

Integrácia – Čo je SOA?

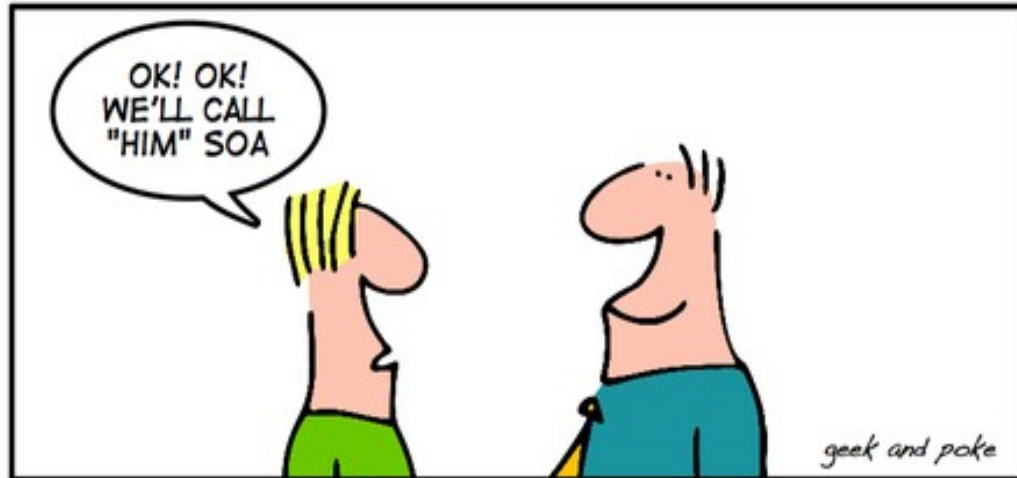
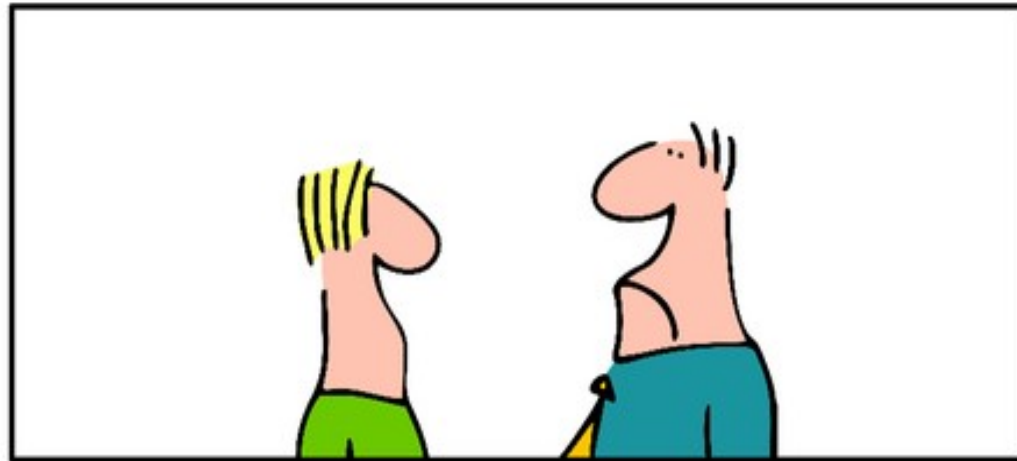
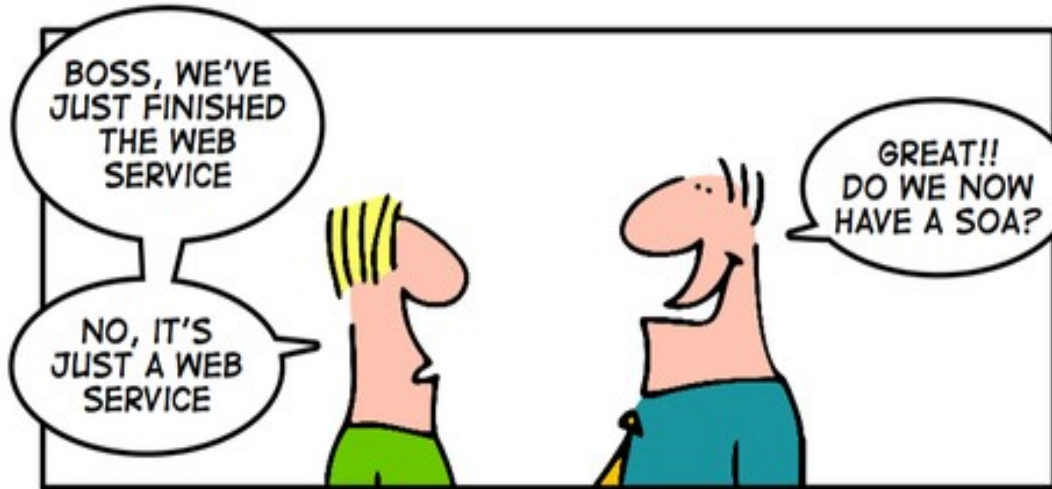
Peter Rybár * Daniel Buchta * Jozef Sivek

Integrácia

- **Systemová Integrácia**
 - Spájanie **komponentov** do jedného **systemu** aby fungoval ako **celok**.
- **Integrácia v IT**
 - Spájanie **rôznych** softvérových aplikácií **fyzicky a funkčne**.
- **Enterprise Application Integration (EAI)**
 - Použitie **softvéru** a **architektonických princípov** na integráciu **sady aplikácií** ⇒ **Architektúra**

SOA

- **Čo je to SOA?**
 - Na túto otázku **je ťažké jednoznačne odpovedať**.
 - Rôzni ľudia chápu tento pojem rôznym spôsobom.



HOW TO GET A SOA

SOA

- **SOA** – architektúra orientovaná na služby
 - Pracovná jednotka – **Služba**
 - Výsledkom služby je **zmena stavu** poskytovateľa alebo konzumenta
 - Poskytovateľ aj konzument služby:
 - **Softvéroví agenti.**

SOA a služby

- **Služba**
 - **Endpoint**
 - Adresa služby (**URI**).
 - **Kontrakt (rozhranie)**
 - Kolekcia všetkých správ podporovaných službou.
 - **Správa**
 - Základná jednotka komunikácie
 - HTTP správa, SOAP správa, JMS, SMTP, ...

