

Architektura Klient-Server

Klient-server je síťový model, ve kterém jsou jednotlivá zařízení v síti rozdělena do dvou jasně definovaných rolí. Komunikace obvykle probíhá stylem „dotaz-odpověď“ (request-response). Tento model umožňuje centrální správu dat a efektivní sdílení výpočetních zdrojů.

Hlavní role v systému

1. Klient (User-Side)

Klient je zařízení nebo aplikace, která iniciuje kontakt se serverem.

- **Funkce:** Odesílá požadavky, přijímá odpovědi a prezentuje data uživateli.
- **Příklady:** Webový prohlížeč (Chrome, Firefox), e-mailový klient (Outlook), mobilní aplikace.

2. Server (Service-Side)

Server je výkonný počítač nebo program, který neustále běží a čeká na příchozí požadavky.

- **Funkce:** Zpracovává požadavky, provádí výpočty, spravuje databáze a odesílá výsledky klientům.
- **Vlastnost:** Jeden server může obsluhovat stovky až miliony klientů současně.

Jak probíhá komunikace?

Celý proces lze rozdělit do čtyř základních kroků:

1. **Požadavek (Request):** Klient pošle přes síť zprávu serveru (např. "Chci zobrazit stránku google.com").
2. **Zpracování:** Server přijme požadavek, ověří identitu klienta a vyhledá potřebná data.
3. **Odpověď (Response):** Server pošle data zpět klientovi (např. HTML kód stránky).
4. **Zobrazení:** Klient přijatá data interpretuje a zobrazí uživateli.

Typy Klient-Server architektur

Model se dělí podle toho, jak moc práce odvádí klient a jak moc server:

Typ	Popis
Thin Client (Tenký klient)	Klient slouží jen pro zobrazení, téměř veškerý výpočet probíhá na serveru.
Thick Client (Tlustý klient)	Klient sám provádí většinu logiky a výpočtů, server používá jen pro ukládání dat.
2-Tier (Dvouvrstvá)	Klient komunikuje přímo s databázovým serverem.
3-Tier (Třívrstvá)	Mezi klienta a databázi je vložen aplikační server (nejčastější u webových aplikací).

Výhody a nevýhody

Výhody:

- **Centralizace:** Data jsou na jednom místě, což usnadňuje zálohování a správu práv.
- **Škálovatelnost:** Výkon lze zvýšit upgradem serveru nebo přidáním dalších serverů.
- **Bezpečnost:** Server může kontrolovat přístup ke zdrojům.

Nevýhody:

- **Single Point of Failure:** Pokud selže server, celá síť (všichni klienti) přestane fungovat.
- **Náklady:** Výkonné servery a jejich správa jsou drahé.
- **Zahlcení:** Pokud se připojí příliš mnoho klientů naráz, server se může zpomalit nebo spadnout (DoS).

Srovnání: Klient-Server vs. Peer-to-Peer (P2P)

Zatímco klient-server má jasnou hierarchii, v modelu **P2P** jsou si všechna zařízení rovna a každé může být zároveň klientem i serverem (např. BitTorrent).

Vlastnost	Klient-Server	Peer-to-Peer (P2P)
—	—	—
Správa	Centrální	Decentralizovaná
Závislost	Vysoká na serveru	Nízká (síť funguje i při výpadku uzlu)
Použití	Webové stránky, firemní sítě	Sdílení souborů, kryptoměny

Související pojmy: Server, IP adresa, HTTP, P2P, DoS útok, Databáze, Cloud Computing.

From:

<https://www.serviceit.cz/> - **IT ENCYKLOPEDIE**

Permanent link:

https://www.serviceit.cz/doku.php?id=client_server_architecture

Last update: **2025/12/31 20:17**

