Database Architecture

Database Design

Department of Computer Engineering

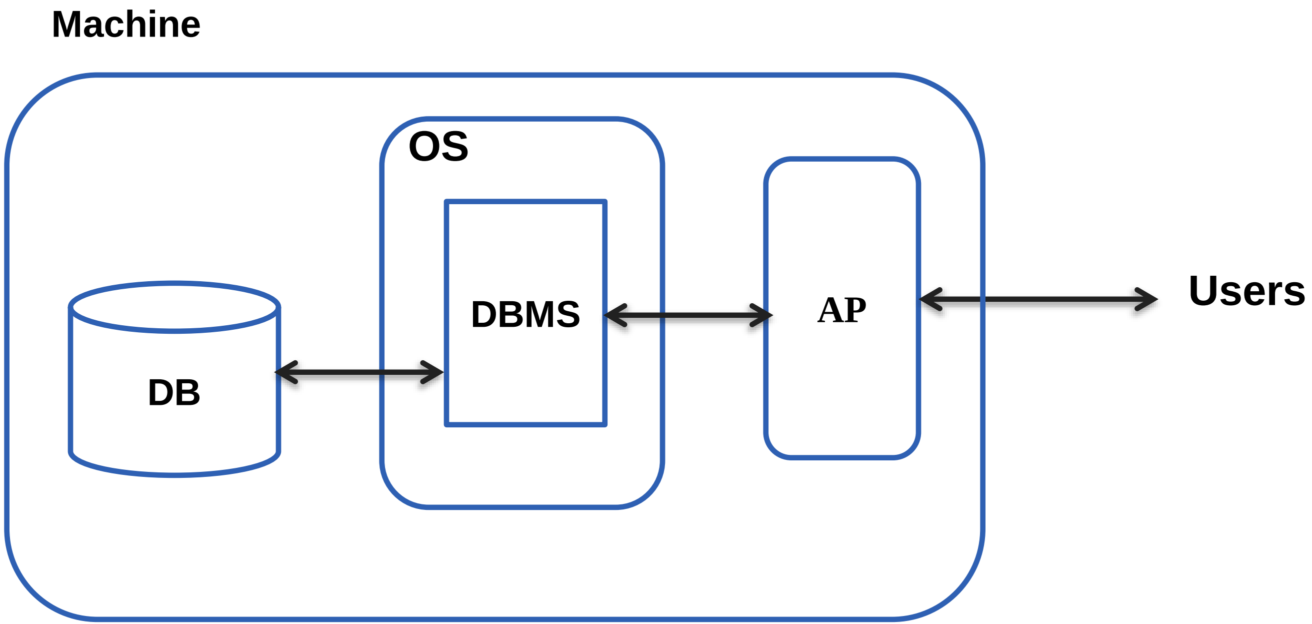
Sharif University of Technology

Maryam Ramezani maryam.ramezani@sharif.edu

Database Architecture

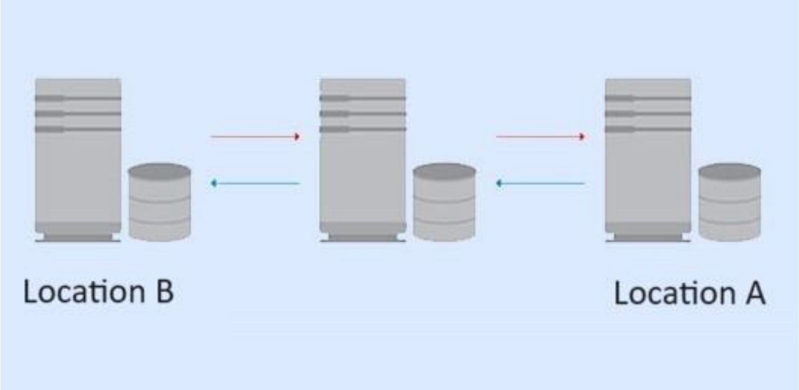
- Components and their relations
- Types:
  - Centralized architecture

