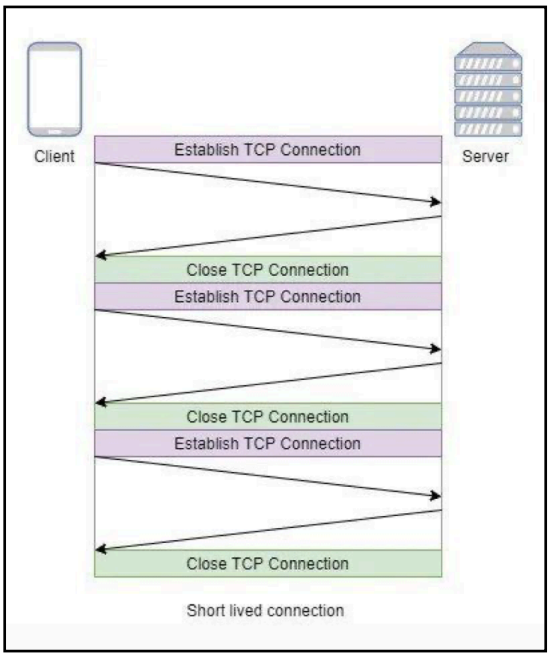
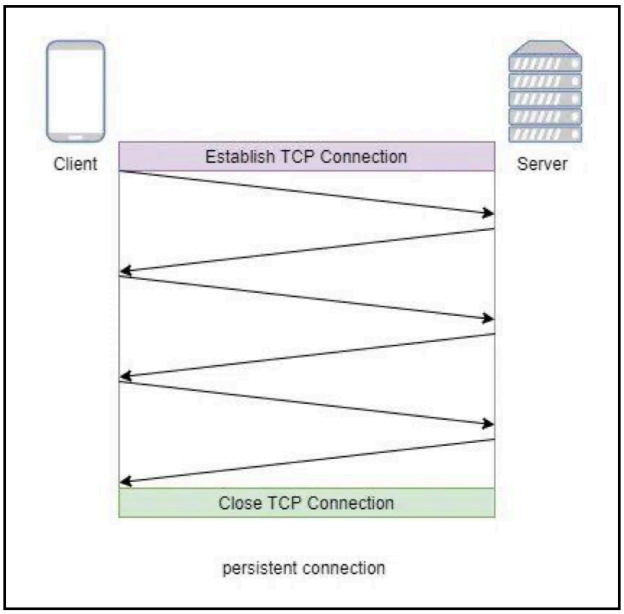
HTTP protokol - stavové kódy, autentizace, verze, HTTPS a SSL

