

# Chapter 11: Physical Architecture Layer Design



# Objectives

- Understand the different physical architecture components.
- Understand server-based, client-based, and client–server physical architectures.
- Be familiar with cloud computing and Green IT.
- Be able to create a network model using a deployment diagram.
- Be familiar with how to create a hardware and software specification.
- Understand how operational, performance, security, cultural, and political requirements affect the design of the physical architecture layer.

# Introduction

- Most modern systems span two or more networked computers
- The physical architecture layer design specifies:
  - How the system will be distributed across the computers
  - What hardware and software will be used
- Most systems design is constrained by existing systems and networks
- Physical architecture design is demanding
  - Knowledge of key factors is essential
  - Nonfunctional requirements play a key role



# Elements of the Physical Architecture Layer

- Purpose is to decide which applications run on what hardware
- Process:
  - Understand the software and hardware options, then
  - Choose from the available alternatives, based on:
    - Cost of acquisition
    - Cost of development
    - Ease of development
    - Interface capabilities
    - Control & security
    - Scalability

# Architectural Components

- Software components
  - Data storage
  - Data access logic
  - Application logic
  - Presentation logic
- Hardware components
  - Clients (computers, handhelds, cell phones, etc.)
  - Servers (mainframes, minis, micros, rack mounted)
  - Networks to connect all computers (Dial-up, always-on, medium or high speed, leased phone lines, cell phone links, satellite links)

# Server-Based Architectures

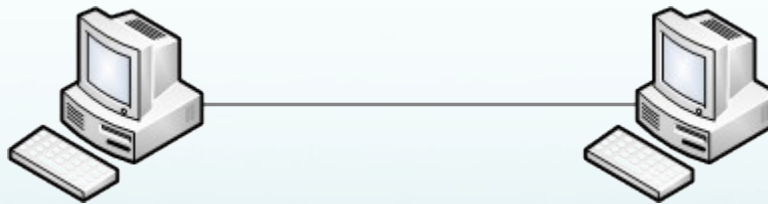
- The server performs all four application functions
- The client (usually a terminal with display and keyboard) captures keystrokes and sends them to the server for processing
- Older tech; sometimes hanging on tunneled over Internet



# Client-Based Architectures

- Clients are personal computers on a network
- Server is a file server on the same network
- Simple to develop, but quickly overloaded
  - All data is downloaded to the client for processing
  - Network traffic may become excessive
  - Client may not have enough computing power

Data Access Logic  
Application Logic  
Presentation Logic



Data Storage

# Client-Server Architectures

- Balance processing between client and server
- Predominant architecture in modern systems
- Amount of client processing varies
  - Thin clients do only presentation logic
  - Thick clients do presentation and application
- Highly scalable at incremental cost
- More complex since applications must be written for both client and server

Application Logic (Thick client)  
Presentation Logic



Data Storage  
Data Access Logic  
Application Logic  
(Thin client)



# Client-Server Tiers

- Client-server architecture tiers are defined based on how the logic is partitioned:
  - 2-tier: one server responsible for data storage and access; client responsible for application & presentation logic
  - 3-tier: data storage and access logic on one server, application logic on another; client responsible for presentation logic
  - n-tier: application logic split among two servers, data logic on another
    - Common in e-commerce applications
    - Better load balancing
    - More scalable than 2 or 3 tier systems
    - Places higher demands on the network

# Selecting a Physical Architecture

- Cost of infrastructure (initial acquisition and future growth)
- Cost of development
- Ease of development
- Interface capabilities
- Control and security
- Scalability (changes in capacity; upgrades)

# Architecture Characteristics

	<b>Server-Based</b>	<b>Client-Based</b>	<b>Client-Server</b>
Cost of infrastructure	Very high	Medium	Low
Cost of development	Medium	Low	High
Ease of development	Low	High	Low-Medium
Interface capabilities	Low	High	High
Control and Security	High	Low	Medium
Scalability	Low	Medium	High



# Cloud Computing

- Treat IT as a commodity or utility
  - Server is in the “cloud”
  - Client is on the desktop
- The “cloud”
  - A data center, internal or external; or
  - A service provided by a vendor
  - An umbrella technology that includes:
    - Virtualization
    - Service-oriented architectures
    - Grid computing

