



Informační systémy

Architektury informačních systémů

Doc. Ing. Radek Burget, Ph.D.

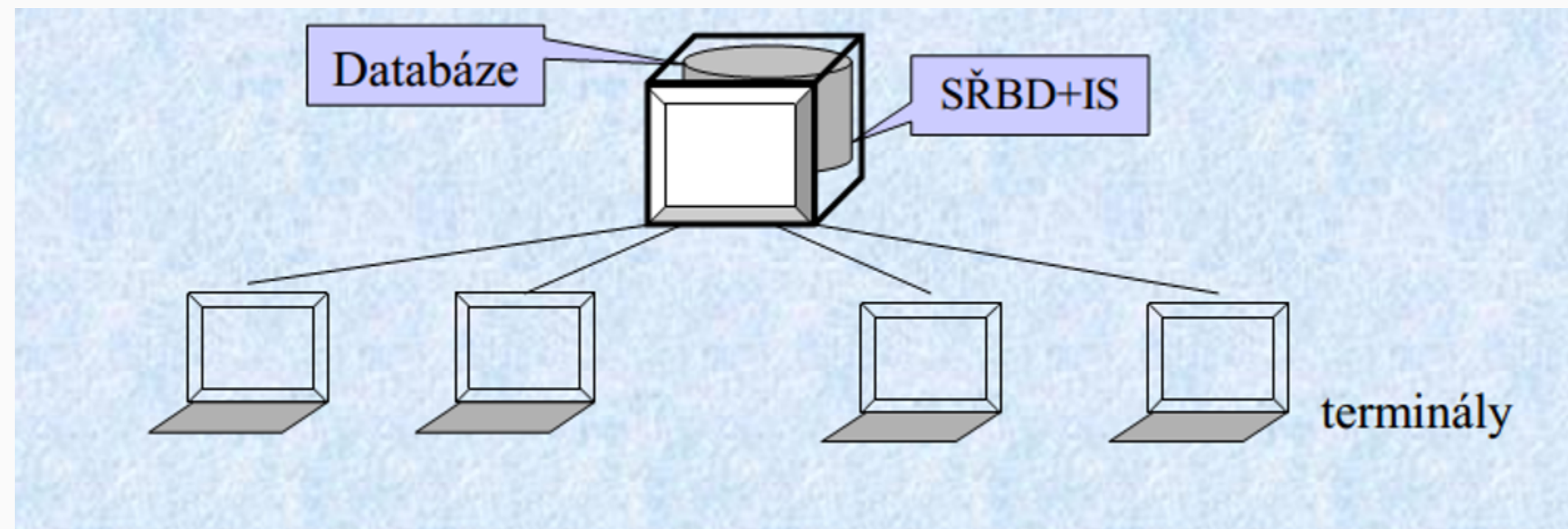
burgetr@fit.vutbr.cz

Architektury – uložení a zpracování dat

- Centrální informační systémy
- Lokální síť
- Klient-server
- Monolitické architektury
 - Třívrstvá architektura
- Distribuované informační systémy
 - Architektury založené na službách

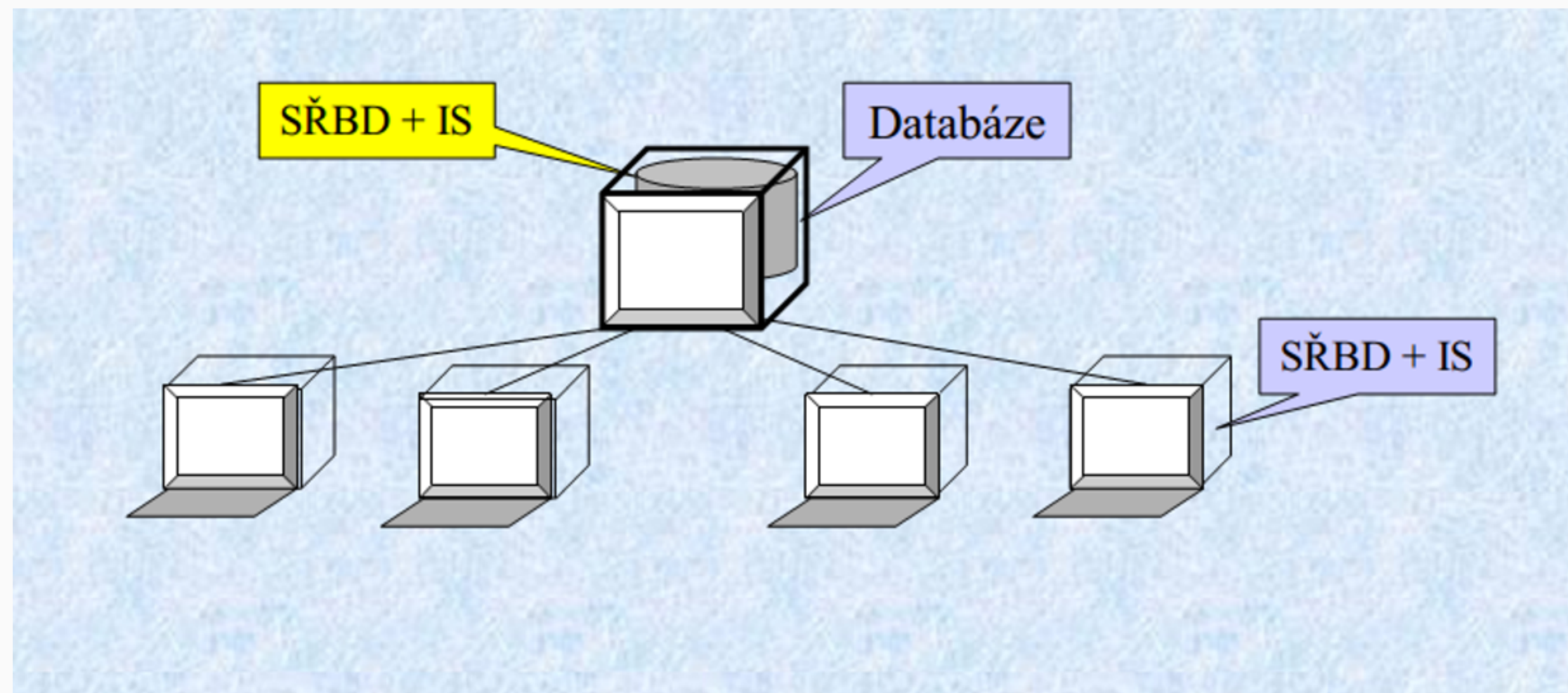
Centrální informační systémy

- Centrálního počítač (*mainframe*) s databází a aplikacemi
- Aktivace aplikačních programů z terminálů (*pracovních stanic*)
- Z hlediska architektury není použita síťová komunikace (není klient)



Lokální síť

- Zavedení lokálního klienta (osobní počítač – PC)
- Aplikace na PC, databáze na speciálním serveru v rámci lokální sítě