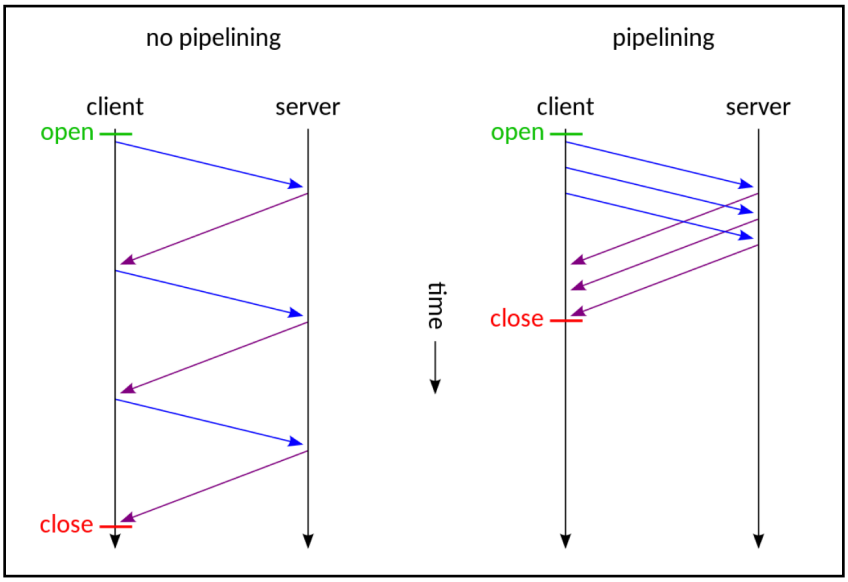
HTTP protokol

- **HTTP:**
  - Základní protokol webové komunikace
  - Bezstavový protokol:
    - Každý požadavek je nezávislý
    - Server neuchovává informace mezi požadavky
    - Výhody a nevýhody:
      - + Snadná škálovatelnost
      - - Nutnost použití cookies pro uchování stavu
  - Vytvořeno v roce 1991 **Timem Berners-Leem**
  - Cílem = sdílet informace na internetu
- **HTTP/1.0:**
  - První oficiální verze z roku 1996
  - Jednoduchá omezení
  - Pouze metoda **GET**, **POST**
  - Každý požadavek = nové TCP spojení

