System Analysis and Design

Eighth Edition

Alan Dennis, Barbara Wixom, Roberta M. Roth

Chapter 7

Architecture Design

Objectives

- Describe the fundamental components of an information system.
- Describe client–server, server- based, and mobile application architectures.
- Describe how cloud computing can be incorporated as a system architecture component.
- Explain how operational, performance, security, cultural, and political requirements affect the architecture design.
- Create a hardware and software specification.

Introduction

- Architecture design
  - Plans for how the system will be distributed across computers and what hardware and software will be used for each computer
- Hardware and software specification
  - Describes the hardware/software components in detail to aid those responsible for purchasing those products

Elements of an Architecture Design

- Objective is to assign the software components of the information system to the hardware devices of the system in the most advantageous way
- The major architectural components of any system are the software and the hardware
- Software systems can be divided into four basic functions:
  1. ***Data storage***: Most information systems require data to be stored and retrieved
  2. ***Data access logic***: the processing required to access stored data
  3. ***Application logic***: the logic documented in the D F Ds, use cases, and functional requirements
  4. ***Presentation logic***: the display of information to the user and the acceptance of the user's commands

Three Primary Hardware Components

1. ***Client computers***: Input-output devices employed by users (e.g., PCs, laptops, handheld and mobile devices, smart phones)
2. ***Servers***: Larger multi-user computers used to store software and data
3. ***Network***: Connects the computers

Client–Server Architectures

- Client-server architectures balance the processing between client devices and one or more server devices
- Generally, clients are responsible for the presentation logic
- The server(s) are responsible for the data access logic and data storage
- Application logic location varies depending on the C-S configuration chosen

Four Benefits of Client-Server

1. Scalable: it is easy to increase or decrease the storage and processing capabilities of the servers
2. Can support different types of clients and servers through middleware
   - ***Middleware*** is a type of system software designed to translate between different vendors' software
3. The presentation logic, the application logic, and the data processing logic can be independent
   - If the thin client-server architectures use Internet standards
4. If a server fails, only the applications requiring that server are affected – highly reliable

Client–Server Tiers

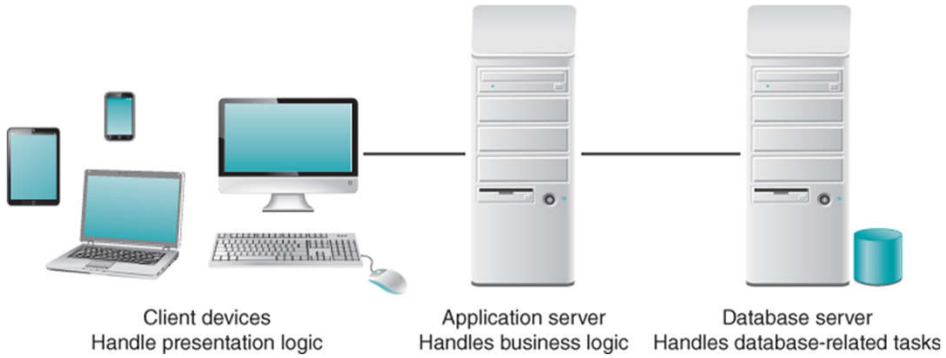
- Two-tiered
- Three-tiered
- n-tiered

Two-Tiered Client-Server Architecture

- Thick client – most of application logic on the client side (shown here)
- Thin client – little application logic on the client side; most shifted to server side

Three-Tiered Client-Server Architecture

- Adds “specialized” servers
- Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface
- The application logic may consist of one or more separate modules running on a workstation
- A relational DBMS running on a database server contains the data access logic and data storage



