

Úvod do Web Services

Základy webových služeb a jejich implementace
na platformě OS/2

Jarda Kačer
jarda@kacer.biz

Český Warpstock 2008
Brno, 20.-21.9.2008

Co je to webová služba?

- Část business logiky přístupná přes standardní internetové protokoly
- Pro komunikaci se používá XML
- Nezávislá na platformě a programovacím jazyce
- Nemá nic společného s HTML, webovým prohlížečem apod.

Proč používat webové služby

- Konečně standard, na kterém se shodli všichni
 - Předtím CORBA, RMI, EJB, DCOM, ...
- Používají se standardní protokoly a formáty dat
- Podporuje Java, .NET, teoreticky jakýkoliv programovací jazyk
- Ideální pro spojování heterogenních systémů
 - Používá se hodně pro integrace systémů

Základní vlastnosti

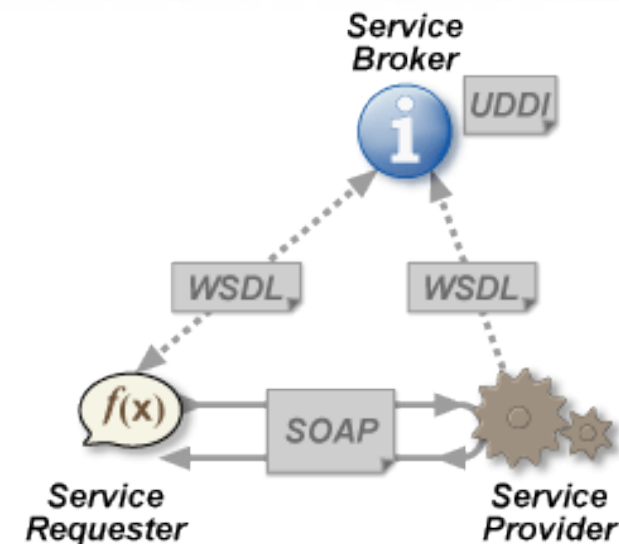
- Distribuované, volně vázané
- Samopopisující se – viz dále WSDL
- Synchronní i asynchronní
- Podporuje RPC – Remote Procedure Call
- Nejsou zaměřeny prezentačně vůči uživateli, ale programově vůči strojovému použití
 - Lze vidět jako API pro ostatní aplikace

Srovnání s tradičním RPC

- Tradiční RPC
 - V rámci jedné firmy
 - Pro konkrétní jazyk
 - Volání procedur
 - Obvykle daný transportní protokol
 - Těsně vázané
 - Efektivní zpracování na čas i velikost
- Webová služba
 - Lze i mezi firmami
 - Nezávislé na jazyce
 - Posílání zpráv (XML)
 - Snadno vyměnitelné transportní protokoly
 - Volně vázané
 - Poměrně neefektivní zpracování (XML)

Základní prvky architektury

- Klient
 - Využívá službu
- Poskytovatel
 - Poskytuje službu
- Registry
 - Uchovává informace o umístění služeb
 - Nepovinné, používá se zřídka



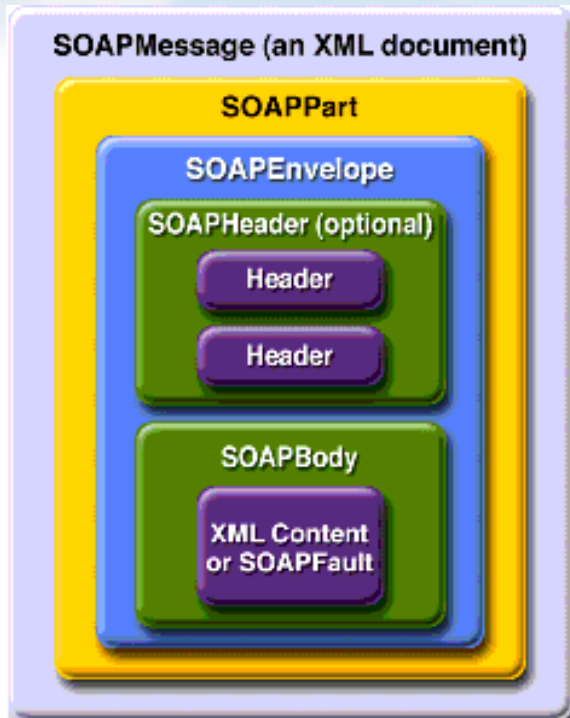
Dva základní typy WS

- SOAP
 - Simple Object Access Protocol
 - Základní prvek: Operace
- REST
 - Representational State Transfer
 - Základní prvek: Zdroj
 - Přístupný přes základní operace HTTP: GET, POST, PUT, DELETE