SOA - Pravidlá

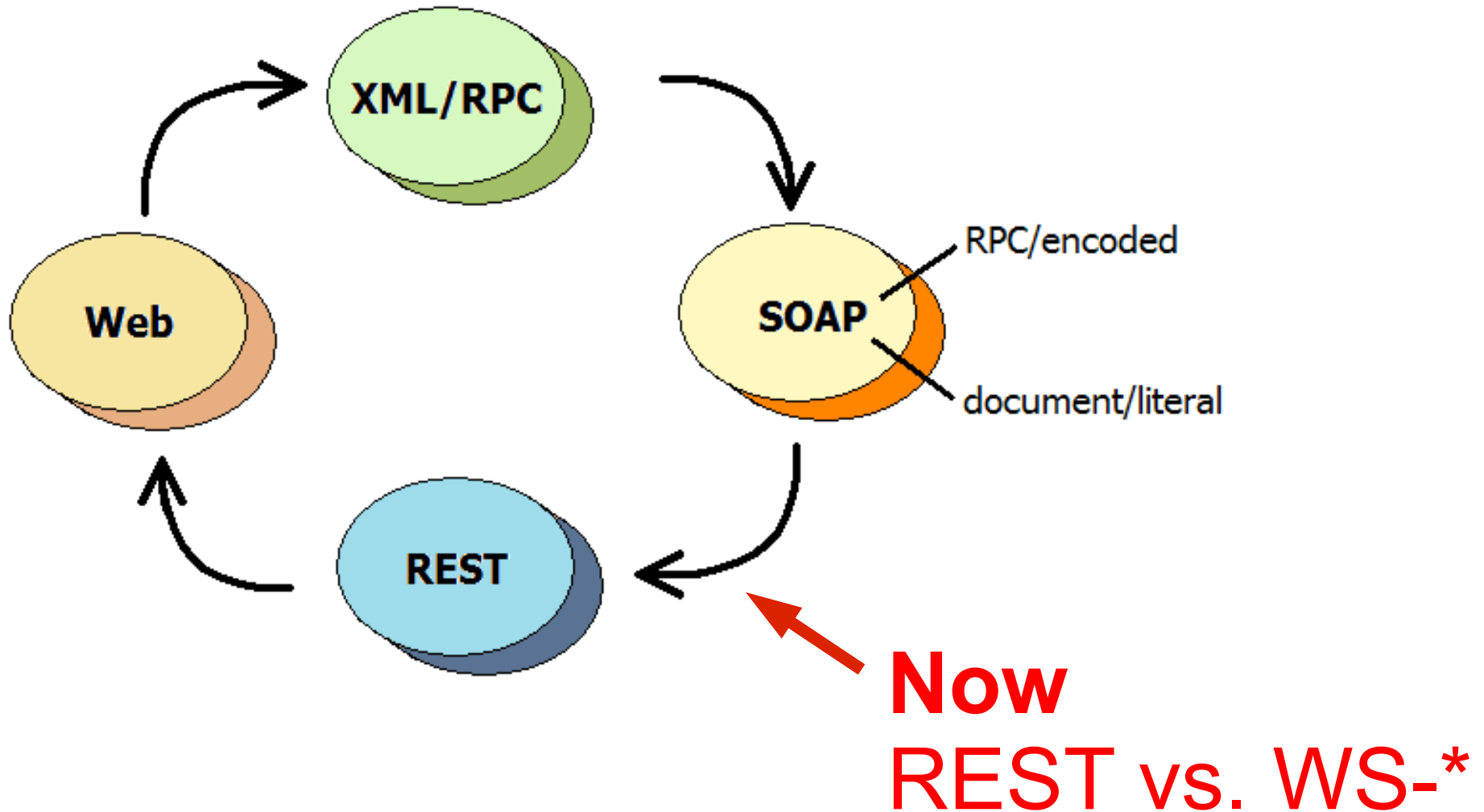
- **SOA musí spĺňať nasledujúce pravidlá:**
 - **Správy sú popisné** – hovoria čo sa má vykonať, nie ako sa to má vykonať.
 - **Otvorený formát správ** – formát ktorý chápu všetky zúčastnené strany.
 - **Rozšíriteľnosť správ a rozhraní** – inak by poskytovateľ a konzument služby boli viazaní na určitú verziu.
 - **Objaviteľnosť služieb** – flexibilný mechanizmus, nemusí sa jednať o centralizovaný register.

SOA – implementácie

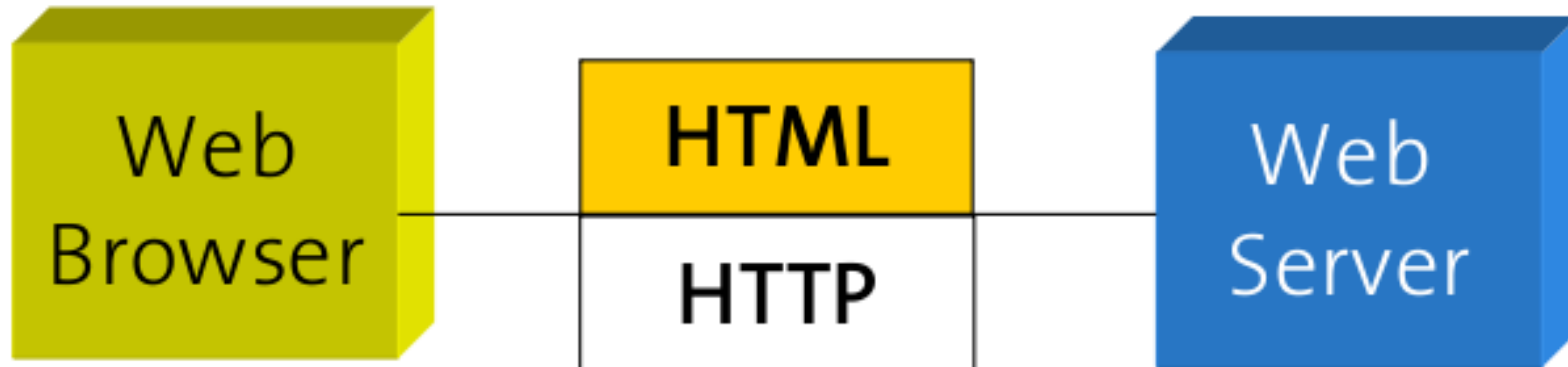


- **WEB** (1990)
- CORBA (1991)
- XML-RPC (1998)
- **WS-*** (1998)
 - SOAP – RPC/literal
 - SOAP – Document/literal (2001)
- **REST** (2000)