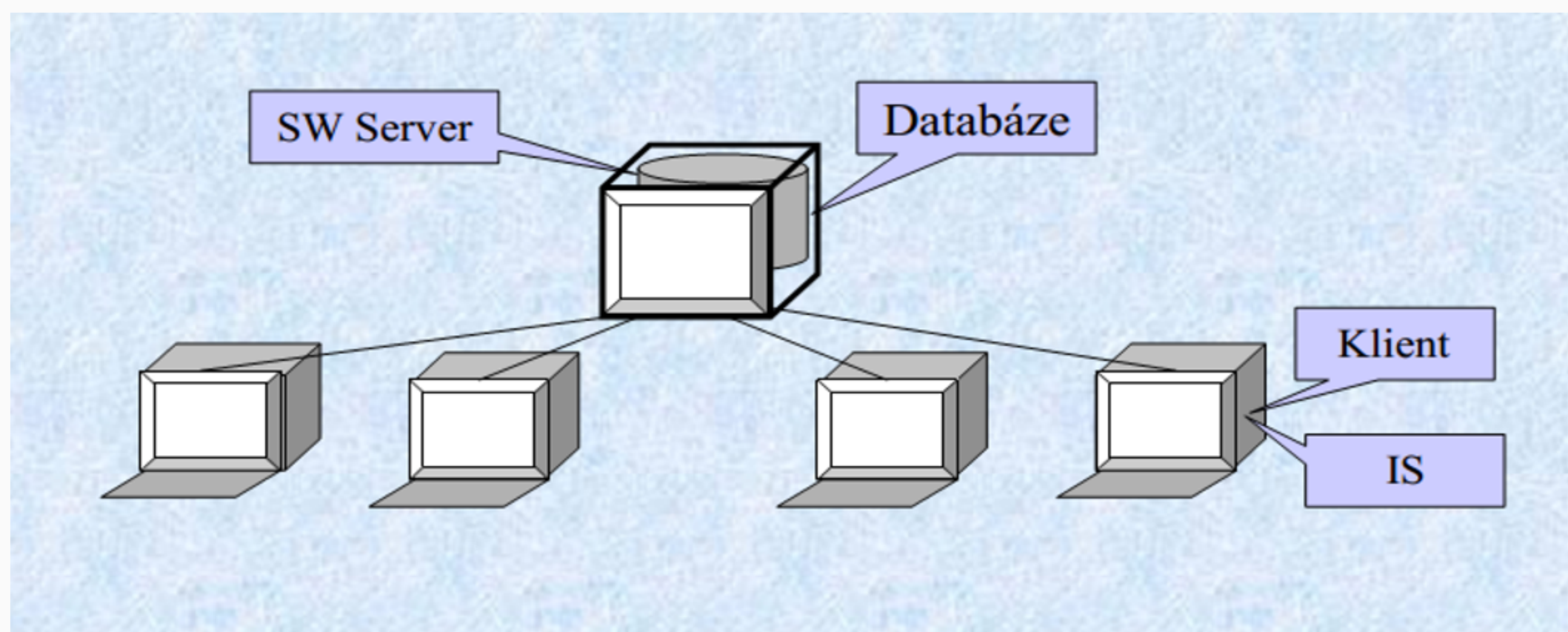


Lokální síť

- Není použita globální síť a standardní protokoly Internetu a TCP-IP
- Snížení rychlosti přenosů, bezpečnosti a zabezpečení integrity
- Vstupuje otázka ***izolovanosti transakcí***, tj. možnosti ***víceuživatelského přístupu***

Architektura klient-server (dvouvrstvá)

- Užity dva druhy oddělených výpočetních systémů **klient** a **server**.
- **Tloušťka** klienta odpovídá jeho "**inteligenci**"



Architektura klient-server

- Na nižší úrovni použita síťová komunikace standardizovaná protokoly Internetu TCP/IP
- Chování klienta a serveru rovněž standardizováno
 - Server specializovaný pro databázové dotazy
 - Po síti se přenášejí pouze dotazy a výsledky
- Ve vyšších vrstvách aplikačních protokolů se nejčastěji komunikuje ***serializovanými daty***, případně v SQL

Třívrstvá architektura

- *(three-tier architecture)*
- **Prezentační vrstva** – **vizualizuje** informace pro uživatele, většinou formou grafického uživatelského rozhraní, může kontrolovat zadávané vstupy, neobsahuje však zpracování dat
- **Aplikační vrstva** – jádro aplikace, logika a funkce, výpočty a zpracování dat
- **Datová vrstva** – nejčastěji databáze. Může zde být ale také (síťový) souborový systém, webová služba nebo jiná aplikace.

Terminologická odbočka

- **Tier** – fyzická vrstva – jednotka nasazení (deployment)
 - Fyzické členění systému – klient, aplikační server, DB server
 - Tomu odpovídá volba technologií pro realizaci jednotlivých částí
- **Layer** – logická vrstva – jednotka organizace kódu
 - Obvykle řešena v rámci aplikační vrstvy
 - *Data layer* – část řešící komunikaci s databází
 - *Business layer* – část implementující logiku aplikace
 - *Presentation layer* – komunikace s klientem

