

Architektura Softwarových Systémů

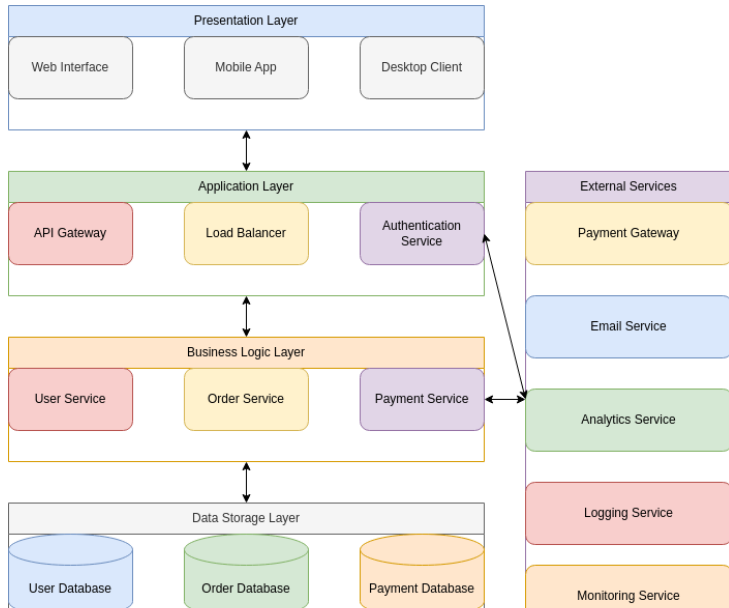
Shrnutí klíčových konceptů

GitHub Copilot

16. prosince 2025

- 1 Úvod do architektur SW systémů
- 2 Architektonické styly
- 3 Pohledy na SW architekturu
- 4 Modelování a dokumentace
- 5 Kvalitativní atributy
- 6 Architektonické vzory
- 7 Datová architektura
- 8 Zdroje