# UBIQUITOUS COMPUTING AND THE INTERNET OF THINGS

- General computing devices
  - Smartphones and tablets may have many different apps that provide all types of computing and communication support
- Specialized computing devices
  - Enchanted objects and specialized devices



# Green IT

- Anything that reduces the environmental impact of IT
- Topics:
  - E-waste (disposal of toxic materials in old computers)
  - Energy consumption of data centers and desktops
  - The paperless office
- Cloud computing may help to reduce energy consumption and improve the viability of the paperless office



# Infrastructure Design

- Although possible, few designs are from scratch
- Most designs utilize systems already in place
  - Change or improve the existing infrastructure
  - Coordination is difficult, but knowledge of elements is essential
    - Deployment diagram
    - Network model

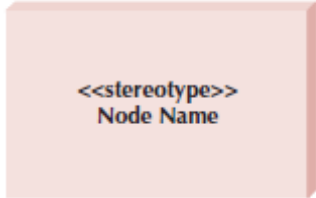
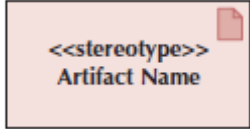
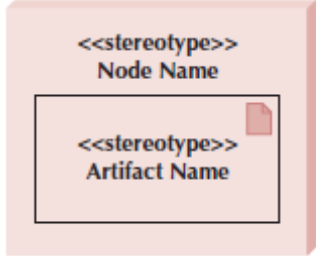
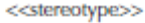


# Deployment Diagram

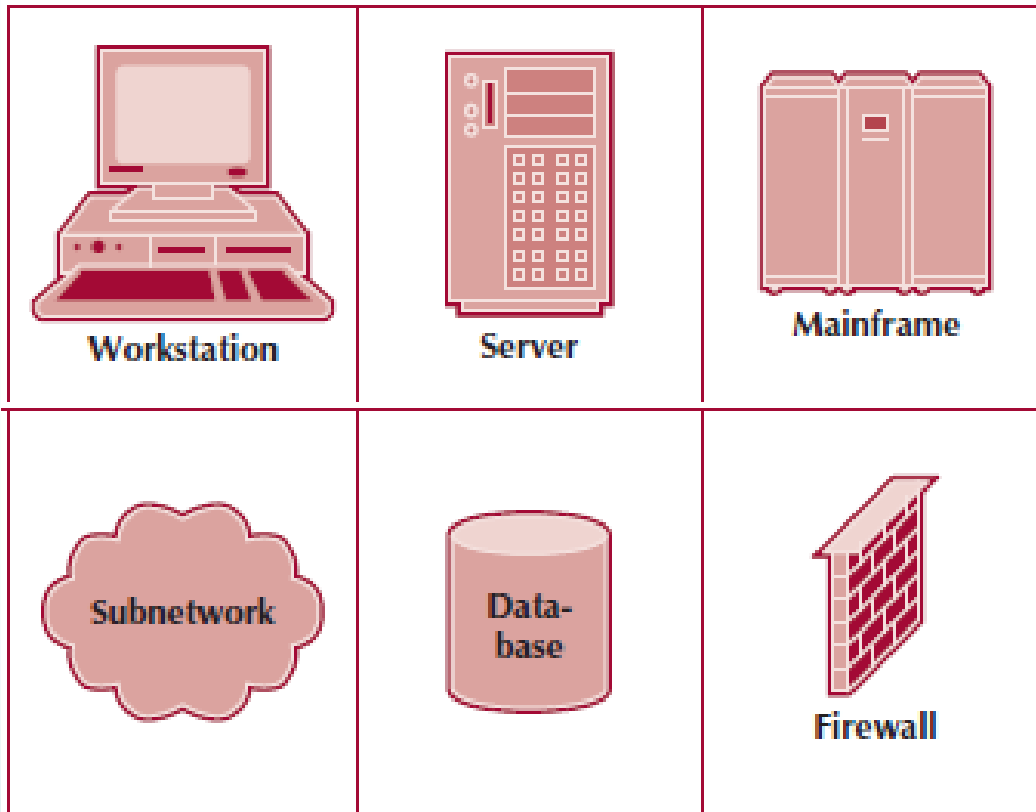
- Represent relationships between hardware components of an information system
- Elements of a deployment diagram
  - **Nodes**: any piece of hardware (e.g. client computers, servers, networks or network devices)
  - **Artifacts**: a piece of the information system which will be installed on a node
  - **Communication paths**: a communication link between the nodes



