

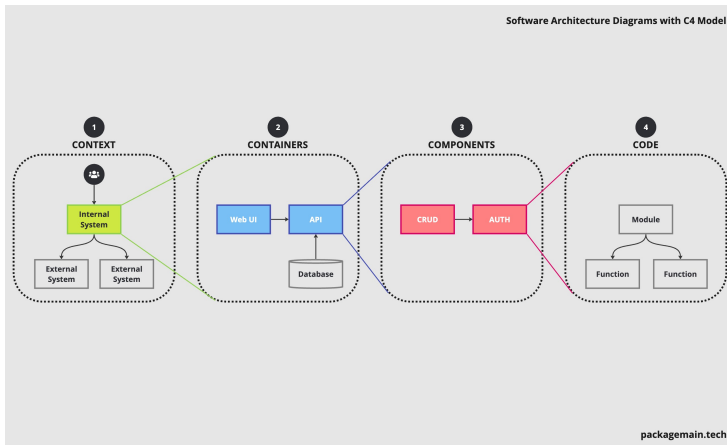
# Architektura softwarových systémů

## Shrnutí konverzace

AI Assistant

16. prosince 2025

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



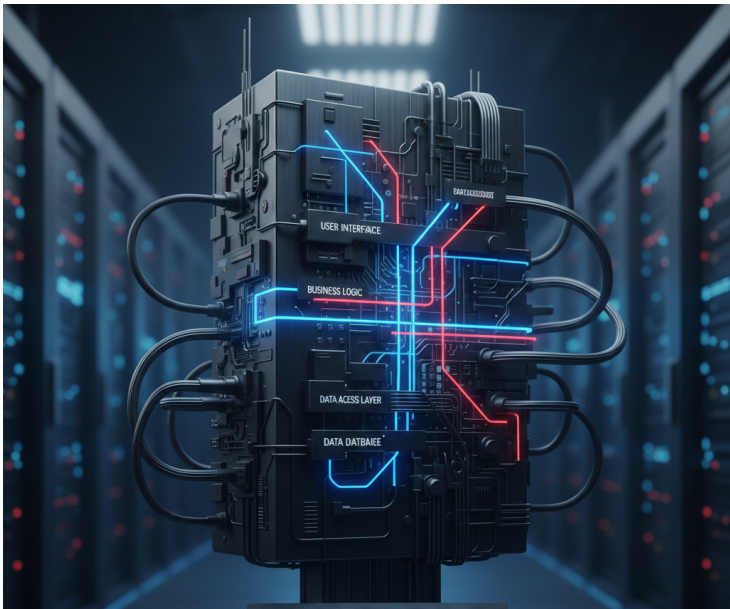
Obrázek: Vizualizace z konverzace

- Softwarová architektura popisuje **vysokou strukturu systému**, komponenty a jejich interakce.
- Cílem je **zajištění kvality, flexibility a udržitelnosti** systému.
- Klíčové role architektury: plánování, dokumentace, rozhodování.

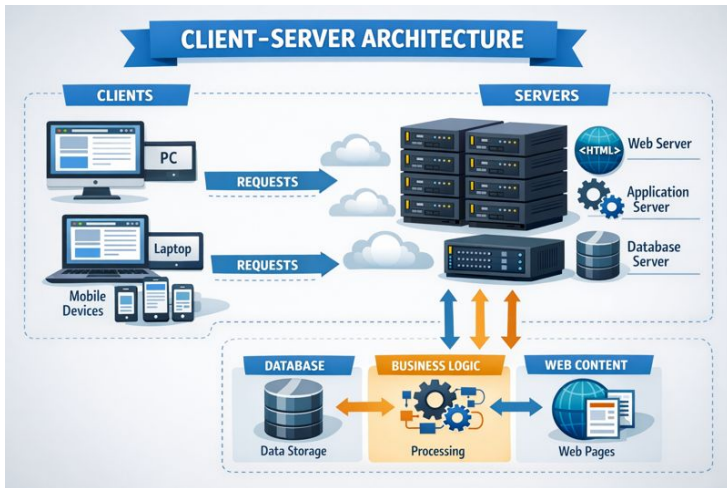
Zdroj: [4]