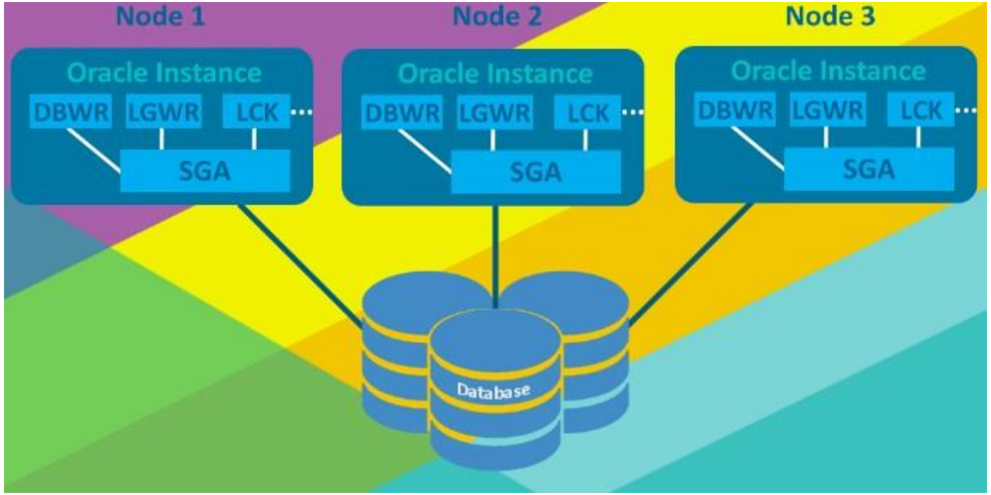
Centralized Architecture



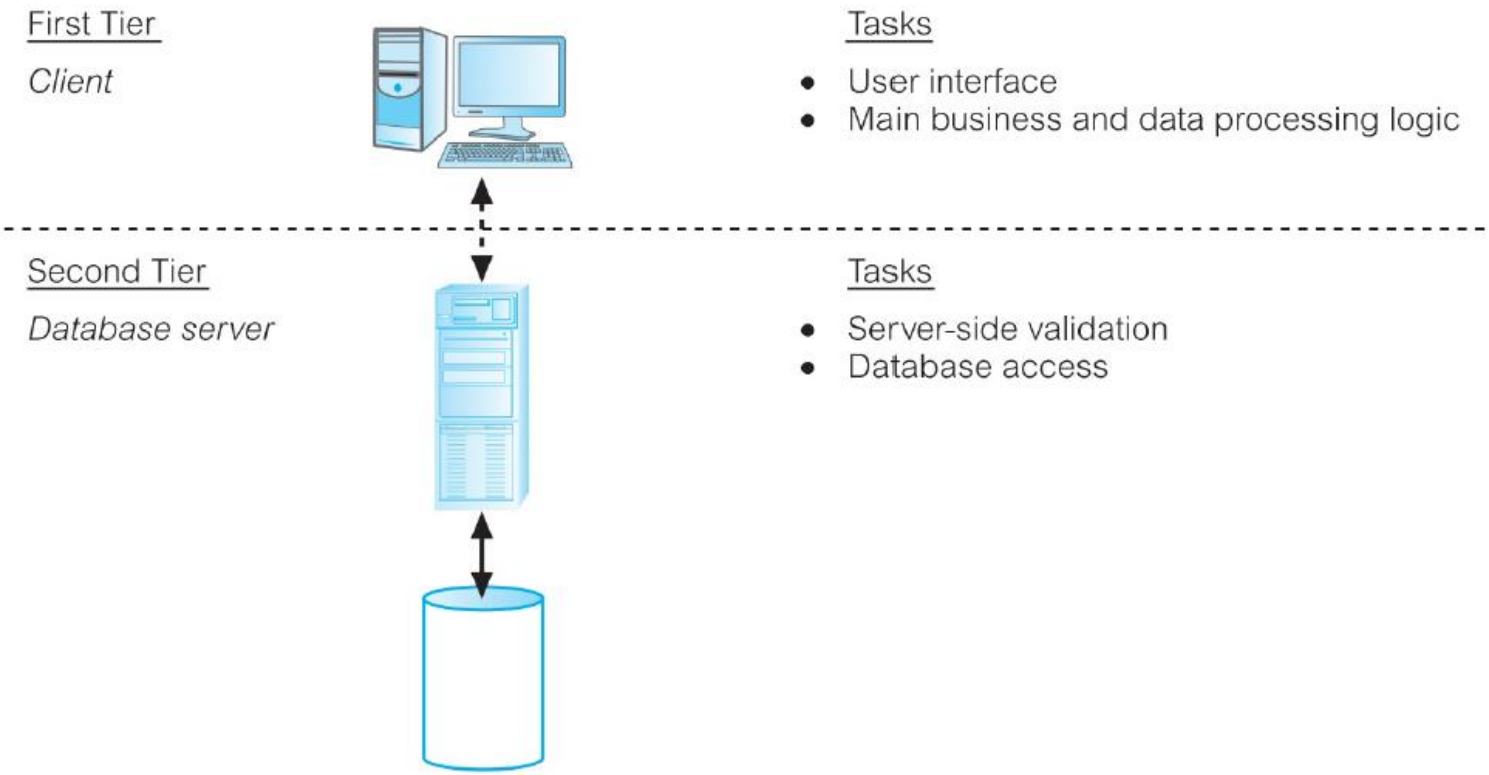
Database Architecture

- ❑ Components and their relations
- ❑ Types:
  - ○ Centralized architecture
  - ○ Decentralized architecture
    - ▪ Distributed
    - ▪ Architecture with parallel processing
    - ▪ Client-server