n-Tiered Client-Server Architecture

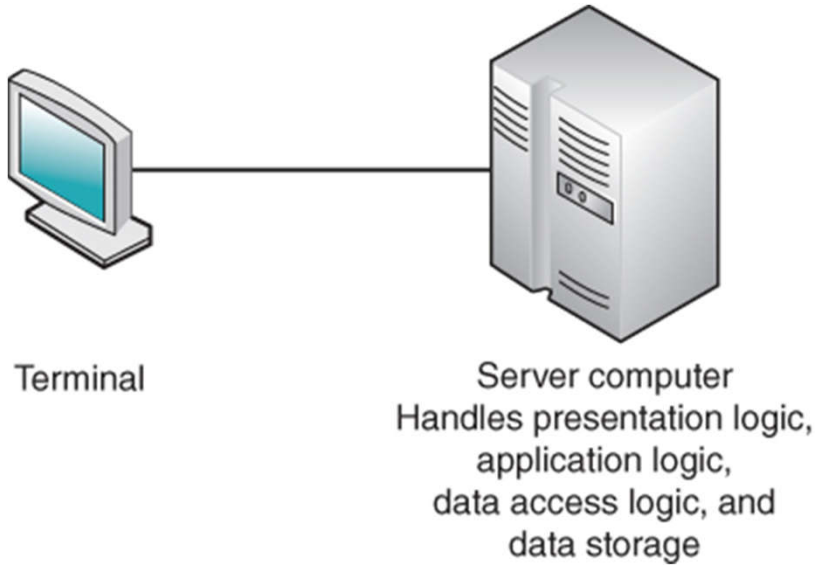
- Adds “specialized” servers
- Distributes the work of the application (the middle tier) among multiple layers of more specialized server computers
- The primary advantage of an n-tiered client–server architecture is that it separates out the processing that occurs to better balance the load

Adding “Tiers” in the Architecture

ADVANTAGES	DISADVANTAGES
• Modular business logic components are shareable across applications • Separating the processing among multiple servers makes it possible to balance the server loads efficiently.	• More tiers place a higher load on the network. • More difficult to implement since the servers must communicate effectively.

Server-Based Architecture

- The very first computing architectures were server-based
- The server was a mainframe computer and the client was a “dumb” terminal
- This remarkably simple architecture often works very well
- Server-based system architectures are common today in situations where systems process very high transaction volumes and strong security is required



Zero Client (a.k.a. Ultrathin Client)

- Zero client is a server-based computing model that is often used today in a virtual desktop infrastructure (VDI)
- Typically a small box that connects a keyboard, mouse, monitor, and Ethernet connection to a remote server
- The server hosts everything
- Benefits:
  - Reduced power usage
  - Less expensive devices
  - No vulnerability to malware
  - Efficient and secure
  - Administration is easy
  - Limits nonbusiness uses of the client computer

Mobile Application Architecture

- Rich client – involves processing on the mobile device using its resources. Presentation logic, business logic, and data access logic on the client side
- Thin Web-based client – business and data access logic on the server side; always connected to server
- Rich Internet application – browser-based; uses some technologies on client device to provide a rich user interface

Mobile Application Options

- Native app – written to run on specific device with specific operating system
- Cross-platform frameworks – develop in web-based technologies and use framework to deploy to multiple devices
- Mobile Web app – browser-based; platform independent
  - Most limited user experience

Advances in Architecture Configurations

- Advances in hardware, software, and networking have given rise to many new architecture options
- Virtualization: Creation of a virtual device or resource
- Cloud computing: Computing resources obtained as a service

Virtualization

- Server virtualization involves partitioning a physical server into smaller virtual servers
- Software is used to divide the physical server into multiple virtual environments
- A physical server device can be used to provide many virtual servers that are independent of each
- Storage virtualization involves combining multiple network storage devices into what appears to be single storage unit
- A storage area network (SAN) uses storage virtualization to create a high-speed subnetwork of shared storage devices

Cloud Computing

- Cloud computing – everything from computing power to computing infrastructure, applications, business processes to personal collaboration can be delivered as a service wherever and whenever needed
- Cloud computing can be implemented in three ways: private cloud, public cloud, and hybrid clouds
- The “cloud” can be defined as the set of hardware, networks, storage, devices, and interfaces that combine to deliver aspects of computing as a service

Advantages of Cloud Computing

- Elasticity: the resources allocated can be increased or decreased quickly, based on demand
- Cloud customers can obtain cloud resources in a straightforward fashion
- Cloud services typically have standardized APIs (application program interfaces)
- Customers are billed for resources as they are used

