REST

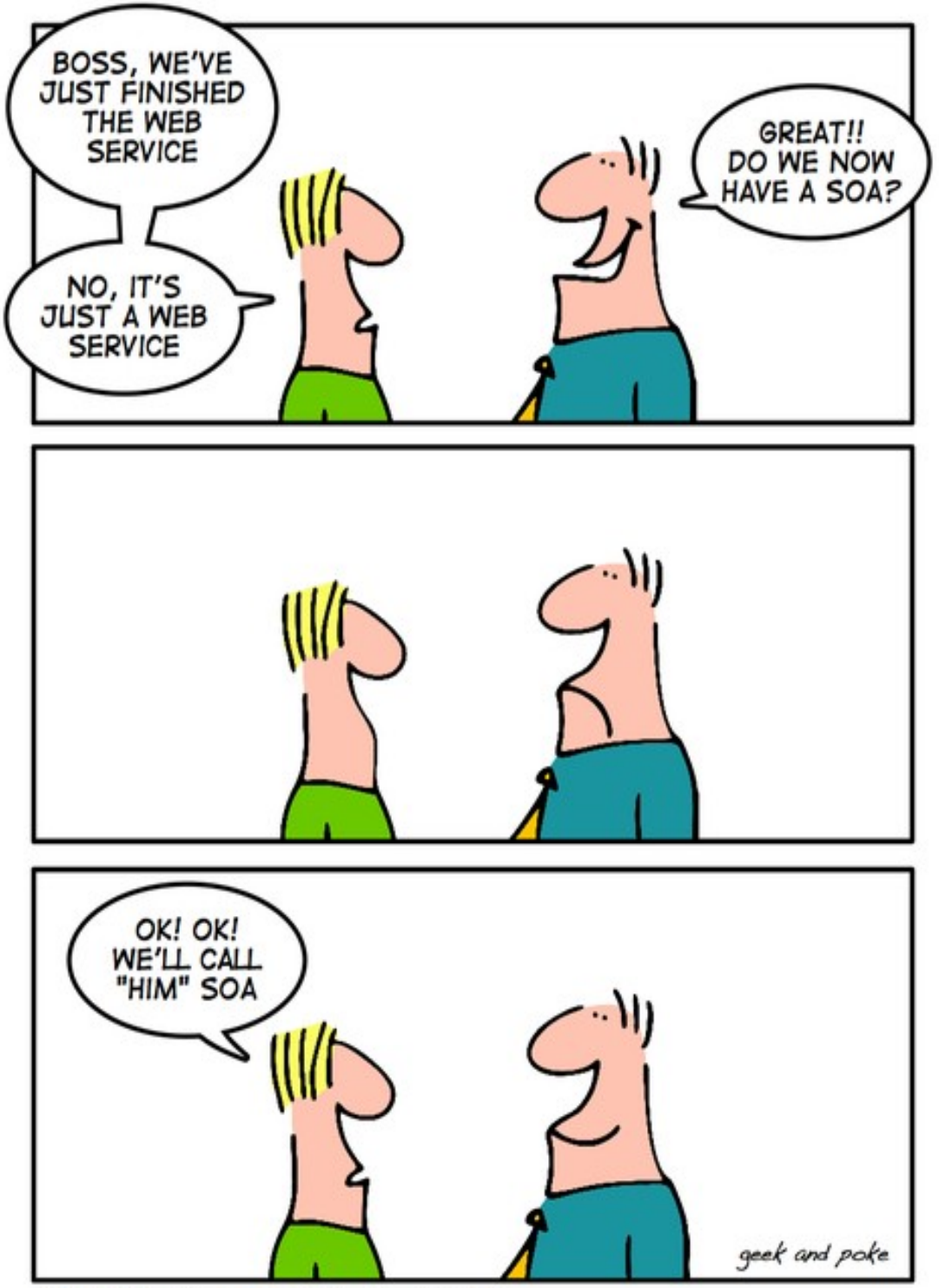
Peter Rybár

CENTAUR

Obsah

- **SOA**
- **REST**
  - REST princípy
  - REST výhody
- **pREST**

- Otázky

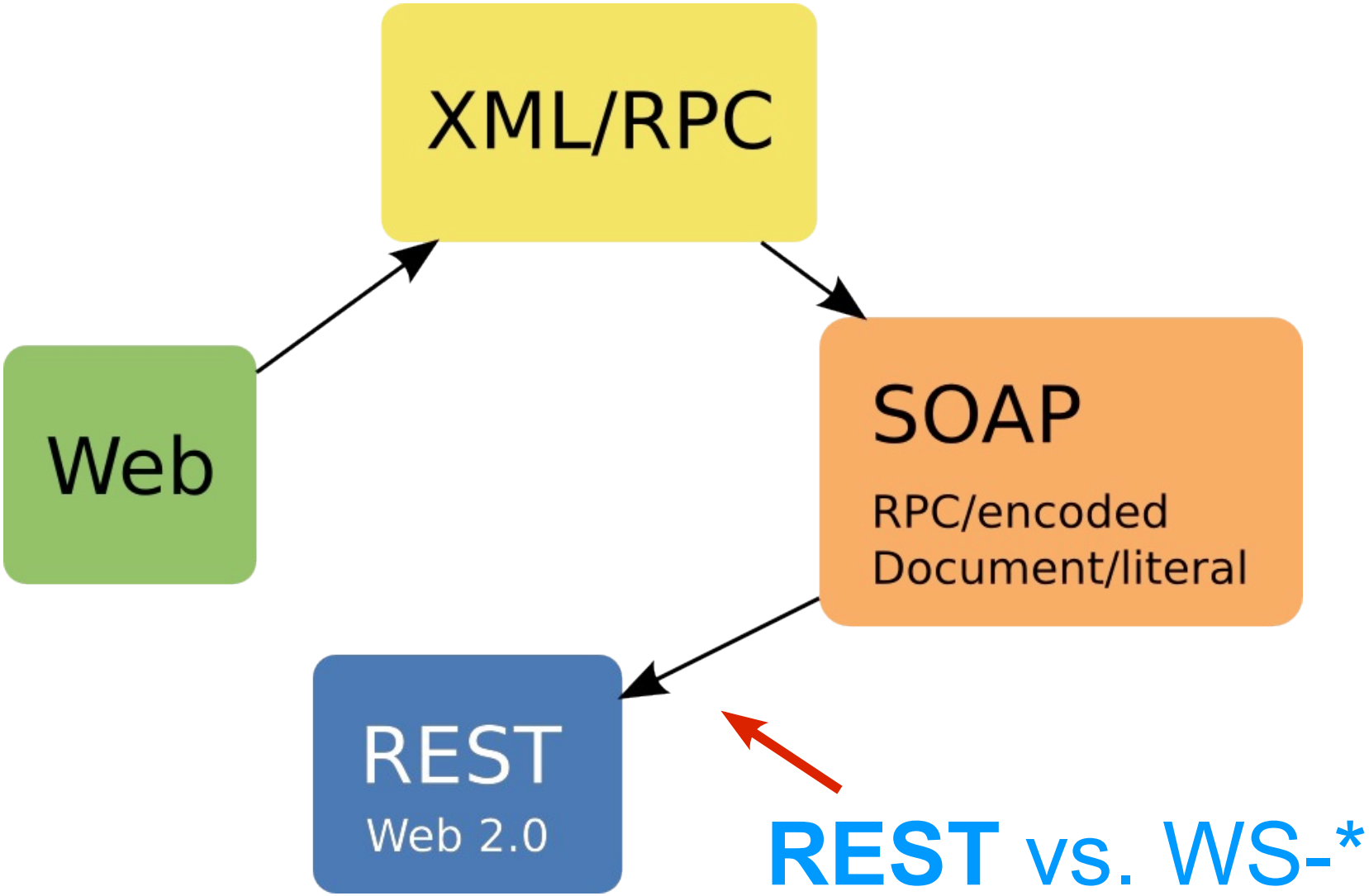


HOW TO GET A SOA