Schéma třívrstvé architektury

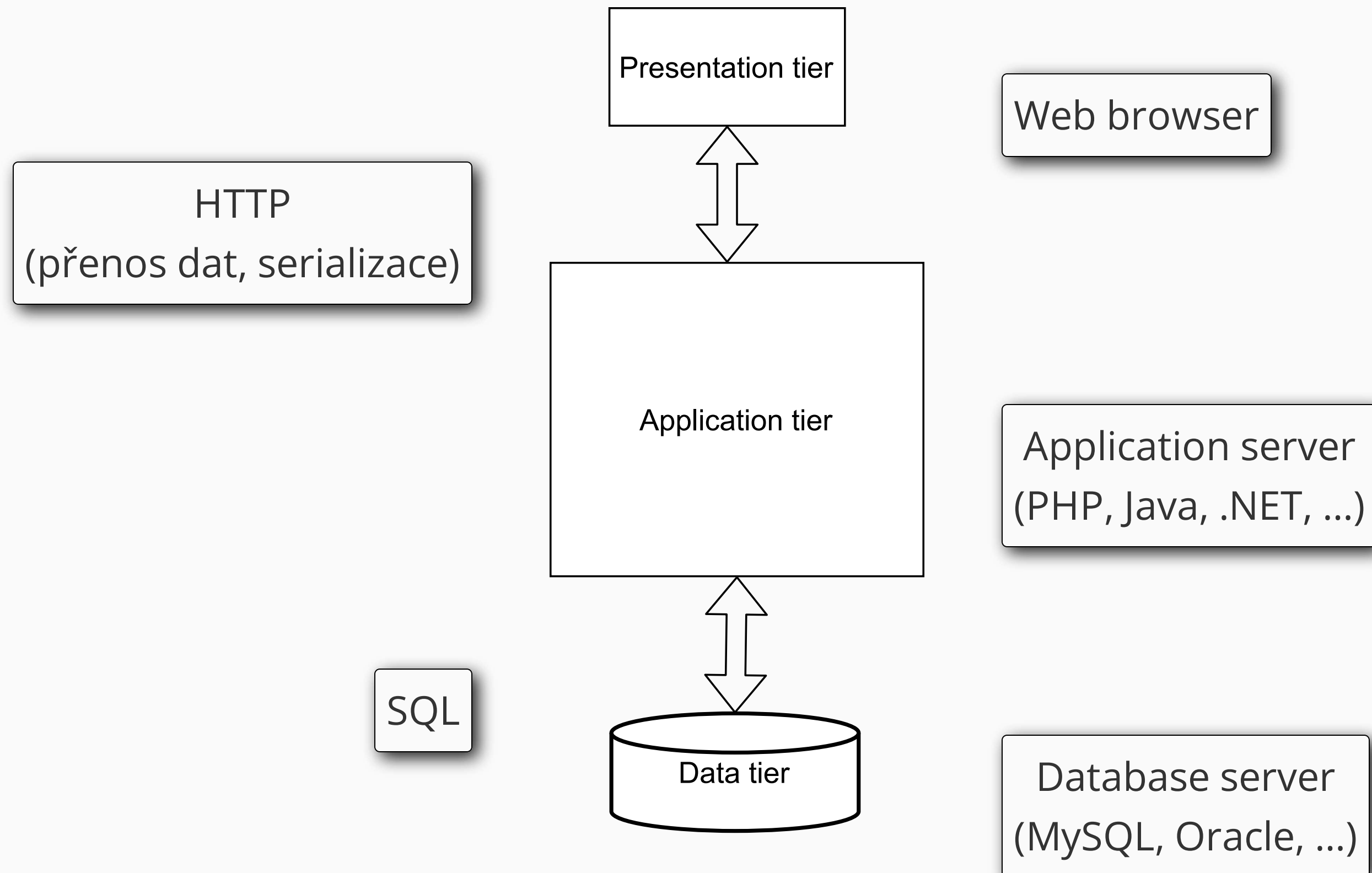
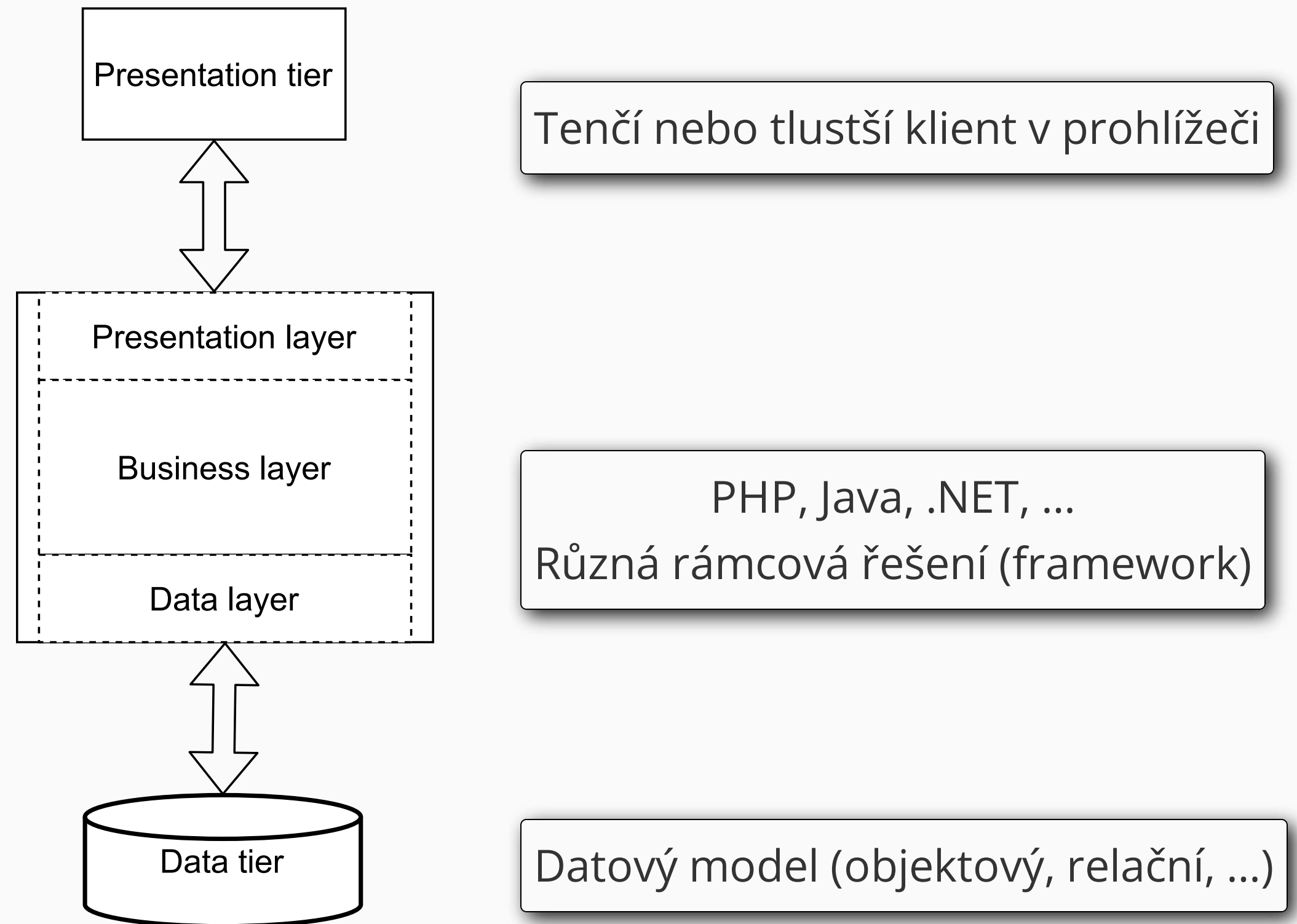


Schéma třívrstvé architektury (II)



Distribuované architektury

- Monolitický systém (typické pro třívrstvou architekturu)
 - Vyvíjí se a nasazuje jako jeden celek
 - + snáze zvládnutelný vývoj, testování
 - - obtížnější a pomalejší nasazování nových verzí
- Distribuované architektury
 - Service-oriented architecture (SOA)
 - Microservices (mikroslužby)
 - Spíše řešeno v rámci Pokročilých informačních systémů

Mikroslužby

- Aplikace je rozdělena na malé části
 - Vlastní databáze (nepřístupná vně)
 - Business logika
 - Aplikační rozhraní (sítové)
- Typicky malý tým vývojářů na každou část (2 pizzas rule)
- Nasazují se odděleně
- + Technologická nezávislost, rychlé aktualizace
- - Testovatelnost, režie komunikace, riziko nekompatibility, řetězové selhání, ...

Návrh a implementace informačních systémů

Shrnutí

Model

- Modelování je prováděno jistým typem systému pro řízení báze dat, tedy, např.:
 - ***relačním modelem*** (_nejčastěji),
 - objektovým modelem,
 - případně jinak.

Databáze

- Databáze pro modelování stavu není podmínkou, specializované IS např. pro řízení výroby v reálném čase používají i jiné typy uchování dat, nicméně databáze jako sídlo stavu modelu je nejčastější.
- Realizace databázového modelu je technologickou a provozní otázkou a může být např.:
 - **monolitický** (lokalizovaný na jediném místě, s jednou databází),
 - **distribuovaný** (s více lokálními databázemi, zde pak vznikají problémy s konzistencí)

Procesy

- Modelovacím prostředím **procesů modelu** je nejčastěji nějaký univerzální programovací jazyk kompilovaný nebo i interpretovaný.
- Snahy o modelování formálnějšími prostředky jako jsou např. různé modifikace automatů nebo **Petriho sítě**. Při modelování procesů se musíme zabývat zejména:
 - udržováním **konzistence** systému,
 - **paralelním během procesů (vícenásobným přístupem)** a vzájemným, ovlivňováním a
 - **transakčním zpracováním**.