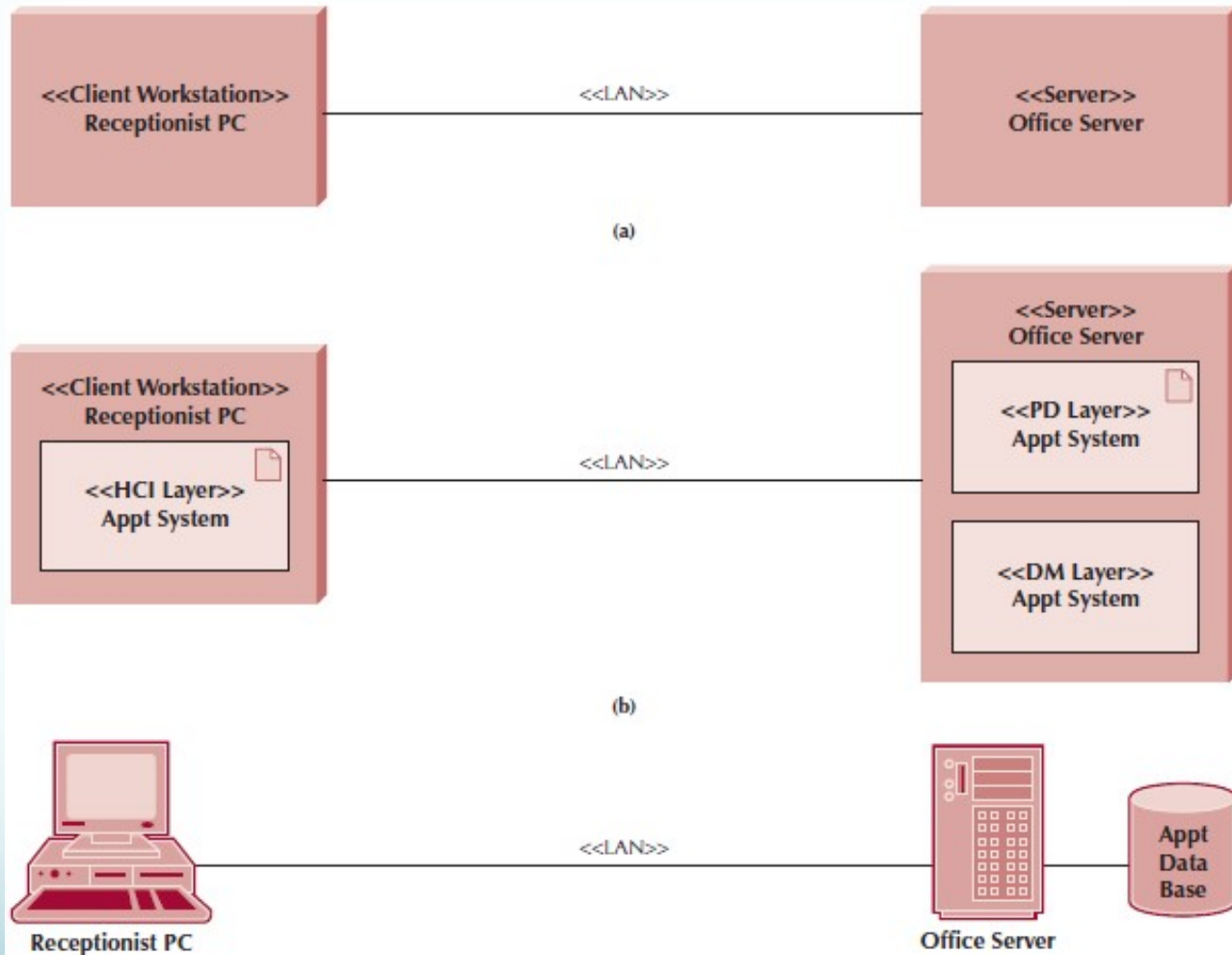
# Deployment Diagram Syntax

<p><b>A node:</b></p> <ul style="list-style-type: none"><li>■ Is a computational resource, e.g., a client computer, server, separate network, or individual network device.</li><li>■ Is labeled by its name.</li><li>■ May contain a stereotype to specifically label the type of node being represented, e.g., device, client workstation, application server, mobile device, etc.</li></ul>	
<p><b>An artifact:</b></p> <ul style="list-style-type: none"><li>■ Is a specification of a piece of software or database, e.g., a database or a table or view of a database, a software component or layer.</li><li>■ Is labeled by its name.</li><li>■ May contain a stereotype to specifically label the type of artifact, e.g., source file, database table, executable file, etc.</li></ul>	
<p><b>A node with a deployed artifact:</b></p> <ul style="list-style-type: none"><li>■ Portrays an artifact being placed on a physical node.</li></ul>	
<p><b>A communication path:</b></p> <ul style="list-style-type: none"><li>■ Represents an association between two nodes.</li><li>■ Allows nodes to exchange messages.</li><li>■ May contain a stereotype to specifically label the type of communication path being represented, (e.g., Lan, Internet, serial, parallel).</li></ul>	