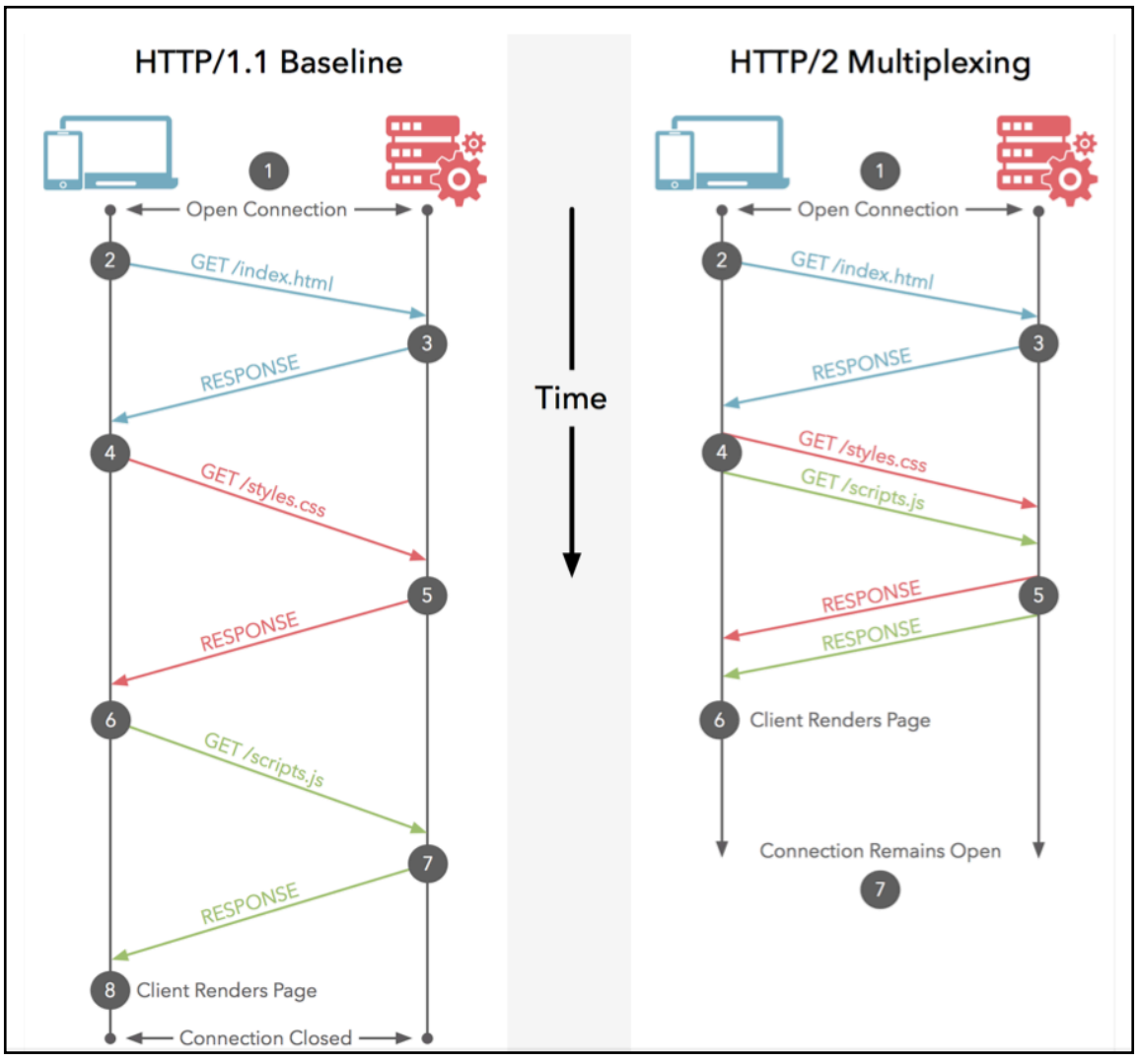


- **HTTP/1.1:**
  - Vylepšená verze z roku 1997
  - Více metod: **OPTIONS**, **PUT**, **DELETE**
  - Přidány funkce jako **keep-alive** a **pipelining**
    - **Pipelining:**
      - Technika, která umožňuje odeslat více HTTP požadavků přes jedno spojení bez čekání na odpovědi
      - **Výhody**: snížení latence, efektivnější využití spojení
      - **Nevýhody**: *Head-of-line blocking* (situace, kdy celá fronta zpráv nebo paketů je zpožděna kvůli zpoždění na jejím začátku.), ne všechny servery je podporují





- **HTTP/2:**
  - Rok vydání: 2015
  - **Binární formát**:
    - Rychlejší zpracování
    - Menší nároky na zdroje
  - **Multiplexing**:
    - Více požadavků přes jedno spojení
    - Řeší problém „*Head-of-line blocking*“
  - **Server Push**:
    - Server může poslat data, aniž by je klient vyžádal
    - Zrychluje načítání stránek



- **HTTP/3:**
  - Rok vydání: 2020
  - Založeno na **QUIC**:
    - Nový transportní protokol
    - Lepší výkon v nestabilních sítích
  - Odstranění „Head-of-line blocking“:
    - Efektivnější zpracování požadavků
    - Rychlejší odezva
  - Větší bezpečnost:
    - Vestavěné šifrování
    - Lepší ochrana proti útokům

Stavové kódy

- **Co to je:**
  - Trojmístné číselné kódy
  - Indikují výsledek požadavku
- **Kategorie:**
  - **2xx -** Úspěch
  - **3xx -** Přesměrování
  - **4xx -** Klientská chyba
  - **5xx -** Serverová chyba
- **Často používáné kódy:**
  - **200 OK -** Úspěšný požadavek
  - **404 Not Found -** Zdroj nenalezen
  - **500 Internal Server Error -** Chyba na serveru

- **Speciální kódy:**
  - **301 Moved Permanently -** Trvale přesměrování
  - **401 Unauthorized -** Neautorizovaný přístup
  - **503 Service Unavailable -** Služba nedostupná

Autentizace

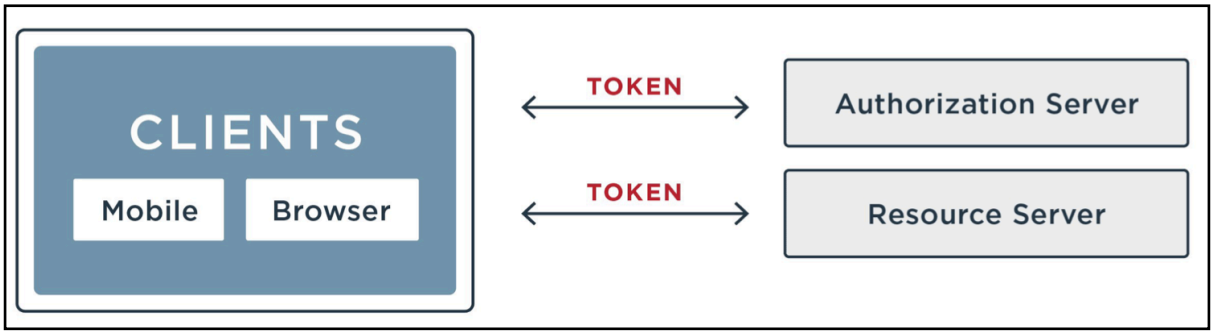
- **Definice:**
  - Autentizace je proces ověření identity uživatele nebo systému
- **Důležitost:**
  - Zajištění bezpečného přístupu k datům a službám
- **Postup autentizace:** záleží
- **Výhody**:
  - Jednoduchost
  - Rychlost
- **Nevýhody**:
  - Nízká bezpečnost (**základní**)
  - Komplexnost (**OAuth**)