Nezbytné znalosti technologie

- Chceme-li se tedy zabývat informačními systémy musíme se zabývat:
- Způsobem vytváření modelů, **modelovacími technikami** a to zejména:
 - **konceptuálním modelováním** jako výchozím prostředkem pro modelování dat (tj. definicí modelu stavu fyzického systému na jisté úrovni abstrakce), převodem konceptuálního modelu na model databázový,
 - **modelováním procesů** a tedy i
 - **univerzálními modelovacími prostředky**, jako je např. UML.

Nezbytné znalosti technologie

- **Databázovými systémy** a jejich použitím a to zejména:
 - různými **typy databázových modelů** a jejich univerzálním rozhraním a ovládáním
 - **transakčním zpracováním** a pojmem transakce,
 - **konzistencí dat**
- **Modelováním procesů** a jejich případnou formalizací a to zejména:
 - **programovacími jazyky** vhodnými pro definici procesů,
 - **formálními metodami definice procesů** a workflow systémy,
 - souvislosti procesů s transakcemi a integritou,
 - metodami **spouštění procesů**.

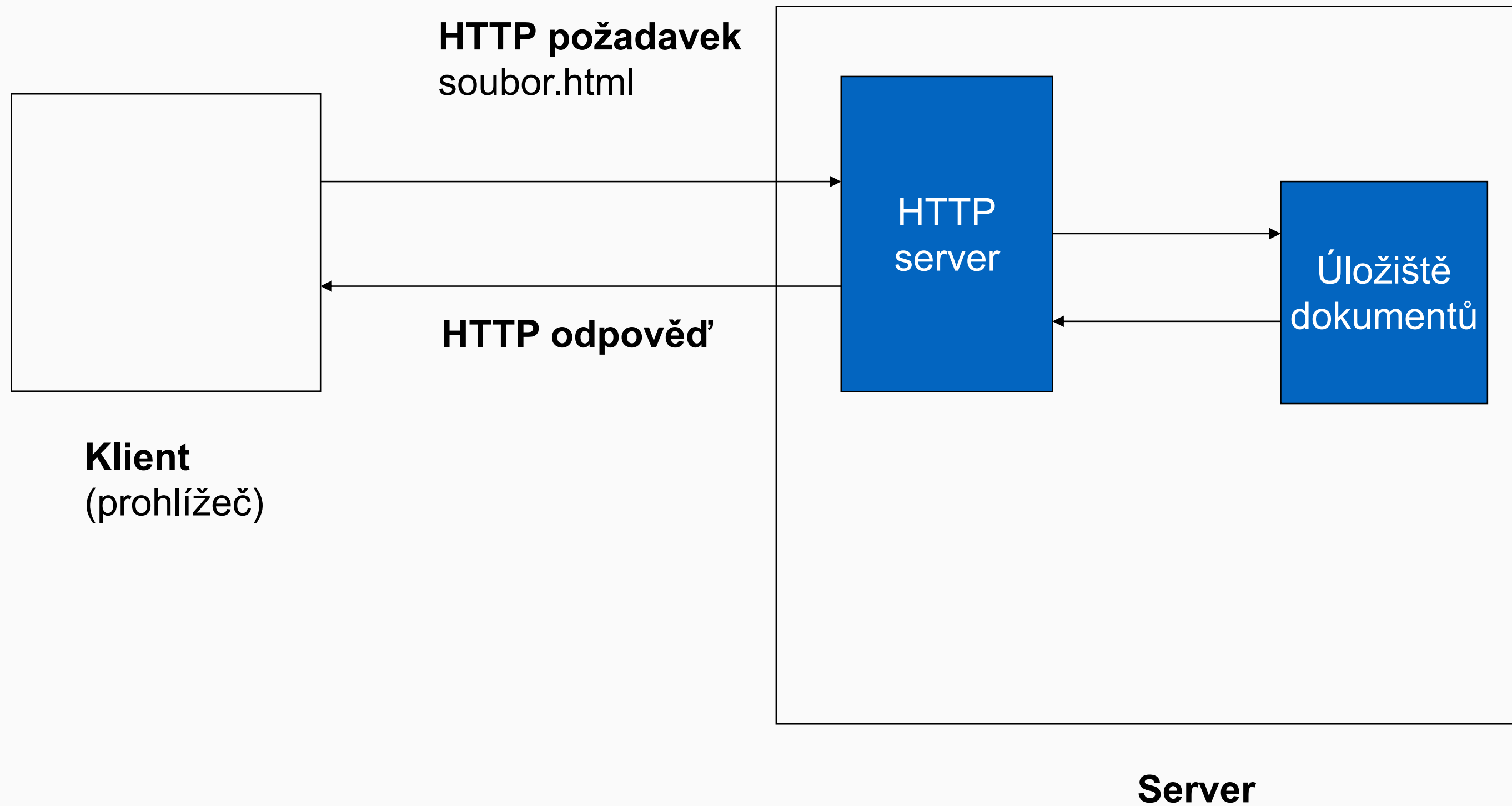
Nezbytné znalosti technologie

- **Počítačovými sítěmi** a to zejména:
 - technologií **klient-server** a vytváření klientské a serverové části informačního systému
 - **internetovými službami**
- **Vizualizací dat** a to zejména:
 - hypertextovou prezentací popsitelnou v *HTML* a pokročilejších technikách využívajících skripty a model DOM.
 - prezentací sloužící pro dolování dat nebo **OLAP technologie**.
- **Mnohé z těchto témat pokrývají jeden nebo více samostatných povinných nebo volitelných předmětů. Zde se zabýváme jejich základy, zopakováním, doplněním a zejména spoluprací při vytváření komplexního systému.**

HTTP a dynamické stránky

Základní stavební kameny informačního systému

Základní scénář komunikace



Protokol HTTP

- Aplikační protokol nad TCP/IP (Hypertext Transfer Protocol)
- Komunikaci začíná klient
 - Naváže spojení se serverem
 - Vyšle HTTP požadavek
- Server reaguje HTTP odpovědí
 - Stav – výsledek vyhodnocení požadavku
 - Požadovaný (nebo chybový) dokument
- Rozlišení typy dokumentů: MIME typ
- Detaily: např. v [přednáškách ITW](#)

HTTP požadavek (request)

- Metoda, URL, hlavičky, tělo (payload)

```
GET /data.html HTTP/1.1
Host: www.fit.vutbr.cz
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:69.0) Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: cs,en-US;q=0.7,en;q=0.3
...
```

```
POST /api/login.php HTTP/1.1
Host: www.fit.vutbr.cz
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:69.0) Firefox/69.0
Accept: application/json, text/plain, */*
Content-Type: application/json
Content-Length: 39
```

HTTP metody

- GET – získání dokumentu
- POST – zaslání dat
- PUT – nahrazení dokumentu
- DELETE – smazání dokumentu
- HEAD, CONNECT, OPTIONS, TRACE, PATCH, ...