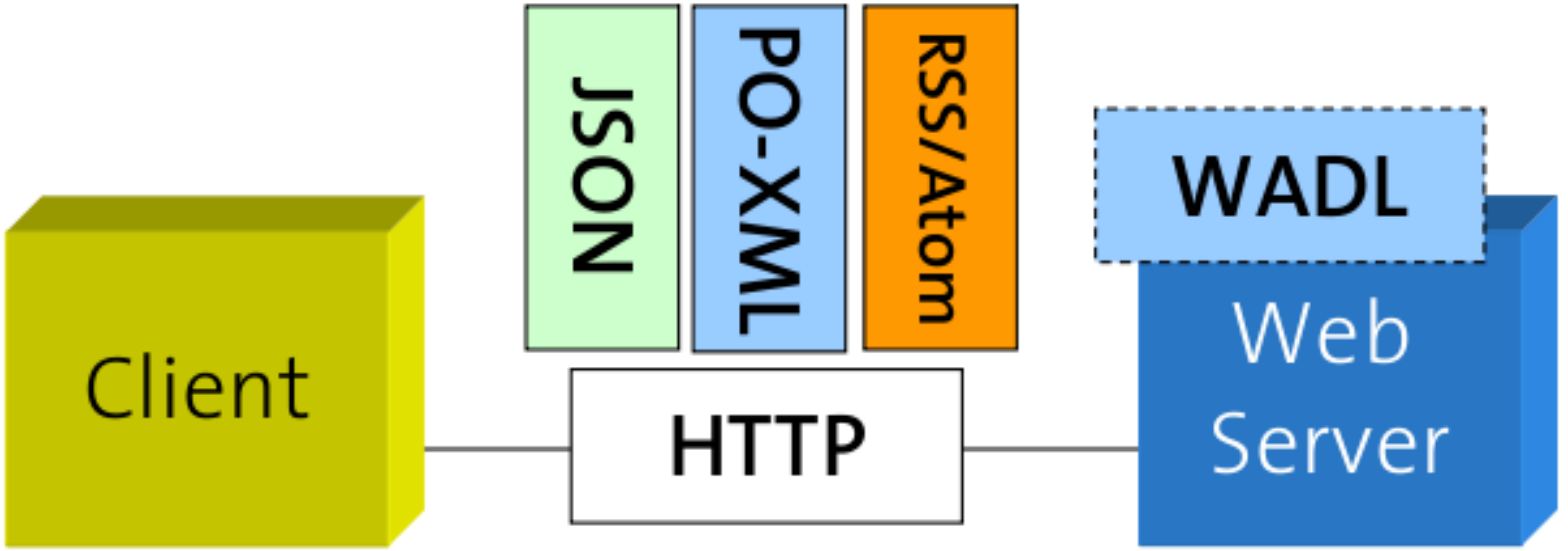
SOA – implementácie

- **WEB** (1990)
- CORBA (1991)
- XML-RPC (1998)
- **WS-*** (1998)
  - SOAP – RPC/literal
  - SOAP – Document/literal (2001)
- **REST** (2000)