SOAP

- Protokol pro definici:
 - Pravidel pro kódování dat
 - Struktury zpráv
 - Napojení na transportní protokol
- Verze 1.1 obecně v používání, existuje v. 1.2
- Zprávy: Obálka, v ní hlavička a tělo, volitelně přílohy pro velké objemy dat
- Přesný tvar konkrétních zpráv je dán WSDL dokumentem dané WS

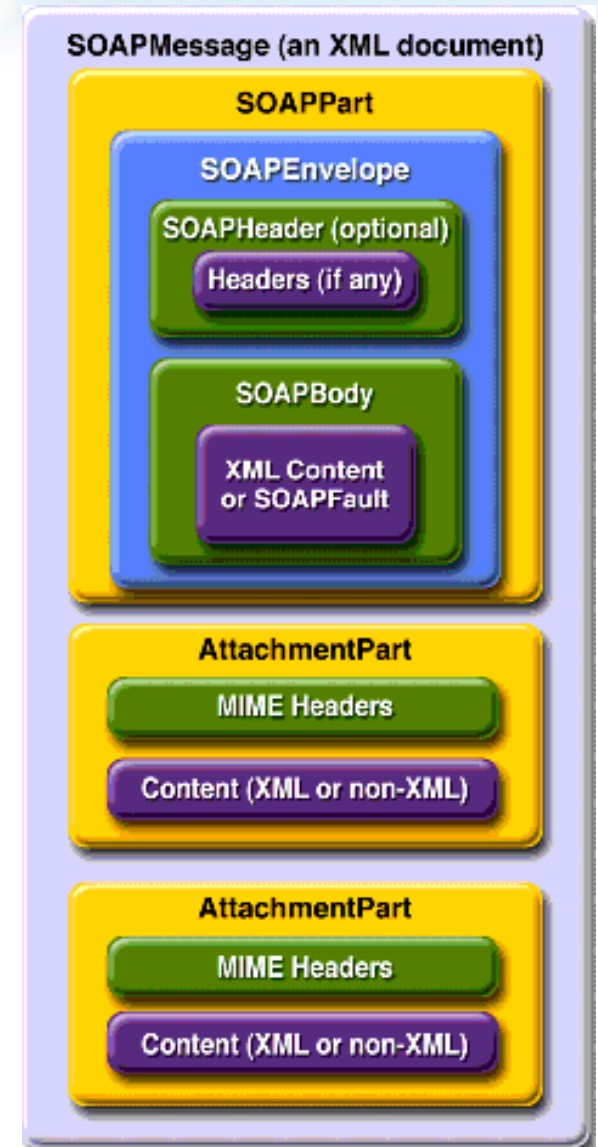
SOAP - zprávy



Bez příloh

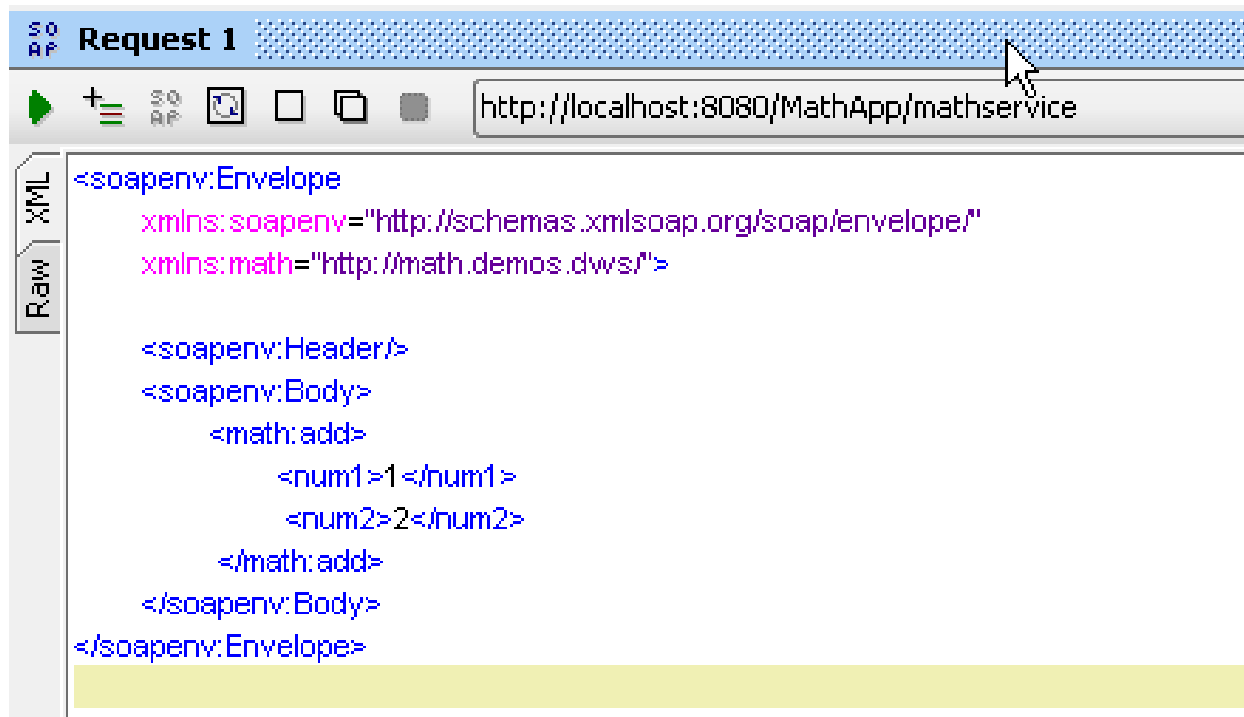


S přílohami
Vhodné pro rozsáhlá data,
například binární soubory
apod.
Použito MIME stejně jako
například v emailech.



SOAP – příklad zprávy