HTTP odpověď (response)

- Stav, hlavičky, tělo (payload)

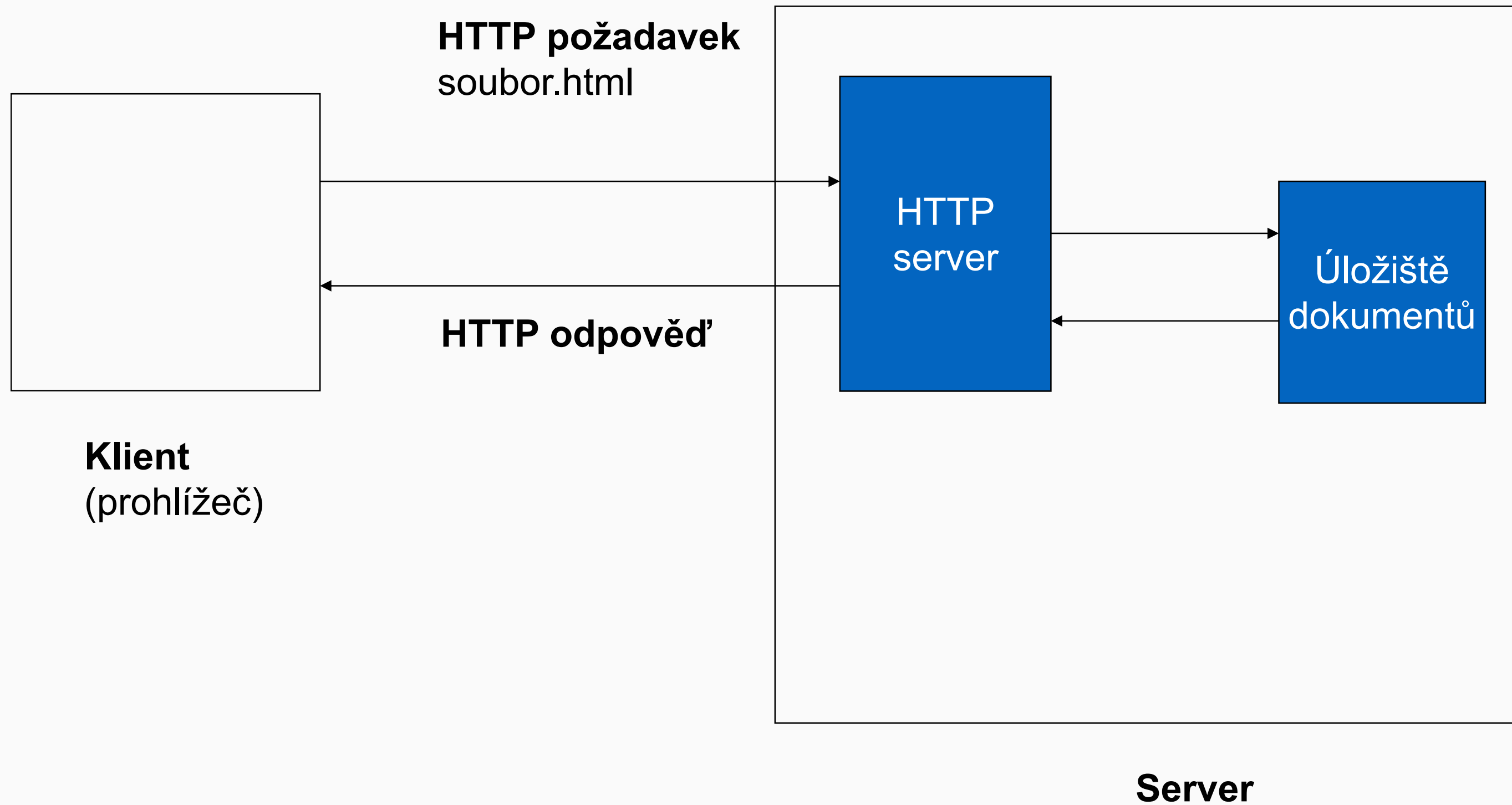
```
HTTP/1.1 200 OK
Date: Wed, 02 Oct 2019 12:11:55 GMT
Server: Apache/2.4.38 (Debian)
Content-type: text/html; charset=utf-8
...

<!DOCTYPE html>
<html>
...
```