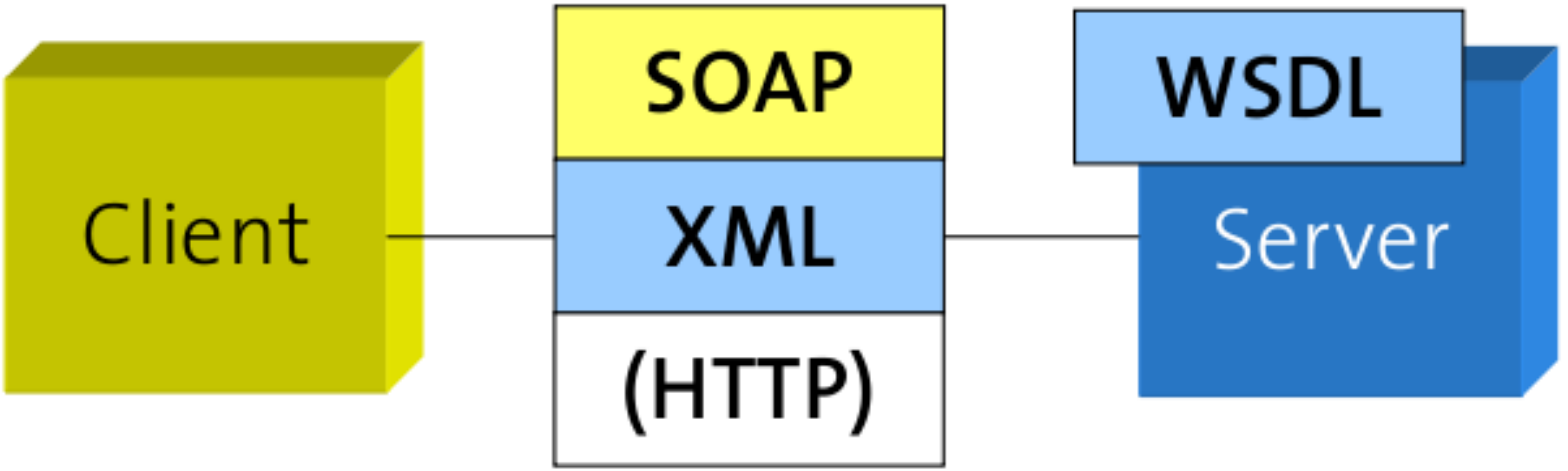
SOA – Web implementácie



RESTful Web Services (2007)



WS-* Web Services (2000)



REST

CENTAUR

Čo je **REST** ?

- **REST**
  - **Re**presentational **S**tate **T**ransfer
  - **Roy Fielding PhD, 2000**
  - **Architektonický štýl**

- REST stojí na princípoch, ktoré umožňujú HTTP byť tak dobre škálovateľný
- REST „je Web“ – nie je tunelovaný cez Web

REST – Princípy

- **Princípy**:

1. **URI** – identifikácia zdroj (všetko je zdroj)
2. **CRUD** – jednotné rozhranie pre všetky zdroje
3. **Reprezentácie** – rôzne podoby správy (MIME)
4. **Bezstavovsť** – umožňuje škálovateľnosť
5. **Hypermédiá** – prelinkovanie médií/reprezentácií

REST – Princípy

- **URI:**
  - Všetko sú **zdroje** ↔ **ROA**
  - Zdroje sú identifikované **URI**
  - Zdroje sú **podstatné mená**

http://example.net/customer

http://example.net/car

http://example.net/shopping-cart

REST – Princípy

- **CRUD**:
  - **jednotné rozhranie** pre prácu so zdrojmi
    - POST – **C**reate, vytvára nový zdroj
    - GET – **R**ead, bezpečná operácia
    - PUT – **U**pdate, idempotentná operácia
    - DELETE – **D**elete, idempotentná operácia

REST – Princípy

- **Reprezentácie:**
  - **Jeden zdroj – viacero reprezentácií**

REST – Princípy

- **Reprezentácie:**
  - **Jeden zdroj – viacero reprezentácií**
    - text/html, application/pdf, image/png
  - Typ reprezentácie je v HTTP hlavičke
    - Request – **Accept**
    - Response – **Content-Type**

REST – Princípy

- **Bezstavovsť**:
  - **HTTP server nepozná stav**
    - **Neexistuje HTTP Session!**
  - Klient udržuje stav cez **linky**
    - Funguje **back button**
    - Funguje **bookmarkovanie**
  - Škálovateľný systém!

