Client/Server Computing

See chapter 7, Stalling's book

Ho Sooi Hock

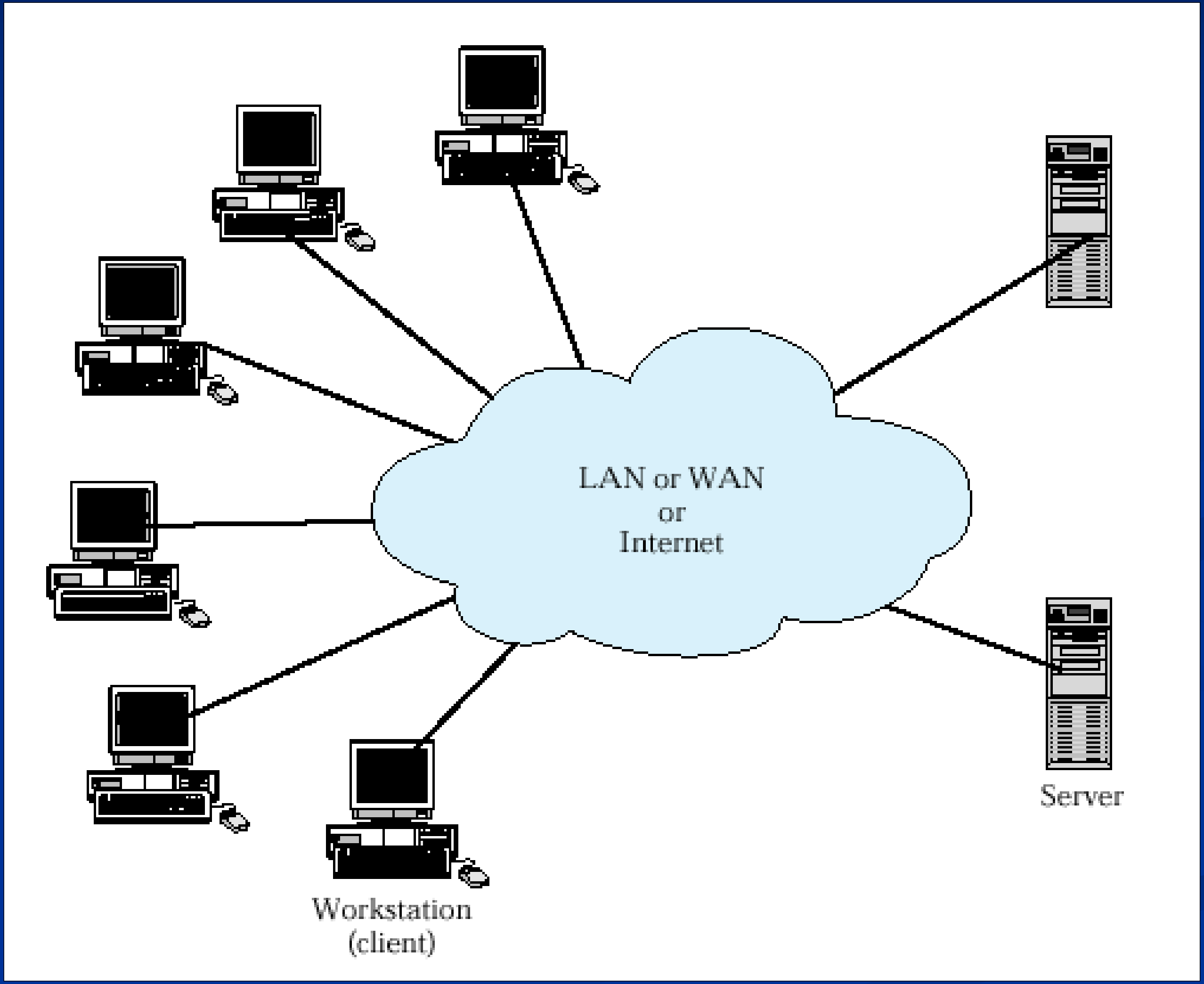
Outlines

- Elements of Client/Server Computing
- Reasons for Growing Interest of Client/Server Paradigm
- Characteristics of Client/Server Computing
- Classes of Client/Server Applications
- Role of Middleware
- Three Middleware Mechanisms

Elements of Client/Server Computing

- An environment populated by clients and servers
  - Clients are generally single-user workstations that provide user friendly interfaces
  - Servers provide a set of shared user services to the clients, e.g. database server
- As a form of distributed computing, networking is the third essential ingredient linking the distributed resources

Client/Server Environment



Why Client/Server Computing is Gaining Popularity?

- Trend # 1:
  - An effort by companies to downsize and streamline processes in a competitive business environment
- Trend # 2:
  - The introduction of the small business units (SBUs) concept

Client/Server computing is a new technical approach providing new ways of organising business, automating tasks and eliminating barriers to information

Characteristics of Client/Server Computing

- It brings user friendly applications to the desktops, giving users greater control over timing and style of computer usage, and allowing departments to be more responsive to local needs
- Enables databases, network management and utility functions to be centralised
- Commitment to open and modular systems by the user organisations and vendors
- Networking is fundamental to the operation, thus good network management and security functions are necessary

