CSC355-Database Systems

Week-4 Lecture-1& 2

Semester-4 Spring 2019

Previous Lecture

- **Three level Architecture**
- **Mapping**
- **Data Independence**
- **Functions of DBMS**

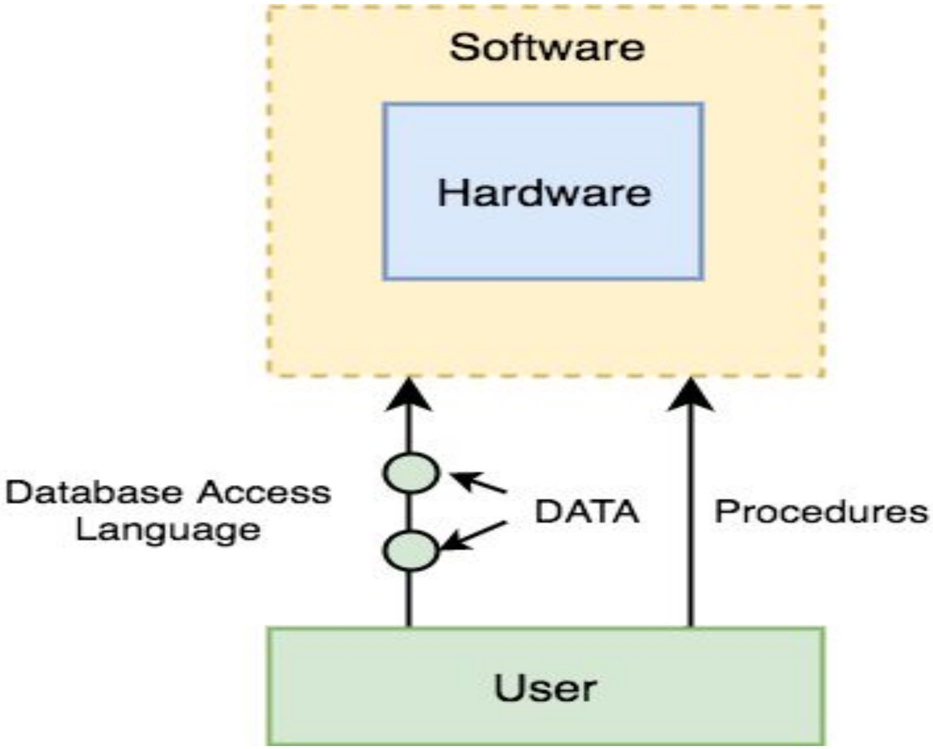
Topics to Cover

- DBMS Environment
- Components of Environment
- Types of DBMS Environments
- Pro’s and Con’s of DB Systems

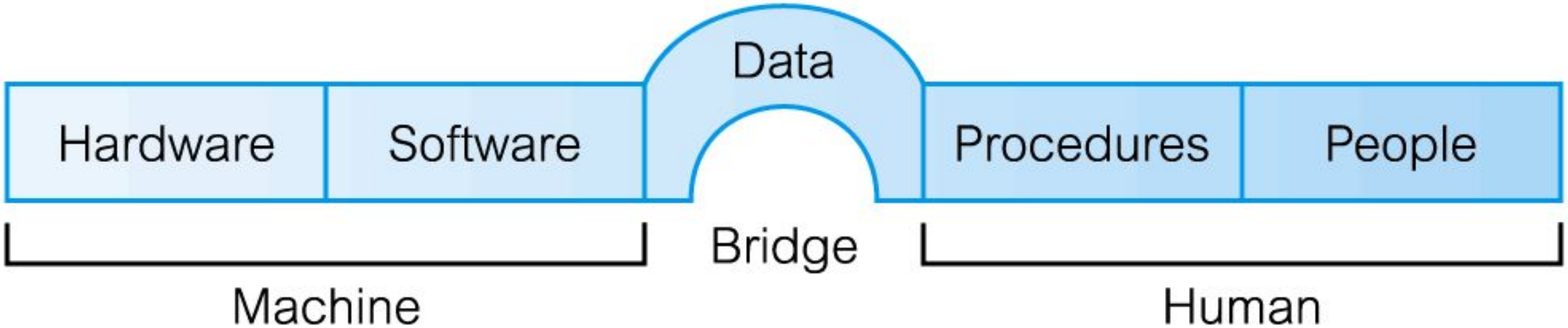
DBMS Environment

- A database environment is a collective system of components that comprise and regulates the group of data, management, and use of data which consist of software, hardware, people, techniques of handling database and the data also.

