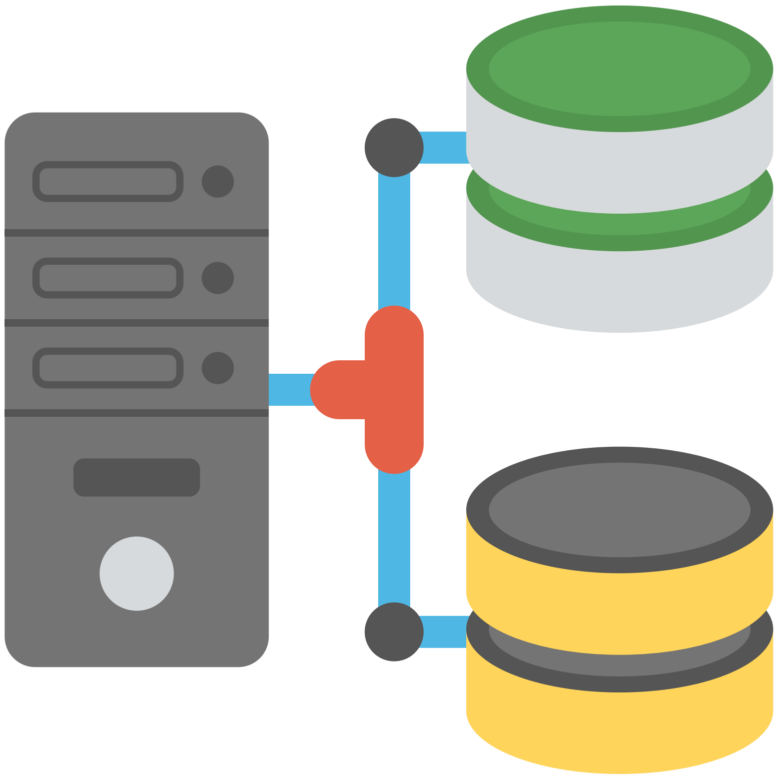
DBMS Architecture

Short Notes



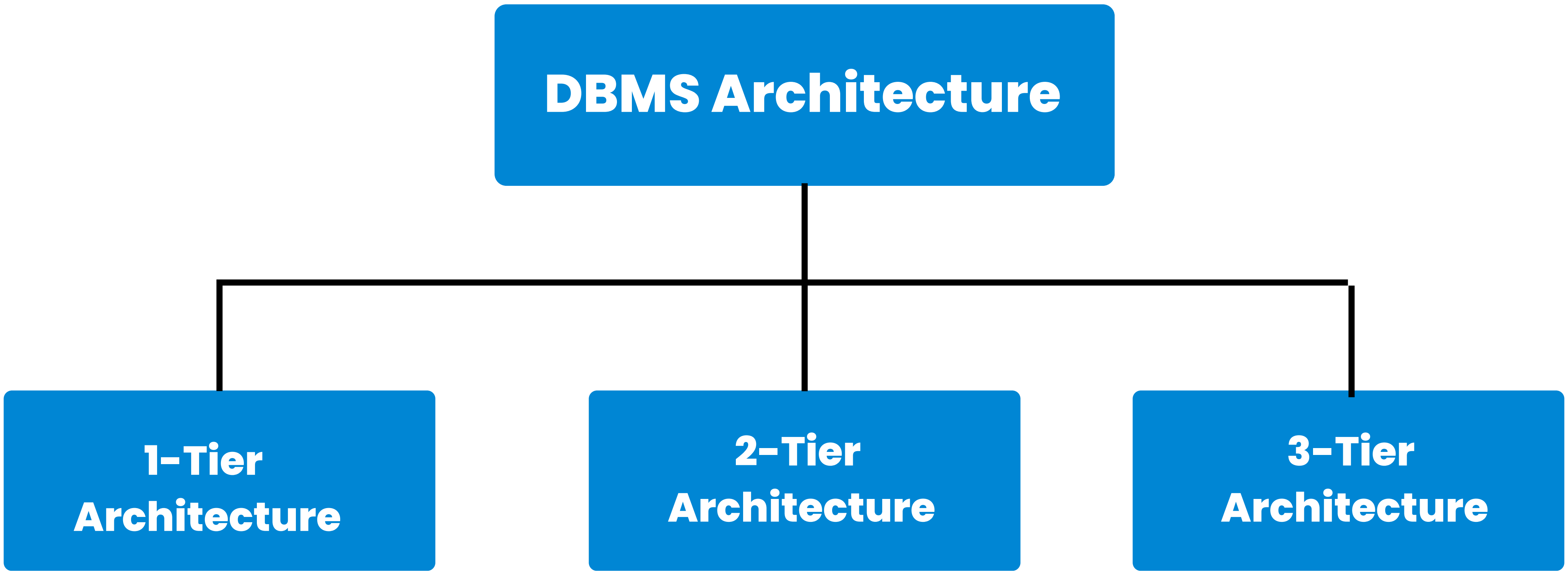
@curious_.programmer

DBMS Architecture

Database architecture refers to the overall design and structure of a database system, including its components, interactions, and relationships.

It involves defining the data models, storage formats, access methods, and other essential elements that make up the database.

The architecture of a database is critical for determining how efficiently and effectively data can be stored, accessed, and manipulated.


DBMS Architecture

1-Tier Architecture

2-Tier Architecture

3-Tier Architecture

@curious_.programmer