Generic Client/Server Architecture

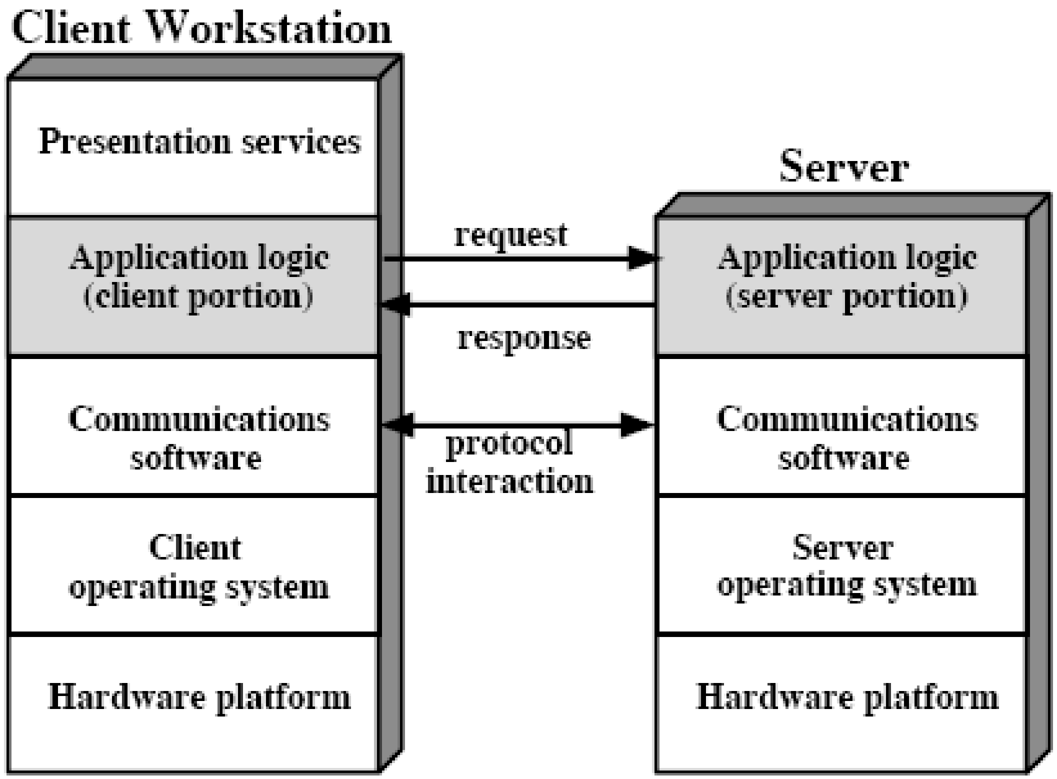


Figure 7.2 Generic Client/Server Architecture

Client/Server Architecture for Database Applications

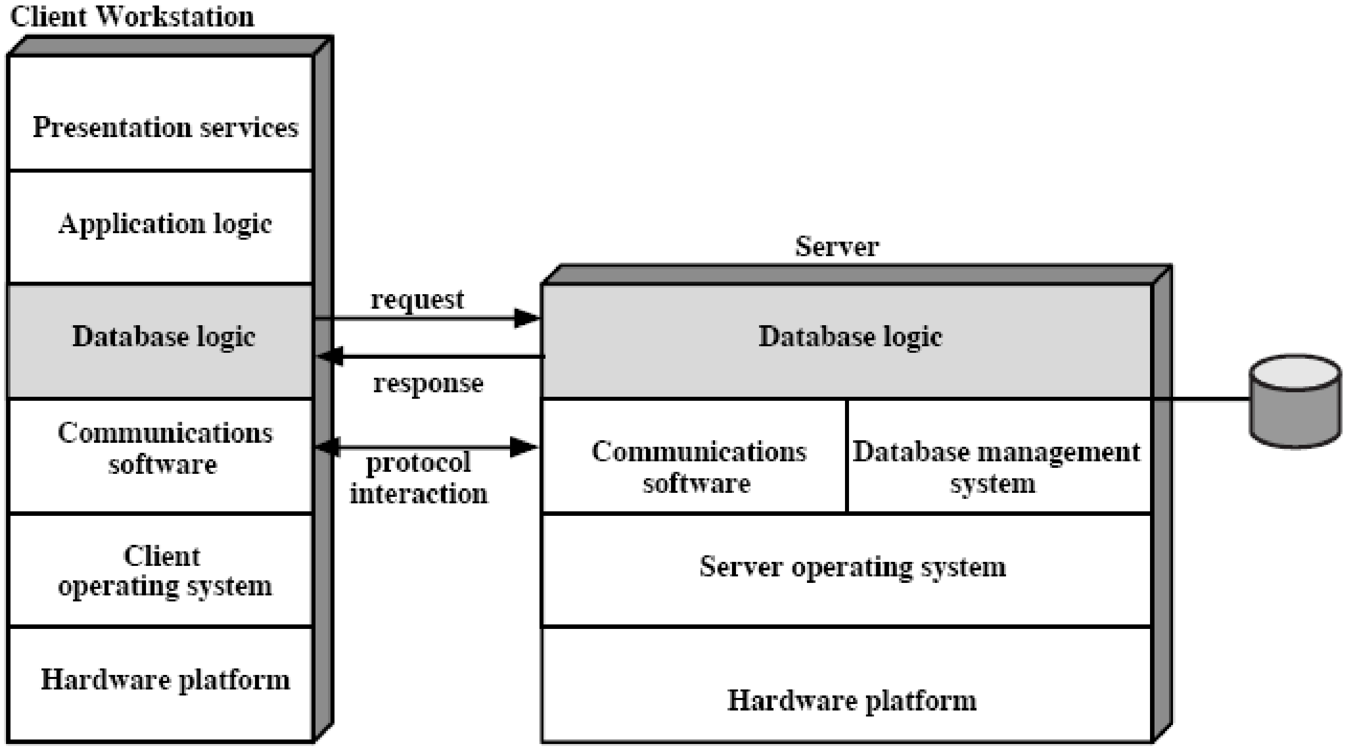
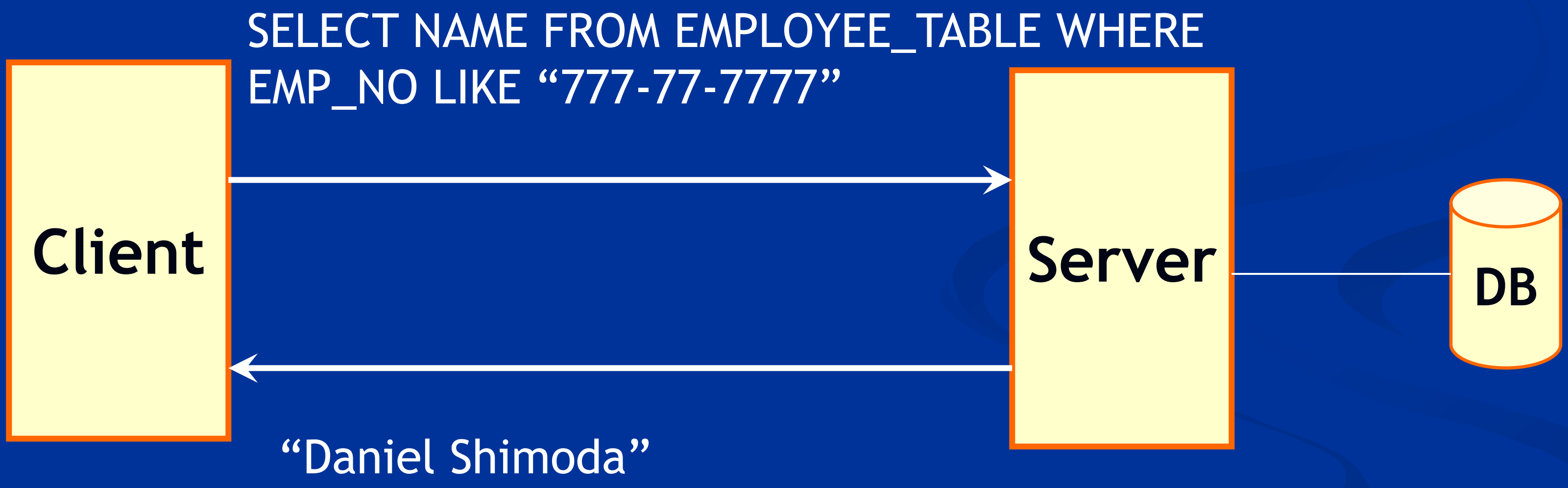


