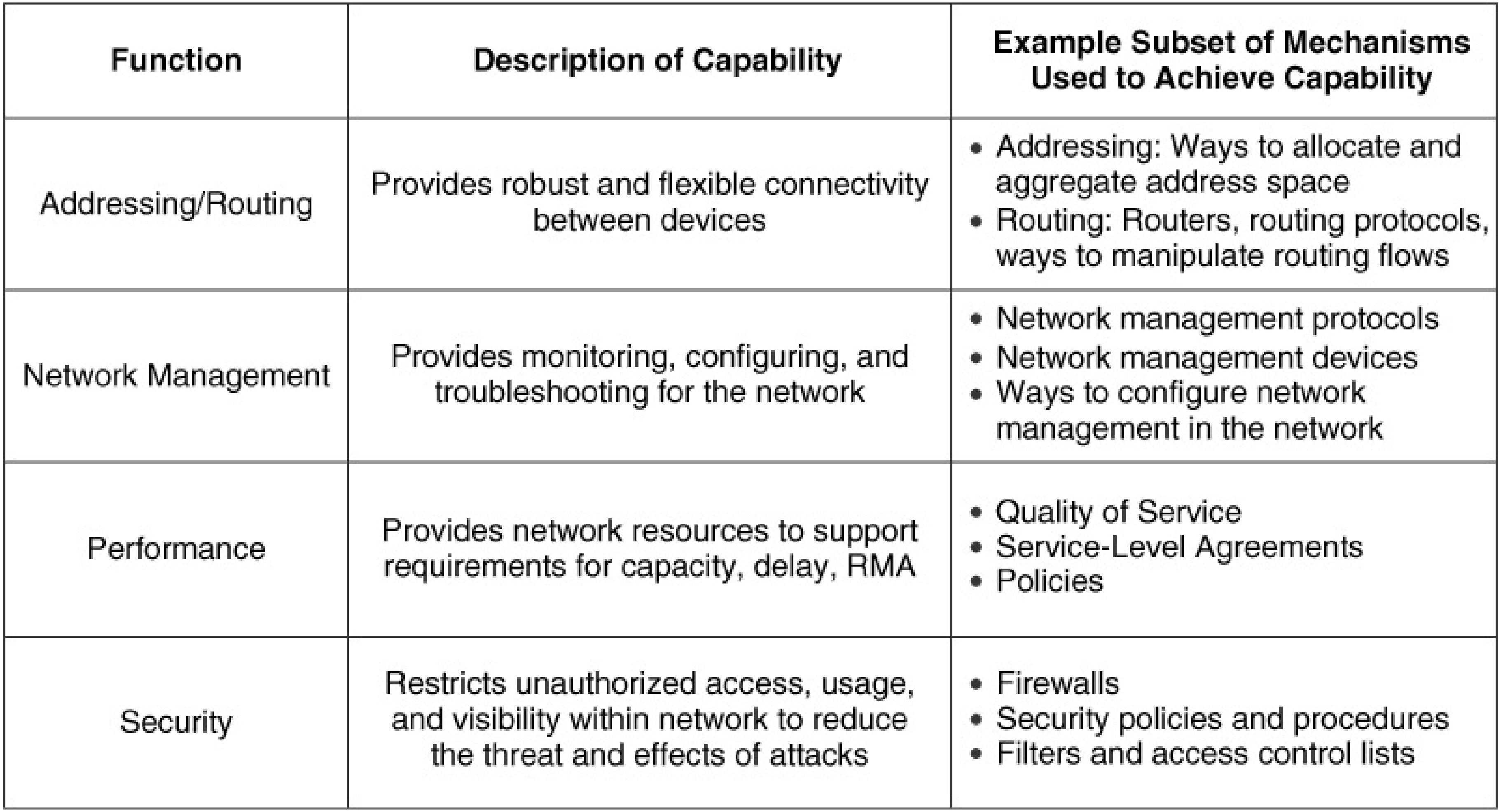
Functional Component architecture is a description of mechanisms how and where each **function** of a network is applied within that network.

Functional Component architecture need to include:

- A set of mechanisms, defined as hardware (transmission, switching, and routing products) and software (e.g., Gateway Protocols, DHCP, DiffServ)
- Where each mechanism may be applied
- A set of internal relationships and dependencies between these mechanisms.

There is a difference between functions (components) and mechanisms.

Network Functional Components and Mechanisms



Function	Description of Capability	Example Subset of Mechanisms Used to Achieve Capability
Addressing/Routing	Provides robust and flexible connectivity between devices	• Addressing: Ways to allocate and aggregate address space • Routing: Routers, routing protocols, ways to manipulate routing flows
Network Management	Provides monitoring, configuring, and troubleshooting for the network	• Network management protocols • Network management devices • Ways to configure network management in the network
Performance	Provides network resources to support requirements for capacity, delay, RMA	• Quality of Service • Service-Level Agreements • Policies
Security	Restricts unauthorized access, usage, and visibility within network to reduce the threat and effects of attacks	• Firewalls • Security policies and procedures • Filters and access control lists

Addressing/Routing Functional Component

Addresses are identifiers to devices at protocol layers (e.g., MAC, IP).

Addressing mechanisms are e.g., subnetting/supernetting, dynamic/private addressing, VLAN, NAT/NAPT.

Routing is applying connectivity information for forwarding packets

Routing mechanisms are e.g., route tables, default routes, route filtering, CIDR, unicast/multicast/broadcast, routing policies, iGP/eGP/BGP gateway protocol selection

Network Management (NM) Functional Component

- **NM** functions consist of monitoring, and control of the network.
- **NM mechanisms** are e.g., monitoring, instrumentation, configuration, in-band/out-of-band mgmt., centralized/distributed mgmt., MIB selection, north-bound integration with Operations Systems.
- FCAPS is often used to describe the NM functions: Fault detection, Configuring network, Accounting functions, Performance monitoring, and Security functions.

Performance Functional Components

Performance management is about allocating network resources to different user applications and management flows based on needs. Prioritizing, scheduling, QoS/SLA administration, are several important ways of managing performance. and controlling

Performance mechanisms are used to configure, provision, operate, and manage resources that control performance e.g., capacity planning, traffic engineering, and other service control mechanisms.

Security Functional Components

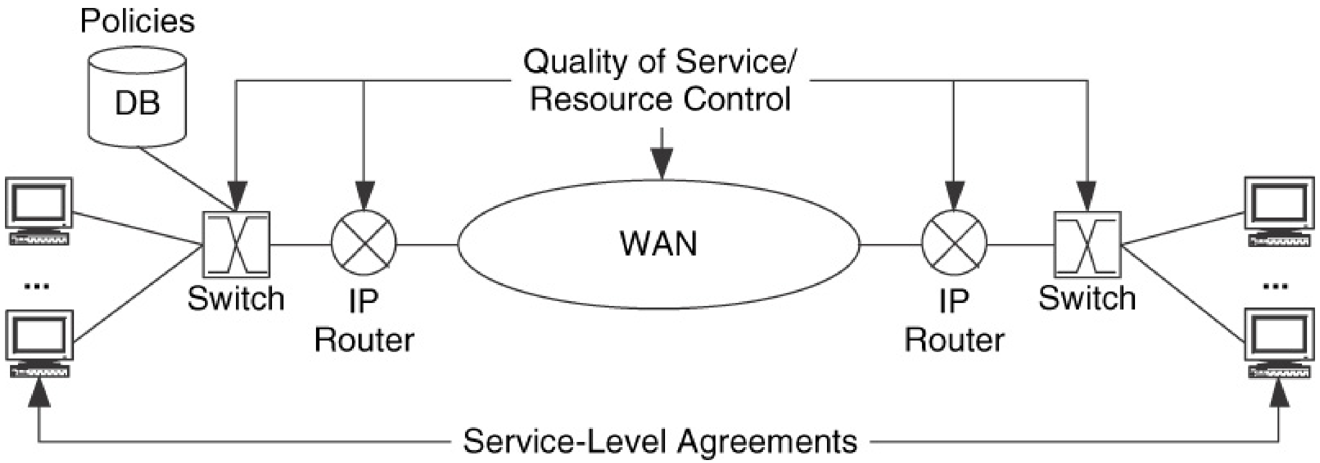
- **Security** component describes how system resources are protected from physical damage, theft, unethical disruption of services (e.g., DOS), unauthorized access, and protecting information flows as well as users' personal information.

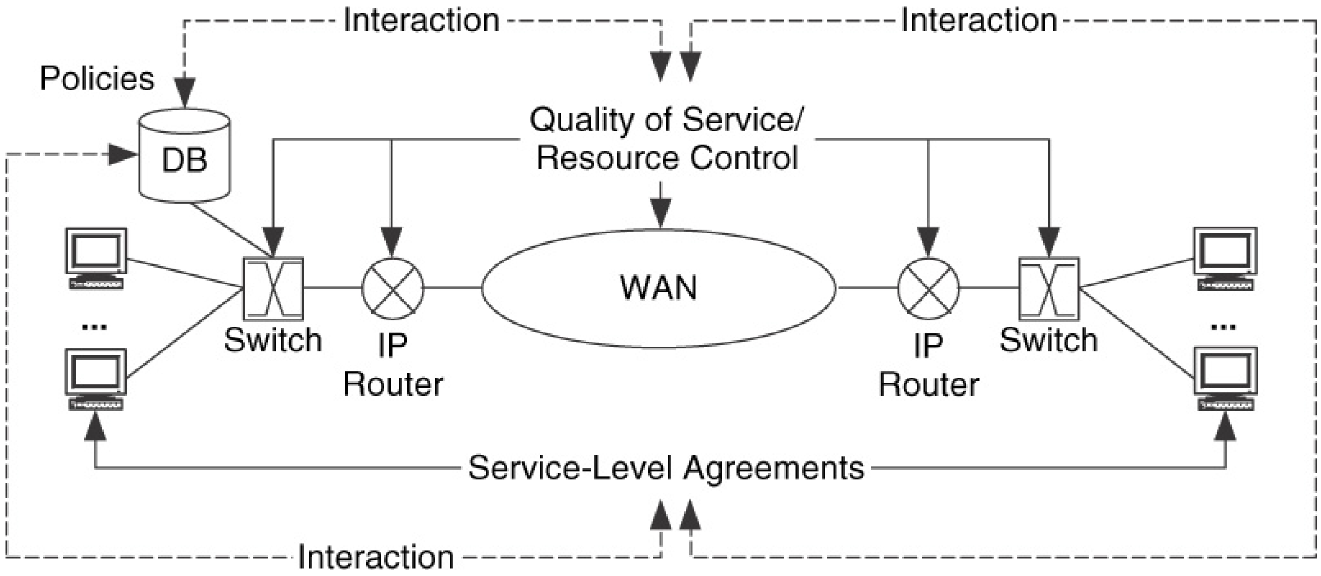
- Examples of **Security mechanisms** are threat analysis, setting security policies and procedures, physical security, application/protocol security, remote access security, border control, tunneling, and encryption.