# Extended Node Syntax



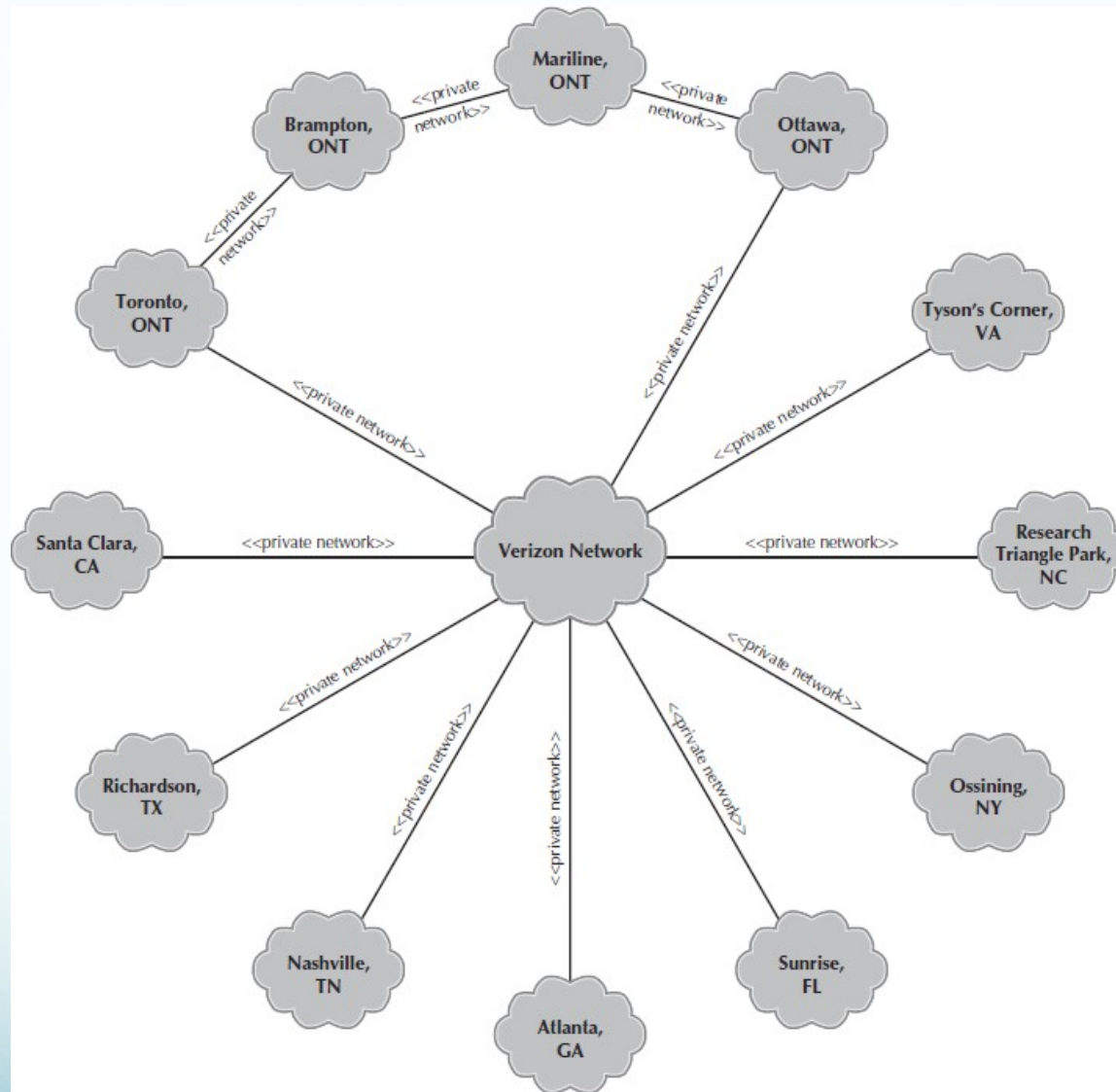
# Sample Deployment Diagrams



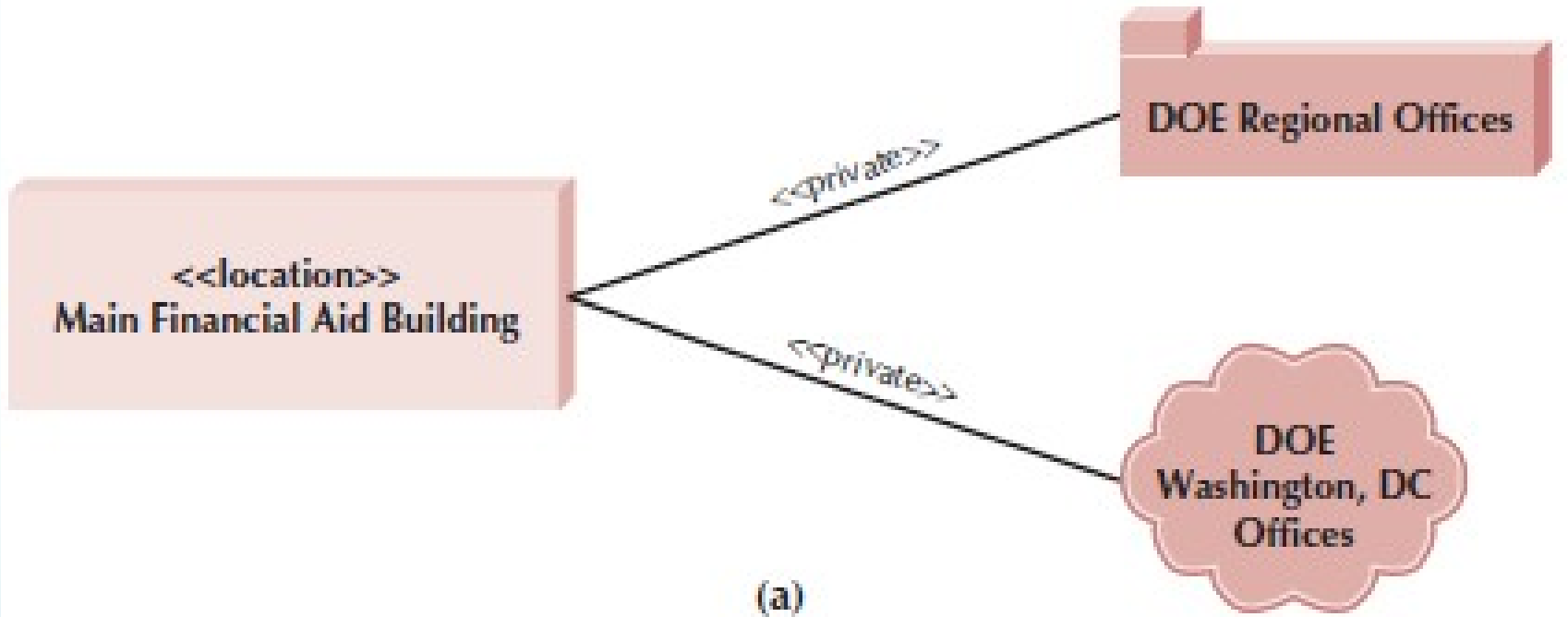
# Network Model

- A network diagram that depicts the major components and their geographic locations in the organization
- Purposes of the network model:
  - To convey the complexity of the system
  - To show how the system's software components will fit together
- Can serve as an aid for specifying hardware and software