Figure 7.3 Client/Server Architecture for Database Applications

A Sample Database Server



SELECT NAME FROM EMPLOYEE_TABLE WHERE EMP_NO LIKE “777-77-7777”

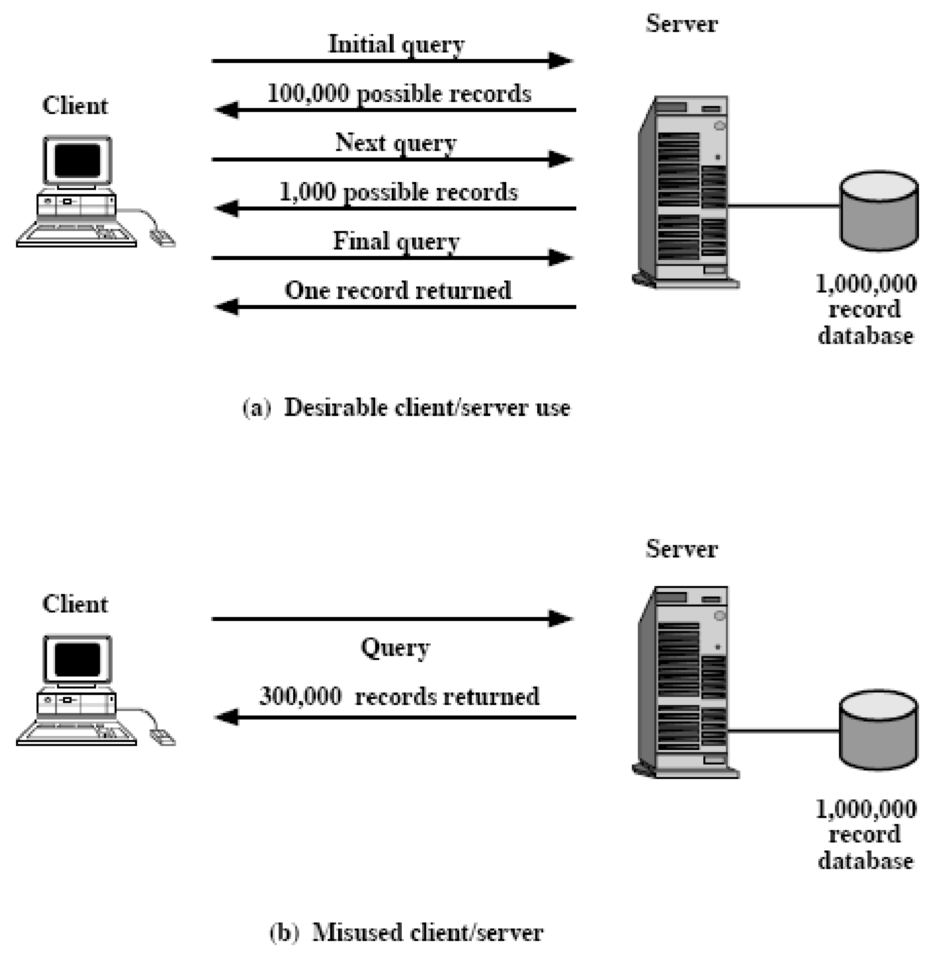
Client

Server

DB

“Daniel Shimoda”