Interactions Among Mechanisms (Performance Example)

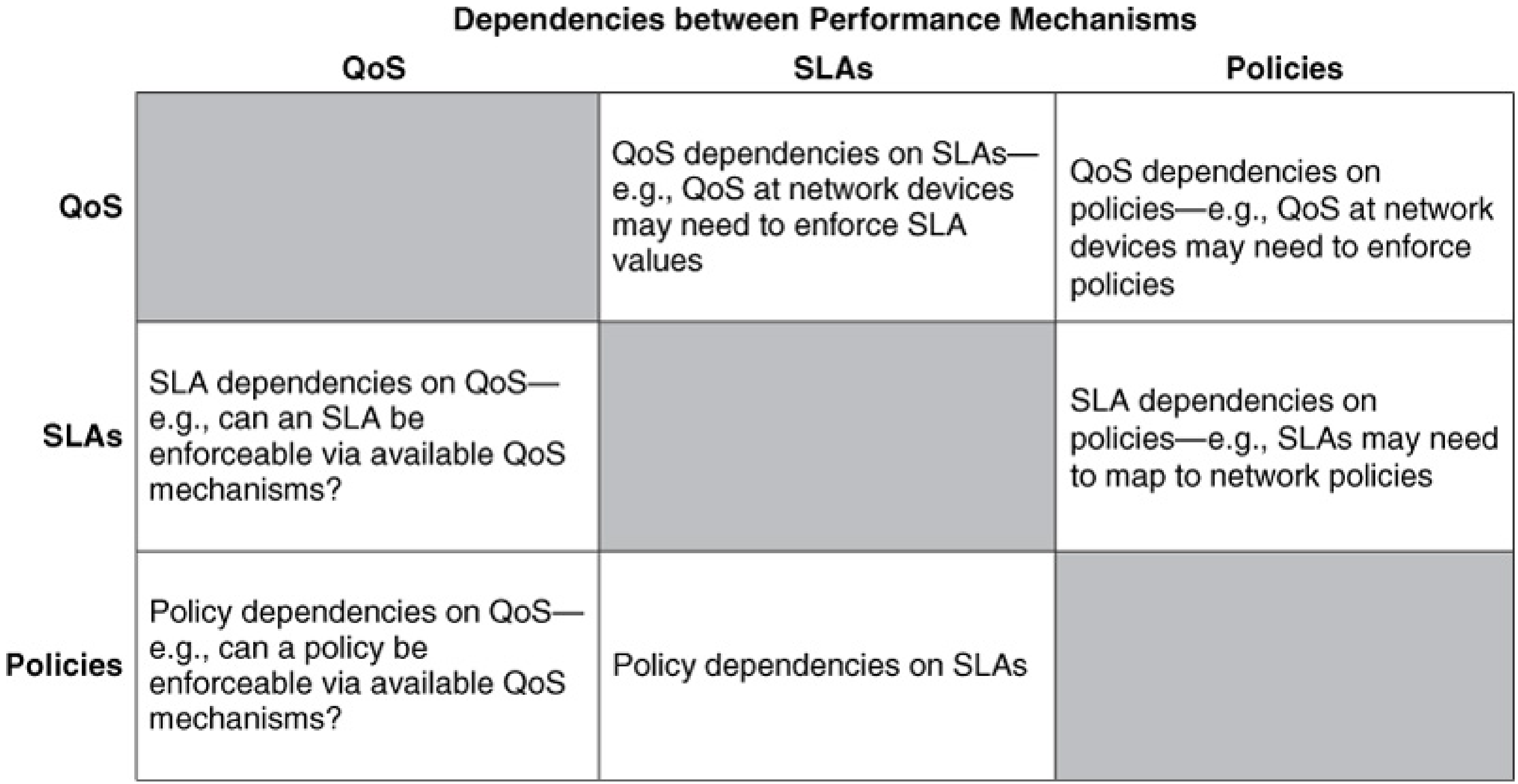
- For performance component architecture, we have different mechanisms: QoS, SLAs, and Policies.
  - SLA → Policies → QoS
  - SLAs are high level agreements with clients
  - Policies are overall directives for the network
  - QoS are mechanisms such as DiffServ, admission control, etc.
- For desired network performance , we need to determine:
  - how and where each mechanism is applied (QoS vs. SLA)
  - how they work together to provide desired overall performance for the system
  - what are the information flows between these QoS, SLAs, and policies mechanisms.
  - What are the constraints? SLAs are constrained by the type and placement of QoS within the network.

Mechanism Interactions - Example Performance





Mechanism Interactions in Performance (Contd.)



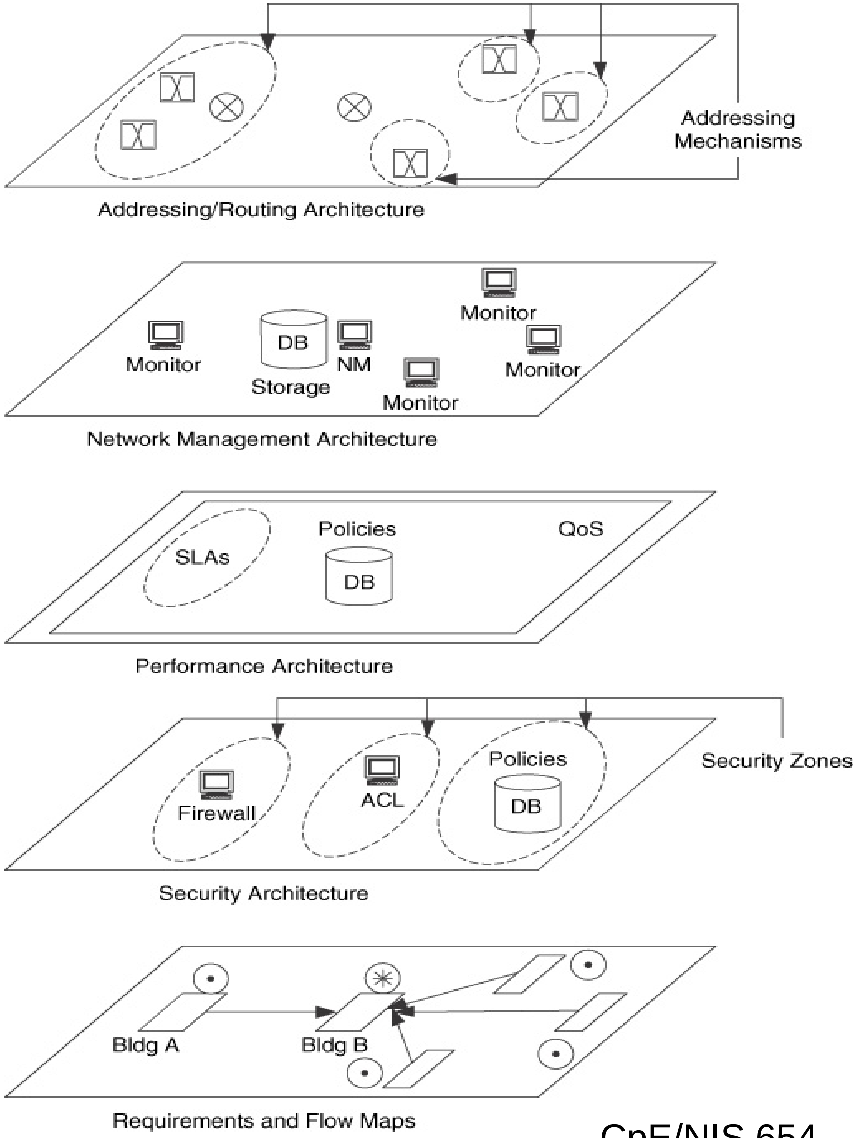
Dependencies between Performance Mechanisms

	QoS	SLAs	Policies
QoS		QoS dependencies on SLAs—e.g., QoS at network devices may need to enforce SLA values	QoS dependencies on policies—e.g., QoS at network devices may need to enforce policies
SLAs	SLA dependencies on QoS—e.g., can an SLA be enforceable via available QoS mechanisms?		SLA dependencies on policies—e.g., SLAs may need to map to network policies
Policies	Policy dependencies on QoS—e.g., can a policy be enforceable via available QoS mechanisms?	Policy dependencies on SLAs	

Mechanism Interactions and Dependencies Other Examples

- Network Management: Here the trade-offs could be:
  - Choosing between centralizing and distributing management capabilities.
  - Choosing between in-band or out-of-band management.
  - dependencies between mechanisms for network monitoring, configuration, and management
- Routing: Here dependencies will be on addressing mechanisms, as proper routing function will depend on how internal and external addressing is done.
- Security: Here policies are constrained by the type of security mechanism (Firewall) and performance (delay) requirements.

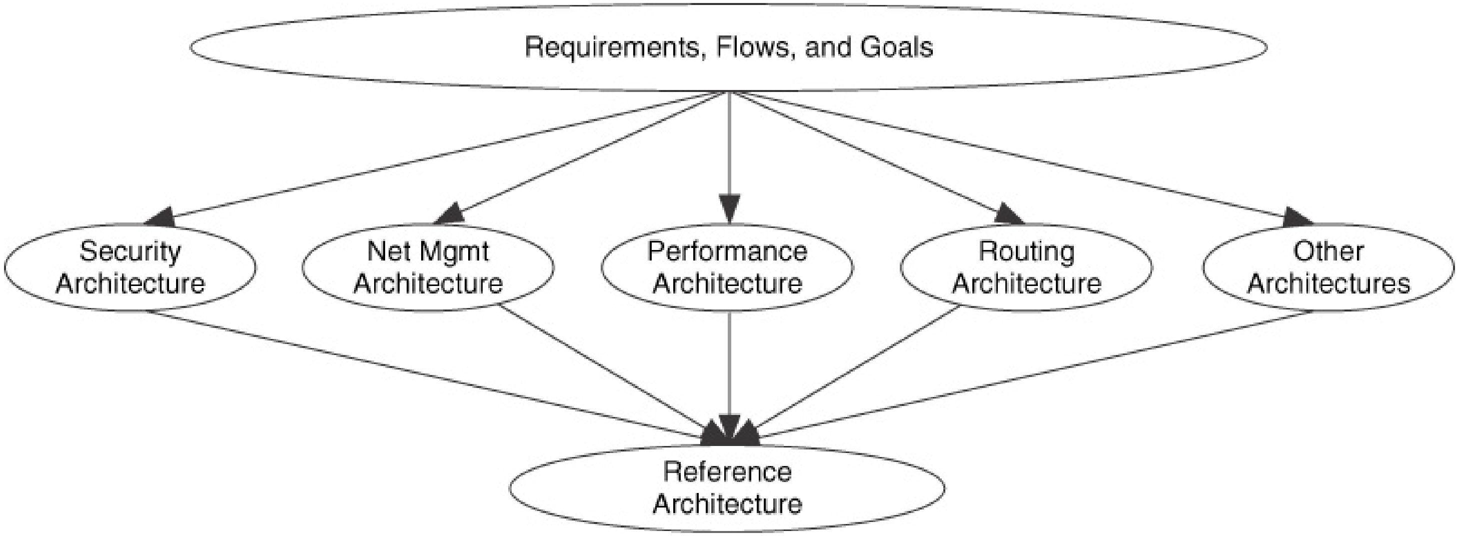
Reference Architecture
Component Architecture Overlaid on Requirements



- A reference architecture contains all of the component architecture (i.e., functions) being considered for the network.
- All of the component architectures for a network are closely coupled to each other.

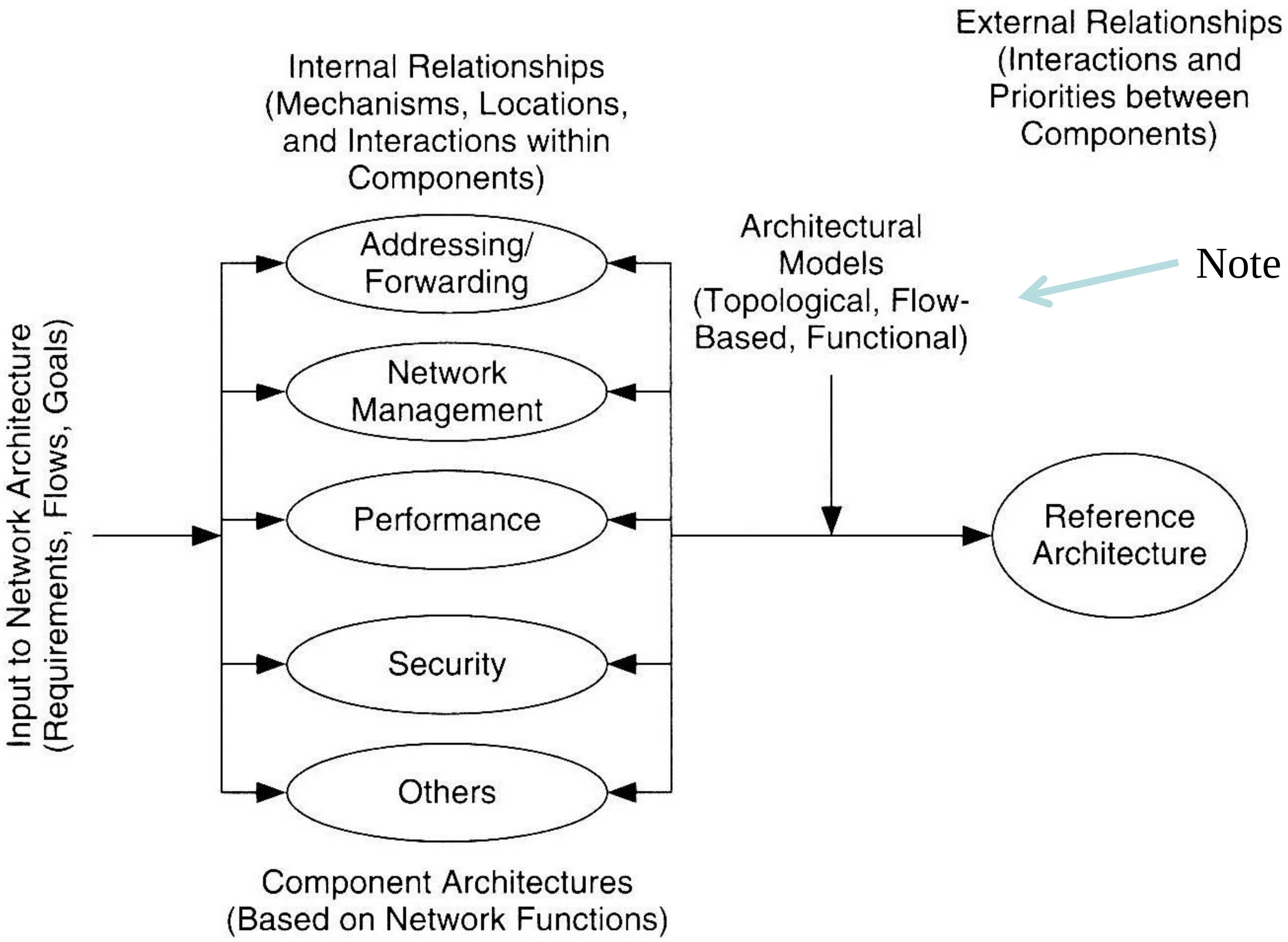
Access Control List
Service Level Agreement
Data Base

Reference Architecture, Component Architecture, and Requirements Relationships



- Based on the requirements, flows, identify:
  - Component architectures
  - Mechanisms within each component
- Define where each mechanism be applied based on the traffic flows.
- Determine and analyze the internal relationships between these mechanisms.