Traditional two-tier Client-Server Architecture



First Tier

Client

Tasks

- User interface
- Main business and data processing logic

Second Tier

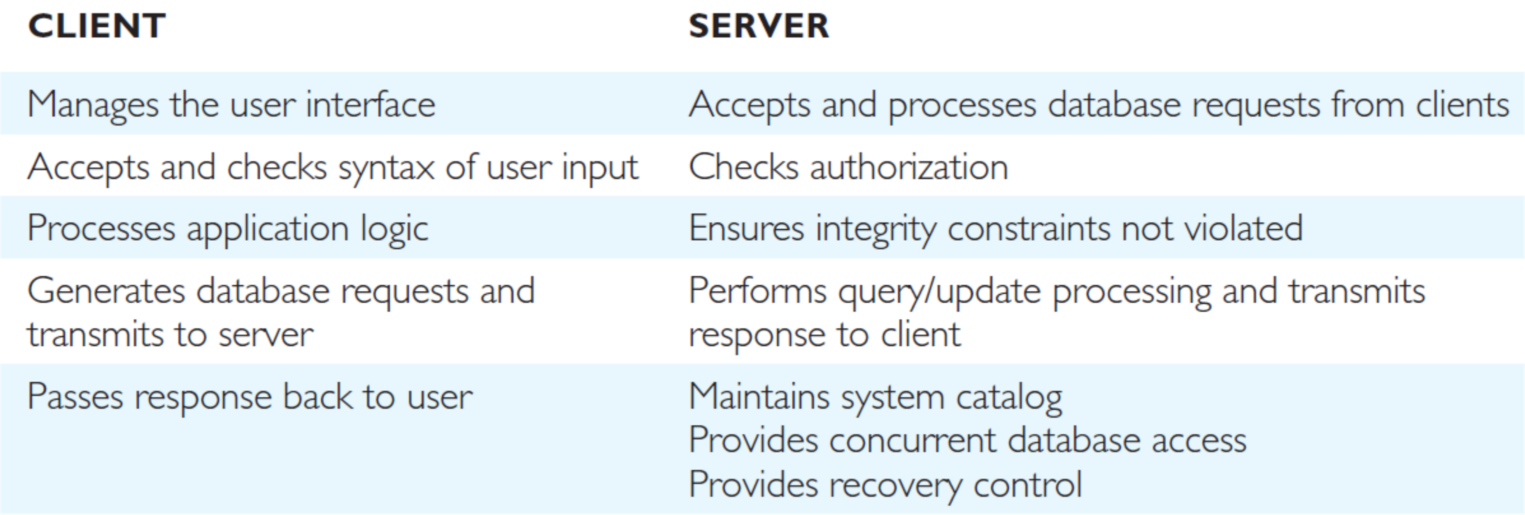
Database server

Tasks

- Server-side validation
- Database access

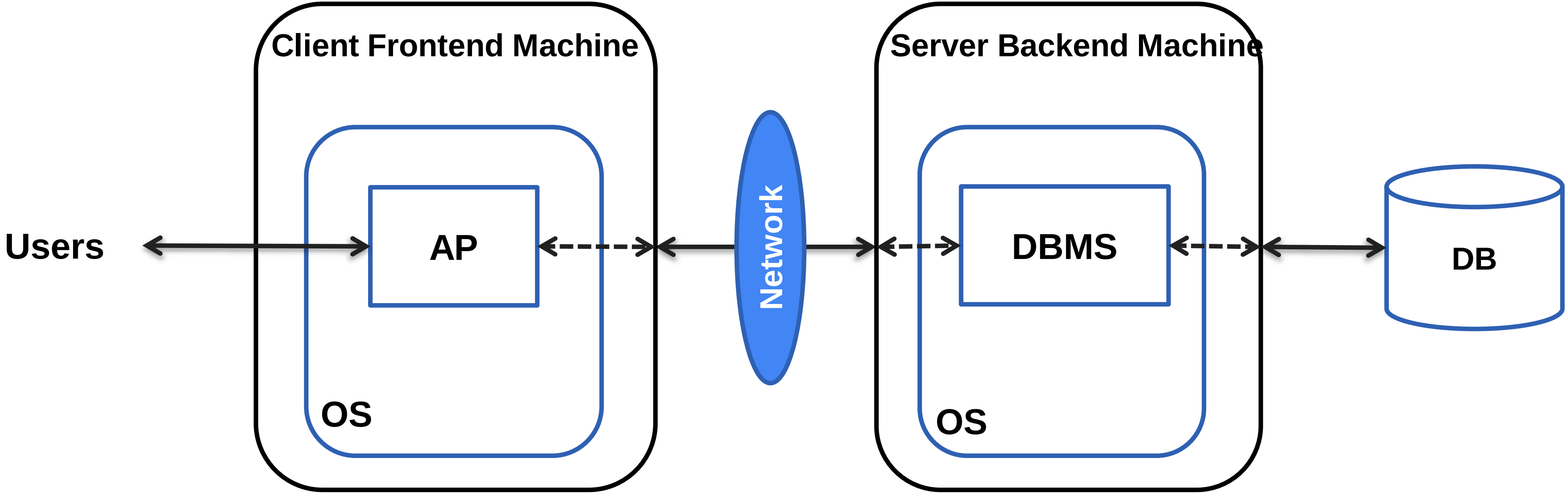
Traditional two–tier Client–Server Architecture

- The client (tier 1) is primarily responsible for the *presentation* of data to the user
  - handle user interface actions and the main business and data application logic
- The server (tier 2) is primarily responsible for supplying *data services* to the client