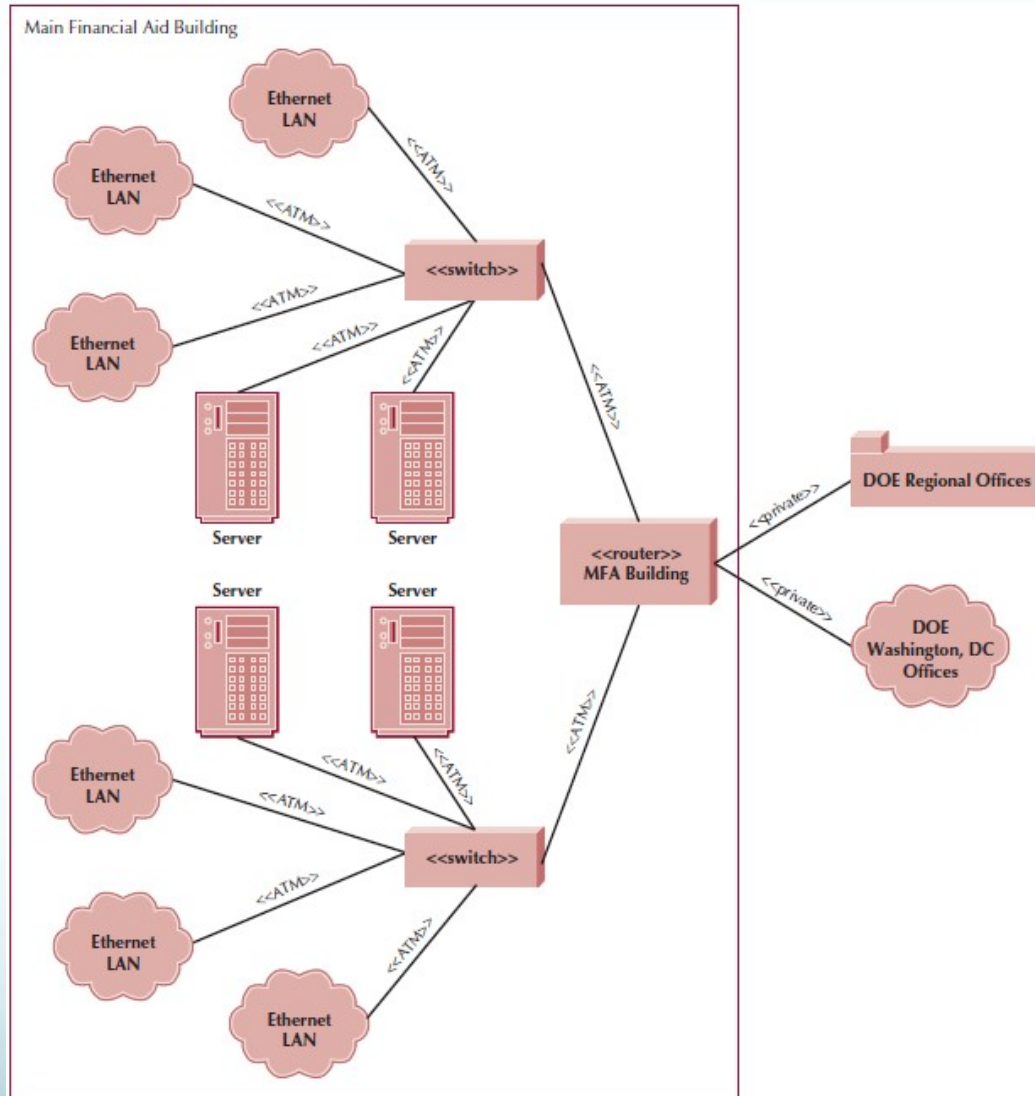
# Network Model



# Sample Network Model



# Diagram With Added Detail



# Hardware & Software Specifications

- Hardware & software needed for the new application is recorded in a specifications document
- Software requirements:
  - Operating system
  - Special purpose software (e.g., DBMS)
  - Include training needed, maintenance, warranties and licensing agreements
- Hardware requirements
  - Use low level network diagram as a starting point
  - Include type & quantity of servers, peripherals, storage & backup devices
  - Describe minimum requirements
  - Use an alternative matrix to evaluate vendor proposals



# Nonfunctional Requirements

- Operational
  - Technical environment
  - System integration
  - Portability
  - Maintainability
- Performance
  - Speed
  - Capacity
  - Availability & reliability
- Security
  - System value
  - Access control
  - Encryption & authentication
  - Virus control
- Cultural & political influence
  - Centralized vs. local control
  - Language differences (keyboard requirements)
- Legal implications
  - Laws & government regulations
  - Global presence requires scrutiny of local laws