Process Model for Component Architecture


Internal Relationships (Mechanisms, Locations, and Interactions within Components)
External Relationships (Interactions and Priorities between Components)
Architectural Models (Topological, Flow-Based, Functional)
Input to Network Architecture (Requirements, Flows, Goals)
Addressing/ Forwarding
Network Management
Performance
Security
Others
Reference Architecture
Component Architectures (Based on Network Functions)

Note Models

Examples of Functional Coupling

Addressing/routing component architecture couples with:

- Network management and security.
  - Based on the mechanism used in the network management and security component architectures, their traffic flows may take separate paths from user traffic flows, a capability that must be incorporated into the addressing/routing component architecture.
- performance.
  - Routing can be configured so that that traffic flows with differing performance requirements take separate paths through the network. (Multiprotocol label switching (MPLS) integrates routing and performance).

Optimizing Reference Architecture

- Use functional relationships and coupling to optimize the sum of all components, or reference architecture.
  - ✓ Balanced across all functions
  - ✓ Weighted to favor one or more functions
  - ✓ Initial analysis dictates these trade-offs

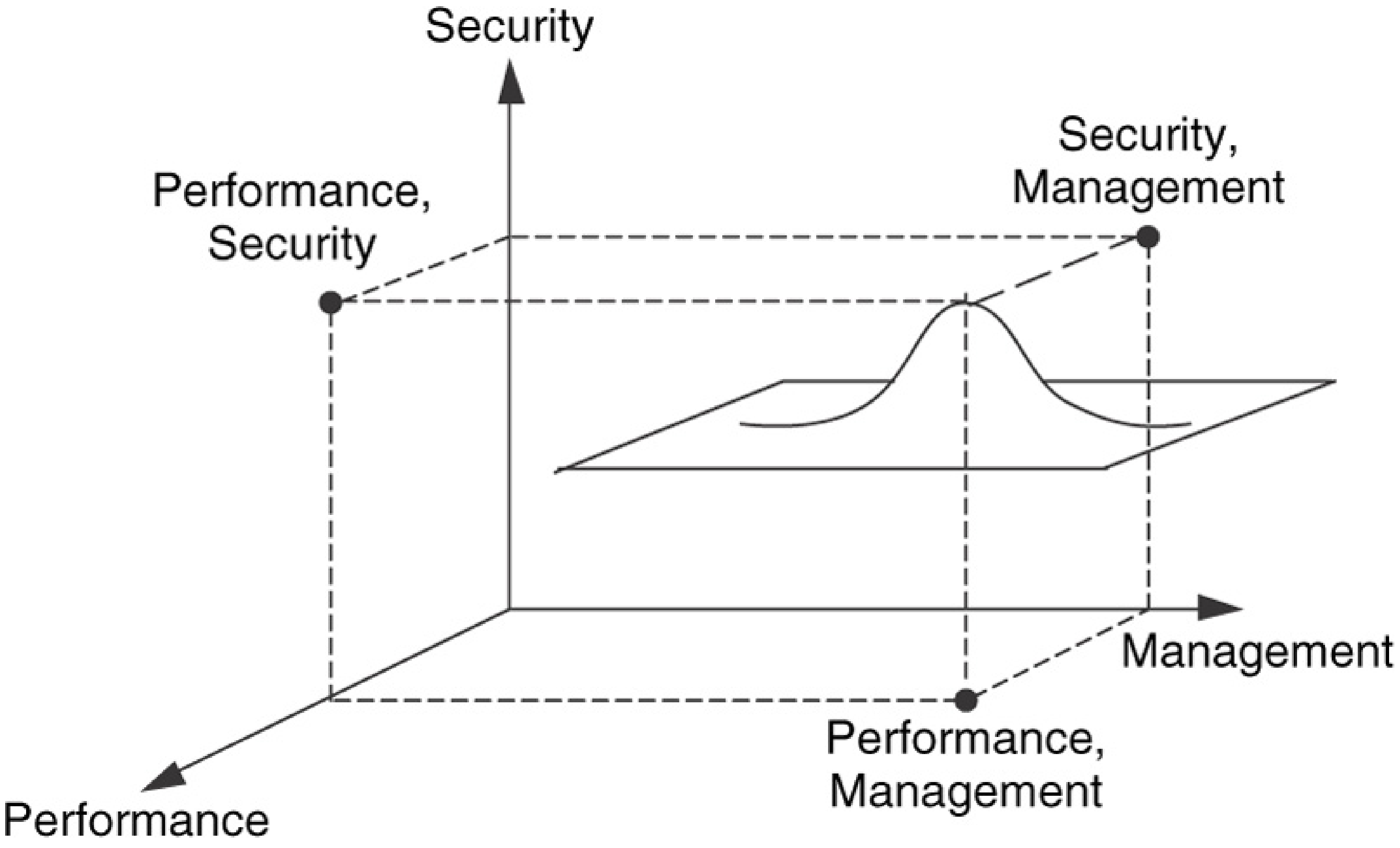
Example:

In a network in which performance is the primary architecture goal, the external relationships between performance and the other functions would be weighted so that trade-offs favor performance (say, over security, network management).

Examples of Interactions Between Pairs of Component Architectures

- Interaction between Performance and Security.
- Interaction between Network Management and Security.
- Interactions between Network Management and Performance.
- Interaction between Addressing/Routing and Performance.

Architectures Are Multidimensional Requiring Functional Trade-offs

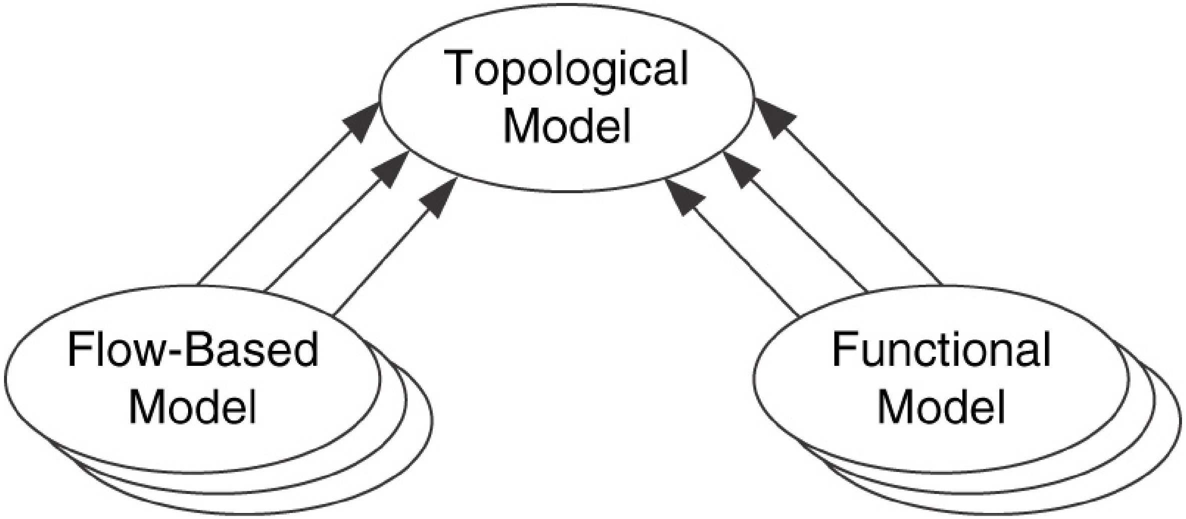


Architecture Models

- Used for developing reference architecture and locating functions within the network.
- Three types of architecture models:
  - Topological models: based on a geographical or topological arrangements and used as starting points.
  - Flow-based models: which are based on a traffic flows.
  - Functional models: which are based on one or more functions or features planned for in the network.

Architecture Models
Topological models

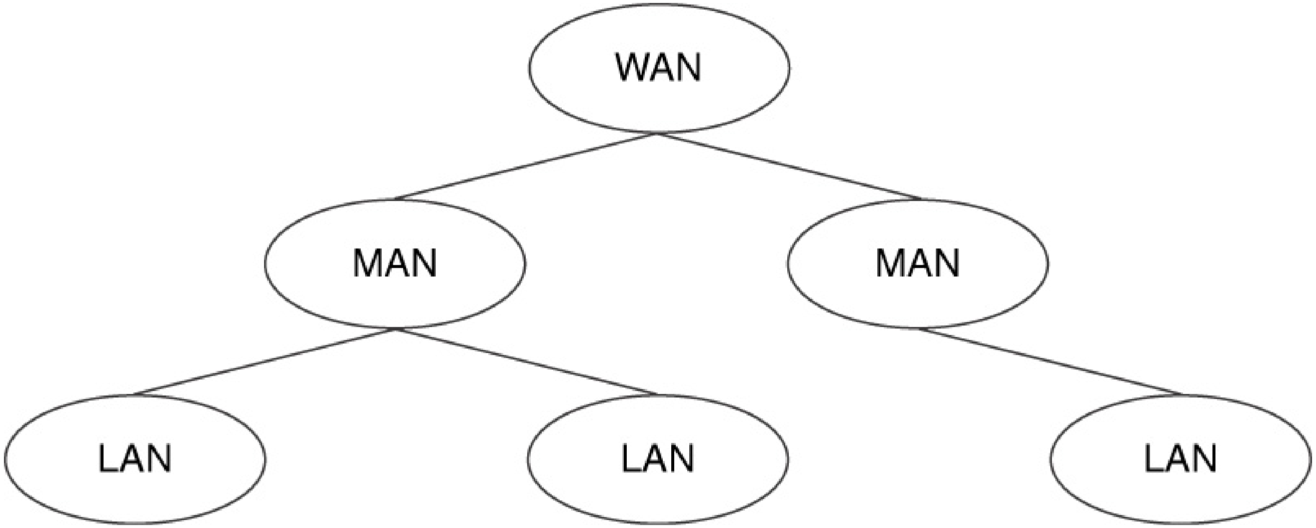
- LAN/MAN/WAN Model
- Access/Distribution/Core Model
- Flow Based Model



LAN/MAN/WAN Architectural Model

Concentrates on LAN/MAN/WAN boundaries:

Focus is on the features and requirements of boundaries, and functions, services, performance, and feature along LAN/MAN/WAN boundaries.