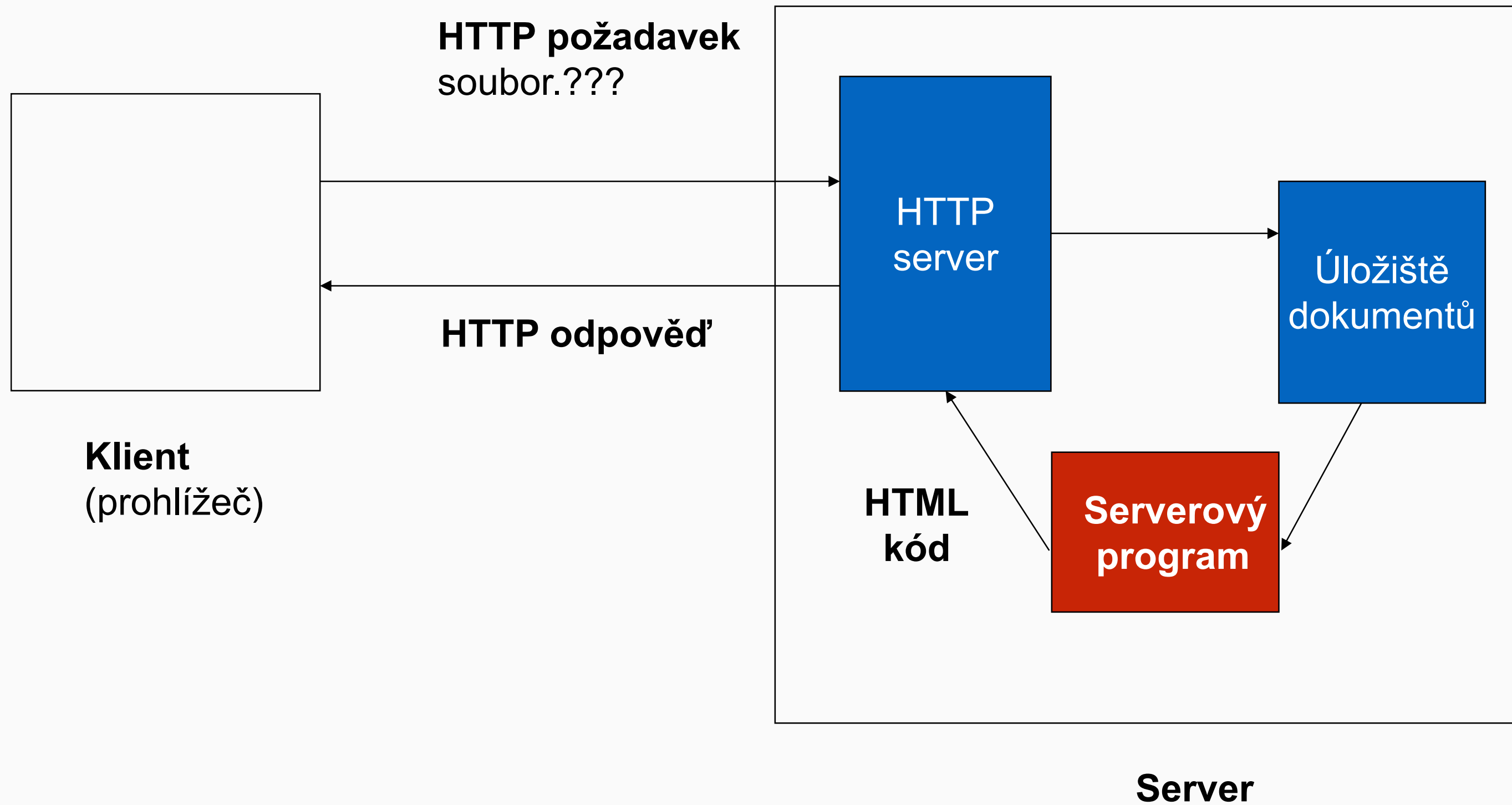
MIME typ obsahu

- Hlavička `Content-type`
- Specifikace typu ve tvaru `třída/typ`
- Standardní typy
 - `text/plain`, `text/html`, `text/xml`
 - `application/json`, `application/x-www-form-urlencoded`
 - `image/jpeg`, `image/png`

Dynamické webové stránky



Dynamické webové stránky



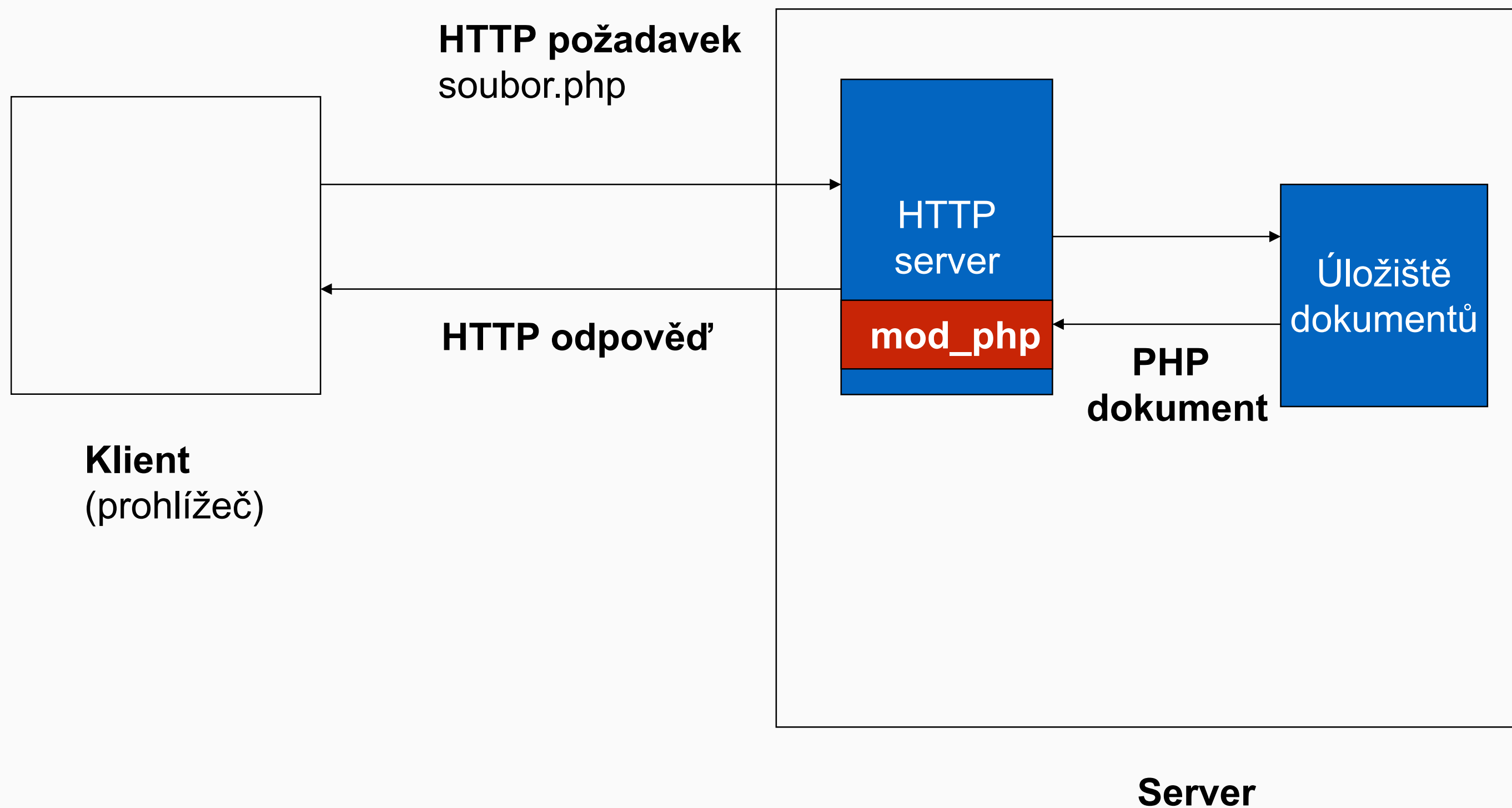
Přímocaré řešení: CGI

- Common Gateway Interface
- Externí program spouštěný HTTP serverem
- Nezávislé na implementačním jazyce
 - Výměna dat přes stdin/stdout a proměnné prostředí
 - Programování v C, PERL, Python, ...
- Velká režie
 - Nový proces pro každý HTTP požadavek
- [Příklad CGI skriptu v PERLu](#)

