**software architecture pattern in python**

يركز على تنظيم هيكل وتنظيم التطبيقات البرمجية بشكل أفضل

**software architecture pattern** the separation of concerns by dividing the software into distinct layers, each with a specific responsibility. This pattern aims to create a system that is independent of frameworks, databases, or any external dependencies, making it highly maintainable and testable. The core principle of software architecture pattern is the Dependency Rule.

* **Model-View-Controller (MVC)**
* **Model-View-Presenter (MVP):**
* **Model-View-ViewModel (MVVM):**
* ***Layered Architecture*:**
* ***Client-server architecture***
* ***Microkernel architecture Pattern:***
* ***Peer to Peer Architecture Pattern:***

1. **Model-View-Controller (MVC):** It separates the application into three parts: Model, which represents the data and operations associated with it, View, which displays the data to the user, and Controller, which manages interactions between the model and the view.

**A black device with a blue line

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

1. **Model-View-Presenter (MVP):** MVP is an architectural pattern that separates an application into three core components: Model, View, and Presenter. مقدم

**A black phone with orange text

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

1. **Model-View-ViewModel (MVVM):** It resembles the MVC pattern but uses the ViewModel pattern to separate view-related operations and data, allowing better management of interface updates.

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer program

Description automatically generated

***Layered Architecture***: Layered Architecture architecture is also called N-Tier architecture since the software is separated processing, data management, and presentation layers

* The Layered Architecture pattern is another way of achieving separation and independence
* Components within the layered architecture pattern are organized into horizontal layers, each layer performing a specific role within the application
* Organizes the system into layers with related functionality associated with each layer. A layer provides.services to the layer above it so the lowest-level layers represent core servicesIare likely to be usedthroughout the system.
* Although the layered architecture pattern does not specify the number and types of layers that must exist in the pattern, most layered architectures consist of four standard layers: presentation, business, persistence, and database

A diagram of a data source

Description automatically generated

***Client-server architecture:*** Where there are 2 entities. It has a set of clients and a server

* Client components send requests to the server, which processes them and responds back.
* When a server accepts a request from a client, it opens a connection with the client over a specific protocol.

A diagram of a server

Description automatically generated

***Microkernel architecture Pattern:*** The microkernel architecture pattern consists of two types of architecture components: a core system and plug-in modules. Application logic is divided between independent plug-in modules and the basic core system.

***Example:***

Let's say we have an online shopping application based on the microkernel pattern. The core can be a simple application that manages database connectivity and manages payments. While delivery modules can be sub-applications that add functionality such as a product rating system, express delivery system, or user management system. Each module represents a specific feature that can be extended or customized separately from the core, providing great flexibility in development.

A diagram of a micro-kernel

Description automatically generated

***Peer to Peer Architecture Pattern:***In the common client-server architecture, multiple clients will communicate with a central server. A Peer-to-peer (P2P) architecture consists of a decentralized network of peers nodes that are both clients and servers. P2P networks distribute the workload between peers, and all peers contribute and consume resources within the network without the need for a server.

***A diagram of a network

Description automatically generated***

*Economical(اقتصادي), here is no central server to maintain (اصلاح)and to pay for , so this type of networks can be more economical.*

*Scalability (التوسيع)— the network is very easy to scale up*