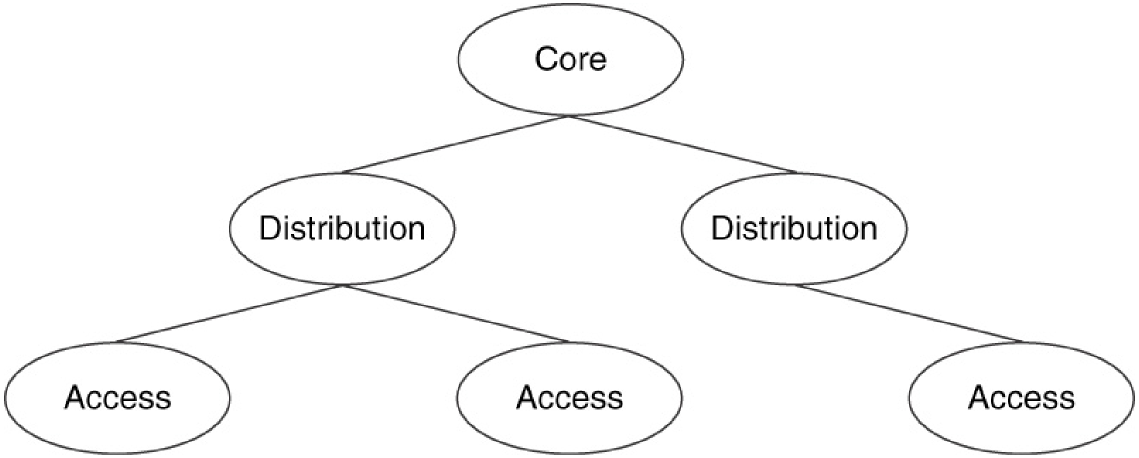
Access/Distribution/Core Architecture Model

Focuses on function instead of location

The access area: is closet to the users and their applications and is where most traffic flows are sourced and sinked.

The distribution area: can be also source and sink flows, but they are more likely to be to or from multi-user devices, such as servers or specialized devices.

The core area: is used for bulk transport of traffic, and flows are not usually sourced or sinked at the core.



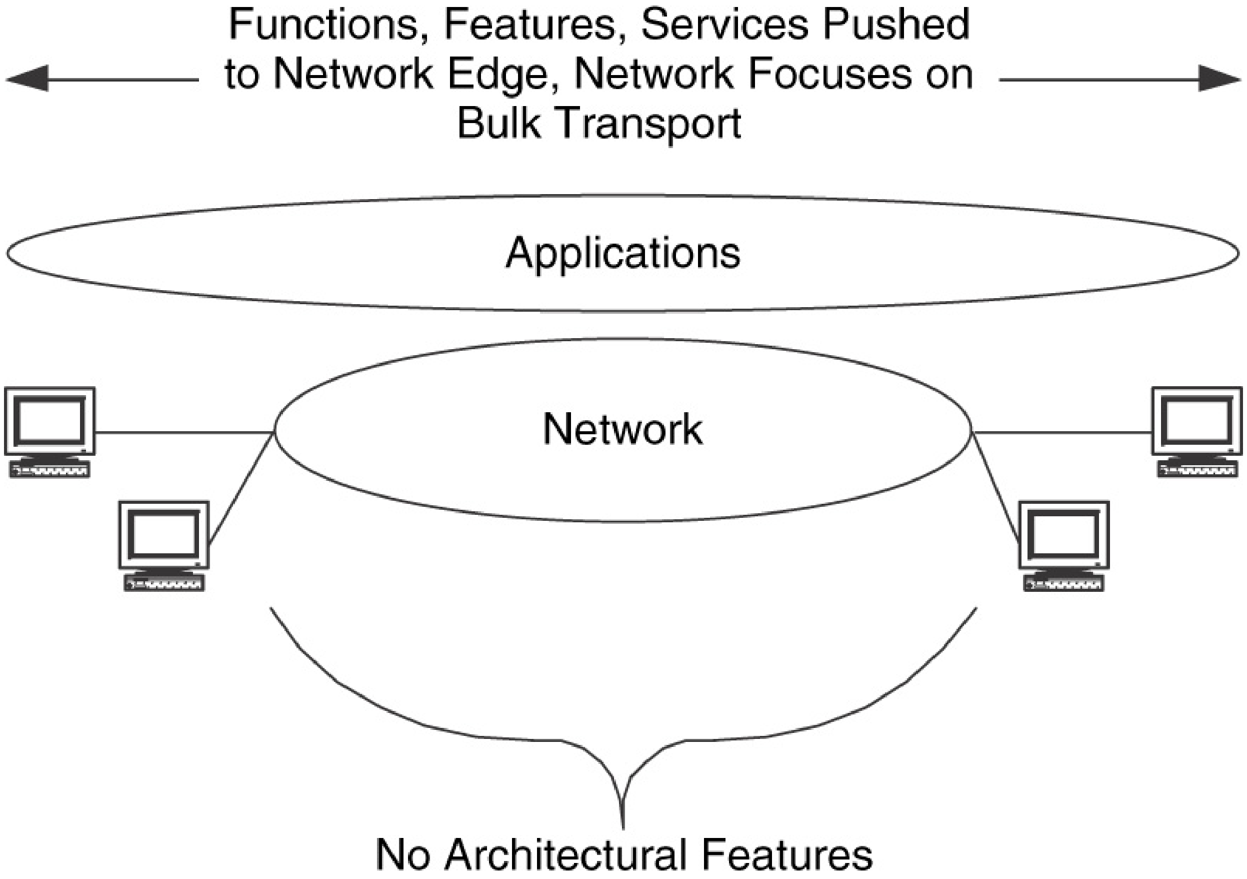
Flow based Architecture Models

There are four flow-based models:

1. Peer-to-peer architectural model
2. Client-server
3. Hierarchical client-server
4. Distributed computing

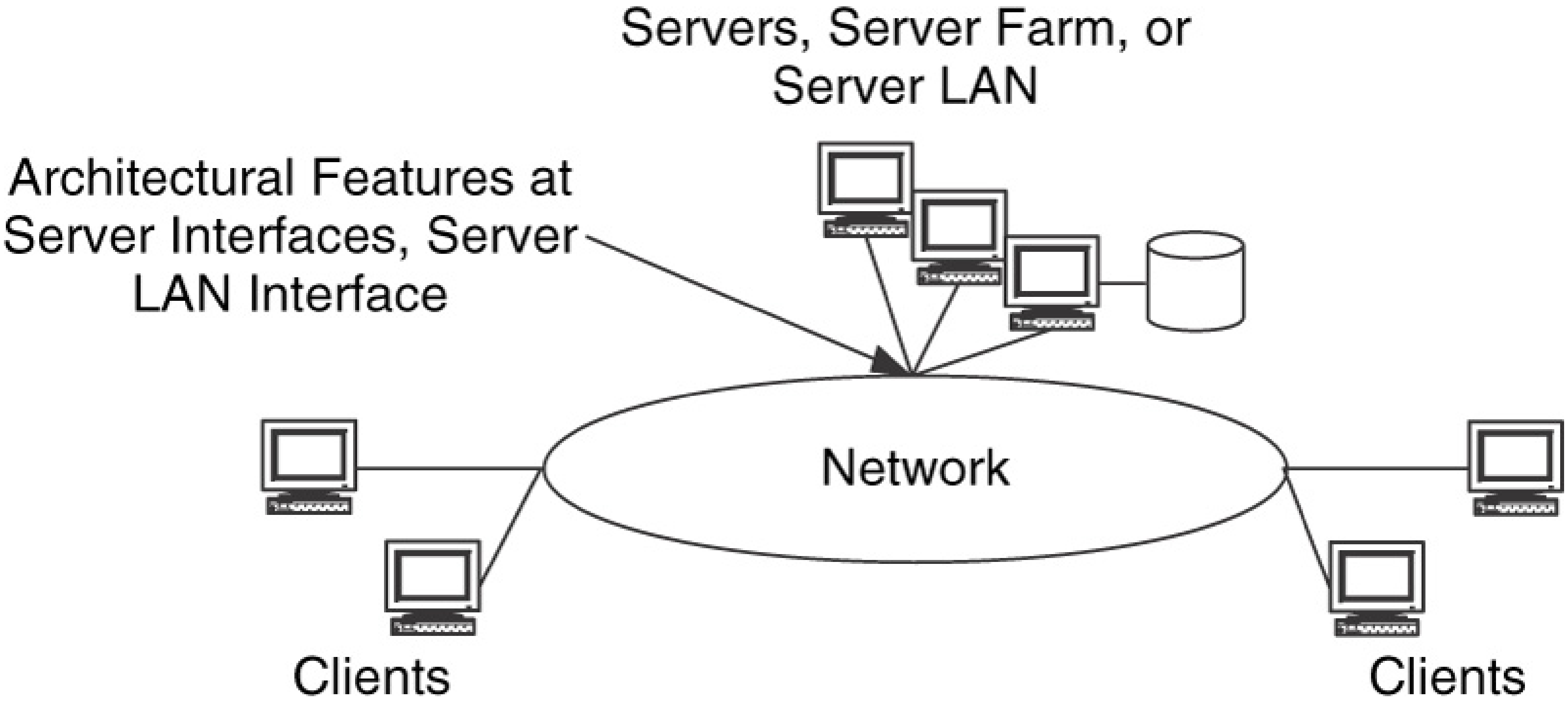
Peer to Peer Flow Based Architecture Model

- In Peer-to-Peer, functions, features, and services are pushed toward the edge of the network, close to users and their devices.
- Helpful in distributing load to the edges


Functions, Features, Services Pushed to Network Edge, Network Focuses on Bulk Transport
Applications
Network
No Architectural Features

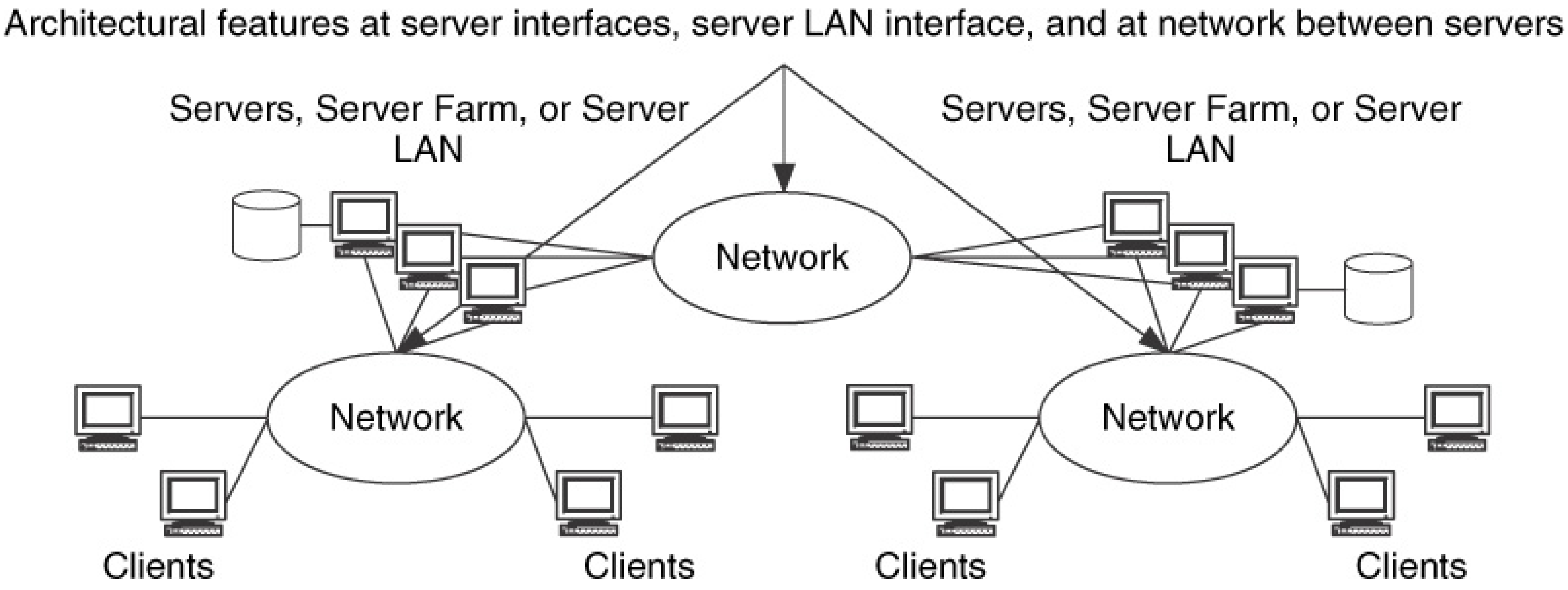
Client Server Based Architecture Model

we can locate functions and features in where flows combine (i.e. at server locations and the interfaces to client LANs).