- **Vrstvená (Layered)** – logické vrstvy, separace zodpovědností.
- **Client–Server** – centralizovaná logika, komunikace přes síť.
- **Monolit** – jeden celek, jednoduchý na začátku, horší škálovatelnost.
- **Microservices** – malé nezávislé služby, škálovatelnost.
- **SOA** – služby s Enterprise Service Bus.
- **Event-Driven** – komunikace přes události, real-time systémy.
- **Hexagonální / Clean Architecture** – oddělení jádra od adaptérů.

Zdroj: [3]



# Client-Server



Obrázek: Client-Server architektura



Obrázek: Microservices architektura



- Pohledy umožňují různým stakeholderům **vidět systém podle svých potřeb**.
- **4+1 Views** (Philippe Kruchten):
  - ① Logický – co systém dělá
  - ② Procesní – chování za běhu
  - ③ Vývojový – modulární struktura kódu
  - ④ Fyzický – nasazení a infrastruktura
  - ⑤ Scénáře / Use Cases – případy použití
- Alternativy: C4 model, ISO/IEC 42010.

Zdroj: [6]

- **Modelování** – vytváření diagramů a modelů pro pochopení systému.
- **Dokumentace** – soubor artefaktů: struktura, rozhodnutí, pohledy.
- Standardy: UML, C4, ArchiMate.
- **ADR** (Architecture Decision Records) – zaznamenání důležitých rozhodnutí.
- Doporučení: udržovat dokumentaci živou, používat nástroje (PlantUML, Structurizr, Archi).

Zdroj: [1]

- **Klíčové atributy:** Dostupnost, Modifikovatelnost, Výkonnost, Bezpečnost, Integrovatelnost, Znovupoužitelnost, Testovatelnost, Uživatelská přívětivost.
- Slouží k definici *nefunkčních požadavků* a ovlivňují architektonická rozhodnutí.

Zdroj: [7]

- **Příklady:** Vrstvená, Client–Server, Event-Driven, Microservices, Hexagonální, Microkernel, Space-Based, CQRS, P2P, Strangler, MVC.
- Vzory řeší opakující se architektonické problémy a podporují škálovatelnost, modularitu a udržitelnost.

Zdroj: [2]

- Popisuje **strukturu, toky, úložiště a správu dat** napříč systémy a organizací.
- **Klíčové komponenty:** datové modely (konceptuální, logické, fyzické), datové toky, úložiště, governance, přístup.
- Důležitá pro: kvalitu, bezpečnost, analytiku, AI/ML, integraci a rozhodování.

Zdroj: [5]

- [1] Software System Design. *Modeling Documentation*. Accessed: 2025-12-16. URL: <https://softwaresystemdesign.com/software-architecture-design/modeling-and-documentation/architecture-documentation-best-practices/>.
- [2] GeeksforGeeks. *Software Architecture Patterns*. Accessed: 2025-12-16. URL: <https://www.geeksforgeeks.org/software-engineering/types-of-software-architecture-patterns/>.
- [3] Software Architecture Guild. *Architecture Styles*. Accessed: 2025-12-16. URL: <https://software-architecture-guild.com/guide/architecture/fundamentals/architecture-styles/>.

- [4] Software Engineering Institute. *Software Architecture*. Accessed: 2025-12-16. URL: <https://resources.sei.cmu.edu/library/subject-areas/software-architecture/>.
- [5] SAP. *What is Data Architecture*. Accessed: 2025-12-16. URL: <https://www.sap.com/cz/resources/what-is-data-architecture>.
- [6] Wikipedia. *4+1 architectural view model*. Accessed: 2025-12-16. URL: [https://en.wikipedia.org/wiki/4%2B1\\_architectural\\_view\\_model](https://en.wikipedia.org/wiki/4%2B1_architectural_view_model).
- [7] Wikipedia. *FURPS*. Accessed: 2025-12-16. URL: <https://en.wikipedia.org/wiki/FURPS>.