Desirable and Misused Client/Server



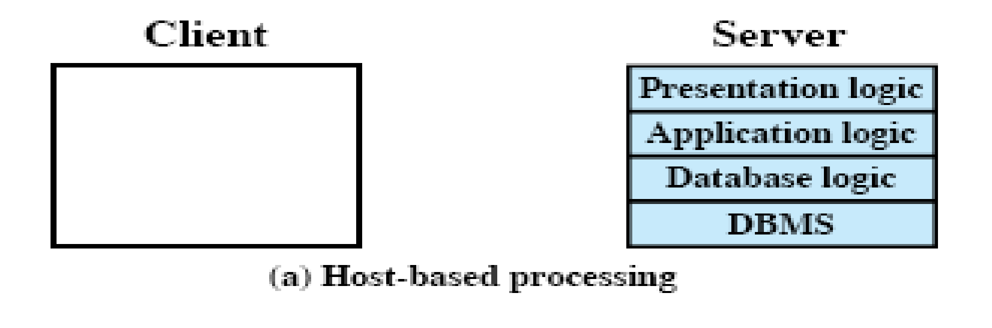
(a) Desirable client/server use

(b) Misused client/server

Figure 7.4 Client/Server Database Usage

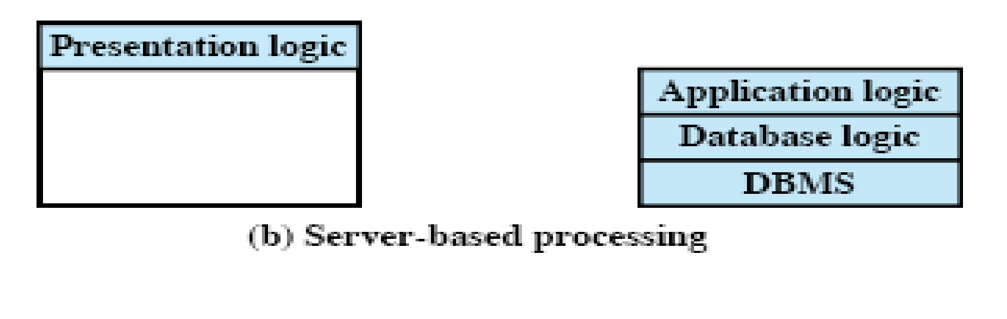
Classes of Client/Server Applications

- Host-based processing



(a) Host-based processing

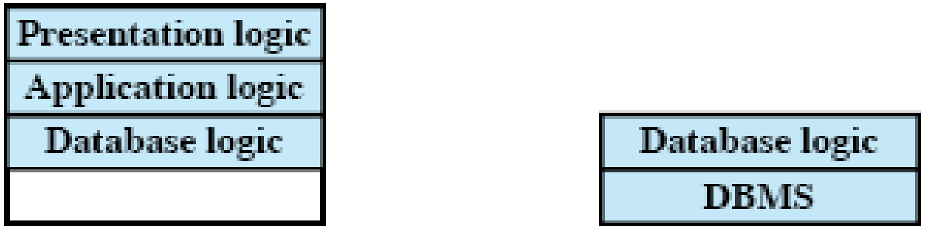
- Server-based processing



(b) Server-based processing

Classes of Client/Server Applications

- Client-based processing

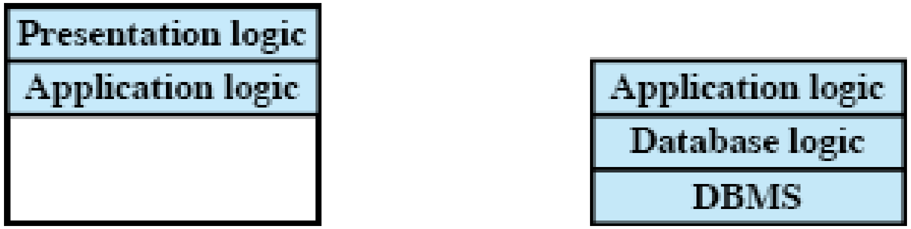


Presentation logic
Application logic
Database logic

Database logic
DBMS

(d) Client-based processing

- Cooperative processing



Presentation logic
Application logic

Application logic
Database logic
DBMS

(c) Cooperative processing

Thin Clients vs Fat Clients*

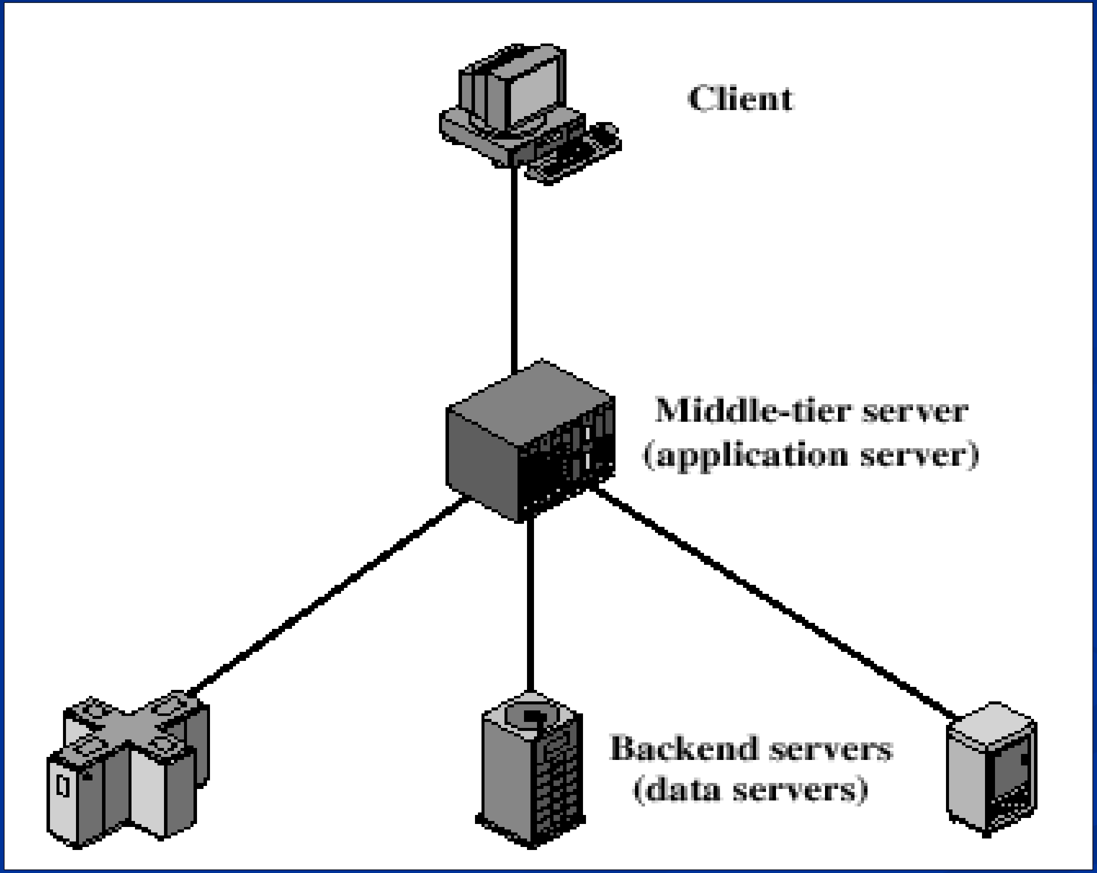
- Thin client is a minimal client. Thin clients utilize as few resources on the host PC as possible. A thin client's job is generally just to graphically display information from the application server. This allows a company the ease of managing their business logic for all applications at a central location.
- Fat client (also known as rich client or thick client) is a client that performs the bulk of the data processing operations. The data itself is stored on the server.