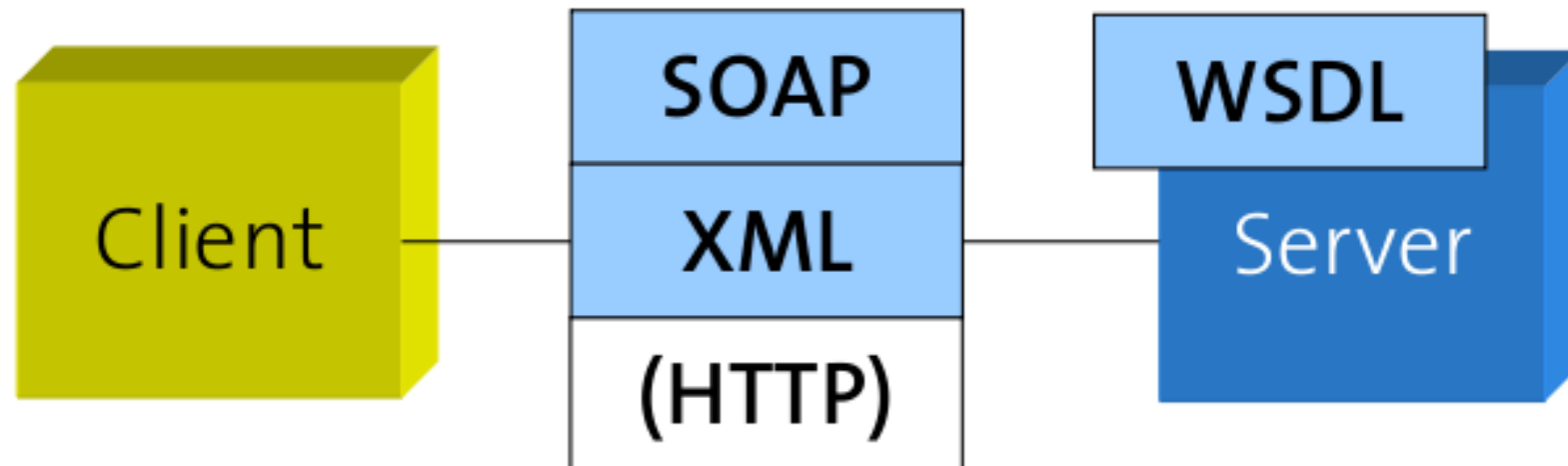
SOA – Web implementácie



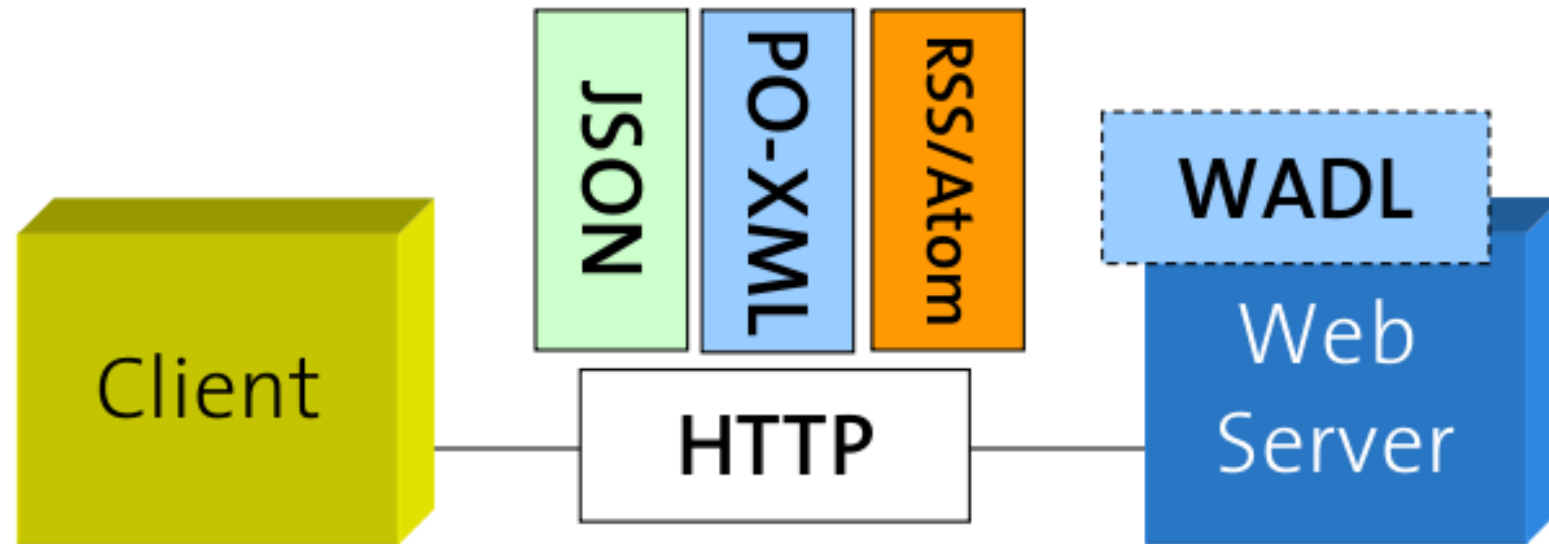
Web Sites (1992)