Servers, Server Farm, or Server LAN
Architectural Features at Server Interfaces, Server LAN Interface
Network
Clients
Clients

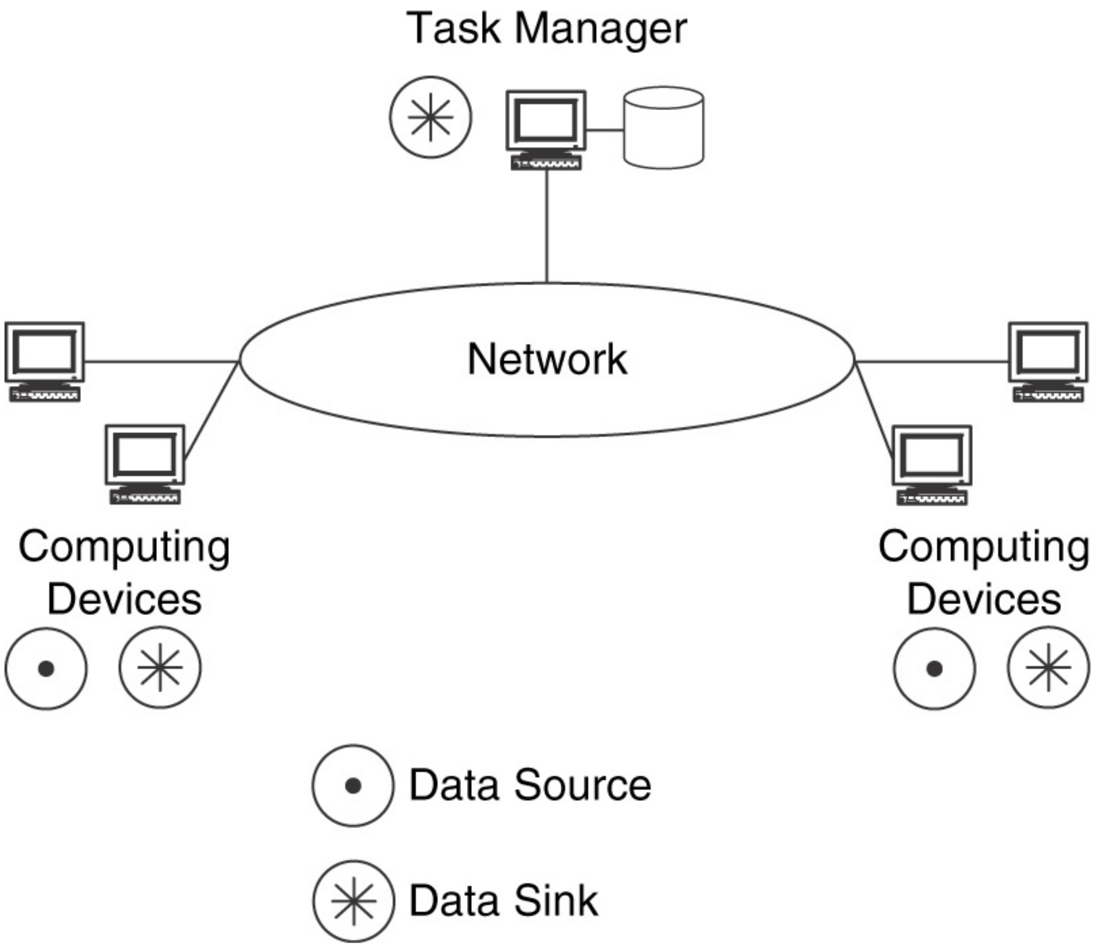
Hierarchal Client Server Based Architecture Model

Same as client server but with network in between the hierarchies


Architectural features at server interfaces, server LAN interface, and at network between servers
Servers, Server Farm, or Server LAN
Servers, Server Farm, or Server LAN
Network
Network
Network
Clients
Clients
Clients
Clients

Distributed Computing based Architecture Model

Sources and sinks are obvious locations for architecture features.



Functional Architecture Models (Contd.)

- Functional architecture models focus on supporting particular functions in the network.
- There are four different functional models:

1- Service-provider architectural model.

2 - Intranet/extranet architectural model

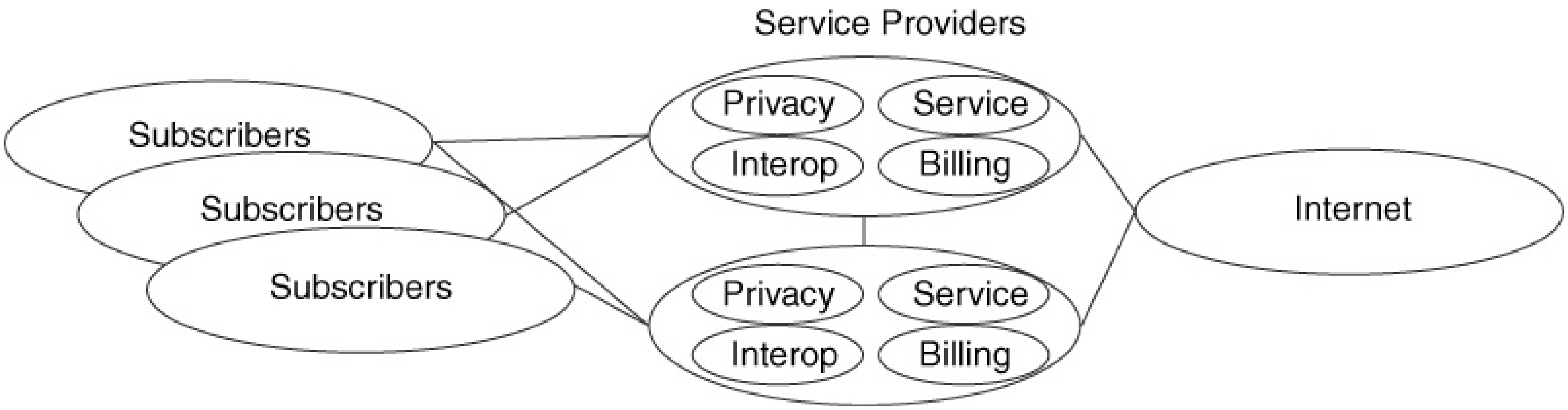
3 - Single-tier/multi-tier performance architectural model

4 - End-to-end architectural model

Service-provider architectural model

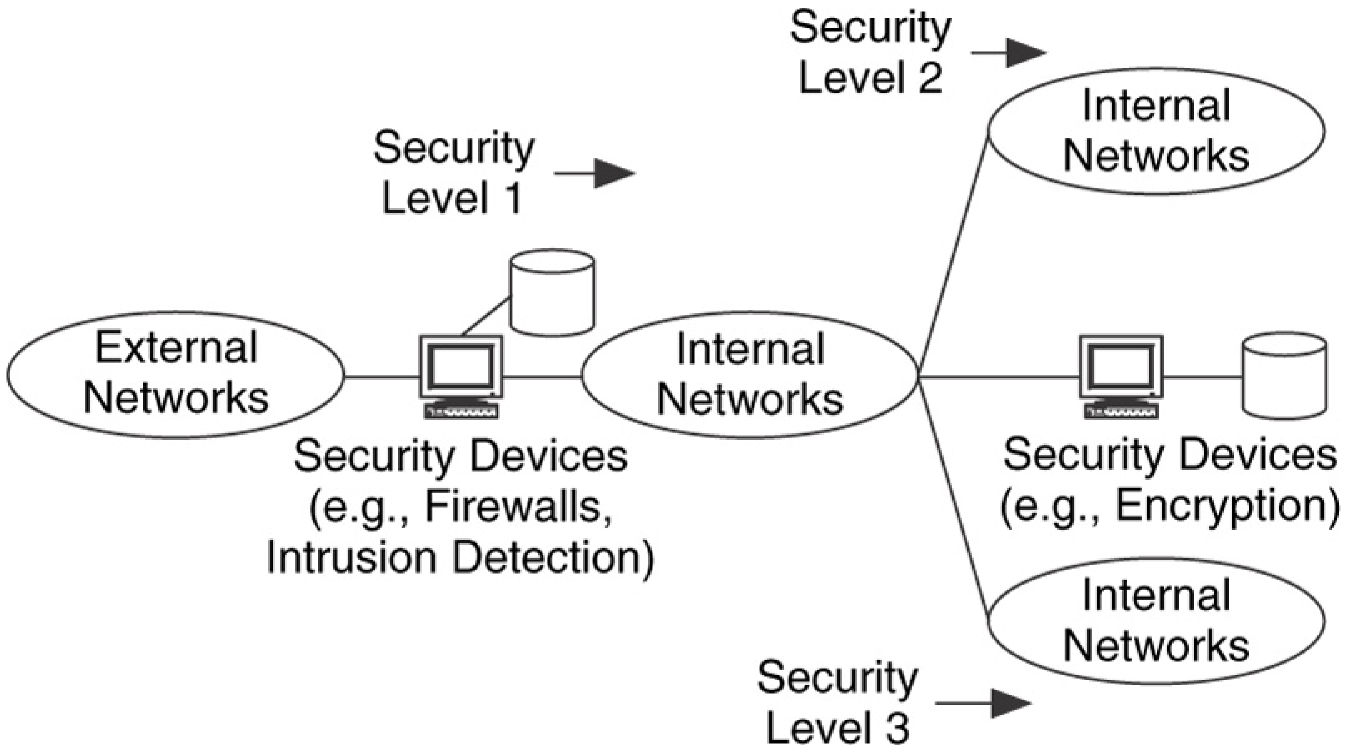
Based on service-provider provided function

- useful for enterprise networks also
- focuses on network providing privacy, security, billing, service delivery to customers



Intranet/extranet architectural model

- Security and privacy distinct for user within intranet and from outside (extranet)
- Separation of users, devices, and applications for intra and inter
- secure access a key consideration.

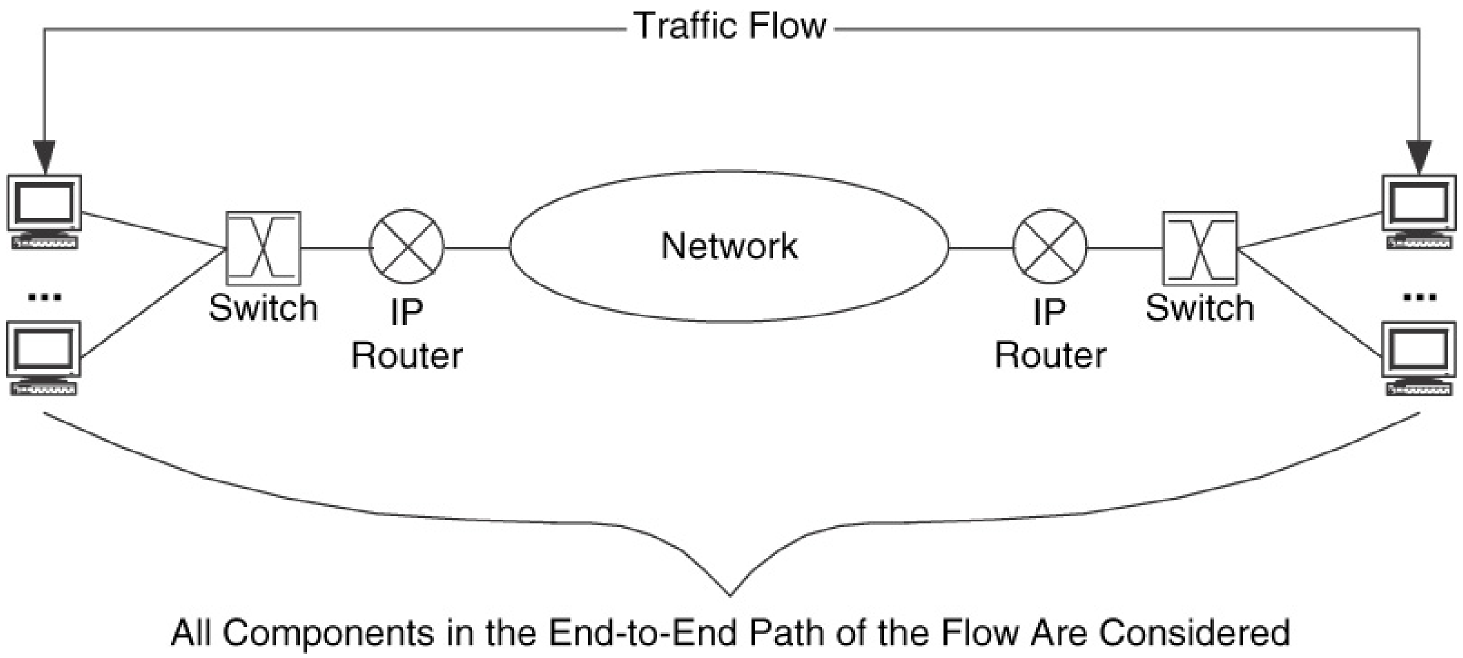


Single-tier/multi-tier performance architectural model

- Tier the Performance architecture
- One tier network for performance
- Multiple tiers of performance
- A combination: part of the network as having a single tier of performance, the other part with multiple tiers of performance, or a combination of all.

