

3-Tier DBMS Architecture

3-tier architecture is the most widely used architecture in real-world large and complex web applications like Facebook. This architecture separates the application into three logically distinct layers:

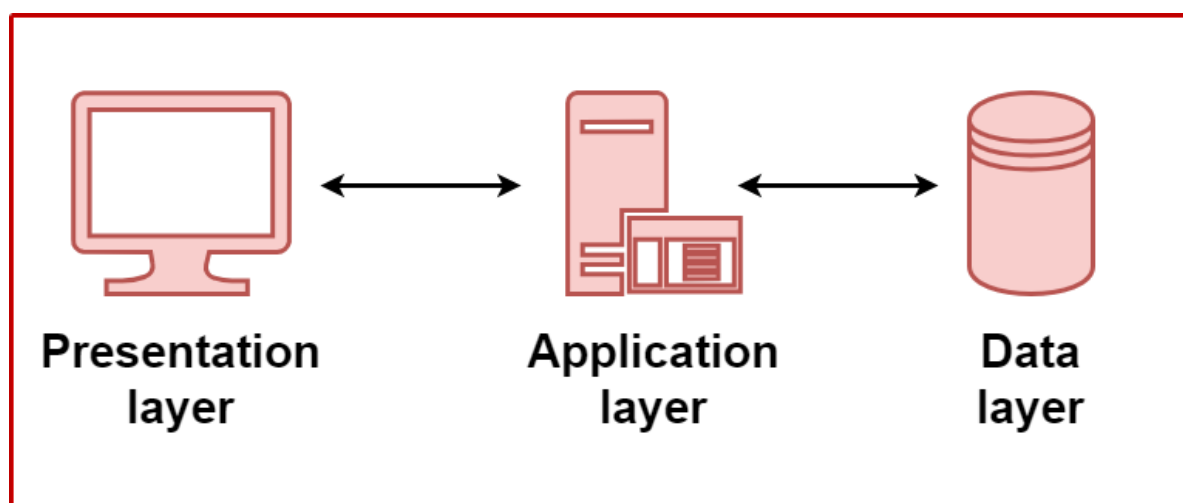
- **Presentation Layer** - Handles the user interface
- **Application Layer** - Manages business logic
- **Data Layer** - Manages data storage and processing

In this architecture, the end user has no idea about the existence of the database beyond the application server. Similarly, the database has no knowledge about any other user beyond the application.

Let us imagine an online shopping scenario. Whenever you go onto a platform like Amazon and perform an operation, such as searching for a product or adding an item to your cart, you interact with the frontend or User Interface (UI) of the application, which is represented by the **Presentation Layer**.

Any operation performed at the presentation layer gets sent to the next layer, the **Application Layer**, as a request. Upon receiving the request, the application layer interprets the request, formulates a corresponding database query, and sends it to the **Data Layer**. The data layer, which stores all the data, receives the query and executes it against the database.

Relevant information, such as product details and user account information, is retrieved, and the results are sent back to the application layer. The application layer processes the data further if necessary (e.g., formatting) and sends the response back to the presentation layer. The presentation layer then presents the results to the user, completing the process.



Advantages of 3-Tier Architecture

- **Scalability:** Enhanced scalability due to the distributed deployment of application servers. Each layer can be adjusted without altering other layers.

- **Security:** The client does not directly interact with the server, providing an added layer of security.
- **Modularity and Maintainability:** Maintenance is simplified due to the separation of responsibilities.
- **Performance:** Individual optimisation of presentation or application tiers is possible, leading to better performance.

Disadvantages of 3-Tier Architecture

- **Increased Complexity:** The introduction of an extra middle layer increases the complexity of the system. Communication points are also doubled.
- **Potential Latency Issues and Bottlenecks:** The added step of processing increases the possibility of latency and bottlenecks, as problems can arise in any layer at any time.
- **Longer Development Time:** Implementing three tiers with different logics and distributed responsibilities results in longer development time.
- **Resource Overhead:** Implementing an extra tier causes resource overhead for development, processing, and maintenance of the architecture.