DBMS Environment



Components of DBMS Environment



Components of DBMS Environment

- The hardware in a database environment means the computers and computer peripherals that are being used to manage a database and the software means the whole thing right from the operating system (OS) to the application programs that includes database management software like M.S. Access or SQL Server.
- Again the people in a database environment include those people who administrate and use the system.
- The techniques are the rules, concepts, and instructions given to both the people and the software along with the data with the group of facts and information positioned within the database environment.

Components of DBMS Environment

1. Hardware:

- The DBMS and the applications require hardware to run. The hardware can range from a single personal computer to a single mainframe or a network of computers.
- The particular hardware depends on the organization’s requirements and the DBMS used.

Components of DBMS Environment

2. Software:

- The software component comprises the DBMS software itself and the application programs, together with the operating system, including network software if the DBMS is being used over a network.
- Typically, application programs are written in a third-generation programming language (3GL), such as C, C++, C#, Java, Visual Basic, COBOL, Fortran, Ada, or Pascal, or using query language such as SQL, embedded in a third-generation language.

Components of DBMS Environment

3. Data

- Perhaps the most important component of the DBMS environment—certainly from the end-users' point of view—is the data.
- We observe that the data acts as a bridge between the machine components and the human components.
- The database contains both the operational data and the metadata, the "data about data."

Components of DBMS Environment

4. Procedures: Procedures refer to the instructions and rules that govern the design and use of the database. The users of the system and the staff who manage the database require documented procedures on how to use or run the system. These may consist of instructions on how to:

- Log on to the DBMS.
- Use a particular DBMS facility or application program.
- Start and stop the DBMS.
- Make backup copies of the database.
- Handle hardware or software failures.
- Improve performance, or archive data to secondary storage.

Components of DBMS Environment

5. People:

- The final component is the people involved with the system.
- Includes database designers, DBAs, application programmers, and end-users.

Types of DBMS Environment

- **Single User**
- **Multi-user**
  - **Teleprocessing**
  - **File Servers**
  - **Client-Server**

Single User Database Environment

- This is the database environment which supports only one user accessing the database at a specific time.
- The DBMS might have a number of users but at a certain time only one user can log into the database system and use it.
- This type of DBMS systems are also called Desktop Database systems.

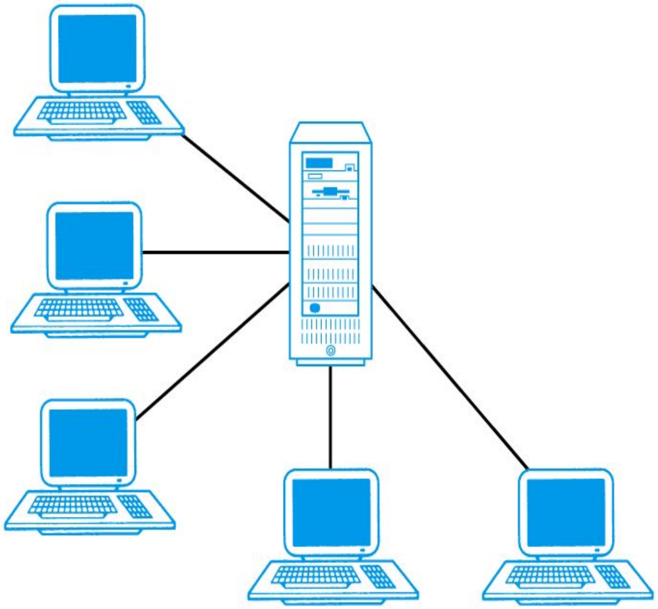
Multi-User Database systems

This is the type of DBMS which can support a number of users simultaneously interacting with the database in different ways. A number of environments exist for such DBMS.