End-to-end architectural model

Focuses on all components in the end-to-end path of a traffic flow. This model is very close to the flow-based perspective of networking.


Traffic Flow
...
Switch
IP
Router
Network
IP
Router
Switch
...
All Components in the End-to-End Path of the Flow Are Considered

Functional Architecture Models Advantages and Disadvantages

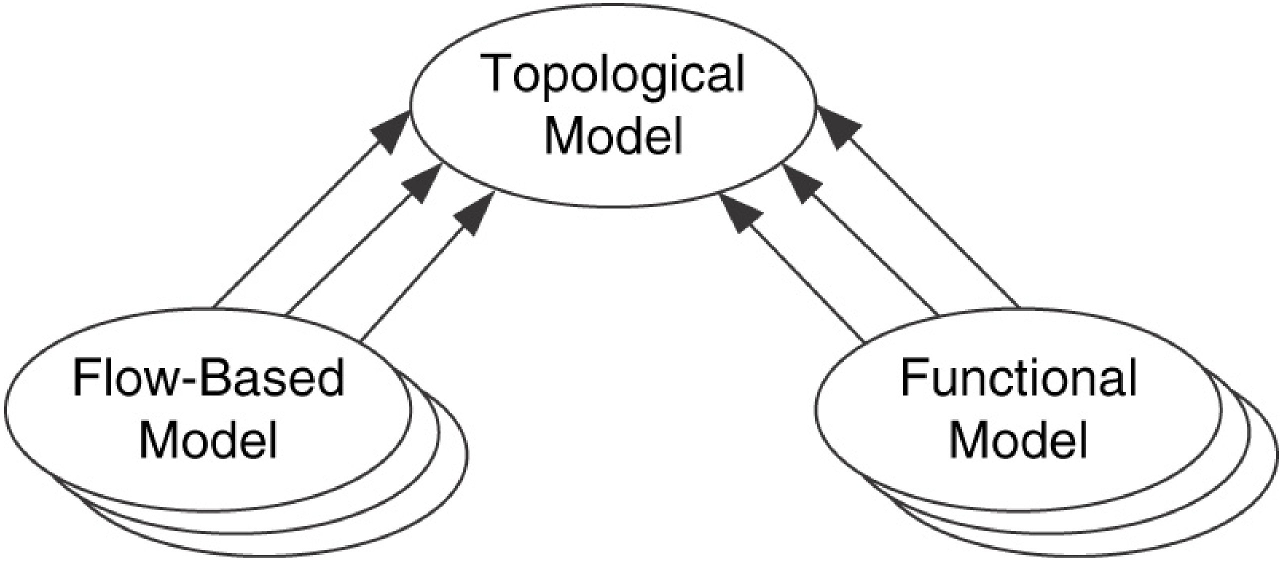
Advantages: Most closely related to the requirements developed during the requirements analysis process.

Disadvantages: Functional models are the most difficult to apply

Because you must understand where each function will be located. For example to apply the end-to-end model, you first have to define where end-to-end is for each set of users, applications, and devices that will be a part of end-to-end.

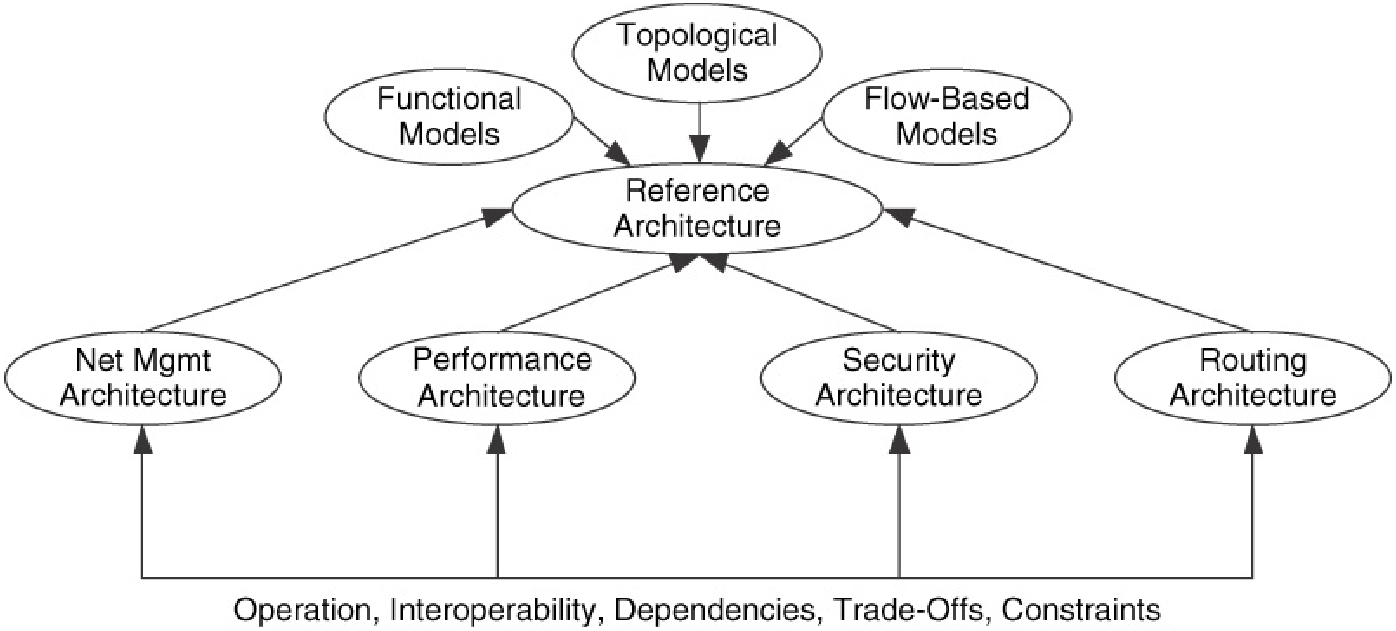
Using the Architecture Models

- A few of the models can be combined to provide a comprehensive architecture view of the network.
- This is usually achieved by starting with one of the topological models and then adding flow-based and functional models as required.
- Functional and Flow-based models complement the topological models.

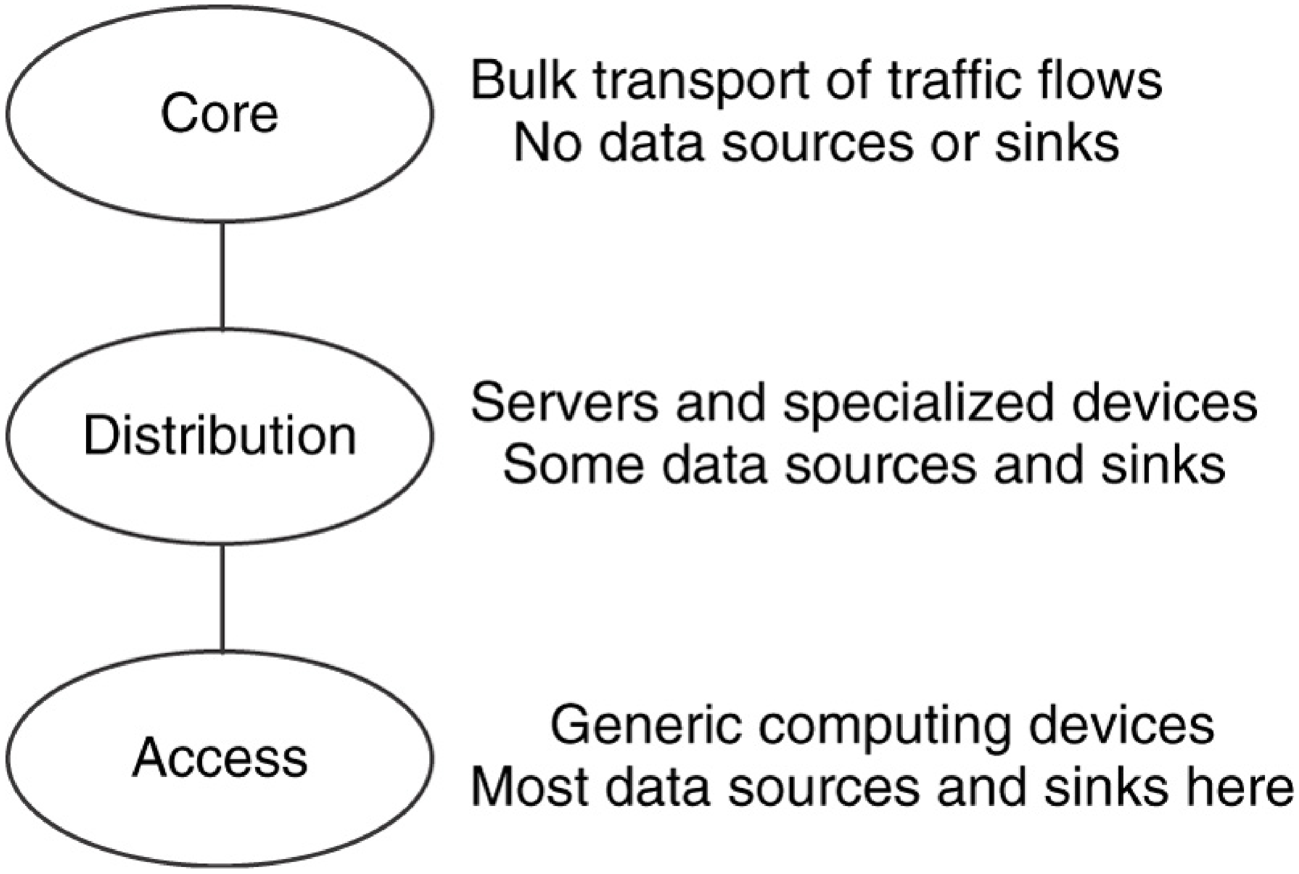


Considerations in Developing Reference Architecture Model

- You need to select proper models to develop your reference architecture.
- In practice, one or two models are typically sufficient for many networks.
- You may choose to apply a model to describe the entire network, with additional models for specific area (e.g. a client-server model for access to a server or a distributed-computing model for computer centers).