  - provide limited business application logic, typically validation that the client is unable to carry out due to lack of information, and access to the requested data, independent of its location

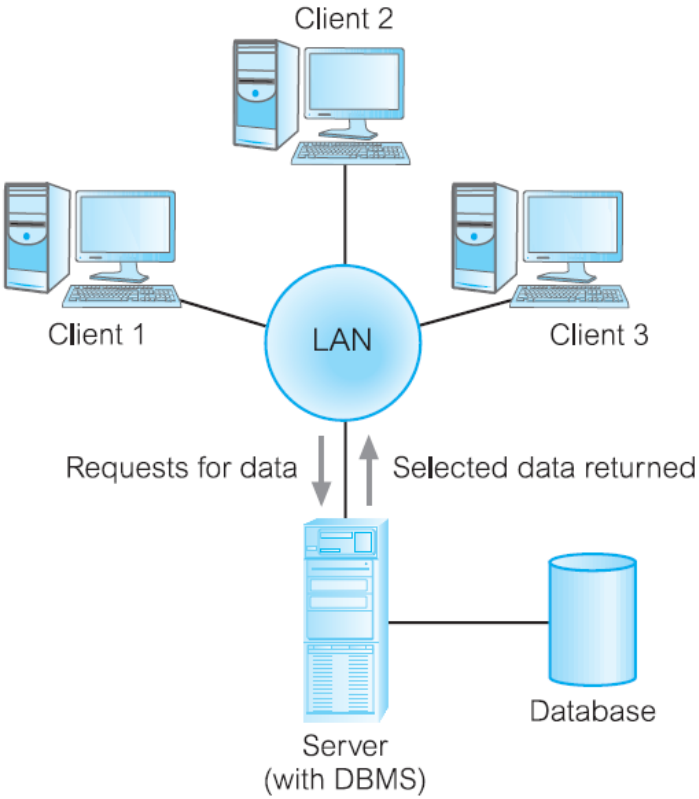


CLIENT	SERVER
Manages the user interface	Accepts and processes database requests from clients
Accepts and checks syntax of user input	Checks authorization
Processes application logic	Ensures integrity constraints not violated
Generates database requests and transmits to server	Performs query/update processing and transmits response to client
Passes response back to user	Maintains system catalog Provides concurrent database access Provides recovery control