Typy autentizace

- **Základní autentizace**:
  - **Mechanismus**:
    - Uživatelské jméno a heslo jsou zakódovány v **Base64** a odeslány na server
      - **Base64**:
        - **Definice**: je metoda kódování dat do ASCII textového formátu. Vytváří 64 znaků z ASCII tabulky k reprezentaci 6 bitů binárních dat
        - **Použití**: v autentizaci, přenosu dat, vložení obrázků do HTML a dalších aplikací, kde je potřeba zakódovat binární data
        - **Výhody**: umožňuje přenos dat přes textové protokoly jako HTTP, SMTP atd.
        - **Nevýhody**: zvětšuje velikost dat o zhruba 33%
  - **Bezpečnost**:
    - Je považována za nejméně bezpečnou, protože data nejsou šifrována
  - **Příklad**:
    - ‘Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==,

- **Digest autentizace**:
  - **Mechanismus**:
    - Používá hashování a několik dalších kroků pro zvýšení bezpečnosti
  - **Bezpečnost**:
    - Je bezpečnější než základní autentizace, ale stále není ideální pro citlivá data
  - **Příklad v cURL**: *curl --digest -u username:password http://example.com*

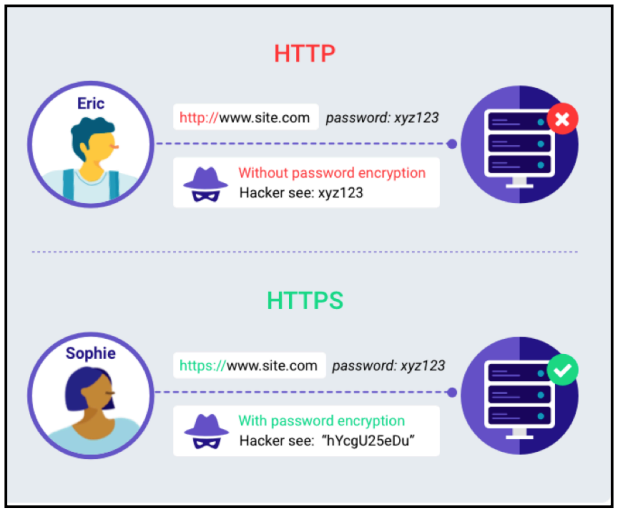
- **Token-Based autentizace**:
  - **Mechanismus**:
    - Po úspěšné autentizaci server vytvoří unikátní token, který je použit pro další požadavky
  - **Bezpečnost**:
    - Je považována za velmi bezpečnou metodu autentizace
  - **Příklad v cURL**:
    - *curl -H "Authorization: Bearer ACCESS_TOKEN" http://example.com*



- **OAuth**:
  - **Mechanismus**:
    - Deleguje autentizaci na externí službu, jako je Google nebo Facebook
  - **Bezpečnost**:
    - Je považována za velmi bezpečnou metodu autentizace
  - **Příklad v cURL**:
    - *curl -H "Authorization: Bearer OAUTH_TOKEN“ http://example.com*