Core/Distribution/Access model from Flow Perspective

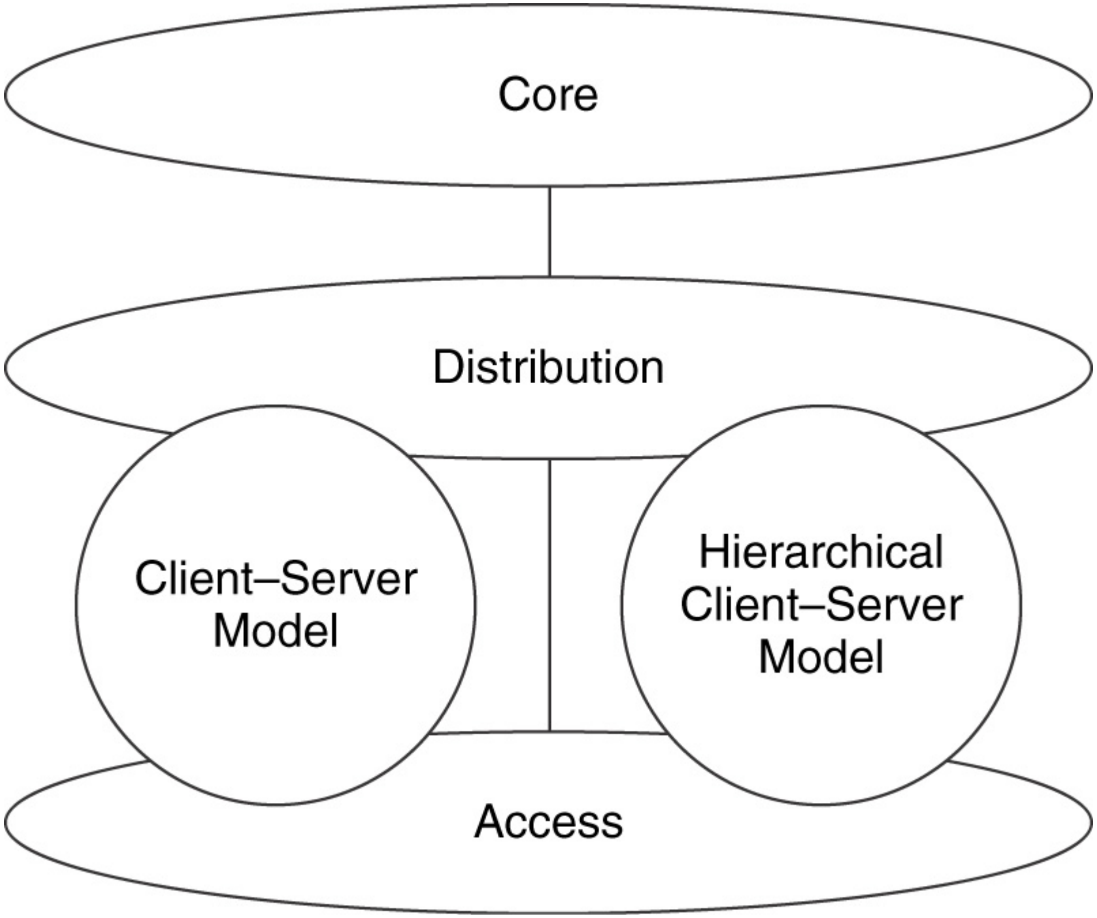


Core: Bulk transport of traffic flows
No data sources or sinks

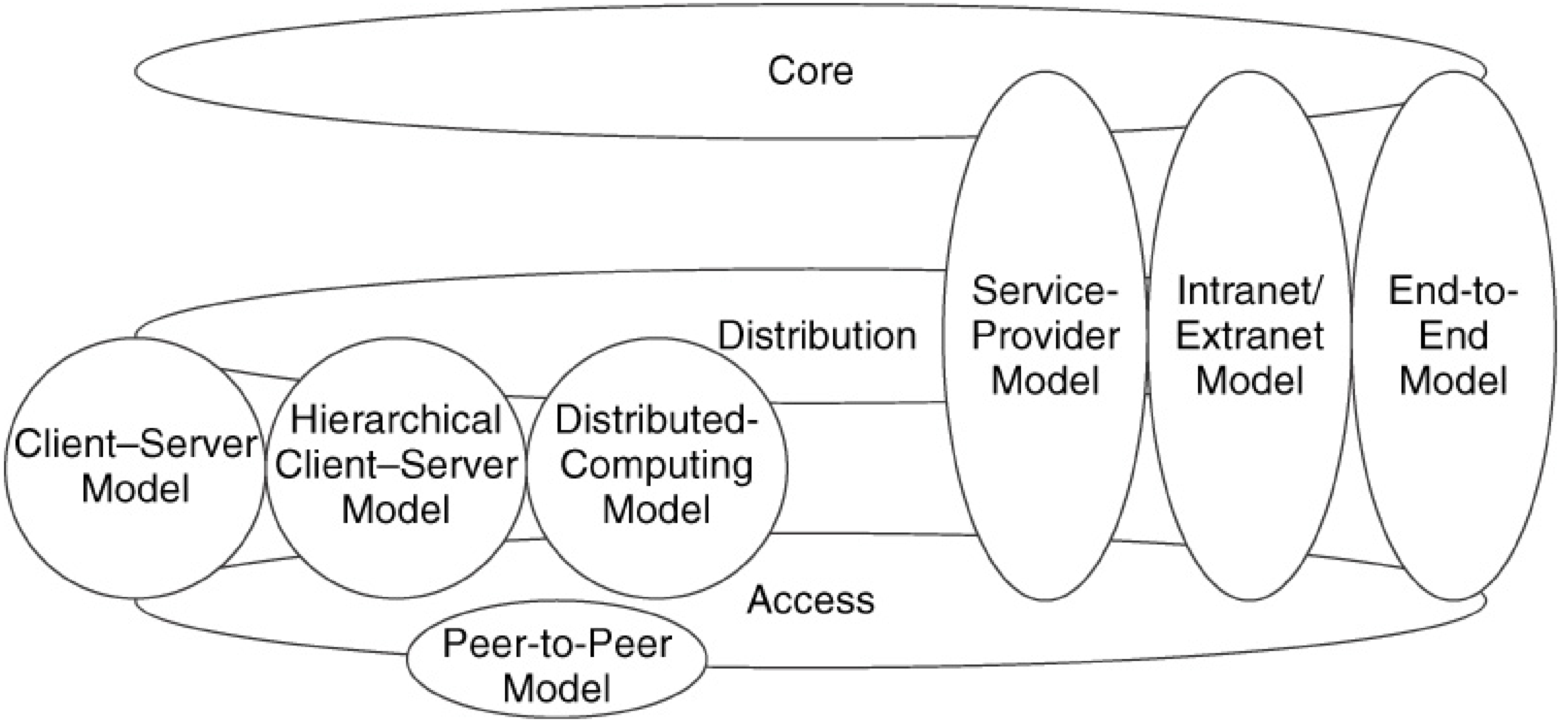
Distribution: Servers and specialized devices
Some data sources and sinks

Access: Generic computing devices
Most data sources and sinks here

Overlapping of Models

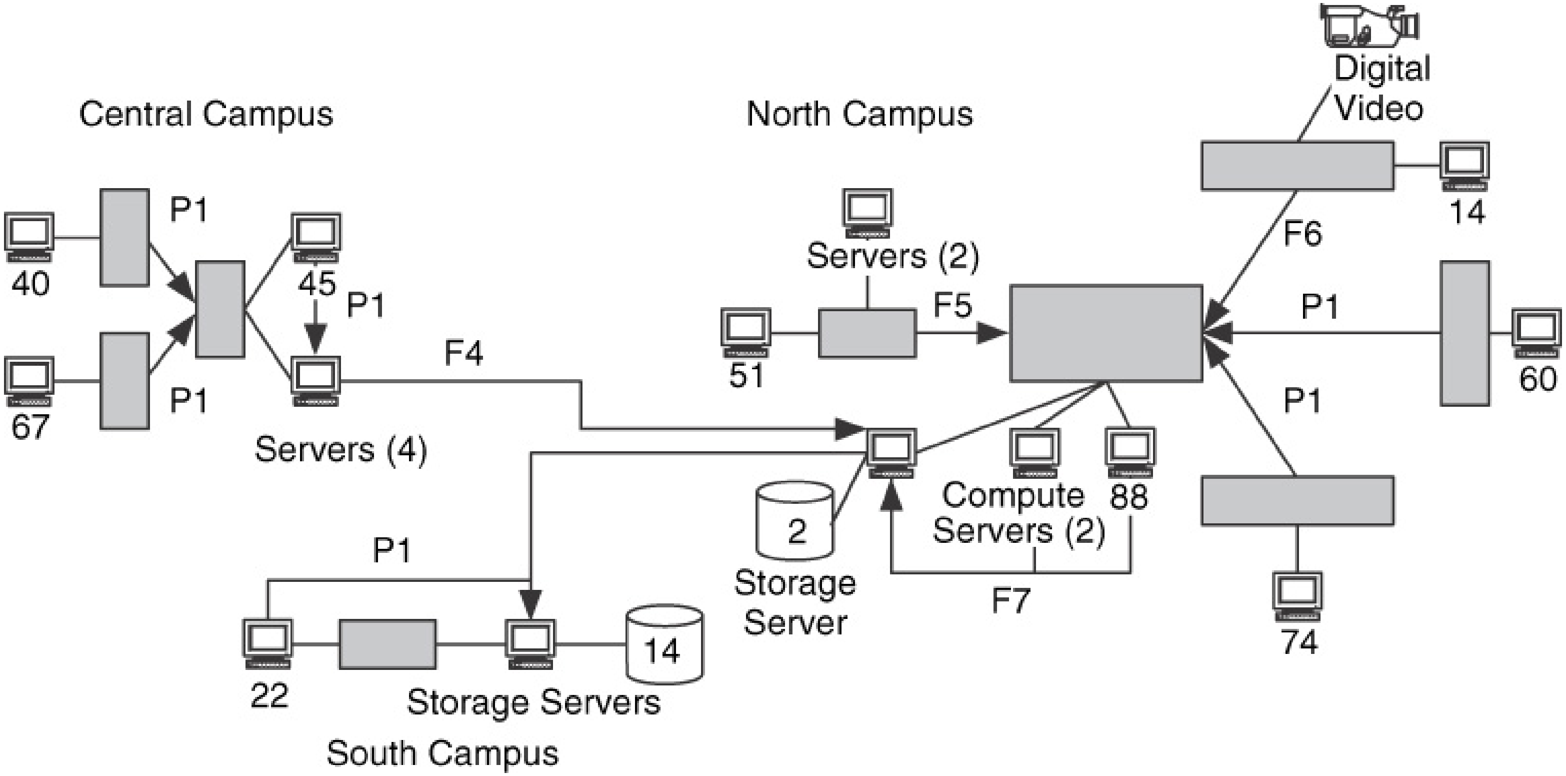


Another View of Overlapping models

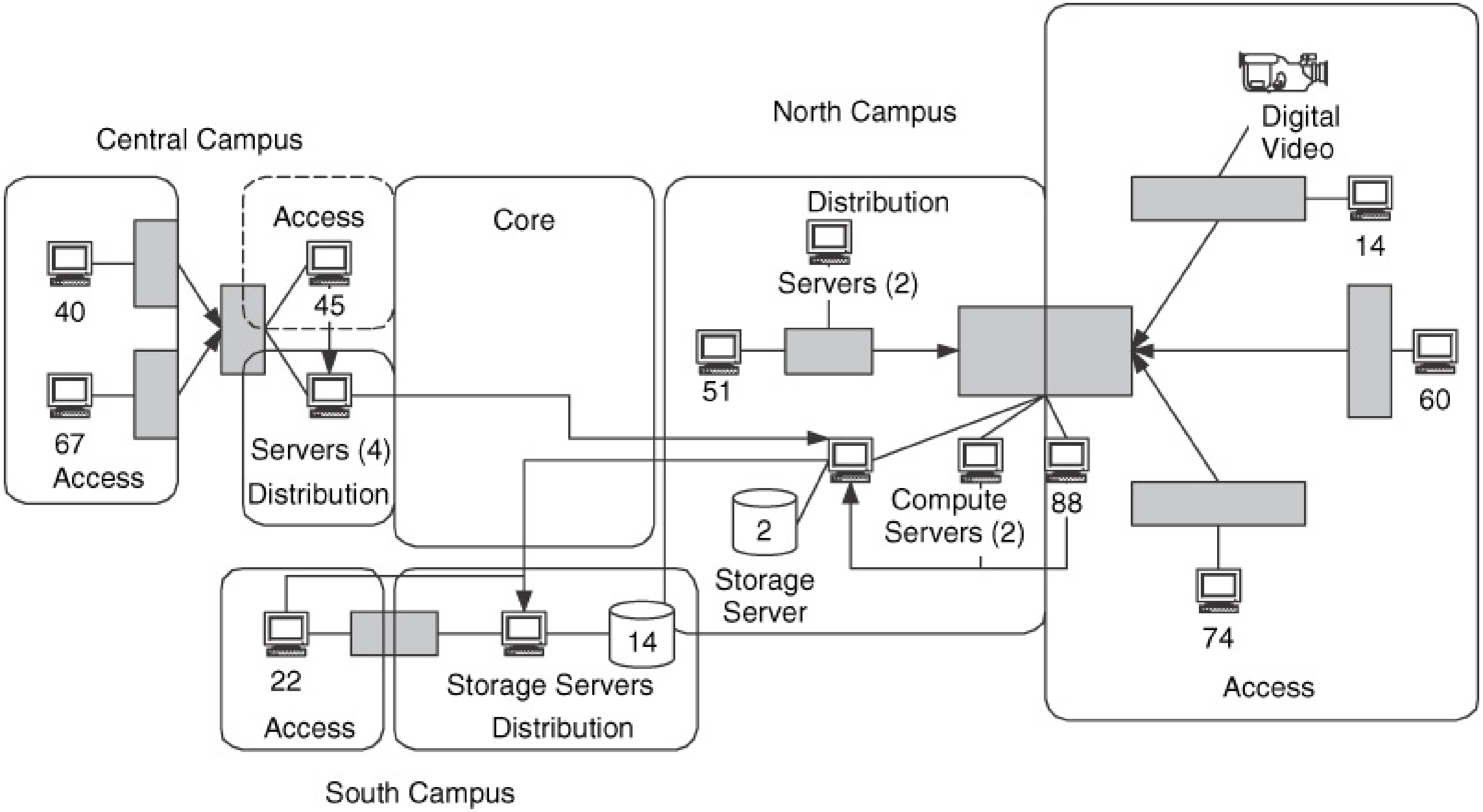


Example: Applying Architecture Models

Reconsider flow maps from storage example discussed in previous Chapter on Flow Analysis