1- Tire Architecture

A single-tier architecture in a database management system (DBMS) is a design in which all the components of the database system, including the user interface, application logic, and database management functions, run on a single machine.

In simpler terms, this means that everything related to the database, such as storing data, accessing data, and processing data, happens on a single computer or server. This approach is sometimes referred to as a "standalone" or "monolithic" architecture.

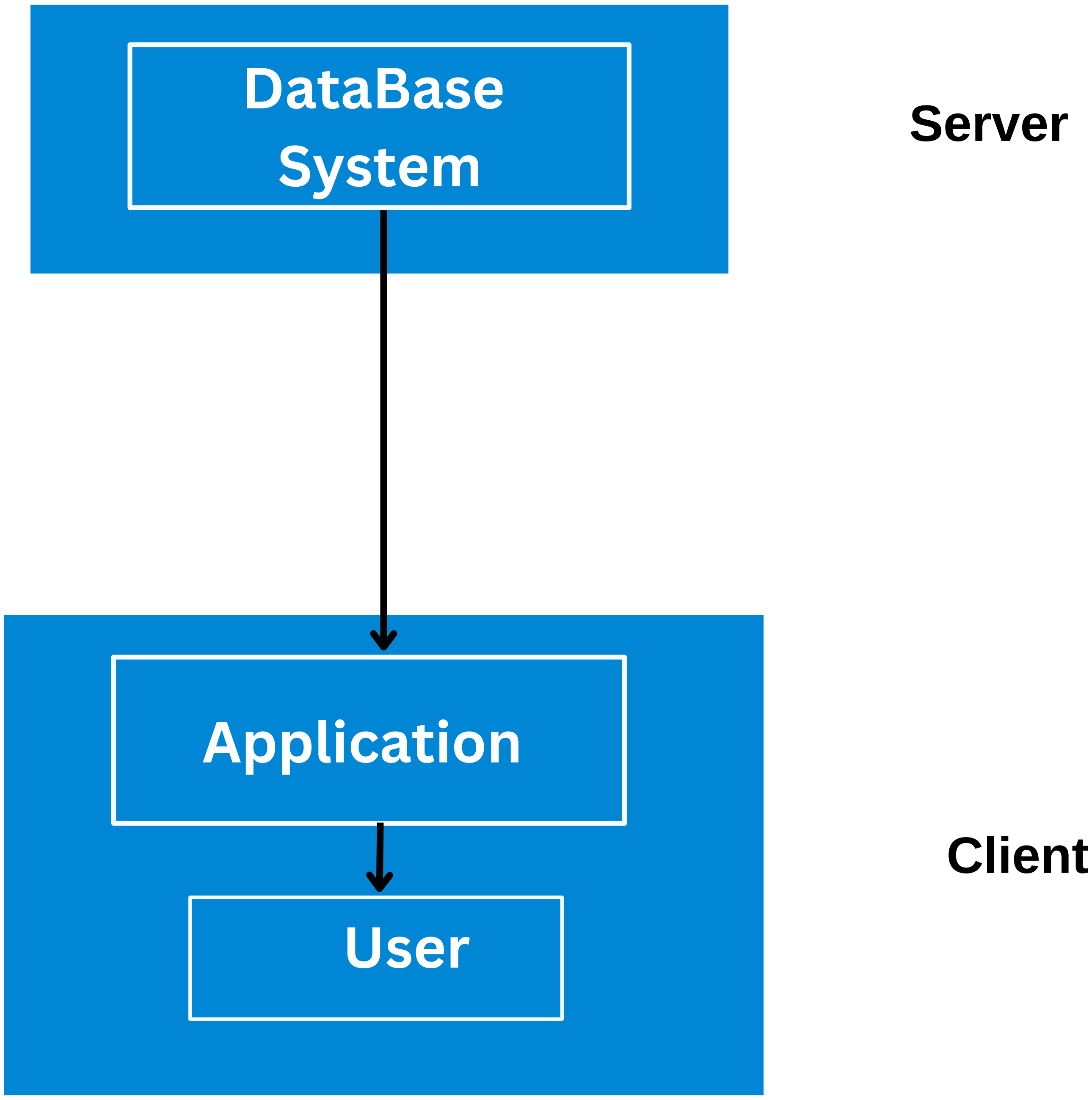
Single-tier architecture is easy to set up and manage, but it can have limitations in terms of scalability, performance, and security.