REST – Princípy

- **Hypermédiá:**
  - Reprezentácie zdrojov – **hypermédiá**
  - **Hypermédiá – obsahujú linky na iné médiá**
  - Zmena stavu klienta – cez linky v hypermédiách
  - Linky poskytuje server

Čo **nie** je REST?

Čo **nie** je REST?

- POX (Plain Old XML) bez SOAP obálky
- Tunelovanie cez HTTP GET
  - http://example.net/api?method=find&id=37
  - http://example.net/api/find/37
- Tunelovanie cez HTTP POST
  - POST http://example.net/api/

```
<method name="find">
    <id>37</id>
</method>
```

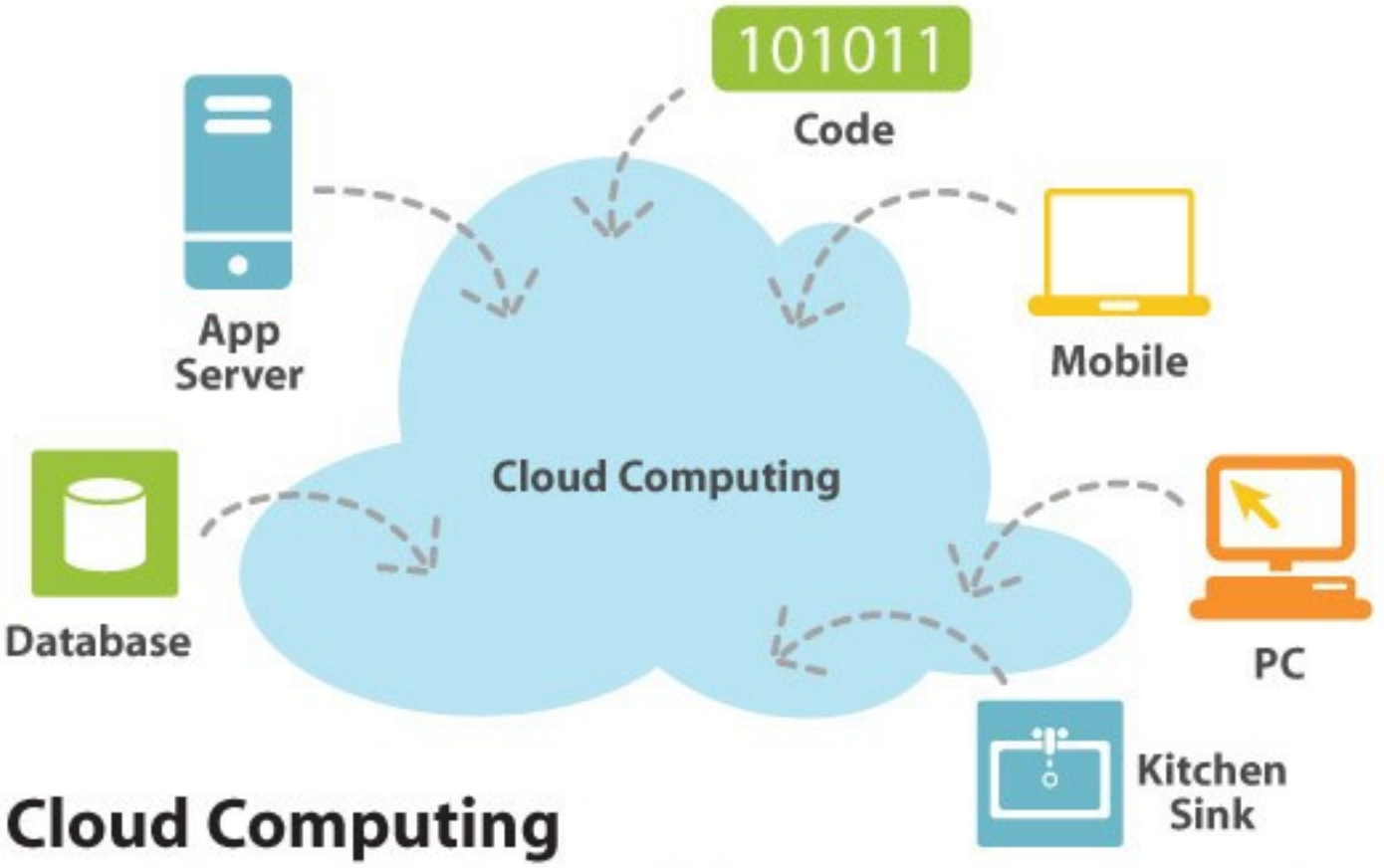
- Ignorovanie HTTP Cache
- Ignorovanie HTTP Status Codes

REST – Výhody

- **Jednoduchý vývoj** => **Nízka cena**
  - Jednotné nemenné rozhranie **CRUD**
  - **HTTP** je **všadeprítomný** – povolený na FW
- **Bezstavová interakcia** => **Škálovateľnosť**
- **Tenká infraštruktúra** => **Ľahká adopcia**
  - Stačí Webový prehliadač
  - **Nie je potreba kupovať drahý WS-* middleware**