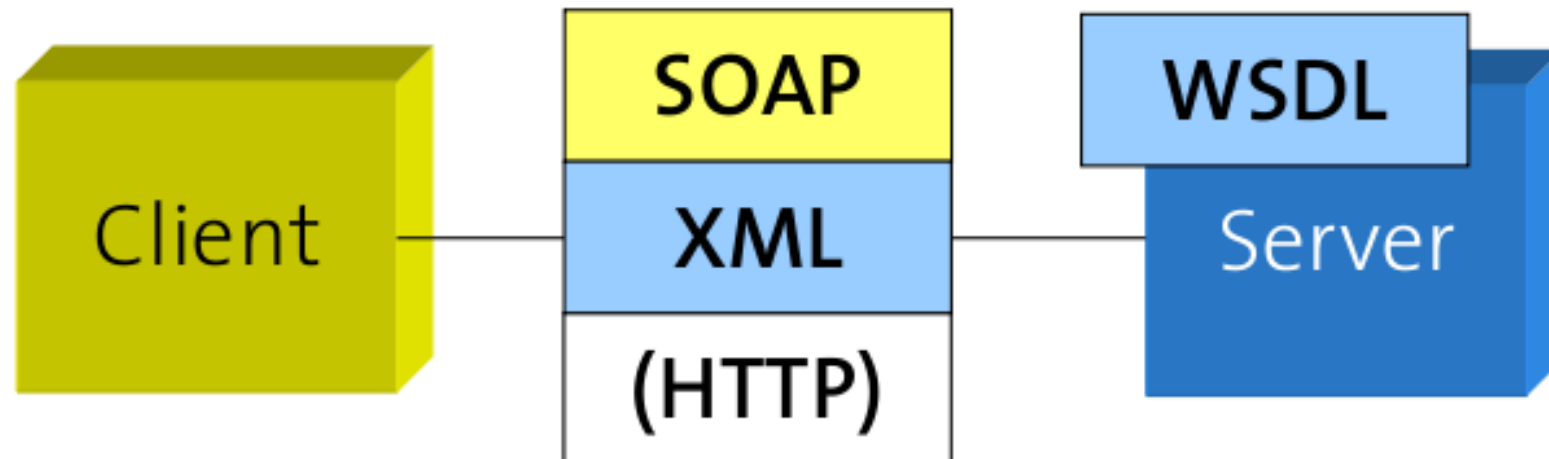
WS-* Web Services (2000)



RESTful Web Services (2007)



WS-* Web Services (2000)



WS-* Standards stack

Interoperability Issues

- Basic Profile
- Advanced Profile
- Single SOAP Binding Profile
- Basic Security Profile
- WS-Security Profile
- WS-Trust Profile
- WS-Transfer Profile
- Transaction Core
- Transaction Metadata (XLANG)
- Reliable, Replicated, Message-Oriented Profiles (RRMR)

Standards Profiles

WS-I Basic Profile

WS-I Advanced Profile

WS-I Single SOAP Binding Profile

WS-I Basic Security Profile

WS-I WS-Security Profile

WS-I WS-Trust Profile

WS-I WS-Transfer Profile

WS-I Transaction Core

WS-I Transaction Metadata (XLANG)

WS-I Reliable, Replicated, Message-Oriented Profiles (RRMR)

Business Process Specifications

Business Process Execution Language (BPEL4People)

Web Services Choreography Description Language (WS-CDL)

Web Services Business Process Modeling Language (WS-BPEL)

Web Services Choreography Description Language (WS-CDL)

Management Specifications

Web Services Management Language (WS-ML)

Web Services Management Language (WS-ML)

Web Services Management Language (WS-ML)

Web Services Management Language (WS-ML)

Metadata Specifications

WS-Policy

WS-PolicyAttachments

WS-PolicyAssertions

WS-PolicyExpressions

WS-PolicyTest

WS-PolicyTestEngine

WS-PolicyTestEngine

WS-PolicyTestEngine

WS-PolicyTestEngine

WS-PolicyTestEngine

WS-PolicyTestEngine

Reliability Specifications

WS-ReliableMessaging

WS-Reliability

Security Specifications

WS-Security

WS-Security Assertions

WS-Security Assertions

WS-Security Assertions

WS-Security Assertions

WS-Security Assertions

WS-Security Assertions

WS-Security Assertions

WS-Security Assertions

WS-Security Assertions

Transaction Specifications

WS-Transaction

WS-Transaction

WS-Transaction

WS-Transaction

WS-Transaction

WS-Transaction

WS-Transaction

WS-Transaction

WS-Transaction

Resource Specifications

Web Services Resource Framework (WS-RF)