* source: Wikipedia, the free encyclopedia.

Three-Tier Client/Server Architecture

- To overcome the limitations of two-tier architecture, a middle tier server (also called gateway) is added between the user machine (typically a thin client) and the backend servers (database servers)
- The middle tier is where the application/business logic of the system resides and it performs a number of different functions like mapping different database queries, integrating results from different data sources and interfacing with backend legacy applications

Three-Tier Client/Server Architecture



Middleware

- A software layer introduced between the application above and communications software, operating systems below to provide a standardised programming interfaces and protocols
- Hides complexities and disparities of network protocols, operating systems and hardware platforms
- All applications use a uniform application programming interface (API)

Role of Middleware in Client/Server Architecture

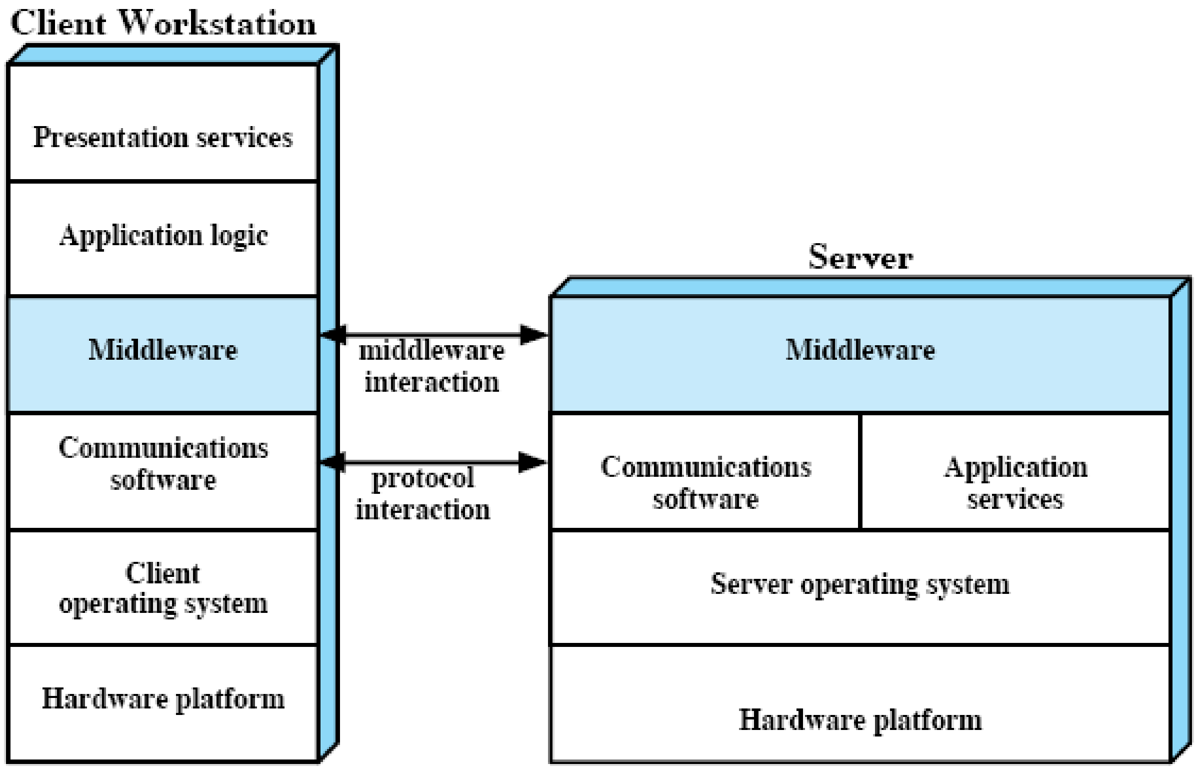


Figure 7.7 The Role of Middleware in Client/Server Architecture

Logical View of Middleware

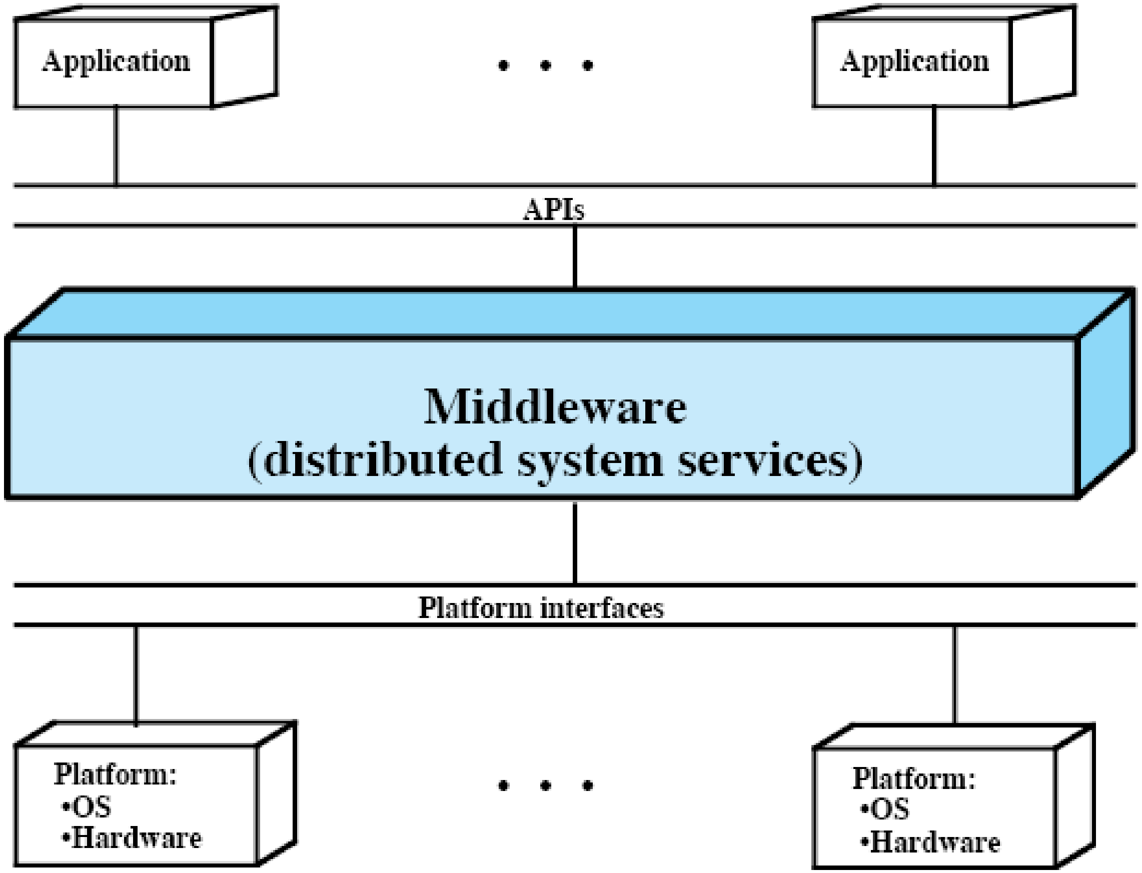
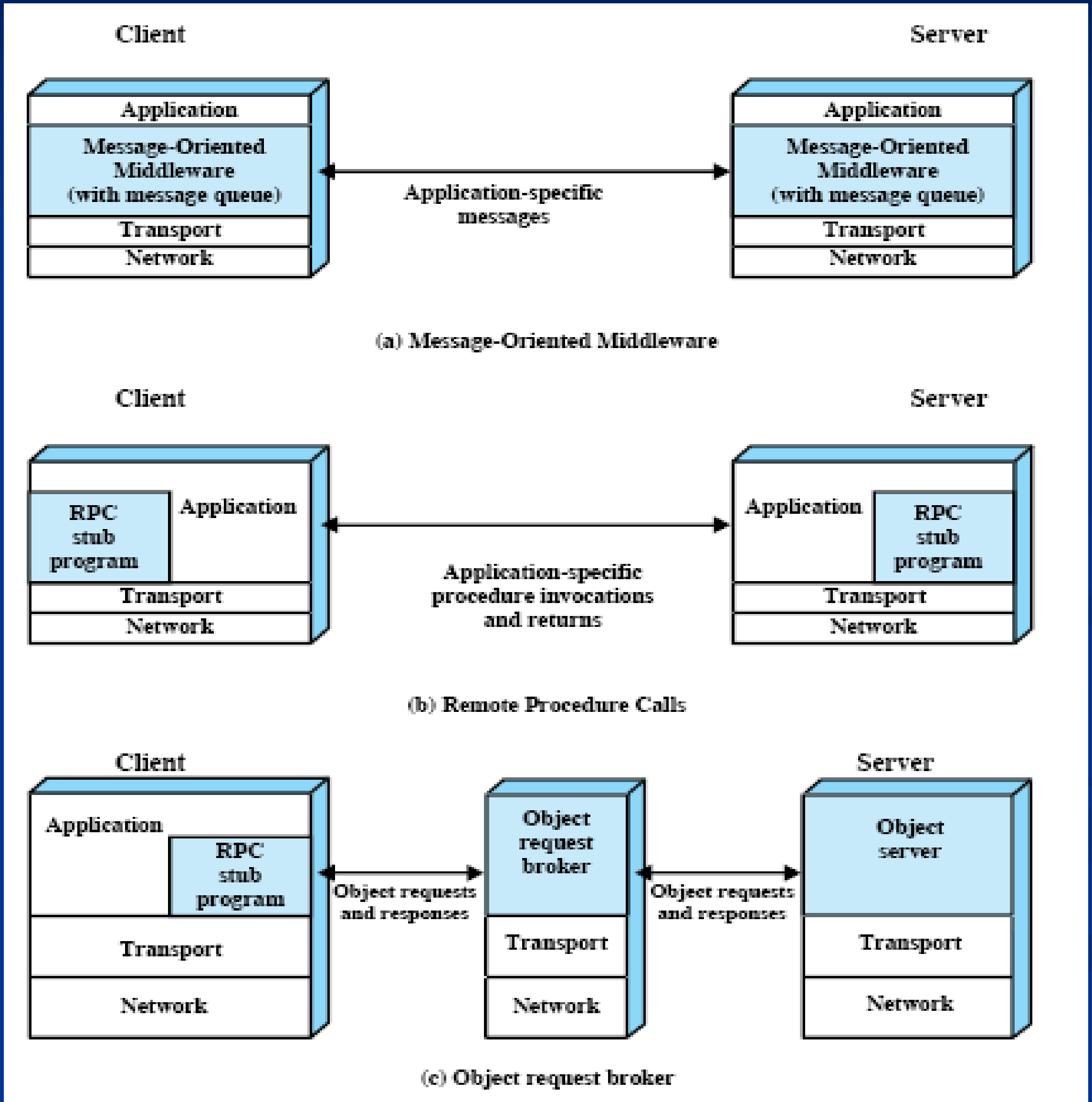