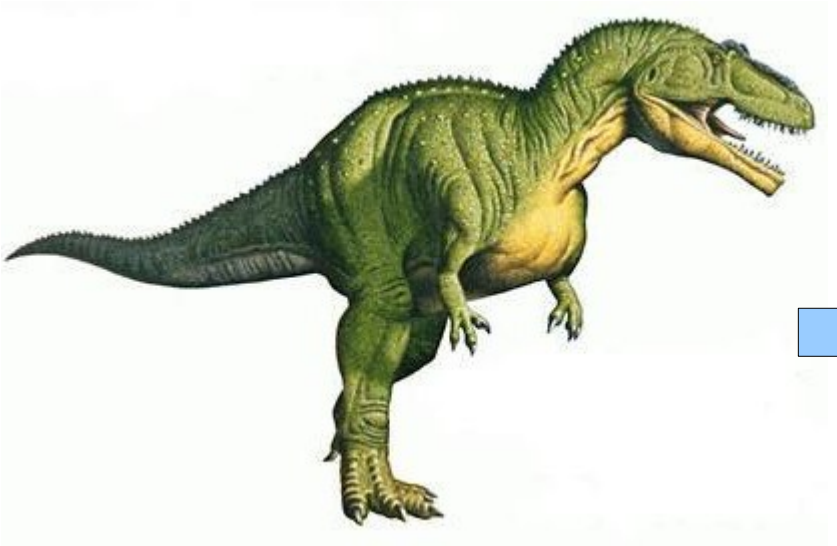
REST – Výhody

- **Fundamentálny prístup => SEO**
- Silná podpora **Web 2.0 –** Google, Amazon (S3) ...



REST – Nástroje

- **Zmena architektúry** => **Zmena nástrojov**
  - Dôraz na jednoduchosť, odľahčenie
  - Jednoduchý stack technológií
  - Jedny nástroje pre vývoj:
    - Web služieb
    - Web aplikácií



pREST

REST Web Framework

Peter Rybár
Centaur a.s.

CENTAUR

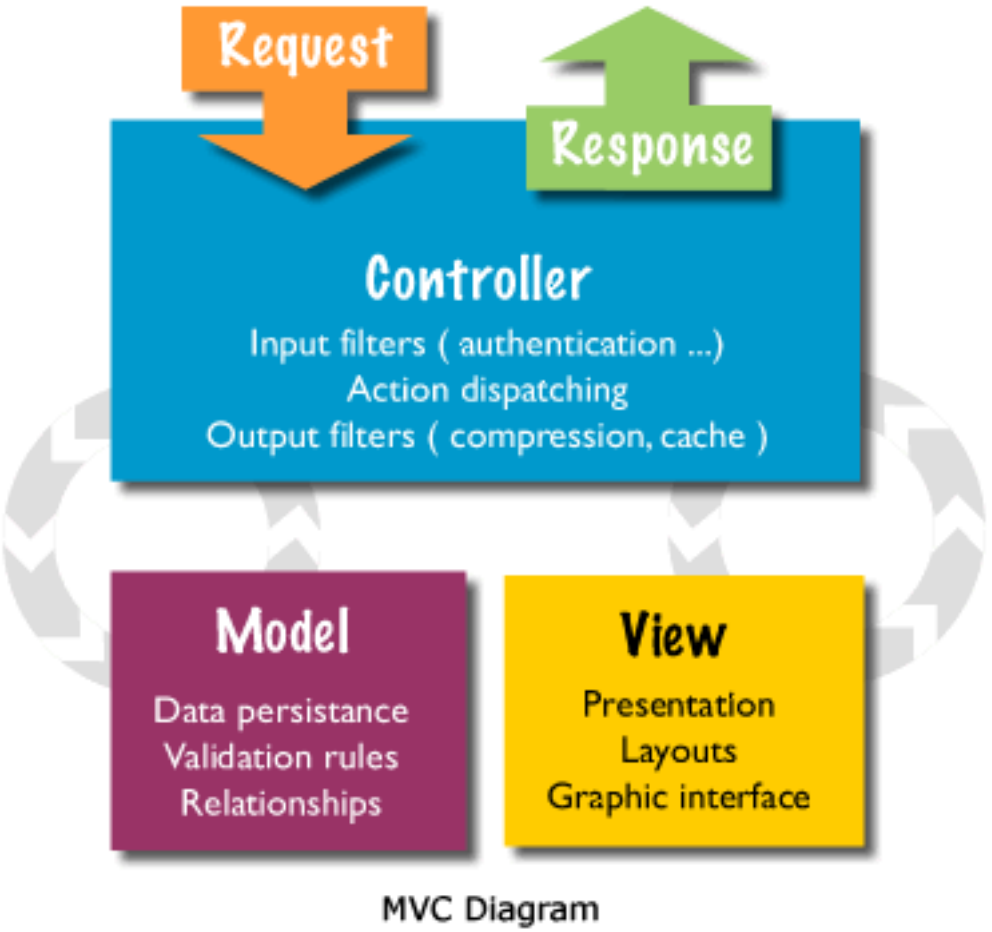
pREST

- Web framework navrhnutý na vývoj
  - Webových aplikácií (AJAX, RIA)
  - Webových služieb
- Pre vývoj architektonickým štýlom **REST**
- Kladie dôraz na:
  - Jednoduchosť vývoja
  - Efektivitu vývoja
  - Modularitu a Extenzibilitu
  - Vysoký výkon

pREST – dva komponenty

- Framework pozostáva z dvoch komponentov
  - **Serverový komponent**
    - Kontróler pre Java servlet kontajner so sadou rozšírení
  - **Klientský komponent**
    - Javascript knižnica, UI Toolkit – je možné ju použiť v kombinácii s ľubovolnou technológiou na strane servera.