WS-Resource

WS-Resource

WS-Resource

WS-Resource

WS-Resource

WS-Resource

WS-Resource

WS-Resource

Messaging Specifications

WS-Addressing

WS-Addressing

WS-Addressing

WS-Addressing

WS-Addressing

WS-Addressing

WS-Addressing

WS-Addressing

WS-Addressing

SOAP

SOAP

SOAP

SOAP

SOAP

XML Specifications

XML

XML

XML

XML

XML

XML

XML

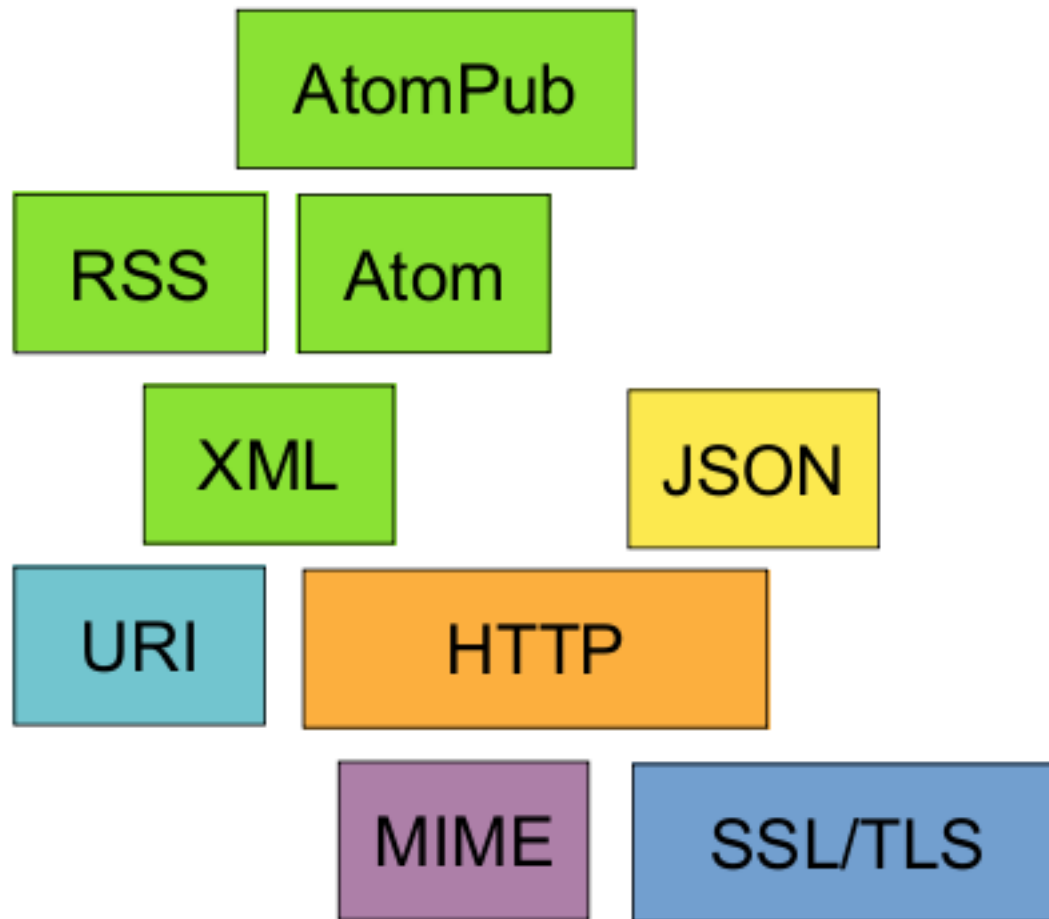
XML

XML

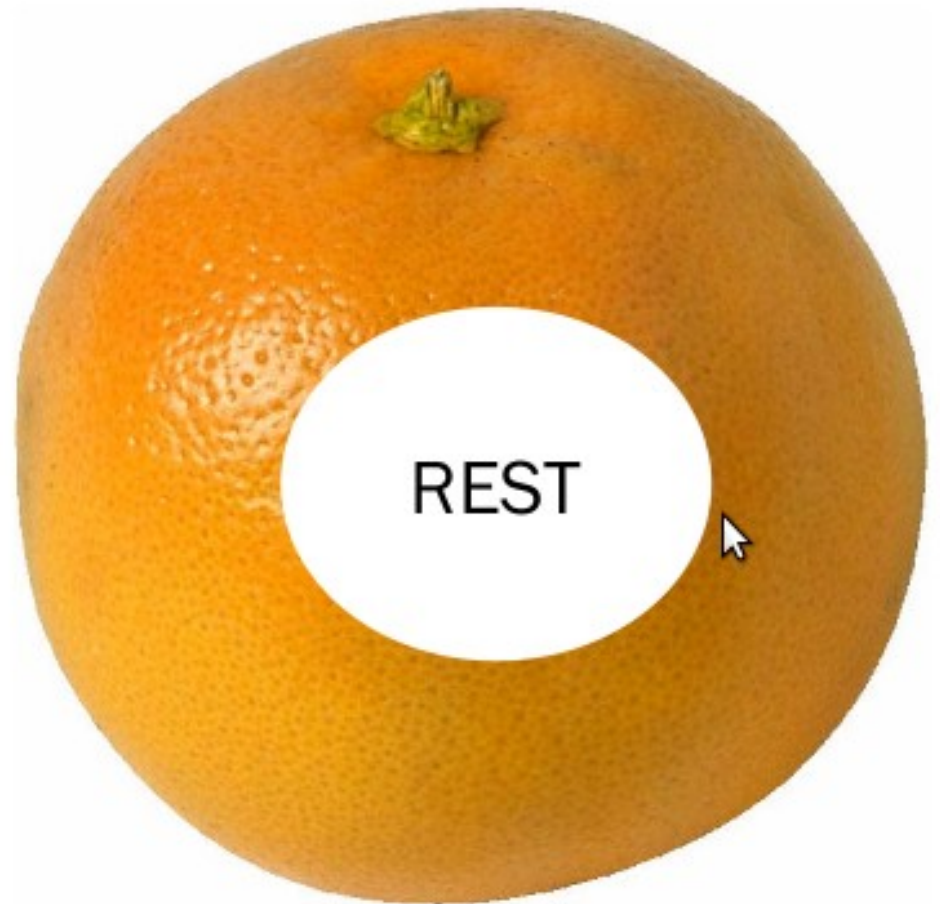
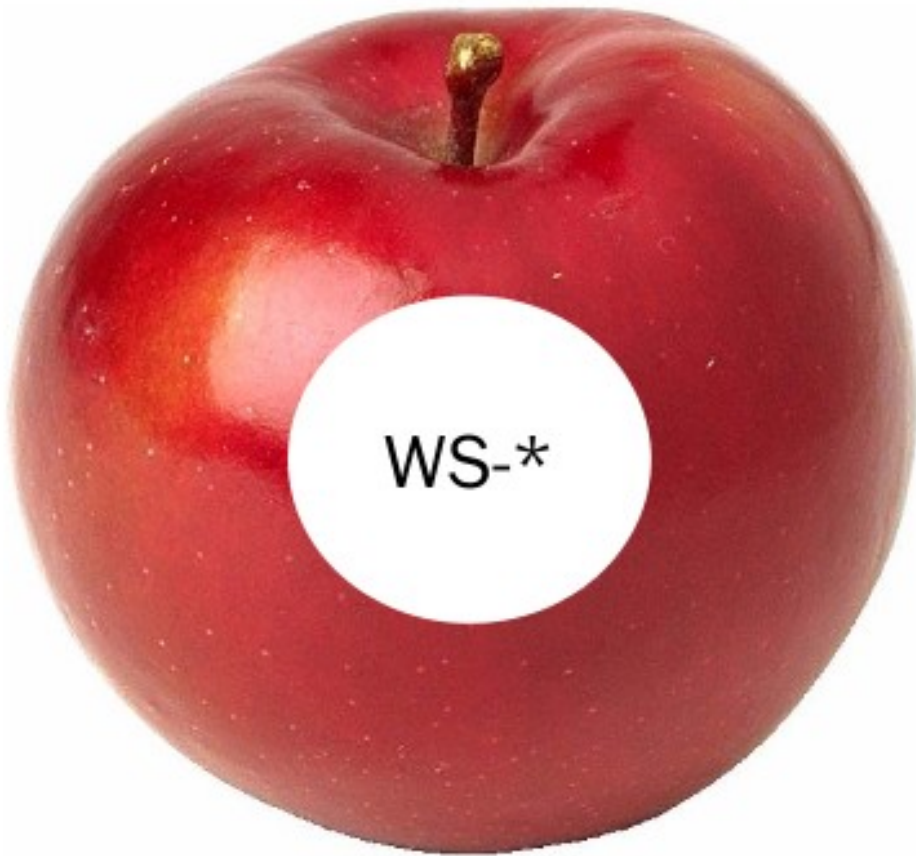
Dependencies