2-tier Client-server Architecture



2-tier Client-server Architecture



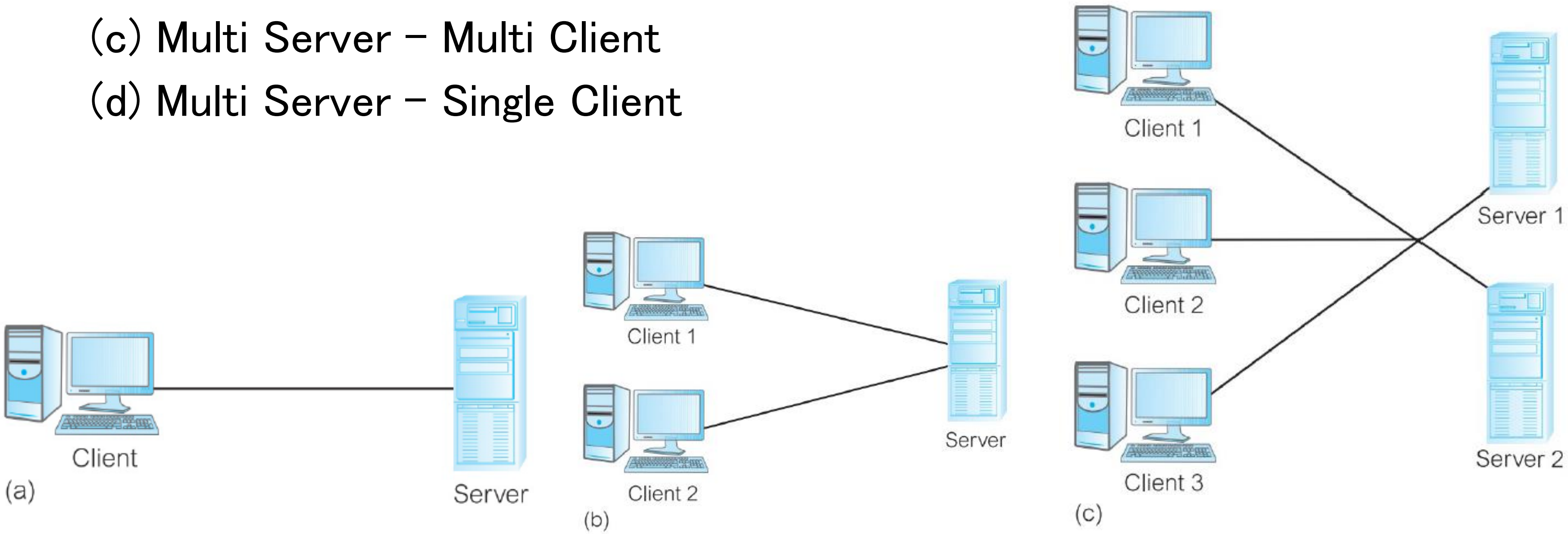
Client-server Architecture

(a) Single Server – Single Client

(b) Single Server – Multi Client

(c) Multi Server – Multi Client

(d) Multi Server – Single Client



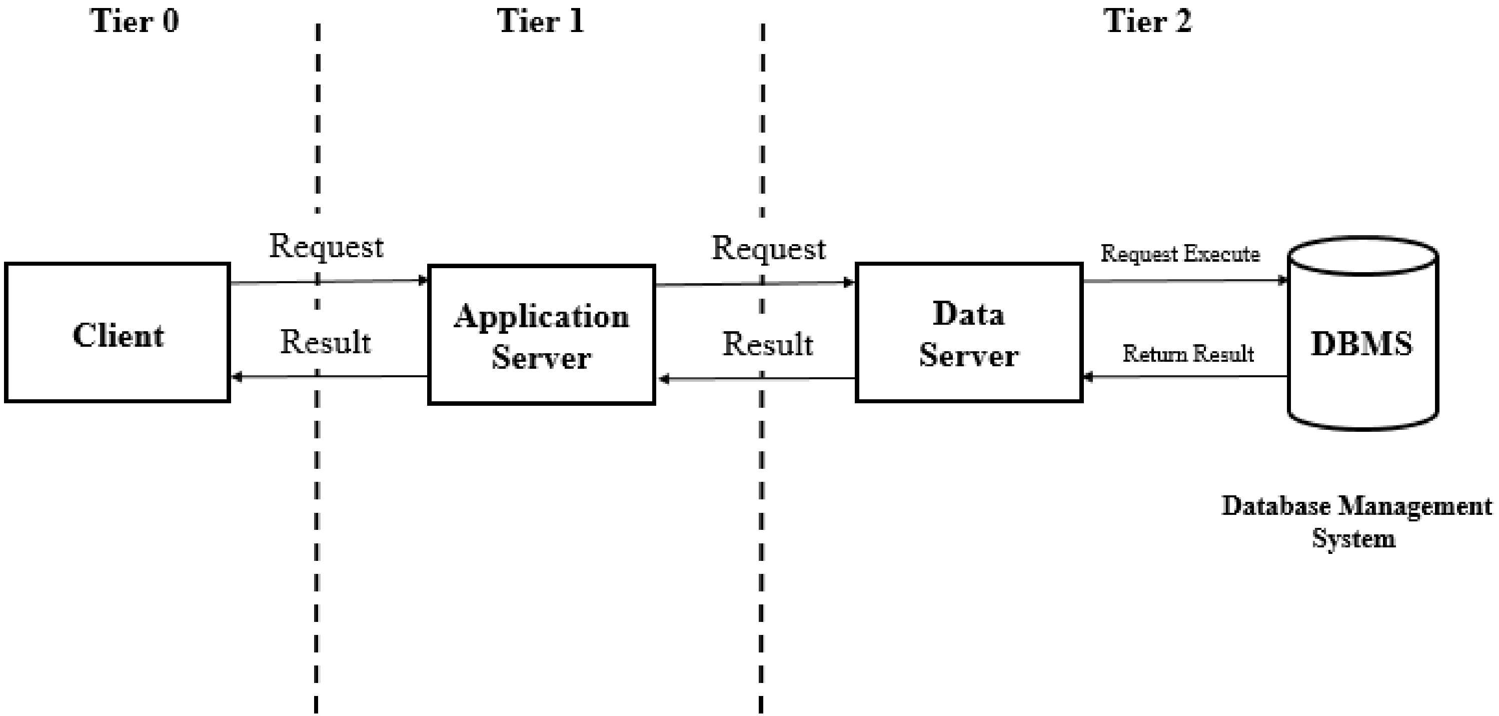
Traditional two-tier Client-Server Architecture

- Problems
  - A “fat” client, requiring considerable resources on the client’s computer to run effectively. This includes disk space, RAM, and CPU power.
  - A significant client-side administration overhead.

Three-Tier Client-Server Architecture

- The user interface layer, which runs on the end-user’ s computer (the thin client).
- The business logic and data processing layer. This middle tier runs on a server and is often called the application server.
- A DBMS, which stores the data required by the middle tier. This tier may run on a separate server called the database server.