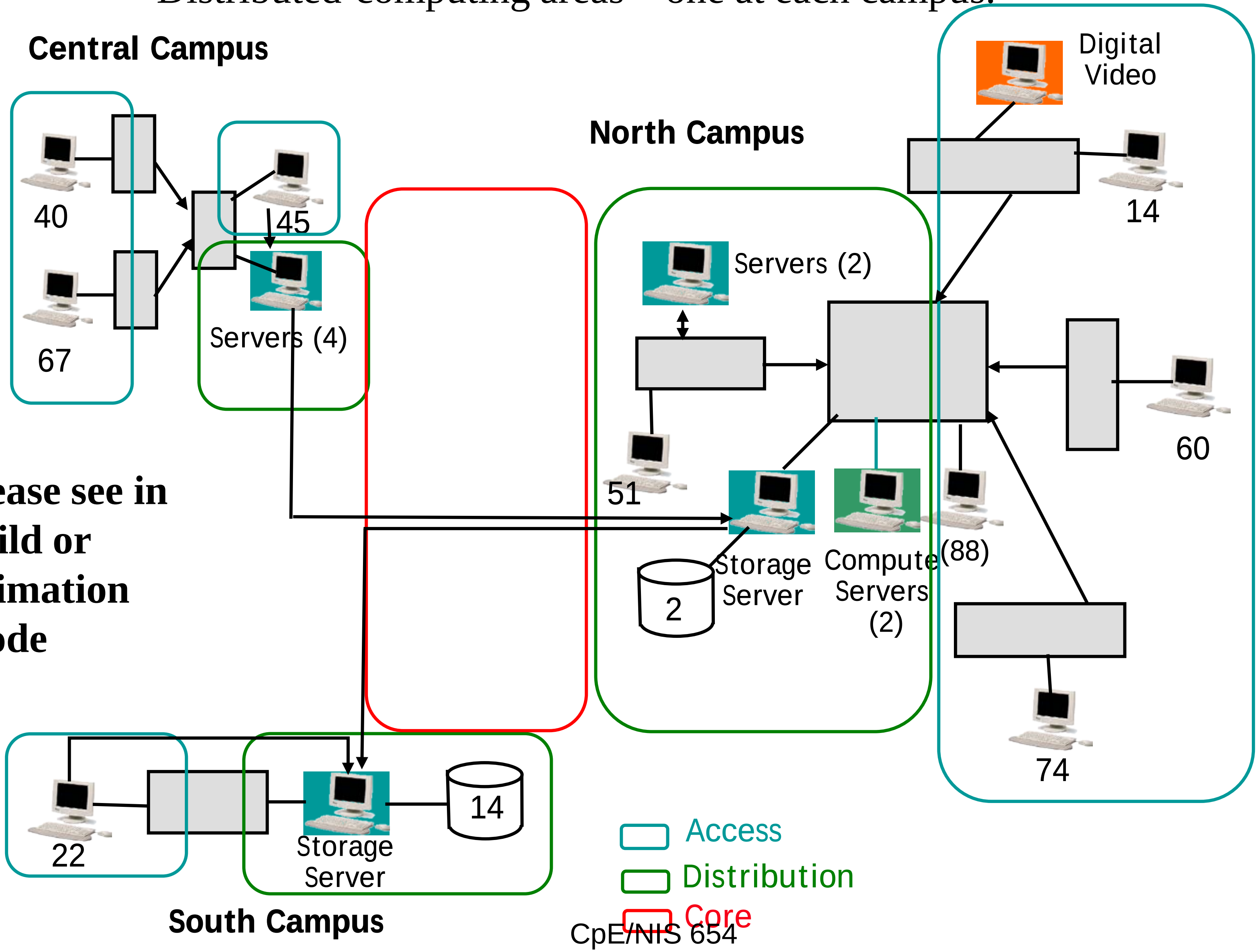
Access/distribution/core Areas for the Storage Example



Access, distribution, and core areas are shown

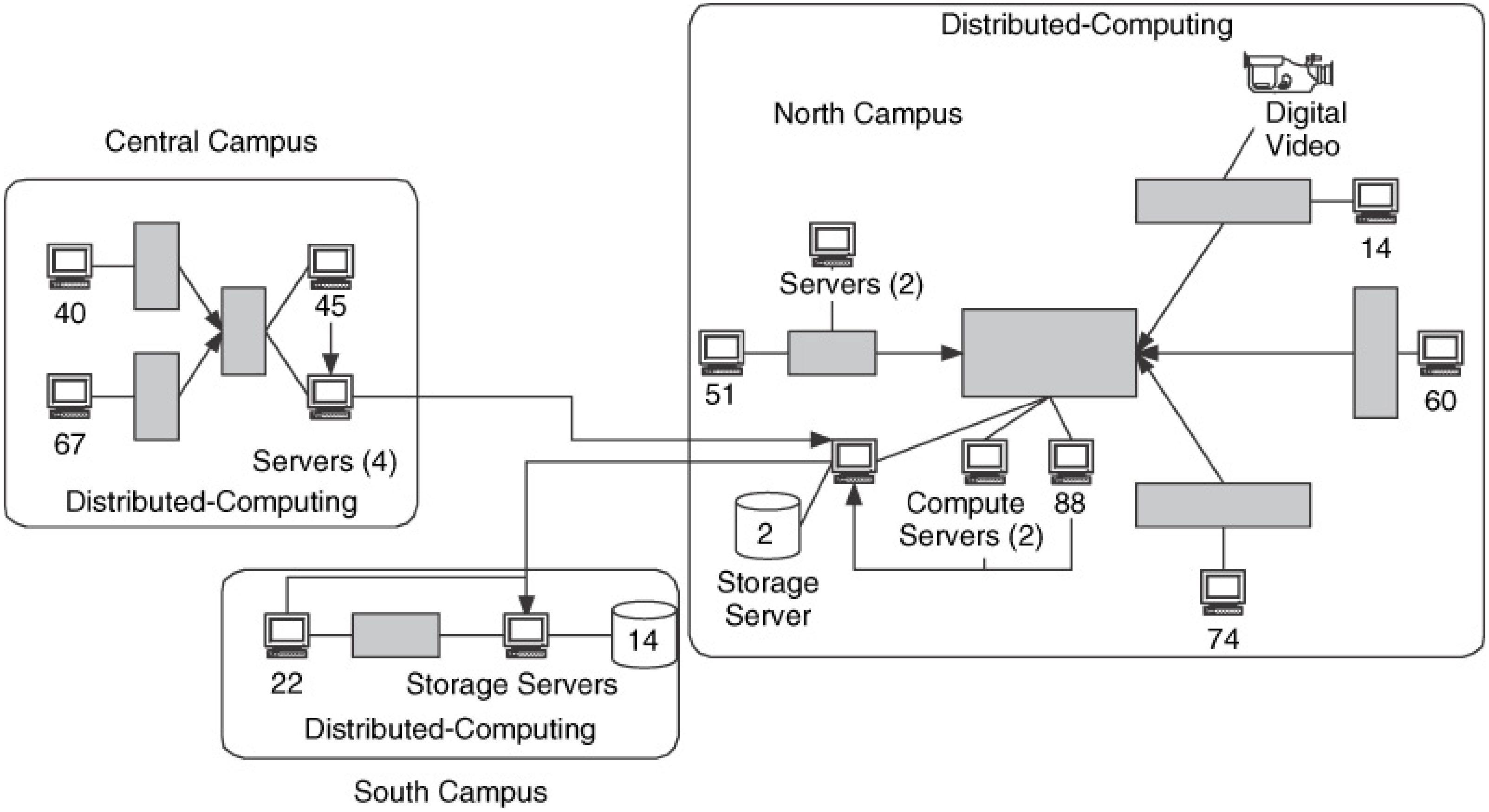
Distributed-computing model from Storage Example

Distributed-computing areas – one at each campus.



Please see in build or animation mode

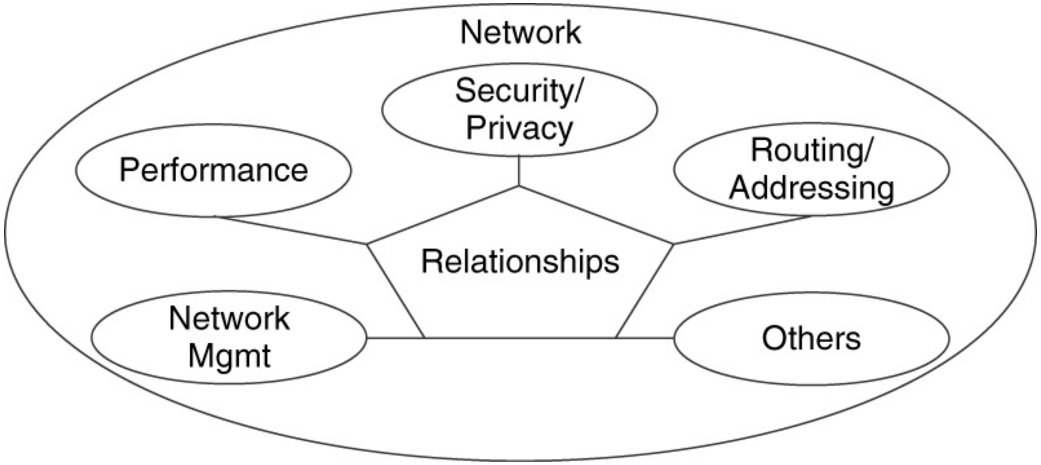
With Distributed Computing model Chosen for Storage Example



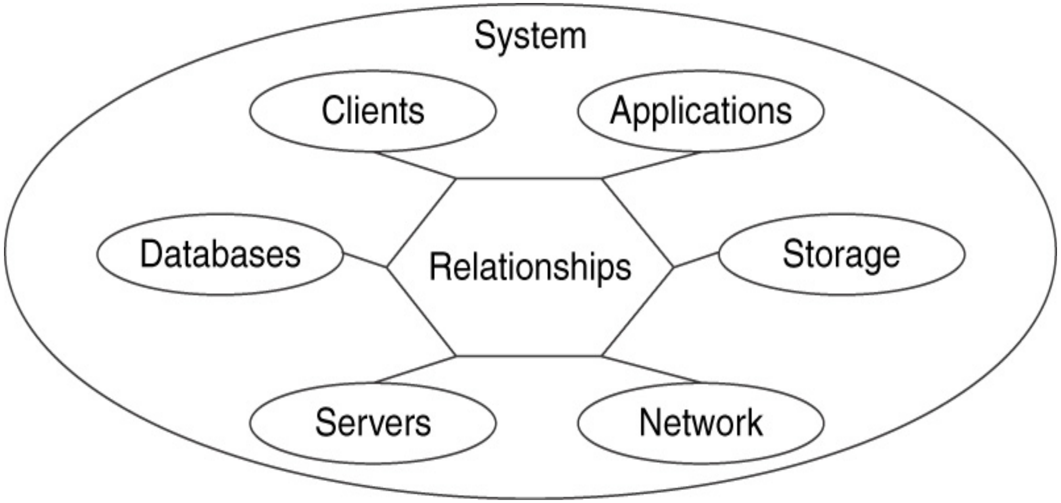
Distributed-computing areas – one at each campus.

Network versus Systems Architecture

Network architecture provides relationships between its functional component.
- Is a starting point for the system architecture.



Systems architecture considers total environment.
- is a superset of a network architecture.
- each subsystem has its own architecture.
- Network Architecture is one subsystem


System
Clients
Applications
Databases
Relationships
Storage
Servers
Network