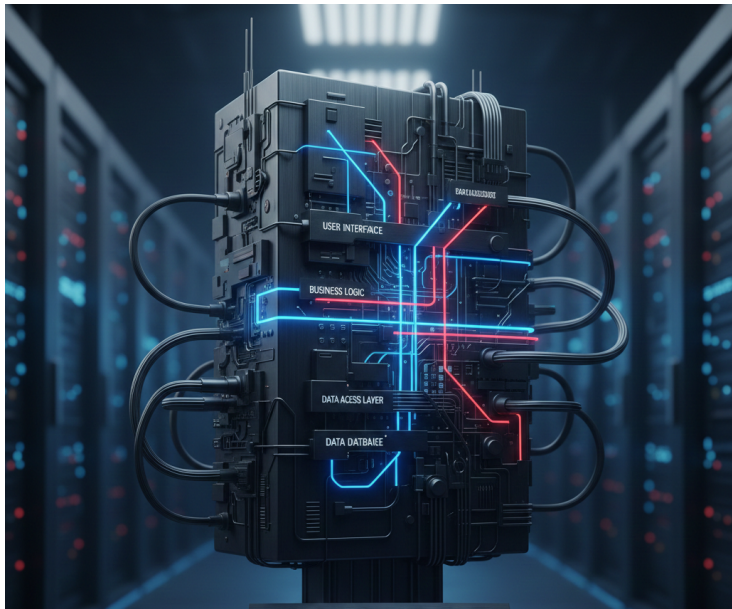
Vizualizace

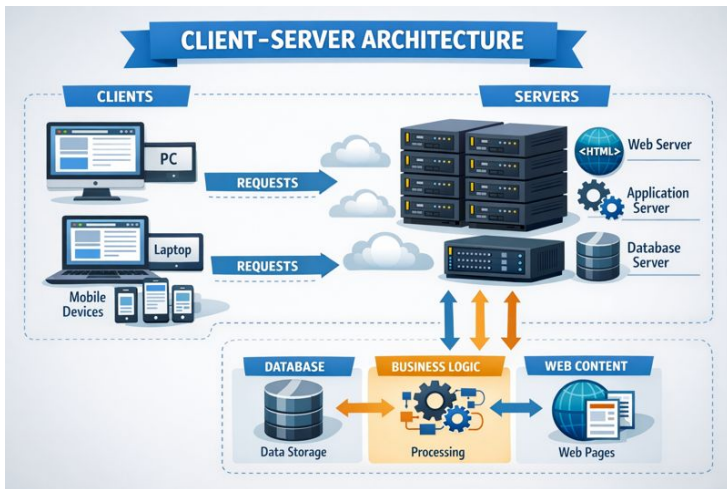


- **SW architektura** popisuje strukturu systému, komponenty a vztahy (Bass, Clements a Kazman 2012).
- Slouží k řízení složitosti a zajištění kvalitativních atributů (výkonnost, bezpečnost, modifikovatelnost).
- **Hlavní cíle:**
 - Komunikace mezi týmy
 - Základ pro rozhodnutí o návrhu
 - Dokumentace a údržba
- Tvoří se na vysoké úrovni abstrakce, bez implementačních detailů (Shaw a Garlan 1996).

- **Monolitický styl:** Vše v jedné aplikaci. Jednoduchý, ale hůře škálovatelný.
- **Klient-server:** Oddělení UI a logiky.
- **Vrstevnatá architektura:** Rozdělení do vrstev (prezentace, logika, data).
- **Pipes & Filters:** Transformace datových toků.
- **Event-driven:** Reakce na události, asynchronní systémy (Richards a Ford 2020).
- **Microservices:** Nezávislé služby, komunikace přes API (Fowler a Lewis 2014).

Monolitický styl





Obrázek: Klient-Server (ChatGPT)

Microservices



Obrázek: Microservices (ChatGPT)

Různé perspektivy pro různé stakeholdery:

- **Logický pohled:** Hlavní komponenty a interakce.
- **Fyzický pohled:** Rozmístění SW na HW (deployment).
- **Procesní pohled:** Dynamické chování a toky dat.
- **Datový pohled:** Struktura a tok dat.
- **Vývojový pohled:** Modulární struktura pro vývoj.

- **Cíle:** Sdílení znalostí, plánování, údržba.
- **Standardy a notace:**
 - UML (Use Case, Class, Component, Deployment diagramy)
 - ArchiMate
 - C4 model (Brown 2023)
- **Klíčové diagramy:** Komponentní, Nasazení, Toků dat.

Kvalitativní atributy (Quality Attributes)

Vlastnosti systému nad rámec funkčnosti (Bass, Clements a Kazman 2012):






- **Dostupnost (Availability):** Odolnost vůči chybám.
- **Modifikovatelnost (Modifiability):** Snadná změna.
- **Výkonnost (Performance):** Rychlost a efektivita.
- **Bezpečnost (Security):** Ochrana dat a přístupu.
- **Integrovatelnost, Znovupoužitelnost, Testovatelnost.**
- **Uživatelská přívětivost (Usability).**

Ověřená řešení opakujících se problémů:

- **MVC (Model-View-Controller):** Oddělení dat, UI a logiky.
- **Layered (Vrstevnatý):** Hierarchické uspořádání zodpovědností.
- **Repository:** Centrální úložiště dat.
- **Broker:** Middleware pro komunikaci.
- **Peer-to-Peer:** Decentralizovaná spolupráce uzlů.

Správa a tok dat v systému:

- **Datové modely:** ER diagramy, relační vs. NoSQL.
- **Úložiště:** Data warehouses, Data lakes.
- **Integrace:** Datové toky.
- **Správa:** Security, zálohování, kvalita dat.

-  Bass, Len, Paul Clements a Rick Kazman (2012). *Software Architecture in Practice*. 3rd. Addison-Wesley Professional.
-  Brown, Simon (2023). *The C4 model for visualising software architecture*. <https://c4model.com/>. Accessed: 2023-10-27.
-  Fowler, Martin a James Lewis (2014). *Microservices*. <https://martinfowler.com/articles/microservices.html>. Accessed: 2023-10-27.
-  Richards, Mark a Neal Ford (2020). *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly Media.
-  Shaw, Mary a David Garlan (1996). *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall.