

5. - Architectural design patterns – MVC, multitier, monolithic, P2P, client / server

Monolithic

Monolithic využívá jednoduchou strukturu, často v jednom nebo dvou souborech. Jde o to, že všechny funkčnosti jsou v jednom layeru a obecně aplikace se nasazuje celá. To znamená, že se neřeší jednotlivé části aplikace, ale aplikace je brána jako jeden celek.

MVC

Model-View-Controller.

Model – Pracuje s informacemi, databáze, data z textových souborů.

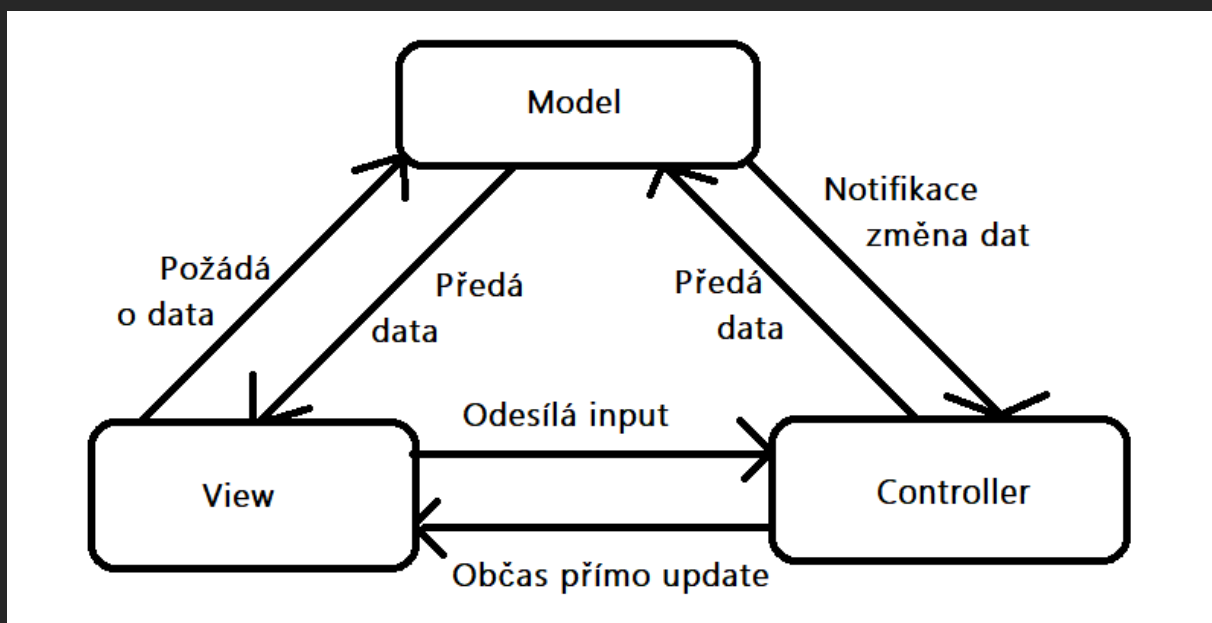
Controller – Pokud Controller potřebuje informace, zeptá se modelu, model odešle zpátky data. Controller posílá poté data do View, který vykresluje data

View – Vykresluje vizuální stránku, například v případě webového dokumentu. Uživatel vidí pouze view a komunikuje pouze s View, který zase posílá data do Controlleru.

MVC spolu komunikuje jakkoliv – Controller může komunikovat s Modelem a View, View může komunikovat s Modelem a Controllerem, Model může komunikovat s View a Controllerem.

View kontrolovat s Modelem může například při vykreslení přihlášeného uživatele.

Na rozdíl od Multi-tieru spolu mohou spolu komunikovat různé, nejen ty, které jsou blízko vedle sebe.



Client – server

Funguje na tom principu, že jeden počítač – klient žádá data nebo program druhého – serveru.

Principově funguje tak, že klient odešle požadavek, server ho zpracuje a zašle zpět odpověď, podle které následně může klient zaslat další požadavek.

Jako příklad se dá vzít například webová stránka – server dostane požadavek na soubor a vrátí ho uživateli, který může poté udělat další request. nebo FTP, která se pomocí klienta připojuje k serveru a mění na něm určité soubory.

Rozdělen na Thin, Hybrid, Thicc.

Thin – program, který se dá spustit před PuTTY a hádáme číslo, tudíž že vše funguje na straně serveru a klient jenom využívá aplikaci. Není žádné ukládání dat, když to vypne, přijde o všechno.

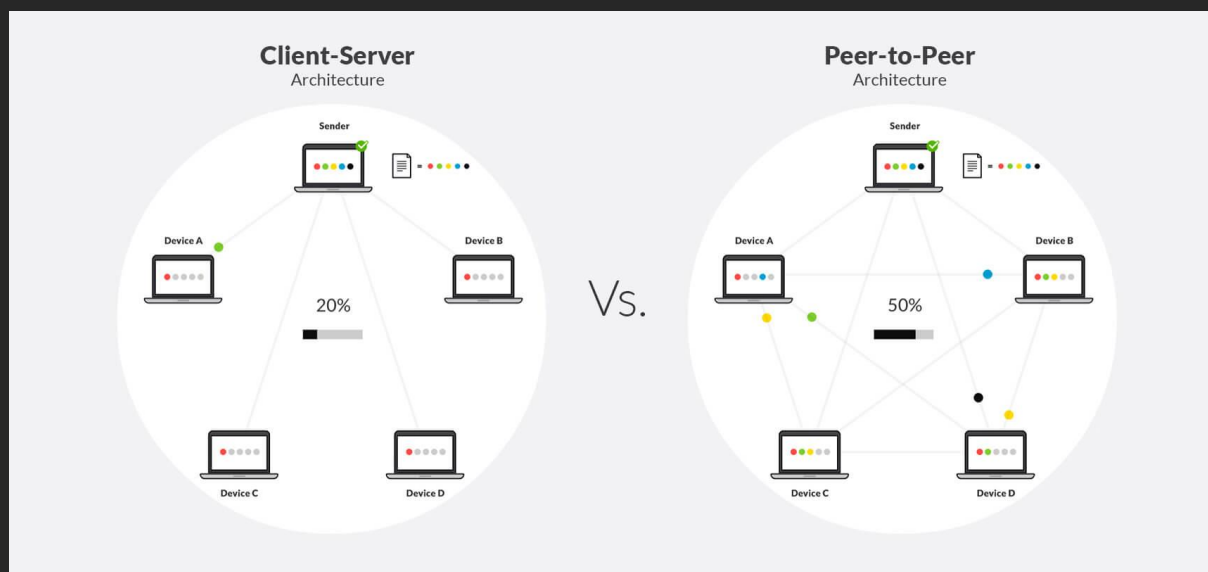
Thicc – Využívá CPU, je možné ukládat data. Příklad – videohry. World Of Tanks – Můžeme jezdit jak singleplayer, tak multiplayer. Můžeme hrát jenom na našich komponentech, našeho CPU, nebo na serveru.

Hybrid – Neukládá data a nikam je neposílá, jedná se o zprostředkovatel. Skype sám o sobě nefunguje, jde se přes prostředníka – Client -> Server <- Client.

Peer to Peer

Jedná se o architekturu, ve které spolu všichni uživatelé komunikují, klienti mezi sebou sdílejí data. Uživatelé třeba v případě Torrentu stahovat části čehokoliv od několika klientů najednou. Komunikace probíhá mezi klienty, jedná se o takovou síť klientů. Pokud odsekne klienta, přijdeme jenom o jeho data, ne o celou aplikaci, nepadne.

Každý klient zároveň může být jak klient, tak server (Jak jsme mohli zkusit u Alfa 4). Peer to Peer má svůj vlastní komunikační protokol, podle kterého spolu uživatelé komunikují. Kounikace může probírat pomocí Berkeleyeho socketu, což je počítačové rozhraní (API), které umožňuje komunikaci mezi různými počítači přes internet nebo ve stejné síti pomocí protokolu TCP.

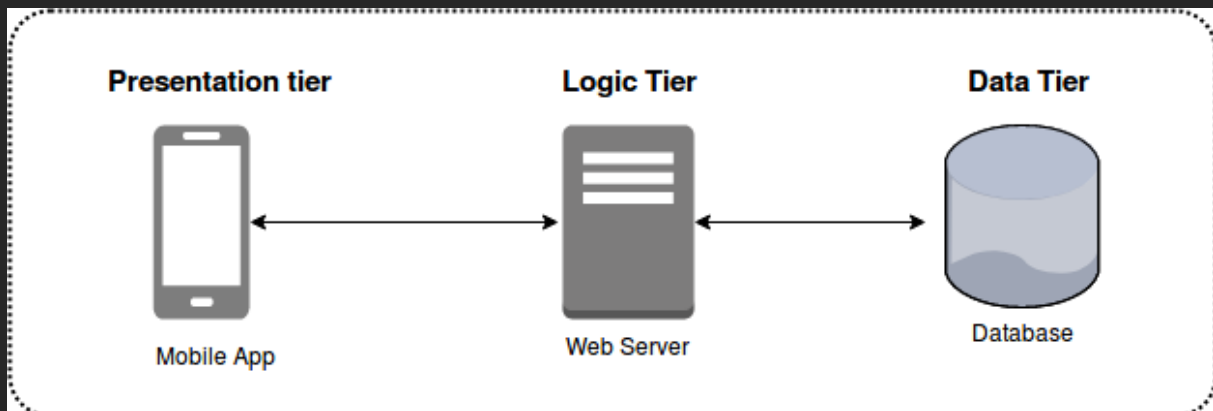


Multitier

Multitier se skládá z nejméně tří vrstev, ale může jich být více. Základními třemi jsou

- A.) Prezentační vrstva (Presentation layer), která se stará o komunikaci s klientem pomocí interface a získává jeho vstupy.
- B.) Aplikační vrstva (Application layer), která řídí byznys logiku aplikace a provádí všechny možné operace
- C.) Datová vrstva (Data / database layer), která zachovává informace a data a zároveň je poskytuje.

Vrstvy spolu mohou komunikovat jen a pouze mezi těmi, které jsou vedle sebe.



Multitier ale neznamenají přesně tři vrstvy, může to být i více vrstev. Tři jsou ovšem nejznámější.