# Operational Requirements

Type of Requirement	Definition	Examples
<b>Technical Environment Requirements</b>	Special hardware, software, and network requirements imposed by business requirements	<ul style="list-style-type: none"><li>• The system will work over the Web environment with Internet Explorer.</li><li>• All office locations will have an always-on network connection to enable real-time database updates.</li><li>• A version of the system will be provided for customers connecting over the Internet via a tablet or smartphone.</li></ul>
<b>System Integration Requirements</b>	The extent to which the system will operate with other systems	<ul style="list-style-type: none"><li>• The system must be able to import and export Excel spreadsheets.</li><li>• The system will read and write to the main inventory database in the inventory system.</li></ul>
<b>Portability Requirements</b>	The extent to which the system will need to operate in other environments	<ul style="list-style-type: none"><li>• The system must be able to work with different operating systems (e.g., Linux, Mac OS, and Windows).</li><li>• The system might need to operate with handheld devices such as a Android and Apple iOS devices.</li></ul>
<b>Maintainability Requirements</b>	Expected business changes to which the system should be able to adapt	<ul style="list-style-type: none"><li>• The system will be able to support more than one manufacturing plant with six months' advance notice.</li><li>• New versions of the system will be released every six months.</li></ul>

# Performance Requirements

Type of Requirement	Definition	Examples
<b>Speed Requirements</b>	The time within which the system must perform its functions	<ul style="list-style-type: none"><li>• Response time must be less than 7 seconds for any transaction over the network.</li><li>• The inventory database must be updated in real time.</li><li>• Orders will be transmitted to the factory floor every 30 minutes.</li></ul>
<b>Capacity Requirements</b>	The total and peak number of users and the volume of data expected	<ul style="list-style-type: none"><li>• There will be a maximum of 100–200 simultaneous users at peak use times.</li><li>• A typical transaction will require the transmission of 10K of data.</li></ul>
<b>Availability and Reliability Requirements</b>	The extent to which the system will be available to the users and the permissible failure rate due to errors	<ul style="list-style-type: none"><li>• The system will store data on approximately 5,000 customers for a total of about 2 MB of data.</li><li>• Scheduled maintenance shall not exceed one 6-hour period each month.</li><li>• The system shall have 99% uptime performance.</li></ul>

# Security Requirements

Type of Requirement	Definition	Examples
<b>System Value Estimates</b>	Estimated business value of the system and its data	<ul style="list-style-type: none"><li>• The system is not mission critical but a system outage is estimated to cost \$50,000 per hour in lost revenue.</li><li>• A complete loss of all system data is estimated to cost \$20 million.</li></ul>
<b>Access Control Requirements</b>	Limitations on who can access what data	<ul style="list-style-type: none"><li>• Only department managers will be able to change inventory items within their own department.</li><li>• Telephone operators will be able to read and create items in the customer file but cannot change or delete items.</li></ul>
<b>Encryption and Authentication Requirements</b>	Defines what data will be encrypted Where and whether authentication will be needed for user access	<ul style="list-style-type: none"><li>• Data will be encrypted from the user's computer to the website to provide secure ordering.</li><li>• Users logging in from outside the office will be required to authenticate.</li></ul>
<b>Virus Control Requirements</b>	Requirements to control the spread of viruses	<ul style="list-style-type: none"><li>• All uploaded files will be checked for viruses before being saved in the system.</li></ul>

# Cultural & Political Requirements

Type of Requirement	Definition	Examples
<b>Customization Requirements</b>	Specification of what aspects of the system can be changed by local users	<ul style="list-style-type: none"><li>• Country managers will be able to define new fields in the product database to capture country-specific information.</li><li>• Country managers will be able to change the format of the telephone number field in the customer database.</li></ul>
<b>Legal Requirements</b>	The laws and regulations that impose requirements on the system	<ul style="list-style-type: none"><li>• Personal information about customers cannot be transferred out of European Union countries into the United States.</li><li>• It is against U.S. federal law to divulge information on who rented what videotape, so access to a customer's rental history is permitted only to regional managers.</li></ul>

# Summary

- Elements of the Physical Architecture Layer
- Cloud Computing
- Green IT
- Ubiquitous computing and the internet of things
- Infrastructure Design
- Hardware & Software Specifications
- Nonfunctional Requirements