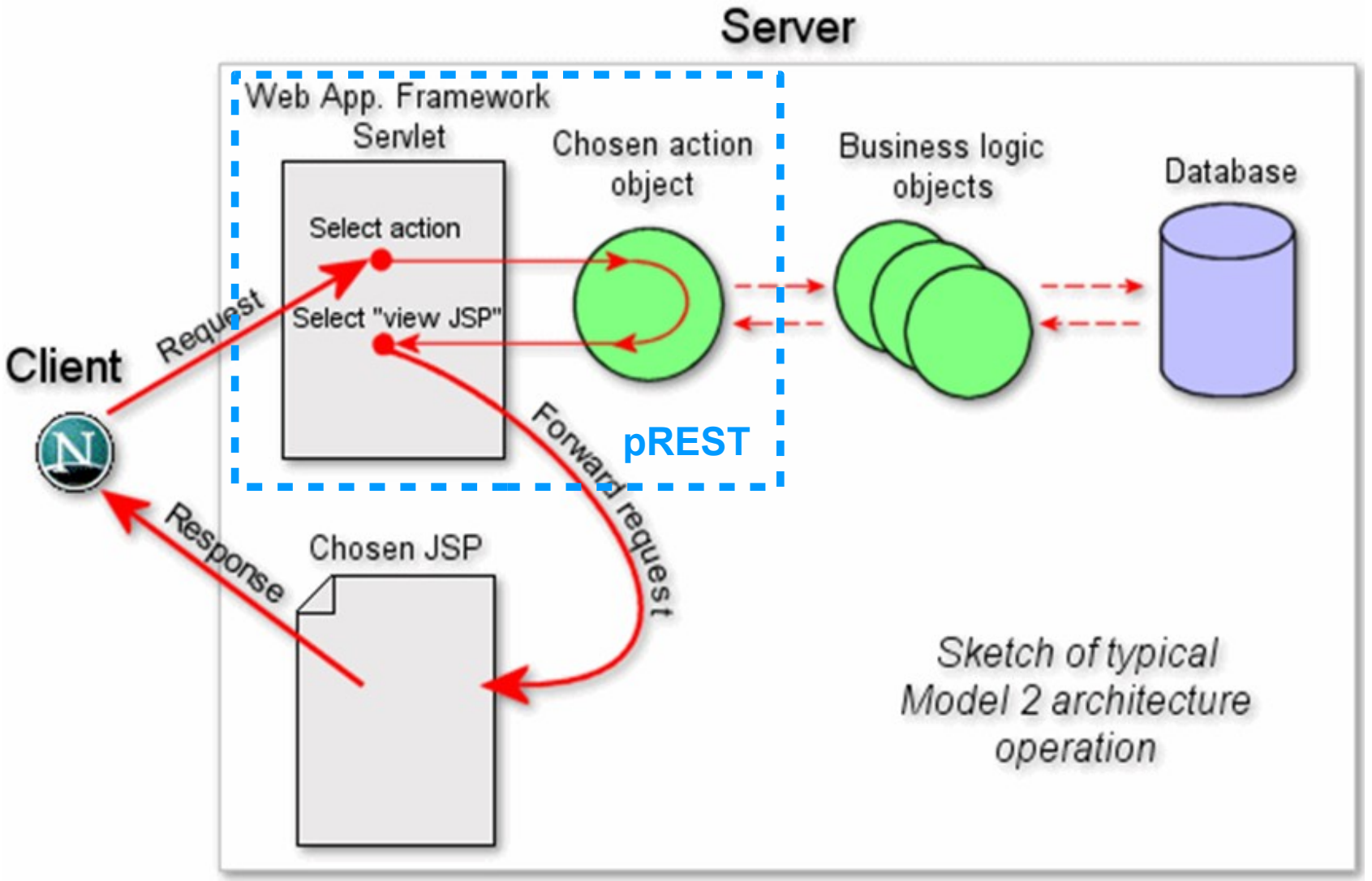
pREST – server



MVC Diagram

- **Kontróler**
  - Základná funkčná jednotka
  - Multiaction – nie komand
  - Akcie – mapované na verejné metódy kontrólera