HTTPS

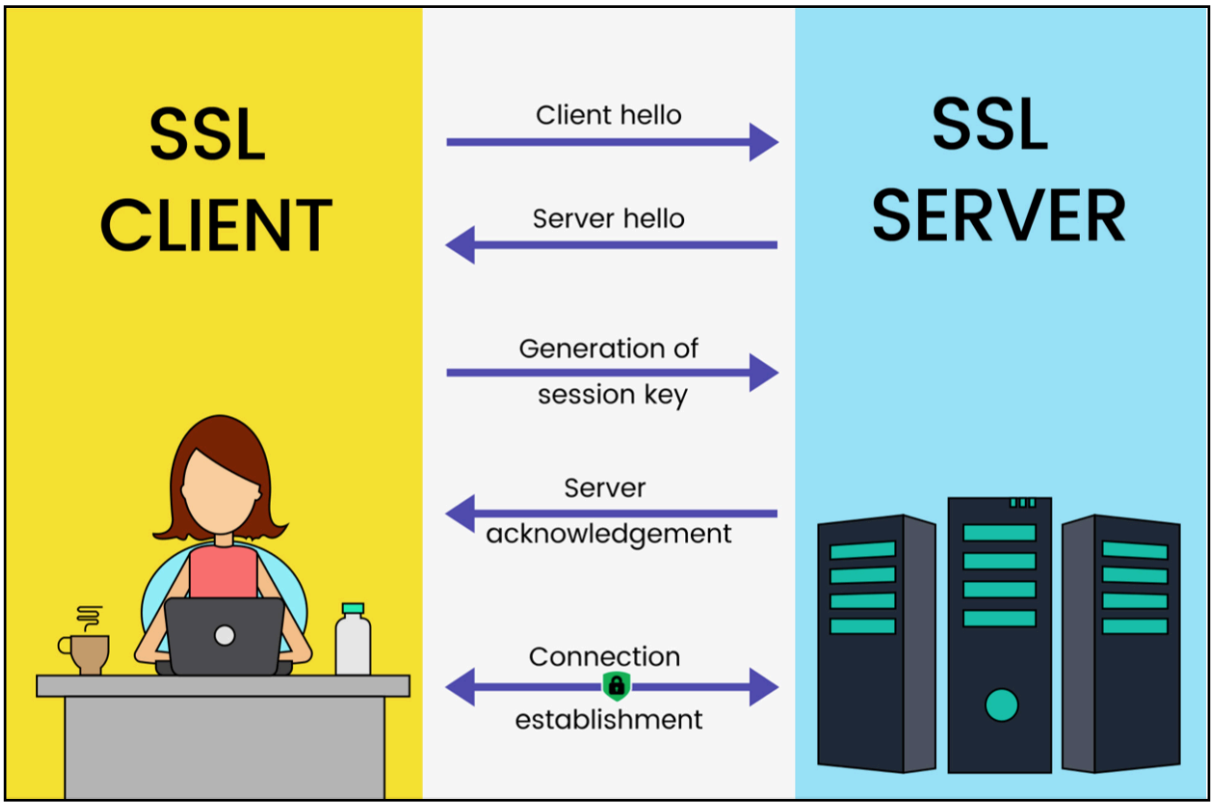
- **Definice**:
  - HTTPS (Hypertext Transfer Protocol Secure) je rozšíření HTTP s vrstvou zabezpečení
- **Použití**:
  - Pro zabezpečený přenos dat mezi klientem a serverem
- **Jak funguje**:
  - Používá **SSL/TLS** protokoly k šifrování dat
- **Výhody a nevýhody**:
  - **Výhody**: Zabezpečení, důvěryhodnost
  - **Nevýhody**: Pomalejší rychlost, náročnější na zdroje



SSL a TLS

- **SSL (Secure Sockets Layer)**:
  - Protokol pro zabezpečení komunikace na internetu
- **TLS (Transport Layer Security)**:
  - Následník **SSL**, modernější a bezpečnější
- **Jak fungují**:
  - Používají asymetrickou kryptografii a digitální certifikáty k zabezpečení komunikace

- **Jak funguje SSL handshake?**:
  1. Klient pošle „ClientHello“ zprávu s podporovanými verzemi **SSL/TLS**
  2. Server odpoví „ServerHello“ a vybere verzi a šifrovací metody
  3. Server pošle svůj digitální certifikát
  4. Klient ověří certifikát a pošle „Finished“ zprávu



Certifikáty a autority

- **Digitální certifikát**: Elektronický „průkaz totožnosti“ serveru
- **Certifikační autorita (CA)**:
  - Třetí strana, která vydává a ověřuje digitální certifikáty
- **Jak získat certifikát**:
  - Požádat o něj důvěryhodné certifikační autority
  - Ověření domény a vlastnictví
  - Platba a vydání certifikátu
- **Jak se stát autoritou**:
  - Splnění přísných bezpečnostních a provozních standardů
  - Audit od externího auditora
  - Získání důvěry od prohlížečů a operačních systémů
- **Jak zaručíme bezpečnost**:
  - Pravidelné audity
  - Ověření a revokace certifikátů