- **Teleprocessing**
- **File-server**
- **Client-server**

Teleprocessing

- This type of Multi user database systems processes the user requests at a central computer.
- All requests are carried to the central computer where the database is residing, transactions are carried out and the results transported back to the terminals (literally dumb terminals).
- There is now an identifiable trend in industry towards **downsizing**, that is, replacing expensive mainframe computers with more cost-effective networks of personal computers that achieve the same, or even better, results.



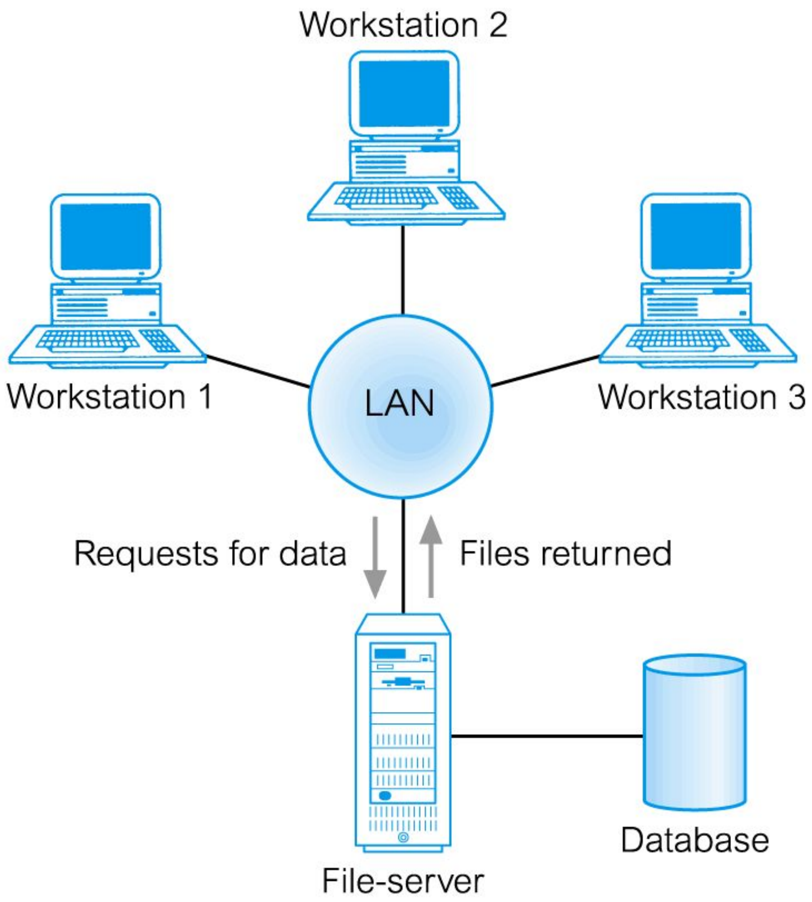
File-Server

- A file server is used to maintain a connection between the users of the database system.
- Each client of the network runs its own copy of the DBMS and the database resides on the file server.
- Now whenever a user needs data from the file server it makes a request the whole file containing the required data was sent to the client.
- At this stage it is important to see that the user has requested one or two records from the database but the server sends a complete file, which might contain hundreds of records.

File-Server

- Now if the client after making the desired operation on the desired data wants to write back the data on the database he will have to send the whole file back to the server, thus causing a lot of network overhead.
- The server does not have lots of actions to do rather it remains idle for lots of the time in contrast with that of the teleprocessing systems approach.
- **Disadvantages include:**
  - Significant network traffic.
  - Copy of DBMS on each workstation.
  - Concurrency, recovery and integrity control more complex.

File-Server



Workstation 2
Workstation 1
LAN
Workstation 3
Requests for data
Files returned
File-server
Database

Client-Server

- Client–server refers to the way in which software components interact to form a system.