It may not be suitable for larger, more complex systems that require distributed processing or multi-user access.

2- Tire Architecture

Two-tier architecture in DBMS, also known as client-server architecture, is a type of database architecture that consists of two layers or tiers: the client layer and the server layer.

The client layer is the interface that allows users or applications to interact with the database. It includes the user interface, application logic, and presentation layer. The client layer sends requests to the server layer and receives responses from it.



The server layer is responsible for managing the database and providing services to the client layer. It includes the database management system, data storage, and database server.

The server layer processes requests from the client layer, retrieves or updates data from the database, and sends responses back to the client layer.

The advantage of two-tier architecture is its simplicity and ease of use.

It is suitable for small to medium-sized applications with a limited number of users. However, it may not be scalable or secure enough for large, complex systems that require high performance and reliability.

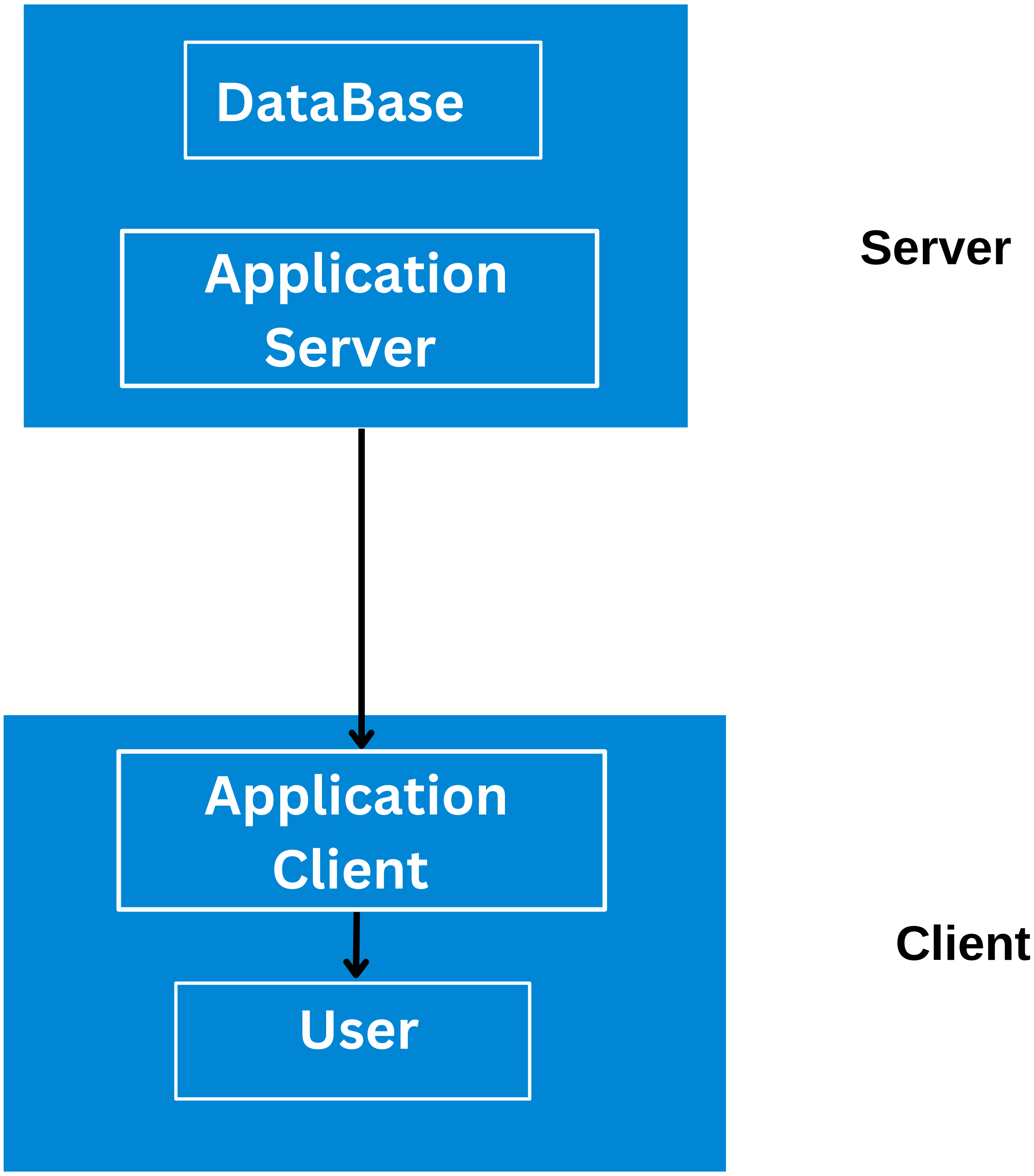
3- Tire Architecture

The three-tier architecture in DBMS, also known as the three-layer architecture, is a design pattern that separates the user interface, application logic, and database management into distinct layers.

The first tier, also known as the presentation layer, is where the user interface resides.

This layer contains the graphical user interface (GUI) and provides a way for the user to interact with the application. The second tier, also known as the application layer, contains the application logic.

This layer processes user requests from the presentation layer and interacts with the database management system (DBMS) in the third tier to retrieve and manipulate data.



The second tier, also known as the application layer, contains the application logic. This layer processes user requests from the presentation layer and interacts with the database management system (DBMS) in the third tier to retrieve and manipulate data.

The third tier, also known as the data storage layer, contains the DBMS and the database itself. This layer is responsible for storing, retrieving, and managing data in the database.

By separating the layers, the three-tier architecture makes it easier to maintain and scale the application. Changes to one layer do not affect the others, allowing developers to make modifications without disrupting the entire system.

Additionally, the architecture provides better security by limiting direct access to the database and only allowing access through the application layer.

PDF Uploaded On Telegram

Curious Coder

67,119 subscribers

share link

https://t.me/Curious_Coder

description

Do join for coding resources, Handwritten notes & Quizzes! 🧑‍💻

LINK IN BIO

@curious_.programmer