REST Standards stack



REST vs. WS-*



REST vs. WS-*



WS-*

Middleware
Interoperability
Standards



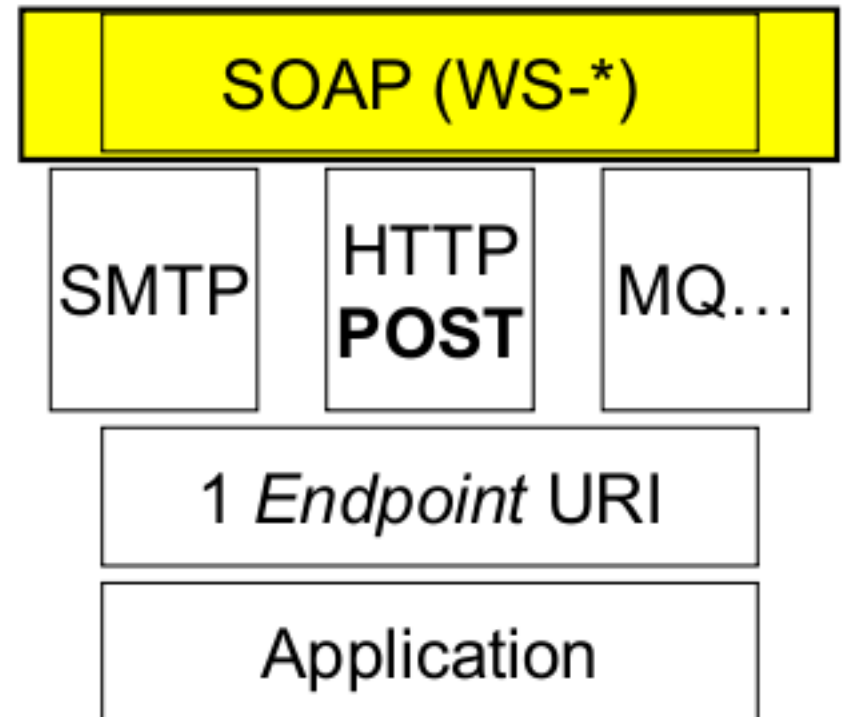
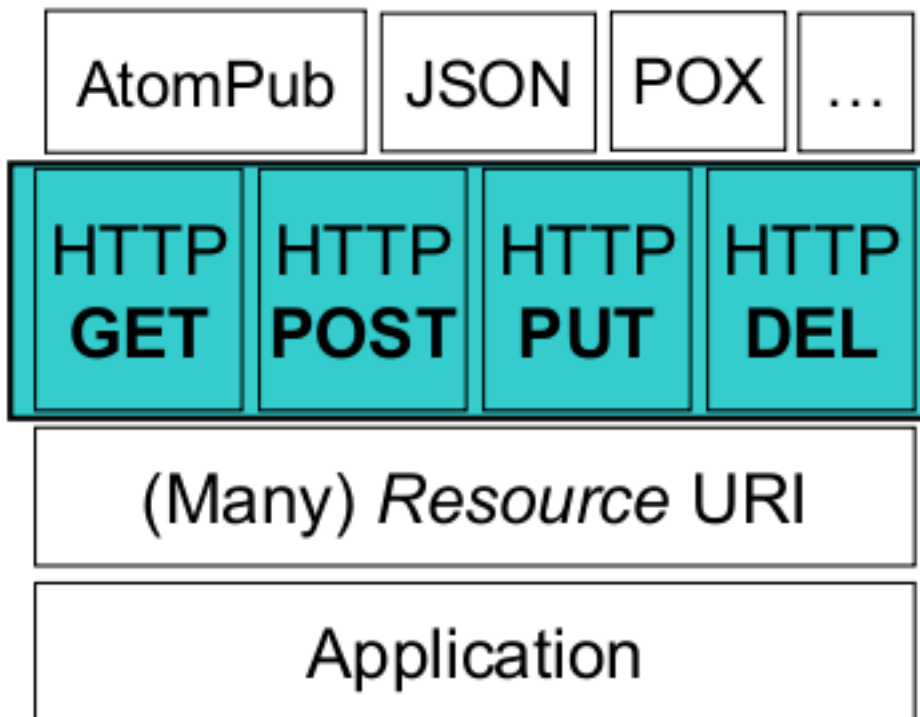
REST

Architectural
style for
the Web

REST vs. WS-*

- Web – svet globálne prístupných informácií
- Aplikácie publikujú dáta na Webe prostredníctvom URI

- Web – univerzálne transportné médium pre informácie
- Aplikácie vzájomne interagujú - zostávajú mimo Web



REST vs. SOAP (WS-*)

Ako priviest' Web naspät' do Web Služieb

SOAP

- **Čo je SOAP?**
- **Započal v 1998 ako remote procedure call protocol** ktorý **mal prejsť cez firewalls** (Simple Object Access Protocol)
- Prvé verzie SOAP:
 - Definovali extenzibilný message formát založený na XML
 - Serializácia do RPC parametrov v XML
 - Transportný protokol HTTP prechádzal cez firewall
- **Posledných 7 rokov** vývoja SOAP:
 - Flexibilnejší prenos XML dát – **document / literal**
 - **Protocol agnostic** – nezávislosť na transportnom protokole

SOAP je iba SOAP

- **SOAP** – protokol “framework” na dosiahnutie potrebnej **interoperability** medzi message-based middleware nástrojmi
- Priemyselný štandard

SOAP Web Services slabosti

- **Priveľká komplexita**
- Problematický štandardizačný proces
- **Skryté problémy**
 - **Absencia architektonickej koherencie**
 - Fragmentácia
 - Design by committee
 - **Feature Bloat** (spájanie konkurujúcich si špecifikácií)
 - Absencia referenčnej implementácie
 - Štandardizácia nad štandardami (WS-I)
- **Nezačína sa už podobat' na CORBA?**
- **Kedy už SOAP WS interoperability začne skutočne fungovať?**
- Naozaj potrebujeme platiť za použitie XML nízkym výkonom?

Na čo sú SOAP WS dobré?