Figure 7.8 Logical View of Middleware

Middleware Mechanisms


Client
Server
Application
Message-Oriented Middleware (with message queue)
Transport
Network
Application-specific messages
Application
Message-Oriented Middleware (with message queue)
Transport
Network

(a) Message-Oriented Middleware

(b) Remote Procedure Calls

(c) Object request broker

Message Passing

- In this mechanism, a client process requires some service, e.g. read a file, sends a message containing a request for service to a server process. The server process honors the request and sends a message containing a reply

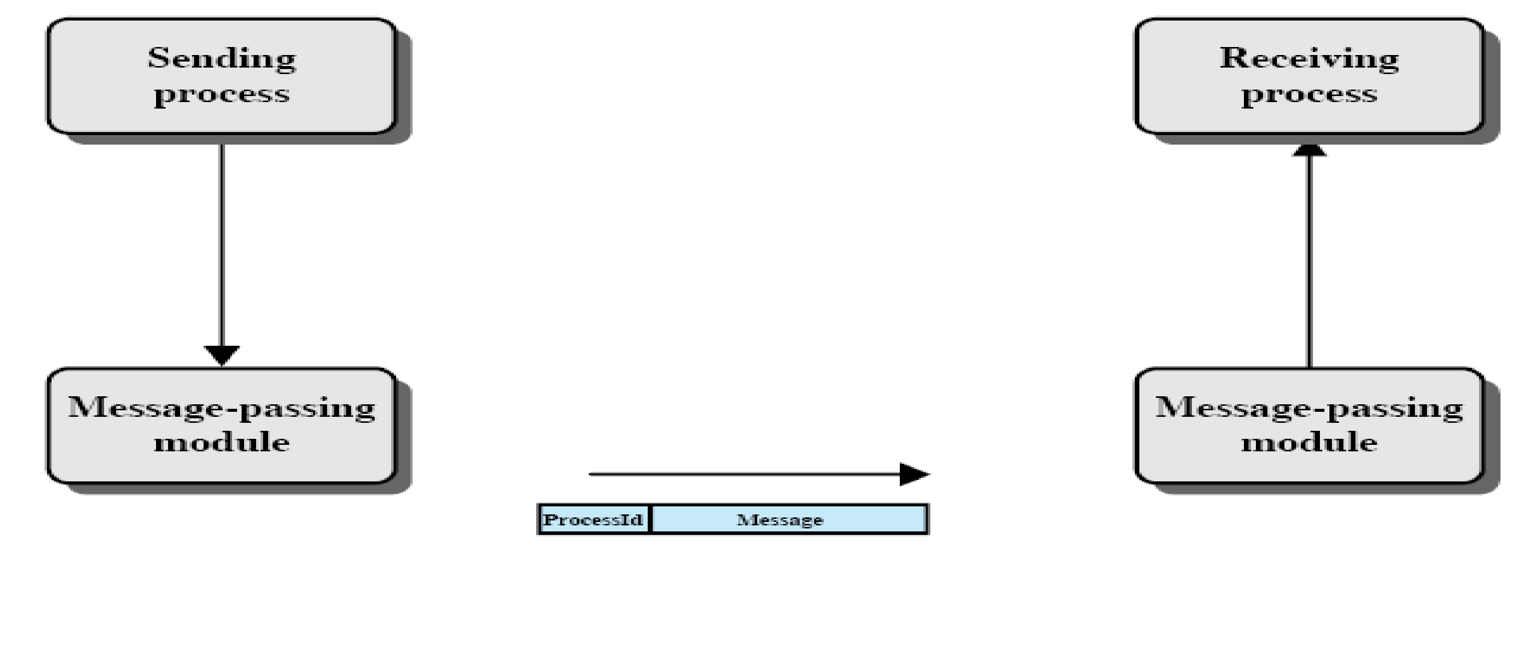


Figure 7.10 Basic Message-Passing Primitives

Remote Procedure Call

- In this mechanism, programs on different machines interact using simple procedure call/return semantics, just as if the two programs were on the same machines