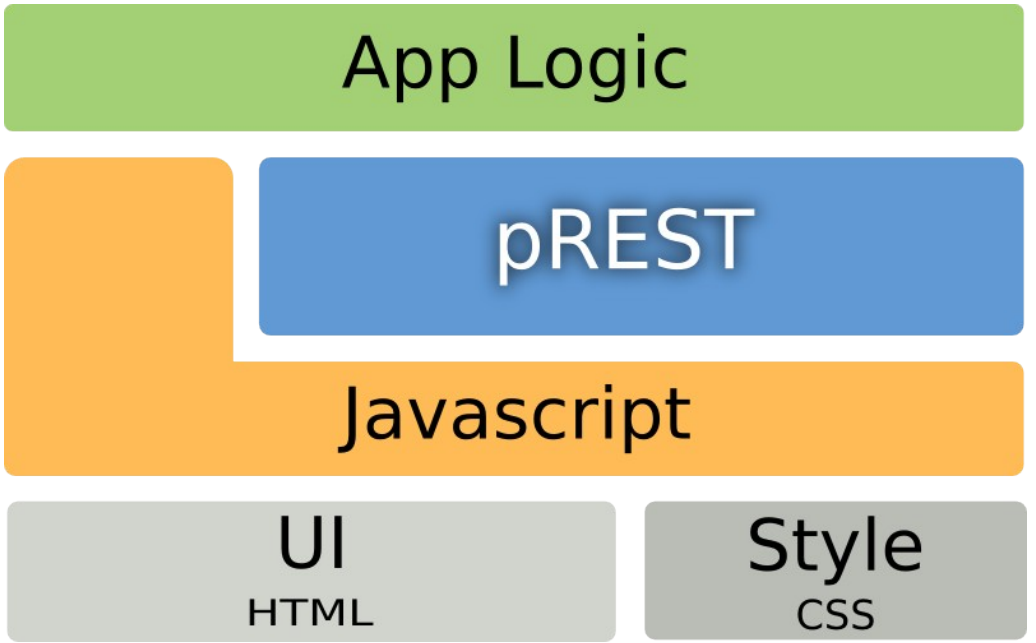
pREST – server



pREST – klient

- **JavaScript knižnica, UI Toolkit** pre tvorbu **RIA**

- Cieľ:
  - **Jednoduchosť**
  - **Efektivita**
  - **Abstrakcia DOM**
  - **Reusability**
  - Konektivita s okolím
  - **Nezávislosť na serverovej platforme**