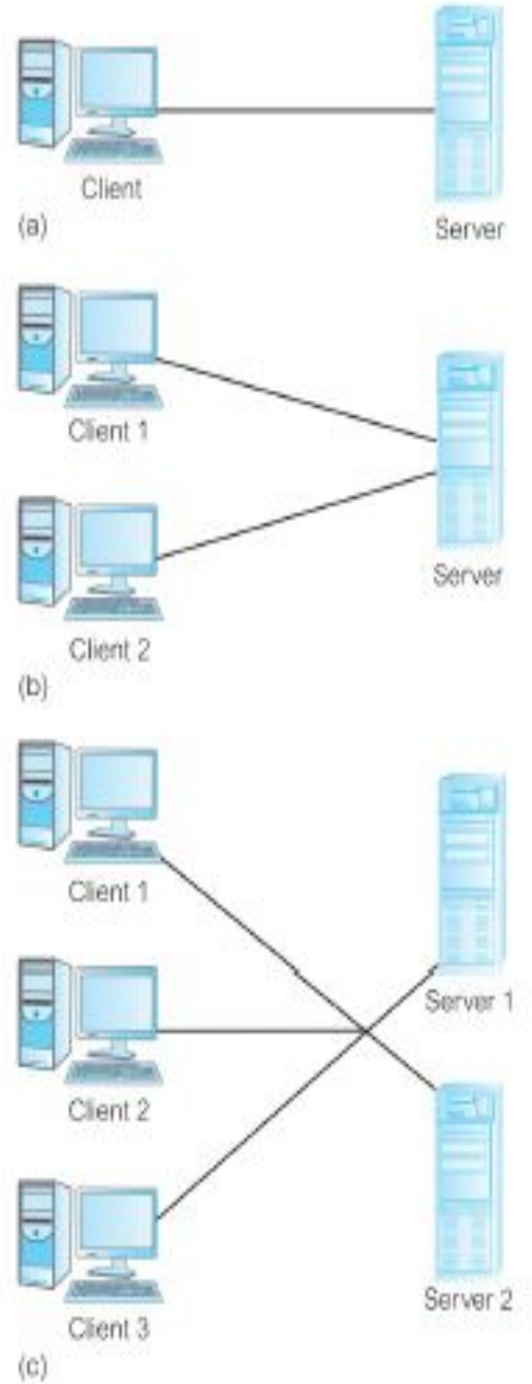