Příklad triviálního SOAP requestu pro sečtení dvou čísel:



The screenshot shows a SOAP client interface. At the top, there is a header bar with the text "Request 1" and a URL "http://localhost:8080/MathApp/mathservice". Below the header, there is a "Raw XML" tab selected, displaying the following XML code:

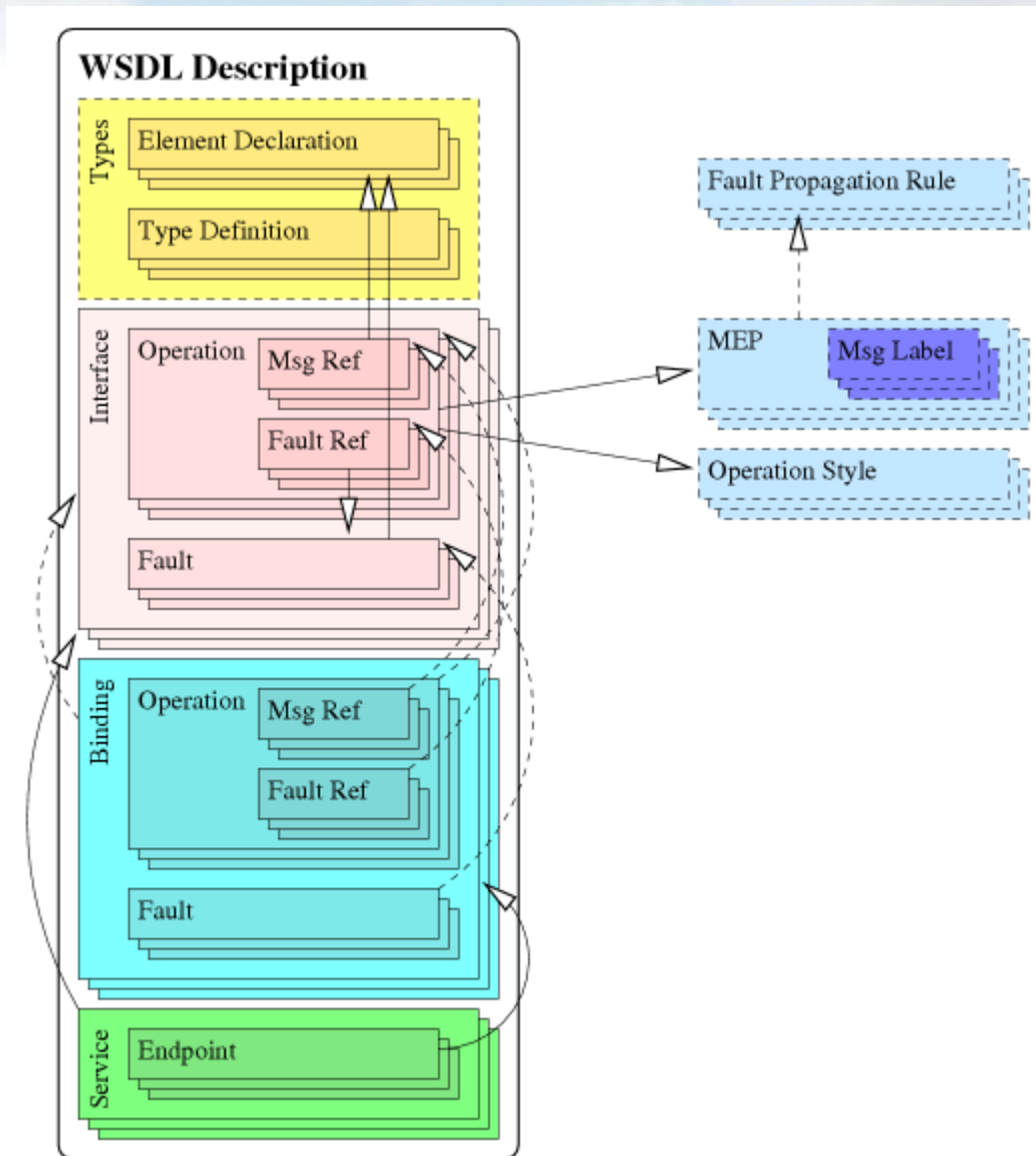
```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:math="http://math.demos.dws/">

  <soapenv:Header/>
  <soapenv:Body>
    <math:add>
      <num1>1</num1>
      <num2>2</num2>
    </math:add>
  </soapenv:Body>
</soapenv:Envelope>
```