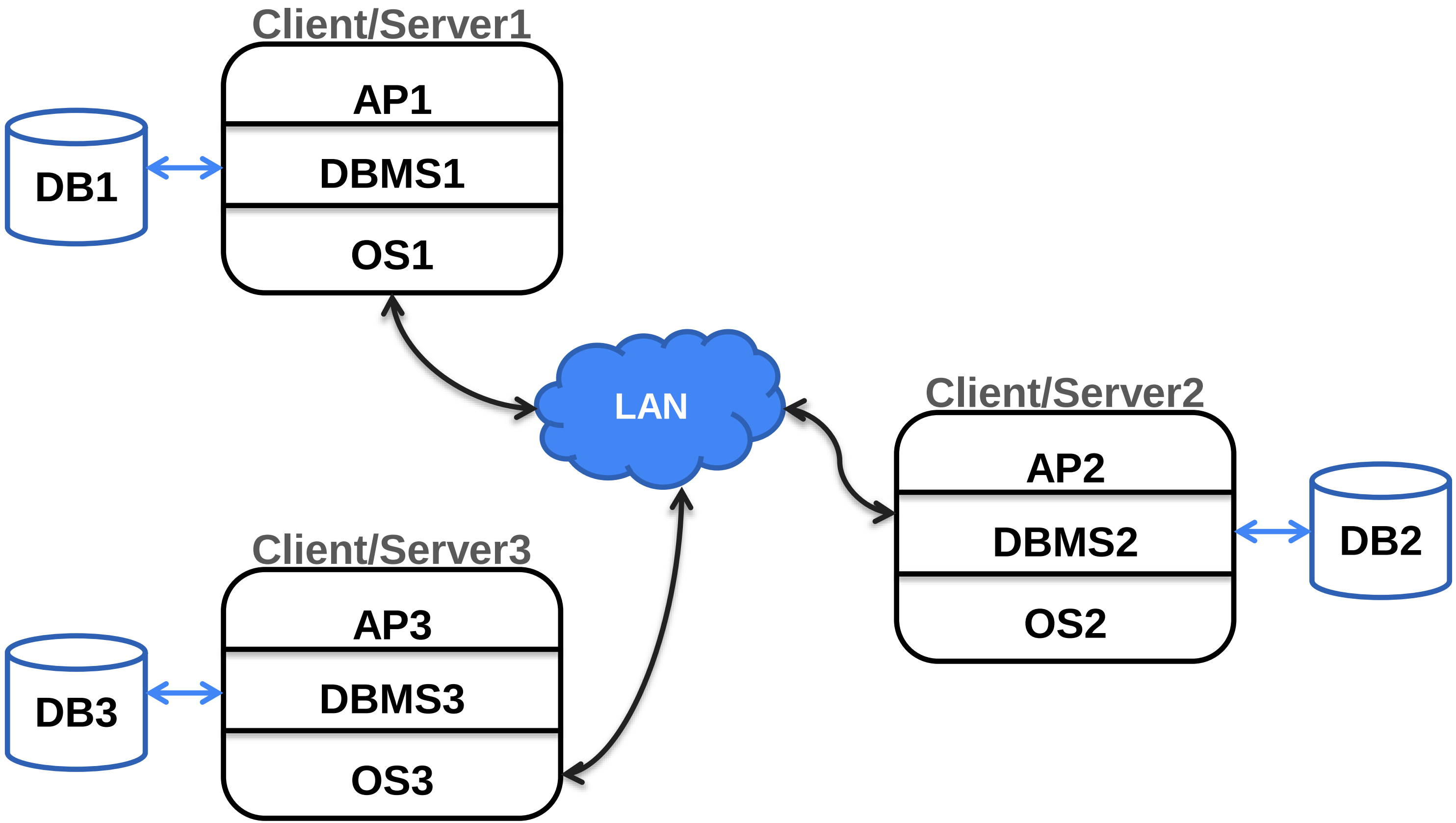
Three-Tier Client-Server Architecture



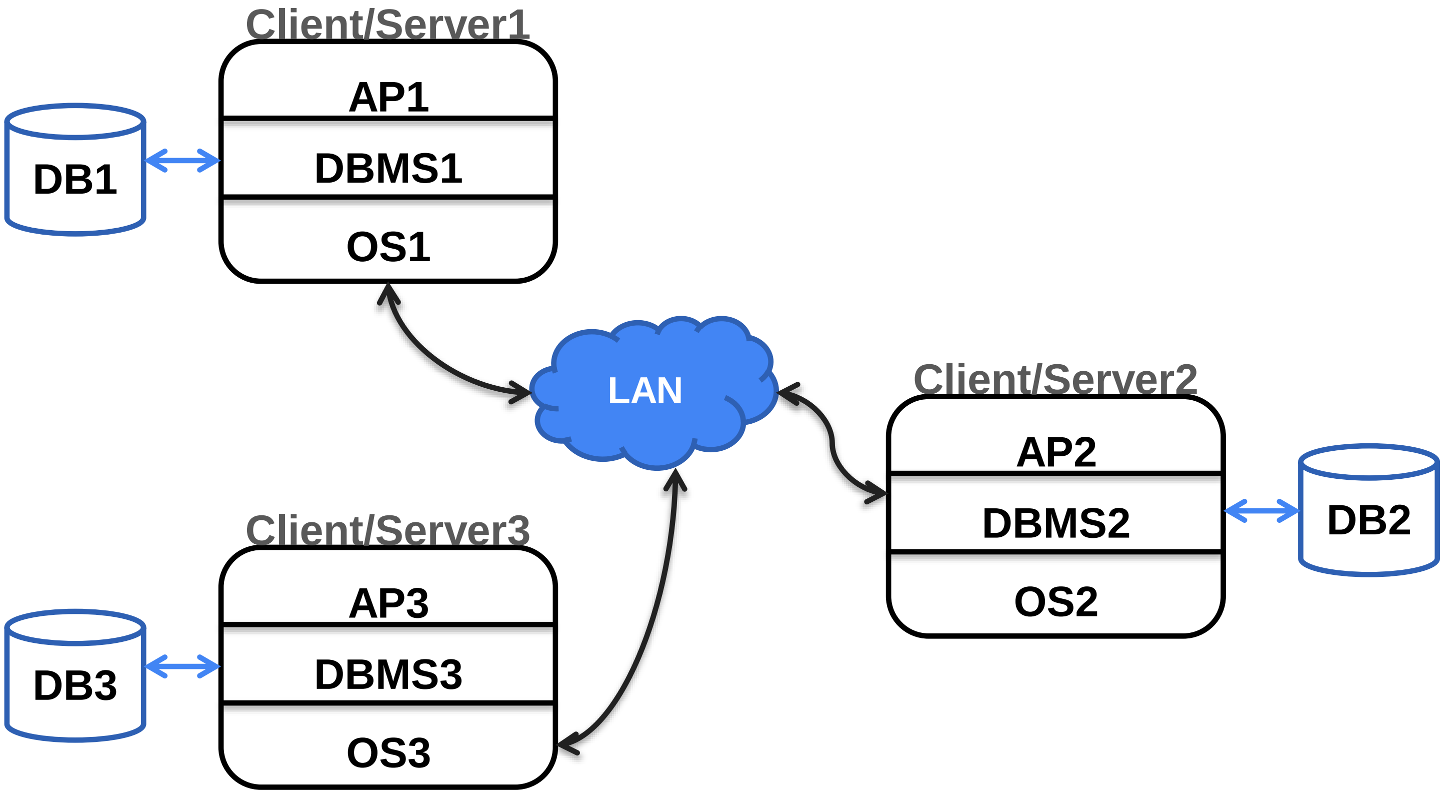
Three-Tier Client-Server Architecture

- Advantages
  - The need for less expensive hardware because the client is "thin."
  - Application maintenance is centralized with the transfer of the business logic for many end-users into a single application server. This eliminates the concerns of software distribution that are problematic in the traditional two-tier client-server model.