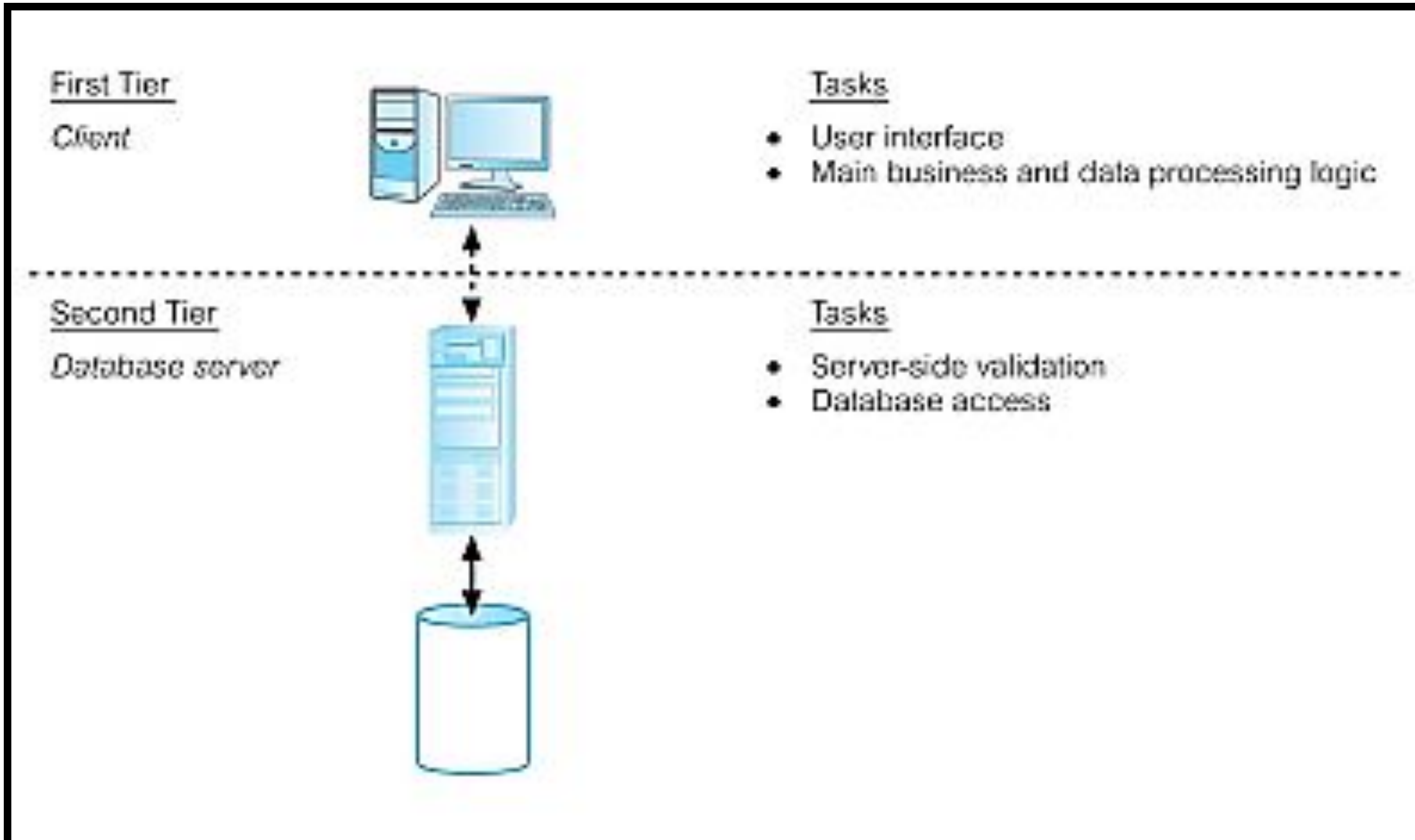