- **SOAP/WSDL** – **komunikačná technológia** poskytujúca interoperabilitu, spoločnú bázu pre messaging aplikácie pracujúce spoločne nad HTTP
- **Legacy reuse a integrácia**
 - Zastaralé systémy nie sú Web-friendly (nepoužívajú HTTP) – používajú multicast, RPC/RMI binárne protokoly, asynchrónny messaging, dávkové hromadné transféry
- **Flexibilita a adaptácia**
 - Ten istý interface je možné naviazať na rôzne protokoly podľa toho ako sa menia biznis a technologické požiadavky

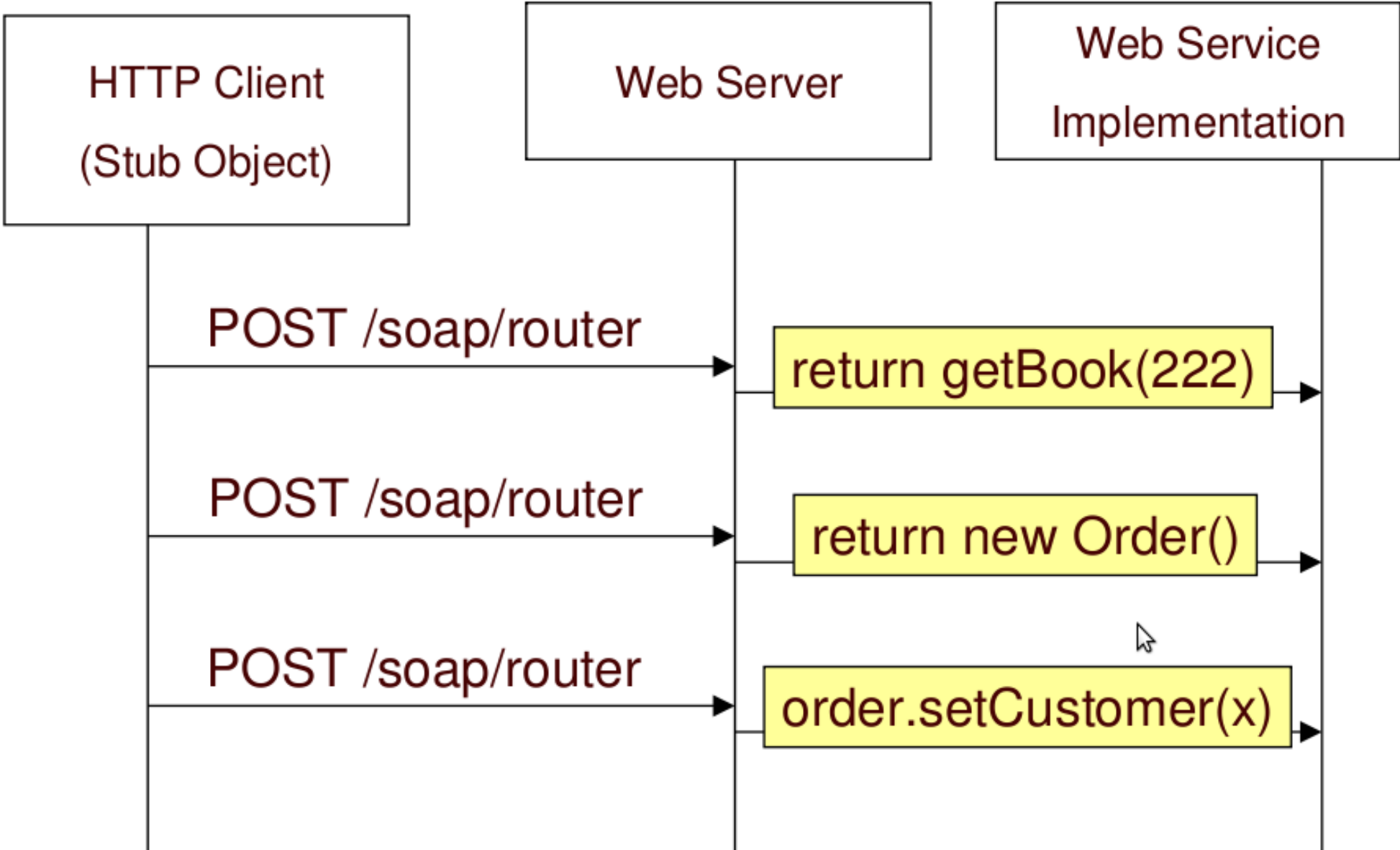
Čo je REST?

- REST je **architektonický štýl Webu**
- REST stojí na princípoch, ktoré umožňujú HTTP byť **tak dobre škálovateľný**

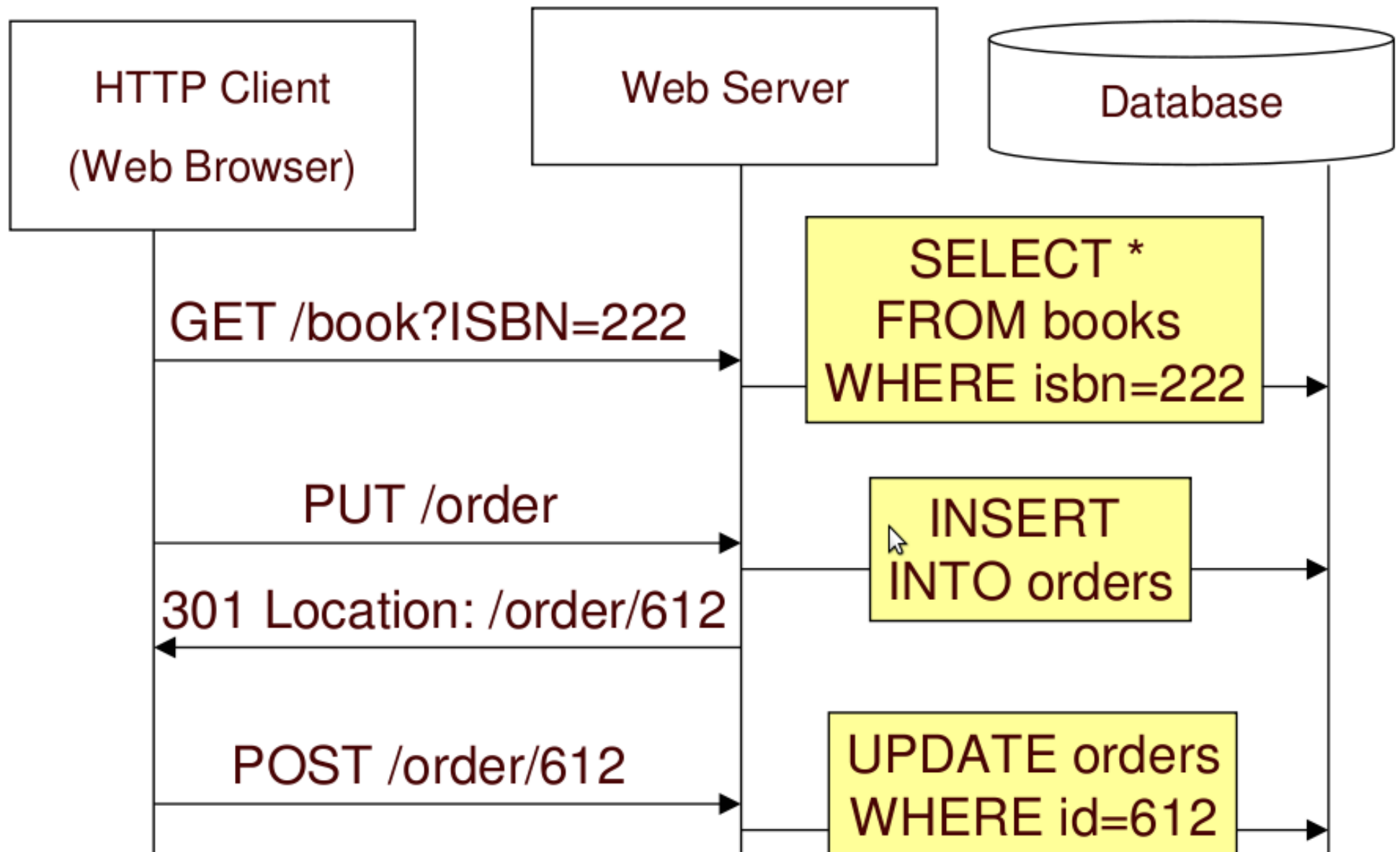
(Roy Fielding PhD, 2000)