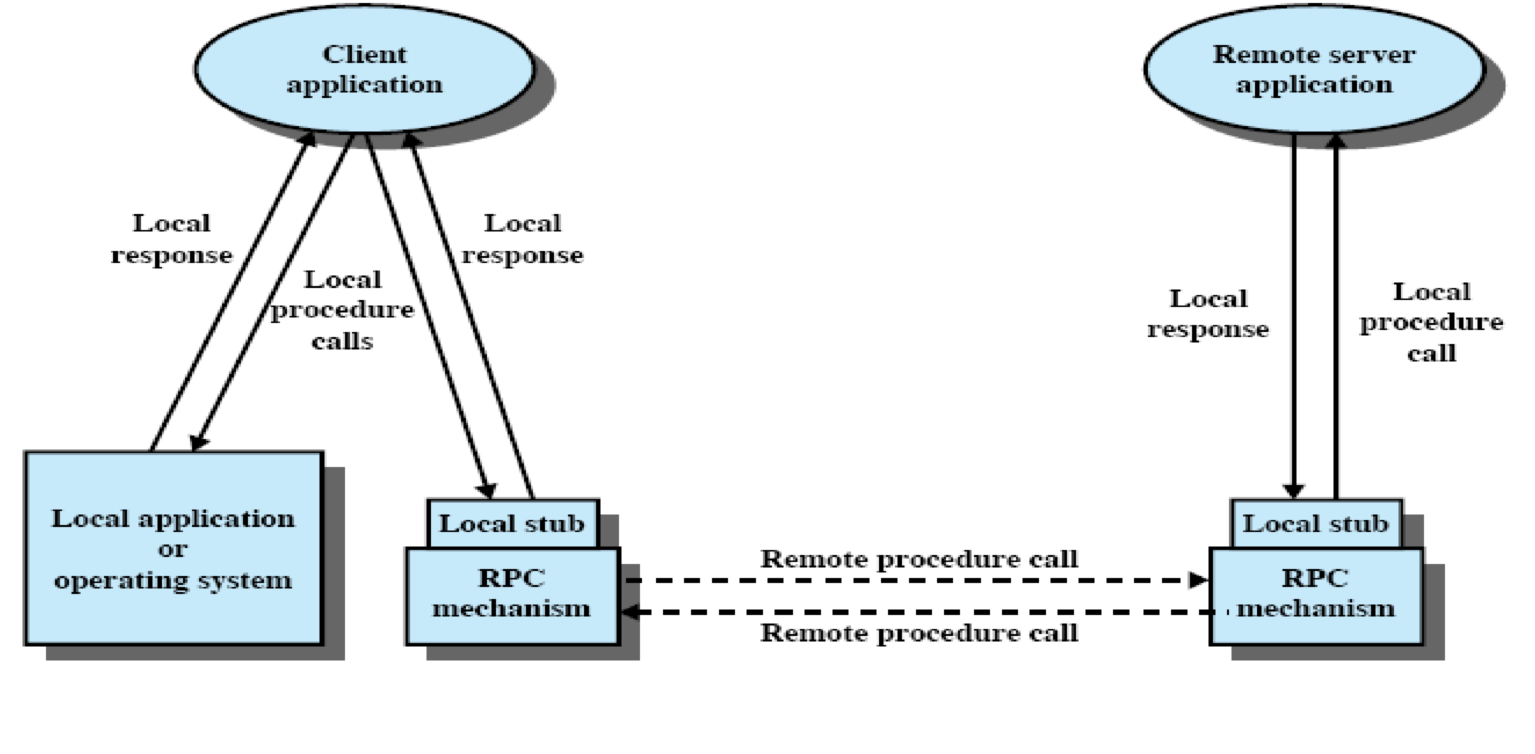


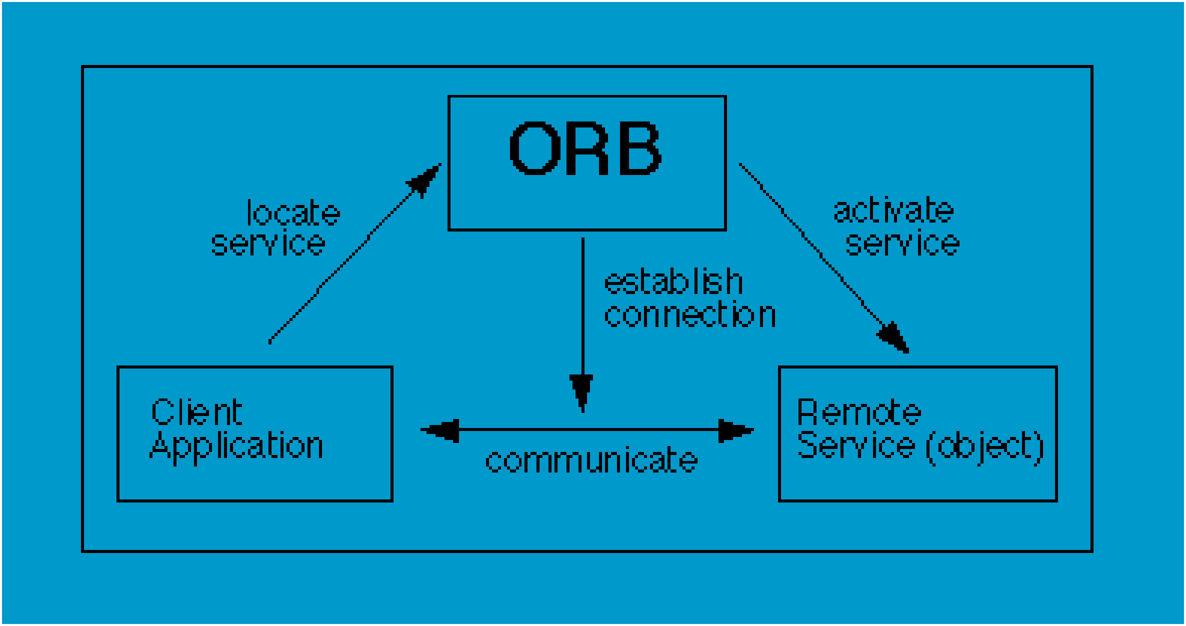
Figure 7.11 Remote Procedure Call Mechanism

RPC Advantages

- Widely accepted, used and understood
- Specifies named operations with designated types
- Because the interface is standard and precisely defined, the communication code can be generated automatically
- Developer can write module which are portable among computers and operating systems with little modification and recoding

Object-Oriented Mechanism

- Clients and servers ship messages back and forth between objects, relying on an underlying message or RPC structure or be developed directly on top of the object-oriented capabilities in the operating systems
- Several competing designs, e.g. COM and CORBA



Summary

- Reasons for the growing interest of client server computing systems
- Characteristics of client/server computing
- Architecture of client/server applications
- Middleware and the three middleware mechanisms

Acknowlegements

This module was taught by Dr. Payam Mamaani Barnaghi since 2005. Most slides have been adopted from his lecture materials and original works from William Stallings with some changes.