Figure 3.4 Alternative client–server topologies: (a) single client, single server; (b) multiple clients, single server; (c) multiple clients, multiple servers.

Traditional Two-Tier Client-Server

- The traditional two-tier client–server architecture provides a very basic separation of these components.
  - Client (tier 1) manages user interface and runs applications.
  - Server (tier 2) holds database and DBMS.

Traditional Two-Tier Client-Server



Traditional Two-Tier Client-Server

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

Traditional Two-Tier Client-Server

Advantages include:

- wider access to existing databases;
- increased performance as server and client resides on different systems.
- possible reduction in hardware costs as It is only the server that requires storage.
- reduction in communication costs as applications carry out part of the operations on the client and send only requests for database access across the network, resulting in less data being sent across the network.
- increased consistency due to the fact that constraints are defined and validated only in the one place

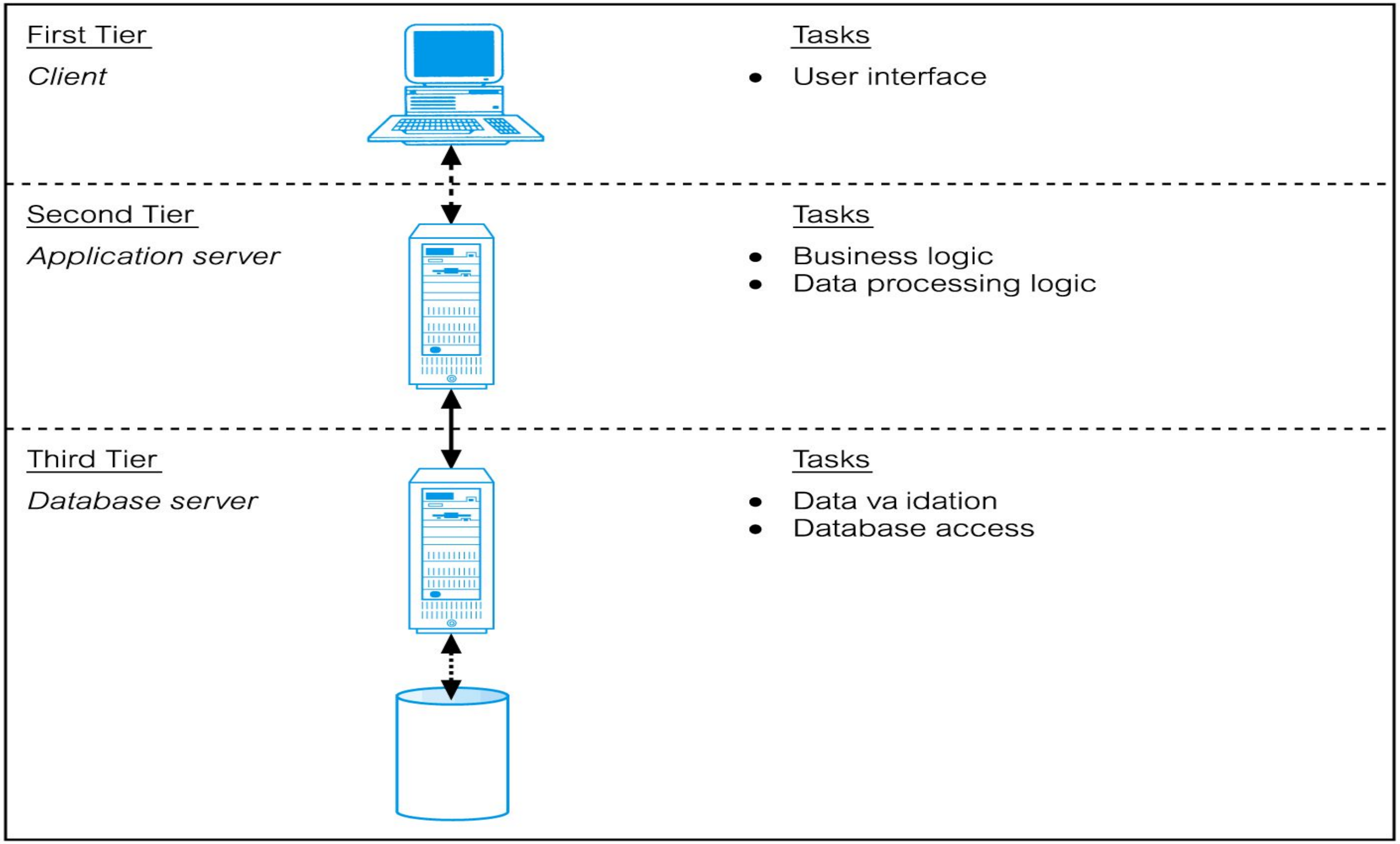
Three-Tier Client-Server

- Client side presented two problems preventing true scalability:
  - ‘Fat’ client, requiring considerable resources on client’s computer to run effectively. This includes disk space, RAM, and CPU power.
  - Significant client-side administration overhead.
- By 1995, three layers proposed, each potentially running on a different platform.

Three-Tier Client-Server

- The user interface layer, which runs on the end-user's computer **(the *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



Three-Tier Client-Server

Advantages:

- ‘Thin’ client, requiring less expensive hardware.
- Application maintenance centralized. This eliminates the concerns of software distribution that are problematic in the traditional two-tier client–server model.
- Easier to modify or replace one tier without affecting others.
- Separating business logic from database functions makes it easier to implement load balancing.