Comparing Architecture Options

- Most systems are built to use the existing infrastructure in the organization, so often the current infrastructure restricts the choice of architecture
- Each of the architectures discussed has its strengths and weaknesses
- Client-server architectures are strongly favored on the basis of the cost of infrastructure
- Cloud computing deserves consideration today

Creating an Architecture Design

- The architecture design specifies the overall architecture and the placement of software and hardware that will be used
- Architecture design is an extraordinarily complex process
- Most organizations are moving to client–server architectures for cost and scalability reasons
- Creating an architecture design begins with the nonfunctional requirements
- There are four primary types of nonfunctional requirements that can be important in designing the architecture
  1. Operational requirements
  2. Performance requirements
  3. Security requirements
  4. Cultural and political requirements

Operational Requirements

Requirement	Definition	Example
Technical Environment	Special hardware, software, and network requirements imposed by business requirements	All office locations have always-on network connection permitting real-time database updates
System Integration	The extent to which the system will operate with other systems	The system will read and write to the main inventory database
Portability	The extent to which the system will need to operate in other environments	The system must operate with mobile devices (Android and iOS)
Maintainability	Expected business changes to which the system should be able to adapt	The system must accommodate new manufacturing plants

Performance Requirements

Requirement	Definition	Example
Speed	Time within which the system must perform its function	Network transaction response time <= 4 seconds
Capacity	Total and peak number of users and the volume of data expected	Maximum of 2000 simultaneous users at peak use times
Availability and Reliability	Extent to which the system will be available to the users and the permissible failure rate due to errors	99% uptime performance

Security Requirements

Requirement	Definition	Example
System Value Estimates	Estimated business value of the system and its data	A complete loss of all system data would cost $20 million
Access Control	Limitations on who can access what data	Inventory item changes can be made only by managers for items in their own department
Encryption and Authentication	Defines what data will be encrypted where and whether authentication will be needed for user access	Data will be encrypted from the user's computer to the Web site to provide secure ordering
Virus Control	Controls to limit viruses	All uploaded files will be checked for viruses before being saved in the system

Encryption and Authentication Requirements

- One of the best ways to prevent unauthorized access to data is encryption
  - A means of disguising information by the use of mathematical algorithms (or formulas)
- There are two fundamentally different types of encryption: symmetric and asymmetric
- A symmetric encryption algorithm is one in which the key used to encrypt a message is the same as the one used to decrypt it
- In an asymmetric encryption algorithm, the key used to encrypt data is different from the one used to decrypt it
- Public key encryption also permits authentication (or digital signatures)

Cultural/Political Requirements

Requirement	Definition	Example
Multilingual	The language(s) the system users will need	The system will operate in English, French, and Spanish
Customization	Specification of what aspects of the system can be changed by local users	Country managers will be able to define new fields in the product database to capture country-specific information
Making Unstated Norms Explicit	Explicitly stating assumptions that differ from country to country	All weights will be stated in pounds and in kilograms
Legal	The laws and regulations that impose system requirements	Personal customer information cannot be transferred from European Union countries to US

Cultural and Political Requirements

- Cultural and political requirements are specific to the countries in which the system will be used
- Global applications often have multilingual requirements, which means that they must support users who speak different languages and write with non-English letters
  - The other challenge often is screen space
- For global applications, the project team will need to give some thought to customization requirements: how much of the application will be controlled by a central group and how much of the application will be managed locally
- Many countries have unstated norms that are not shared internationally
- Legal requirements are imposed by laws and government regulations

Designing the Architecture

- Technical environment requirements, driven by business requirements, often define the application architecture
- If not, other nonfunctional requirements become important

Nonfunctional Requirements and the Architecture Design

Requirements	Server-Based	Thin Client-Server	Thick Client-Server
Operational Requirements			
System Integration Requirements	✓	✓	✓
Portability Requirements		✓	
Maintainability Requirements	✓	✓	
Performance Requirements			
Speed Requirements		✓	✓
Capacity Requirements		✓	✓
Availability/Reliability Requirements	✓	✓	✓
Security Requirements			
High System Value	✓	✓	
Access Control Requirements	✓		
Encryption/Authentication Requirements		✓	✓
Virus Control Requirements	✓		
Cultural/Political Requirements			
Multilingual Requirements		✓	
Customization Requirements		✓	
Making Unstated Norms Explicit		✓	
Legal Requirements	✓	✓	✓