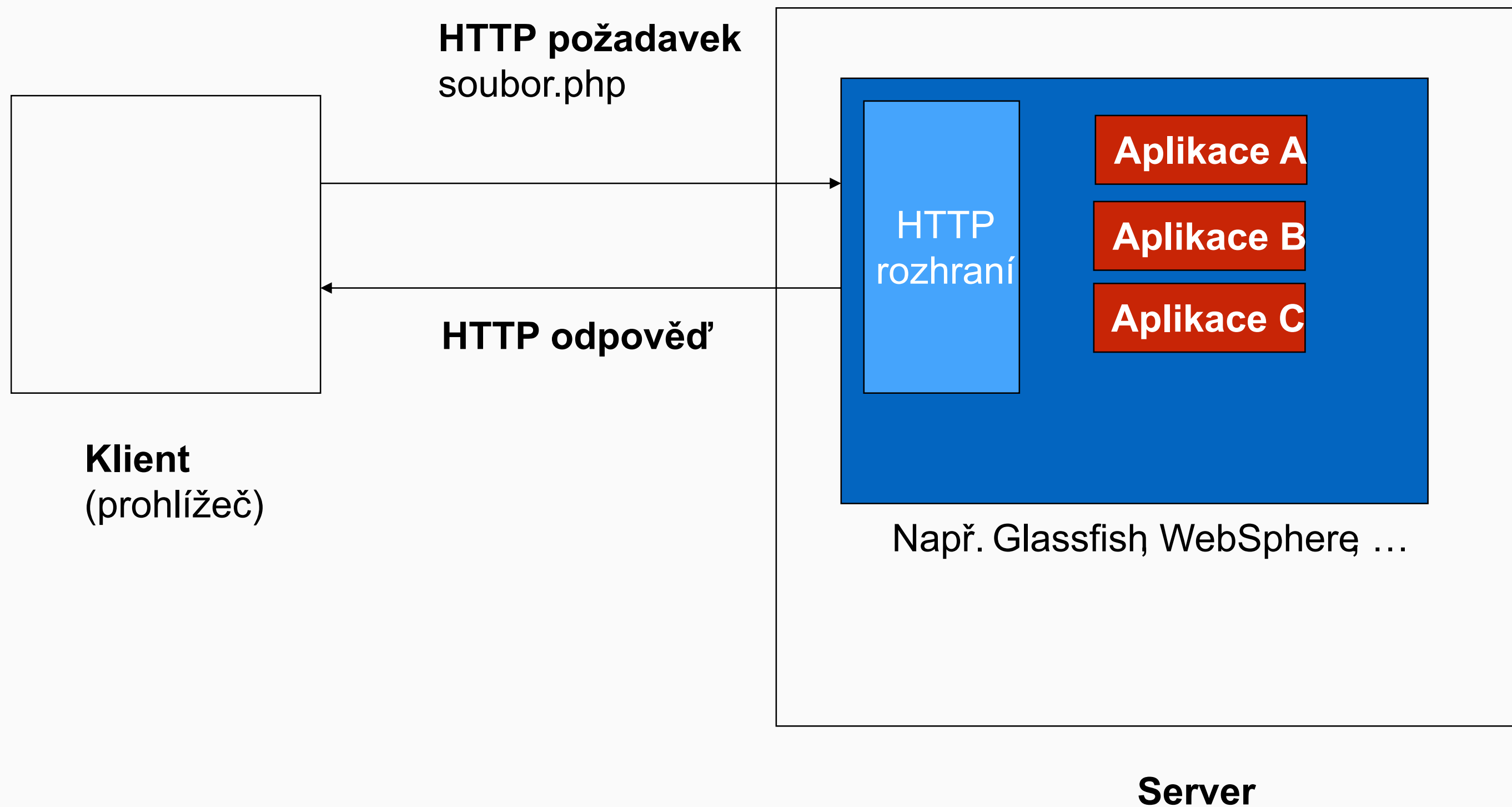
Efektivnější řešení

- FastCGI
 - Trvale běžící proces zpracovává více HTTP požadavků
- Rozšíření běžícího HTTP serveru
 - Rozšiřující moduly
 - Např. PHP (mod_php)
- Specializovaný HTTP server pro specifickou platformu
 - Např. Java EE server (Tomcat, Glassfish, ...)
 - Podobně JavaScript (nodejs), .NET, ...

Moduly HTTP serveru (PHP)



Aplikační server (např. Java)



Správa sezení – kontext

- Protokol HTTP je **bezstavový**.
 - Požadavky jsou vyhodnocovány nezávisle na sobě.
- Potřebujeme rozlišit požadavky pocházející od stejných/různých klientů – **kontext**.
- Je třeba přidat k HTTP mechanismus pro uchování informace o kontextu klientů. Tento mechanismus se nazývá **správa sezení** (*session*).

Správa sezení – princip

- Nově příchozím uživatelům vygenerujeme jednoznačný identifikátor **Session ID**.
 - Při prvním HTTP požadavku od nového klienta (zařízení)
- Tímto identifikátorem se klient prokáže při každém dalším požadavku.
 - Jsme schopni rozlišit požadavky od jednotlivých klientů
- Jak to technicky zabezpečit?

Jak udržet kontext

1. Předávání hodnoty Session ID jako parametr jednotlivých dotazů.
 - Vyžaduje příslušné úpravy na všech místech aplikace, která mohou generovat HTTP dotaz
 - Bezpečnostní problémy (session ID je např. v URL, v HTML kódu, ...)
2. Použití cookies
 - Zabudovaný mechanismus HTTP

Cookies

- Cookie: Malý objem dat, který serverová aplikace může uložit na straně klienta (v prohlížeči)
- Každý cookie má ***jméno*** a ***hodnotu***
- Pro každý cookie je navíc definována ***cesta*** a ***expirace***
 - Přístupovat ke cookies mohou pouze stránky se stejnou cestou jako je stránka, která cookie uložila
 - Lze nastavit jiný adresář (nejčastěji kořenový, aby celá aplikace mohla číst všechna svoje cookie)

Trvanlivost cookies (expirace)

- Lze zadat přesný čas, dokdy má být cookie uložen v prohlížeči – tzv. ***expirace***
- Pokud není expirace zadána, cookie se vymaže se zavřením prohlížeče

Řešení cookies v HTTP

- Server v rámci **odpovědi** na nějaký požadavek použije hlavičky `Set-Cookie`

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: theme=light; Path=/; Domain=.example.com
Set-Cookie: sessionId=abc123; Expires=Wed, 09 Jun 2021 10:18:14 GMT
...
```

- Klient uloží nastavené cookie, při každém dalším **požadavku** odešle všechna relevantní cookie pomocí hlavičky `Cookie`

