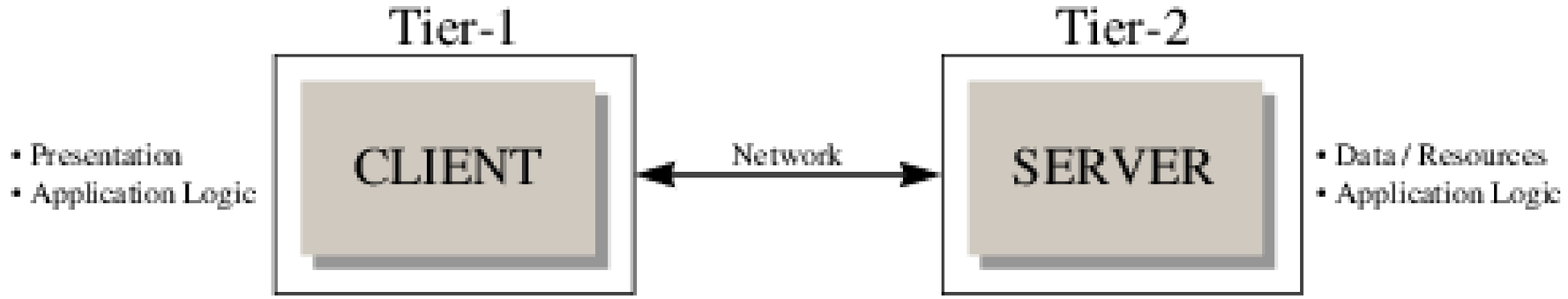
Two-Tier Client-Server Architecture

Software design And architecture

Two-Tier Client-Server Architecture

The two-tier architecture is based on the Client-Server model.
It consists of **Client-Application tier** and **Database tier**.
The Client-Application server communicates directly with the Database server.
Data or information transfer between the two components is fast due to the absence of a middleware.



Two-Tier Client-Server Architecture

The Client-Application contains the codes for interfacing with the user and also for saving data in the database server.
The Client-Application sends the request to the server and it processes the request and sends back with data.
This means client Application handles both the Presentation layer (application interface) and the Application layer (logical operations).
The client-Application layer can be build using languages such as C, C++, Java, Python, PHP, Rails, .NET.

Two-Tier Client-Server Architecture

the database server handles the data management layer. Data management layer consists of data storage(database or file system) and methods for storing and retrieving data from the data storage. Commonly used databases include MySQL, MongoDB, PostgreSQL, SQLite. Hosting is possible on-premises or in the cloud.

Two-Tier Client-Server Architecture

Two-tiered application examples include:

- desktop applications
- games
- music players.

Advantages:

- It is fast and easy to implement
- communication is faster
- It is suitable in an environment where business rules or logic operations are static

Disadvantages:

- It is not easily scalable, thus performance degrades as users scale.
- Data integrity issue may arise due to the server responding to multiple requests at the same time.

Difference Between Two-Tier And Three-Tier Database Architecture

Two-Tier Database Architecture		Three-Tier Database Architecture
It is a Client-Server Architecture.	VS	It is a Web-based application.
In two-tier, the application logic is either buried inside the user interface on the client or within the database on the server (or both).	VS	In three-tier, the application logic or process resides in the middle-tier, it is separated from the data and the user interface.
Two-tier architecture consists of two layers : Client Tier and Database (Data Tier).	VS	Three-tier architecture consists of three layers : Client Layer, Business Layer and Data Layer.
It is easy to build and maintain.	VS	It is complex to build and maintain.
Two-tier architecture runs slower.	VS	Three-tier architecture runs faster.
It is less secured as client can communicate with database directly.	VS	It is secured as client is not allowed to communicate with database directly.
It results in performance loss whenever the users increase rapidly.	VS	It results in performance loss whenever the system is run on Internet but gives more performance than two-tier architecture.
Example – Contact Management System created using MS-Access or Railway Reservation System, etc.	VS	Example – Designing registration form which contains text box, label, button or a large website on the Internet, etc.

References

- https://medium.com/@fleviankanaiza/two-tier-three-tier-architecture-8b02536d3482
- https://www.geeksforgeeks.org/difference-between-two-tier-and-three-tier-database-architecture/
- https://www.researchgate.net/publication/238329276_A_Survey_of_Research_into_Legacy_System_Migration/figures?lo=1

Any Question ?

Thank You