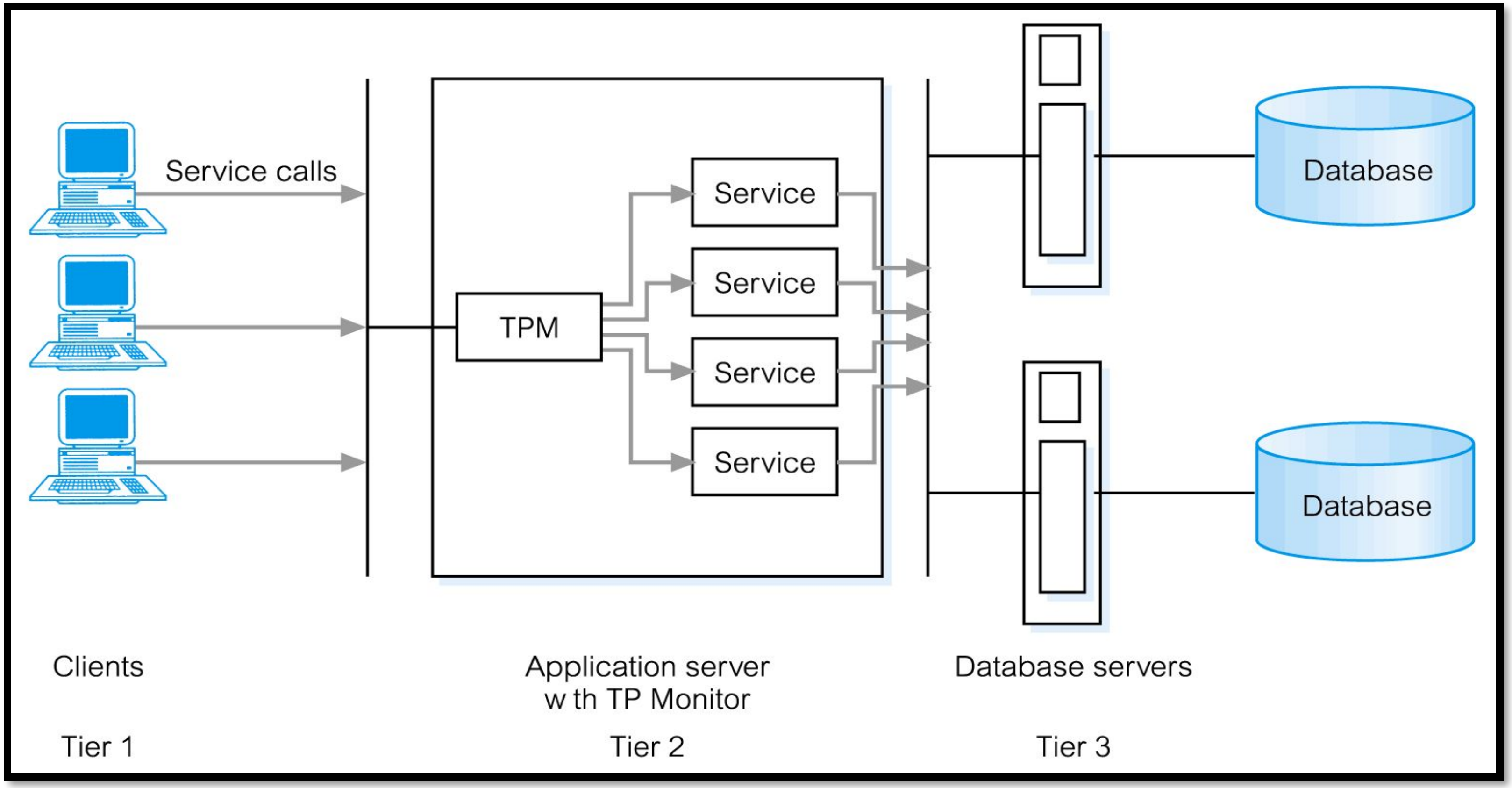
Middleware

- *Middleware* is a generic term used to describe software that mediates with other software and allows for communication between different applications in a heterogeneous system.

Transaction Processing Monitors

- Program that controls data transfer between clients and servers in order to provide a consistent environment, particularly for Online Transaction Processing (OLTP).
- Complex applications are often built on top of several **resource managers** (such as DBMSs, operating systems, user interfaces, and messaging software).
- A TP Monitor forms the middle tier of a three-tier architecture. Some commercial TP monitors : CICS from IBM, oracle TUXEDO and Microsoft Transaction server.
- Its functionality includes logging, recovery, concurrency control etc.

TPM as middle tier of 3-tier client-server



Pros and Cons of DBMS

- The database management system has promising potential advantages.
- Unfortunately, there are also disadvantages. In this section, we examine these advantages and disadvantages.

Advantages of DBMS

Control of data redundancy:

- Traditional file-based systems waste space by storing the same information in more than one file.
- In contrast, the database approach attempts to eliminate the redundancy by integrating the files so that multiple copies of the same data are not stored.
- However, the database approach does not eliminate redundancy entirely.
- Sometimes it is necessary to duplicate key data items to model relationships.

Advantages of DBMS

Data consistency:

- By eliminating or controlling redundancy, we reduce the risk of inconsistencies occurring.
- If a data item is stored only once in the database, any update to its value has to be performed only once and the new value is available immediately to all users.
- If a data item is stored more than once and the system is aware of this, the system can ensure that all copies of the item are kept consistent.
- Unfortunately, many of today's DBMSs do not automatically ensure this type of consistency.