  - The added modularity makes it easier to modify or replace one tier without affecting the other tiers.
  - Load balancing is easier with the separation of the core business logic from the database functions.

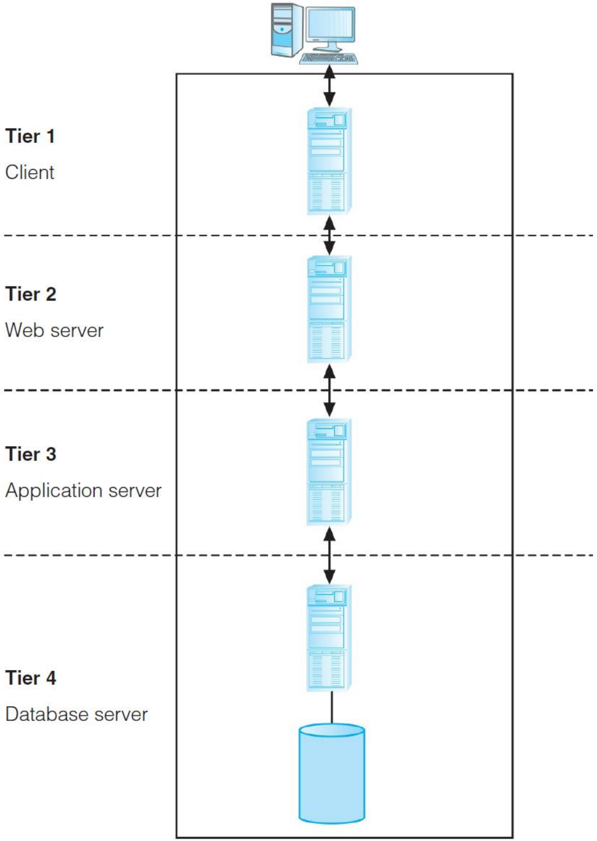
Multi Server – Multi Client



Distributed Architecture



N-Tier Architectures


Tier 1
Client
Tier 2
Web server
Tier 3
Application server
Tier 4
Database server