```
GET /spec.html HTTP/1.1
Host: www.example.org
```

Přístup k hodnotám cookies

- Na straně serveru
 - Server shromáždí hodnoty z HTTP hlaviček a zpřístupní aplikaci
 - Např. v proměnných prostředí, speciální proměnné, apod.
- Na straně klienta
 - JavaScriptové API v prohlížeči

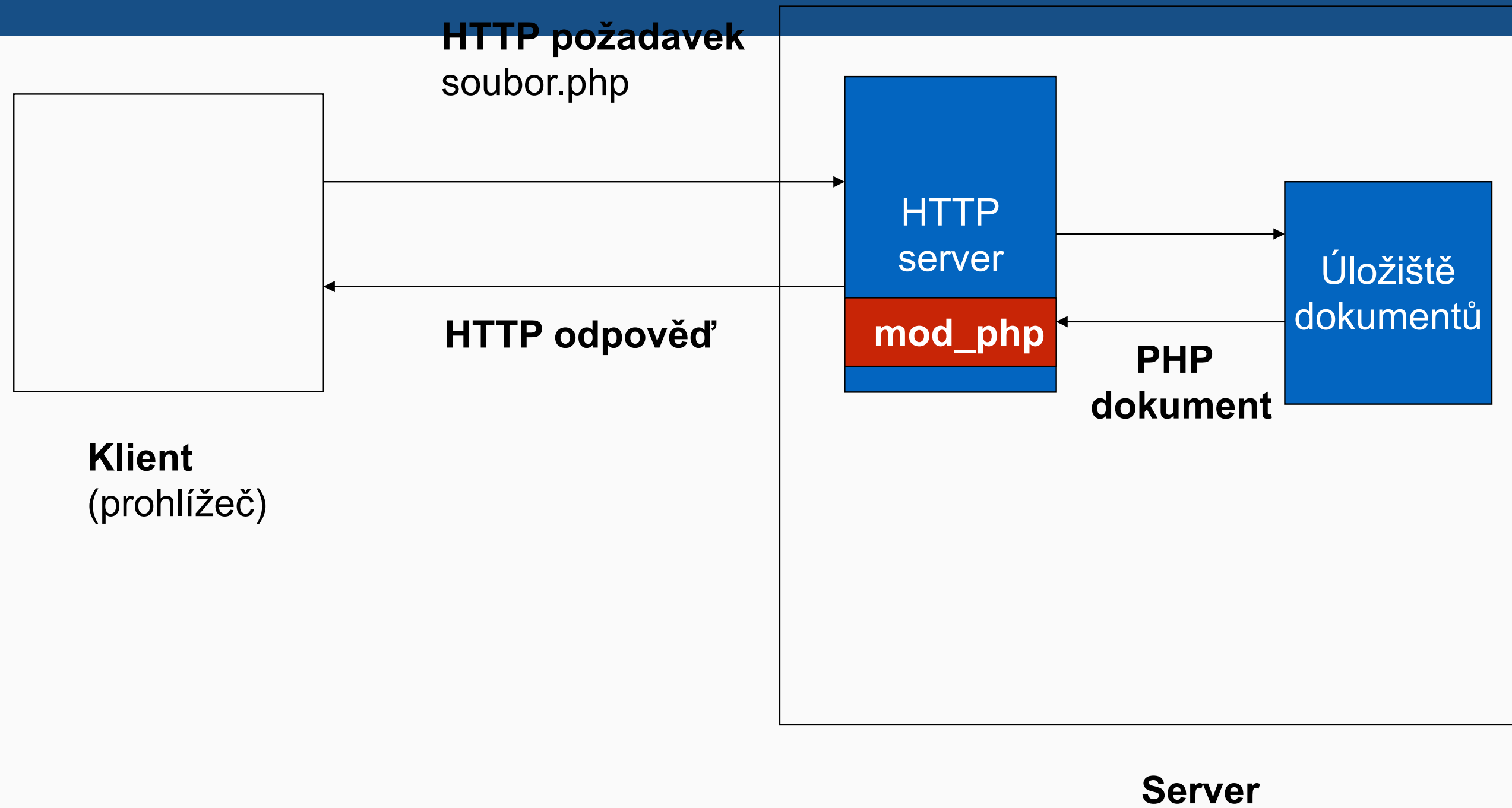
Cookies a Session ID

- Mechanismus řeší pouze identifikaci klienta
 - Rozlišení požadavků jednotlivých klientů
 - Zatím žádná autentizace (přihlášení uživatelů)
- Pozor na bezpečnost
 - Znalost Session ID umožňuje vydávat se za nějakého uživatele

Cookies a Session ID – bezpečnost

- Způsob generování Session ID – předvídatelnost
- Zcizení Session ID (session stealing)
 - Síťový odposlech – šifrování (HTTPS)
 - Útok na klientský prohlížeč (XSS)
 - http-only cookies
 - Útok na klientské zařízení
 - Datové soubory prohlížeče

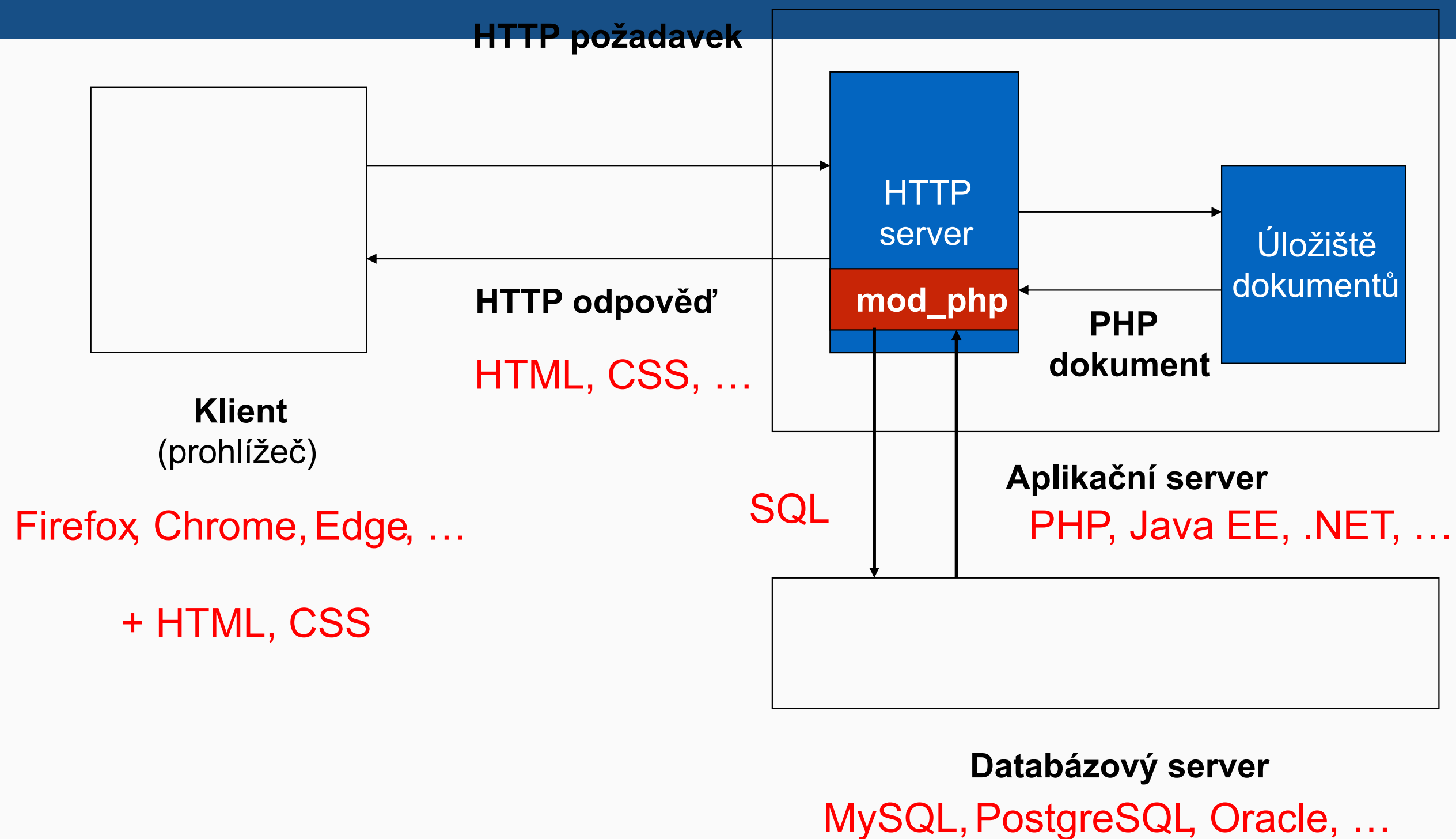
Architektura znovu



Databázová vrstva

Třívrstvá architektura

Základní technologie



Rozšiřující technologie

Co dále?

- Serverová část systému
 - Jazyk PHP
 - Řízení session a HTTP komunikace v PHP
 - Rámcová řešení v PHP (frameworks)
- Databázová vrstva
 - Datové modelování, relační datový model
 - Přístup k relační databázi v PHP (PDO)
- Klientská část
 - Relevantní základy HTML (vstup/výstup) + CSS
 - Klientské skripty (JavaScript)

A to je vše!

Dotazy?