1. Identifikácia zdroja na základe **URI**
2. Jednotné rozhranie pre všetky zdroje – **CRUD**
3. “**Samopopisné**” správy
4. **Hyperlinky** medzi rôznymi typmi médií

Web Service Example (from REST perspective)



RESTful Web Application Example



High REST vs. Low REST

- **Best practices** sa rôznia:
 - **Low REST**
 - HTTP **GET** pre idempotentné požiadavky, **POST** pre všetko ostatné
 - Odpovede v ľubovoľnom MIME Type (**XHTML**)
 - **High REST**
 - Doporučované používanie „pekných“ URI
 - Plné využitie **GET, POST, PUT, DELETE**
 - Využitie dátových MIME Type formátov **RDF, JSON, YAML, ATOM**

REST silné stránky

- **Jednoduchosť**
 - Jednotné nemenné rozhranie – CRUD
- HTTP je **všadeprítomný** – povolený na firewall-och
- **Bezstavová / Synchronna** interakcia
- **Škálovateľnosť potvrdená!**
- **Ľahká adopcia** a tenká infraštruktúra
 - **Stačí Webový prehliadač** – nie je potreba kupovať WS-* middleware
- Fundamentálny prístup
- **Silná podpor zo strany Web 2.0**
 - Google v 2006 **opustil SOAP/WSDL** API

REST slabé stránky

- Nerozhodnosť v použití High REST vs. Low REST
- Priamo neponúka riešenia nad rámec HTTP/SSL
- **Zdanlivá absencia štandardov** – iných než URI, HTTP, XML, MIME, HTML
- **Sémantika/Syntax** je dosť neformálna – **user/human oriented**

REST vs. SOAP - Metodológia návrhu

- Identifikovať zdroje ktoré majú byť vystavené ako služby
 - Definovať “pekné” URL pre adresáciu zdrojov
 - Rozlíšiť read-only zdroje (GET) od modifikovateľných zdrojov (POST, PUT, DELETE)
 - Zabezpečiť prelinkovanie zdrojov linkami v reprezentáciách
 - Implementovať a nasadiť na Web server
- Identifikovať operácie služieb (procedúry) do WSDL dokumentu
 - Definovať dátový model pre obsah komunikačných správ, ktorými služba komunikuje (XML Schema data types).
 - Zvoliť vhodný transportný protokol, bezpečnostné, prístupové a transakčné pravidlá
 - Implementovať a nasadiť na Web service container

REST vs. SOAP - Stavovosť

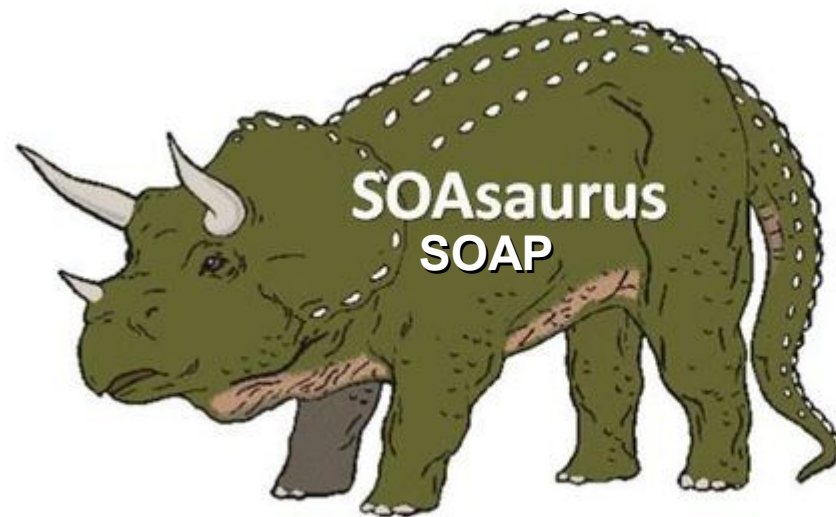
- **REST** – explicitné stavové prechody
 - **Server je bezstavový**
 - Klienti udržujú stav korektne využitím **liniek**
- Techniky pre zavedenie session do HTTP:
 - **Cookies** (HTTP Headers)
 - **URL Re-writing**
 - **Hidden Form Fields**

- **SOAP** – služby majú implicitné stavové prechody
 - **Server udržiava state naprieč správami**
 - Klienti udržujú stav určený **stavovým strojom** služby
- Techniky pre zavedenie session do SOAP:
 - **Session Headers** – neštandardný spôsob

REST vs. SOAP - Security

- REST security je o HTTPS
 - Proven track record (SSL1.0 od 1994)
 - Secure – point to point komunikácia
 - Autentifikačná integrita
 - Encryption
- SOAP security extensions definované vo WS-Security (od 2004)
 - XML Encryption (2002)
 - XML Signature (2001)
 - Implementácie sa práve vynárajú
 - Problémy interoperability
 - Performance?
 - Secure end-to-end komunikácia – Self-protecting SOAP messages

Evolúcia



?