Advantages of DBMS

More information from the same amount of data:

- With the integration of the operational data, it may be possible for the organization to derive additional information from the same data.

Sharing of data:

- Typically, files are owned by the people or departments that use them.
- On the other hand, the database belongs to the entire organization and can be shared by all authorized users. In this way, more users share more of the data.
- Furthermore, new applications can build on the existing data in the database and add only data that is not currently stored, rather than having to define all data requirements again.

Advantages of DBMS

Improved data integrity:

- Database integrity refers to the validity and consistency of stored data. Integrity is usually expressed in terms of **constraints**, which are consistency rules that the database is not permitted to violate.
- Constraints may apply to data items within a single record or to relationships between records.
- For example, an integrity constraint could state that a member of staff's salary cannot be greater than $40,000.

Advantages of DBMS

Improved security:

- Database security is the protection of the database from unauthorized users.
- For example, the DBA has access to all the data in the database; a branch manager may have access to all data that relates to his or her branch office; and a sales assistant may have access to all data relating to properties but no access to sensitive data such as staff salary details.

Advantages of DBMS

Enforcement of standards

- Again, integration allows the DBA to define and the DBMS to enforce the necessary standards.
- These may include departmental, organizational, national, or international standards for such things as data formats to facilitate exchange of data between systems, naming conventions, documentation standards, update procedures, and access rules.

Advantages of DBMS

Economy of scale:

- Combining all the organization's operational data into one database and creating a set of applications that work on this one source of data can result in cost savings.
- In this case, the budget that would normally be allocated to each department for the development and maintenance of its file-based system can be combined, possibly resulting in a lower total cost, leading to an economy of scale.

Advantages of DBMS

Balance of conflicting requirements:

- Each user or department has needs that may be in conflict with the needs of other users.
- Because the database is under the control of the DBA, the DBA can make decisions about the design and operational use of the database that provide the best use of resources for the organization as a whole.
- These decisions will provide optimal performance for important applications, possibly at the expense of less-critical ones.

Advantages of DBMS

Improved data accessibility and responsiveness

- Many DBMSs provide query languages or report writers that allow users to ask *ad hoc* questions and to obtain the required information almost immediately at their terminal.

Increased concurrency:

- Many DBMSs manage concurrent database access and ensure that problems like loss of information and integrity cannot occur.

Advantages of DBMS

Improved maintenance through data independence:

- A DBMS separates the data descriptions from the applications, thereby making applications immune to changes in the data descriptions. This is known as **data independence.**
- The provision of data independence simplifies database application maintenance.

Improved backup and recovery services:

- Backup can be immediate or differed based on the system requirements.

Disadvantages of DBMS

Complexity:

- The provision of the functionality that we expect of a good DBMS makes the DBMS an extremely complex piece of software.
- Database designers and developers, data and database administrators must understand this functionality to take full advantage of it.
- Failure to understand the system can lead to bad design decisions, which can have serious consequences for an organization.

Disadvantages of DBMS

Size:

- The complexity and breadth of functionality makes the DBMS an extremely large piece of software, occupying many megabytes of disk space and requiring substantial amounts of memory to run efficiently.

Cost of DBMSs:

- The cost of DBMSs varies significantly, depending on the environment and functionality provided.
- For example, a single-user DBMS for a personal computer may only cost $100. However, a large mainframe multi-user DBMS servicing hundreds of users can be extremely expensive, perhaps $100,000 or even $1,000,000. There is also the recurrent annual maintenance cost, which is typically a percentage of the list price.

Disadvantages of DBMS

Cost of conversion:

- In some situations, the cost of the DBMS and extra hardware may be relatively small compared with the cost of converting existing applications to run on the new DBMS and hardware.
- This cost also includes the cost of training staff to use these new systems, and possibly the employment of specialist staff to help with the conversion and running of the systems.
- This cost is one of the main reasons why some organizations feel tied to their current systems and cannot switch to more modern database technology.

Disadvantages of DBMS

Greater impact of a failure

- The centralization of resources increases the vulnerability of the system.
- Because all users and applications rely on the availability of the DBMS, the failure of certain components can bring operations to a halt.

Thank you