Hardware Software Specification

- The hardware and software specification is a document that describes what hardware and software are needed to support the application
- First, you will need to define the software that will run on each component
- Next, you must create a list of the hardware needed to support the future system
- Finally, you need to describe, in as much detail as possible, the minimum requirements for each piece of hardware

Sample Hardware and Software Specifications

	Standard Client	Standard Web Server	Standard Application Server	Standard Database Server
Operating System	•Windows 10 Pro	•Linux	•Linux	•Linux
Special Software	•Real Audio •Adobe Acrobat Reader	•Apache	•Java	•Oracle
Hardware	•1 TB disk drive	•8 TB disk drive	•8 TB disk drive	•32 TB disk drive
	•Intel®-Core™ i5-8400 six core processor	•Xeon E5-4600 v4	•Xeon E5-4600 v4	•RAID
	•22-inch LED Monitor			•Xeon 28 core processor
Network	•Always-on Broadband, preferred	•Dual 100 Mbps Ethernet	•Dual 100 Mbps Ethernet	•Dual 100 Mbps Ethernet

Factors in Hardware and Software Selection

Factor	Definition
Functions and Features	What specific functions and features are needed (e.g., size of monitor, software features)?
Performance	How fast do the hardware and software operate (e.g., processor, number of database writes per second)?
Legacy Databases and Systems	How well do the hardware and software interact with legacy systems (e.g., can it write to this database)?
Hardware and OS Strategy	What are the future migration plans (e.g., the goal is to have all of one vendor's equipment)?
Cost of Ownership	What are the costs beyond purchase (e.g., incremental license costs, annual maintenance, training costs, and salary costs)?
Political Preferences	People are creatures of habit and are resistant to change, so changes should be minimized.
Vendor Performance	Some vendors have reputations or prospects that are different from those of a specific hardware or software system that they currently sell.

Chapter Review

- Identify and describe the four basic functions of a software system.
- Describe the basic premise of the client–server architecture.
- Explain the difference between thick clients and thin clients.
- Discuss the “tiered” approach to client–server architectures.
- Explain the architectural options for mobile applications.
- Discuss several ways to create mobile applications and the pros and cons of each.
- Explain the use of nonfunctional requirements in the creation of the architecture design.
- Describe the purpose and typical content of the hardware and software specification.

Key Terms

- 24/7
- Access control requirements
- Application logic
- Application program interfaces
- (API)
- Architectural components
- Architecture design
- Asymmetric encryption algorithm
- Authentication
- Availability and reliability requirements
- Blockchain
- Capacity requirements
- Certificate authority (CA)
- Client computers
- Client–server architectures cloud computing
- Cloud computing
- Concurrent multilingual system
- Controls
- Cultural and political requirements
- Customization requirements
- Data access logic
- Data storage
- Discrete multilingual system
- Elasticity
- Encryption
- Encryption and authentication requirements
- Fat client
- Functions
- Graphical user interface (GUI)

Key Terms Continued

- Hardware and software specification
- Invertible
- Legal requirements
- Mainframe
- Maintainability requirements
- Microcomputer
- Middleware
- Mission critical system
- Multilingual requirements
- Network
- n-tier architecture
- Performance requirements
- Portability requirements
- Presentation logic
- Private key
- Public key
- Public key encryption
- Response time
- Rich client
- Rich Internet application (RIA)
- Scalable
- Security requirements
- Servers
- Server-based architecture
- Server virtualization
- Speed requirements
- Storage area network (SAN)

Key Terms Continued

- Storage virtualization
- Symmetric encryption algorithm
- System integration requirements
- System Value
- Technical environment requirements
- Terminal
- Thick client
- Thin client
- Thin Web-based client
- Three-tiered architecture
- Two-tiered architecture
- Ultra thin client
- Unstated norms
- Virtualization
- Virus
- Virus Control Requirements
- Zero client