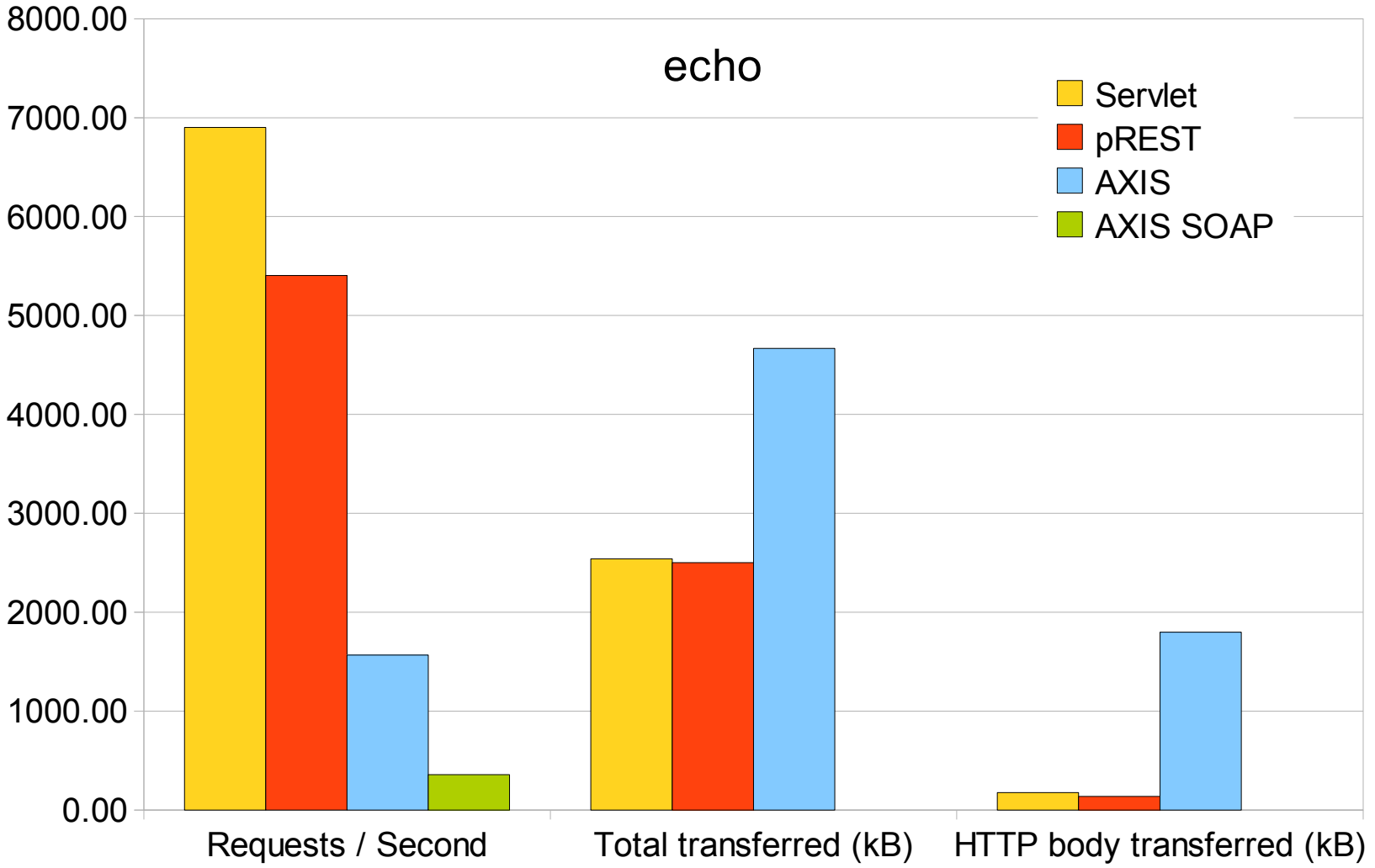


Využiť silu JavaScript-u ako prototypovacieho objektového jazyka

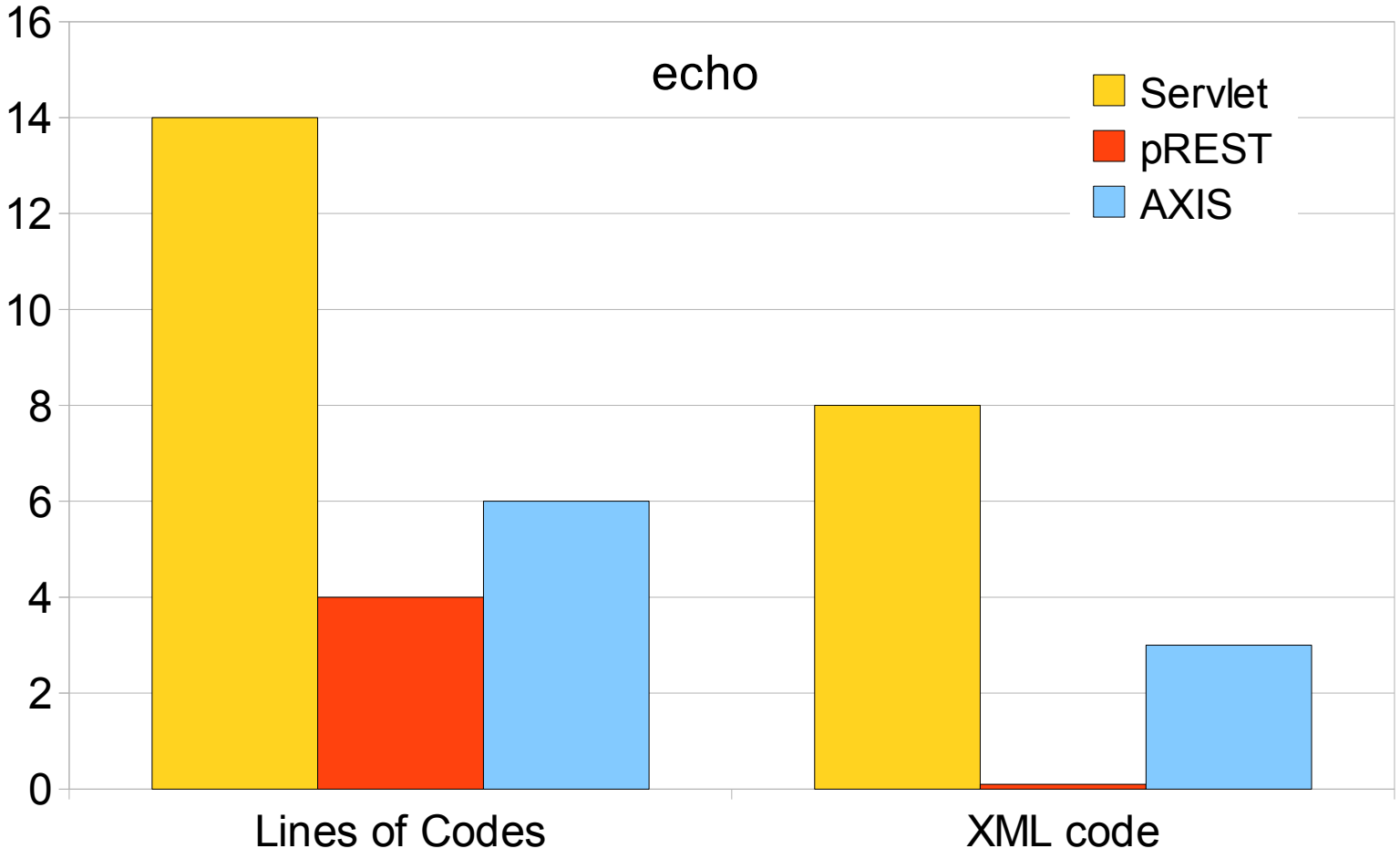
pREST – spĺňa požiadavky

- **Minimálna doba nábehu vývojára do vývoja**
  - Čas rádovo **v hodinách**
- **Horizontálny vývoj aplikácií**
  - Vývojár ovláda iba svoju doménu
  - Vyššia **kvalita kódu**, **efektivita vývoja**
- **Voľná väzba**
  - Technologická nezávislosť
  - **Stabilita**
- **Platformová nezávislosť**
  - Java 5

pREST – výkon, efektivita, testy



pREST – výkon, efektivita, testy



?

CENTAUR

Peter Rybár
peter.rybar@centaur.sk