WSDL

- Web Service Description Language, v. 1.1 a 2.0
- Popisovač WS, může být automaticky generován službou samotnou nebo být statický
- Opět XML formát
- Obsahuje definice:
 - **Datových typů** – XML Schema
 - **Typů zpráv**, které mohou být dále použity
 - **Typů portů (rozhraní)** a jejich **operací**
 - **Vazeb** – napojení portů na transportní protokol
 - **Služeb** – konkrétní **URL** + **port s vazbou**

WSDL - struktura



Struktura dokumentu WSDL 2.0.
Stále se používá starší WSDL 1.1,
praktické rozdíly nejsou velké.

WSDL - příklad

The screenshot shows a WSDL viewer interface with a tree view on the left and a source code area on the right. The tree view is expanded to show the following structure:

- Source: <http://math.demos.dws/>
- Types
 - Imports
 - Messages
 - add
 - addResponse
 - divide
 - divideResponse
 - multiply
 - multiplyResponse
 - subtract
 - subtractResponse
 - Port Types
 - MathWS
 - add
 - divide
 - multiply
 - subtract
 - Bindings
 - MathWSPortBinding PortType="MathWS"
 - soap:binding
 - add
 - divide
 - multiply
 - subtract
 - Services
 - mathservice
 - MathWSPort Binding="MathWSPortBinding"
 - soap:address
 - Extensibility Elements

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.1-->
<types>
  <xsd:schema>
    <xsd:import namespace="http://math.demos.dws/" />
  </xsd:schema>
</types>
<message name="add">
  <part name="parameters" element="tns:add"/>
</message>
<message name="addResponse">
  <part name="parameters" element="tns:addResponse"/>
</message>
<message name="divide">
  <part name="parameters" element="tns:divide"/>
</message>
<message name="divideResponse">
  <part name="parameters" element="tns:divideResponse"/>
</message>
<message name="multiply">
  <part name="parameters" element="tns:multiply"/>
</message>
<message name="multiplyResponse">
  <part name="parameters" element="tns:multiplyResponse"/>
</message>
<message name="subtract">
  <part name="parameters" element="tns:subtract"/>
</message>
<message name="subtractResponse">
  <part name="parameters" element="tns:subtractResponse"/>
</message>
</definitions>
```

```
<message name="subtractResponse">
  <part name="parameters" element="tns:subtractResponse"/>
</message>
<portType name="MathWS">
  <operation name="add">
    <input message="tns:add"/>
    <output message="tns:addResponse"/>
  </operation>
  <operation name="divide">
    <input message="tns:divide"/>
    <output message="tns:divideResponse"/>
  </operation>
  <operation name="multiply">
    <input message="tns:multiply"/>
    <output message="tns:multiplyResponse"/>
  </operation>
  <operation name="subtract">
    <input message="tns:subtract"/>
    <output message="tns:subtractResponse"/>
  </operation>
</portType>
```

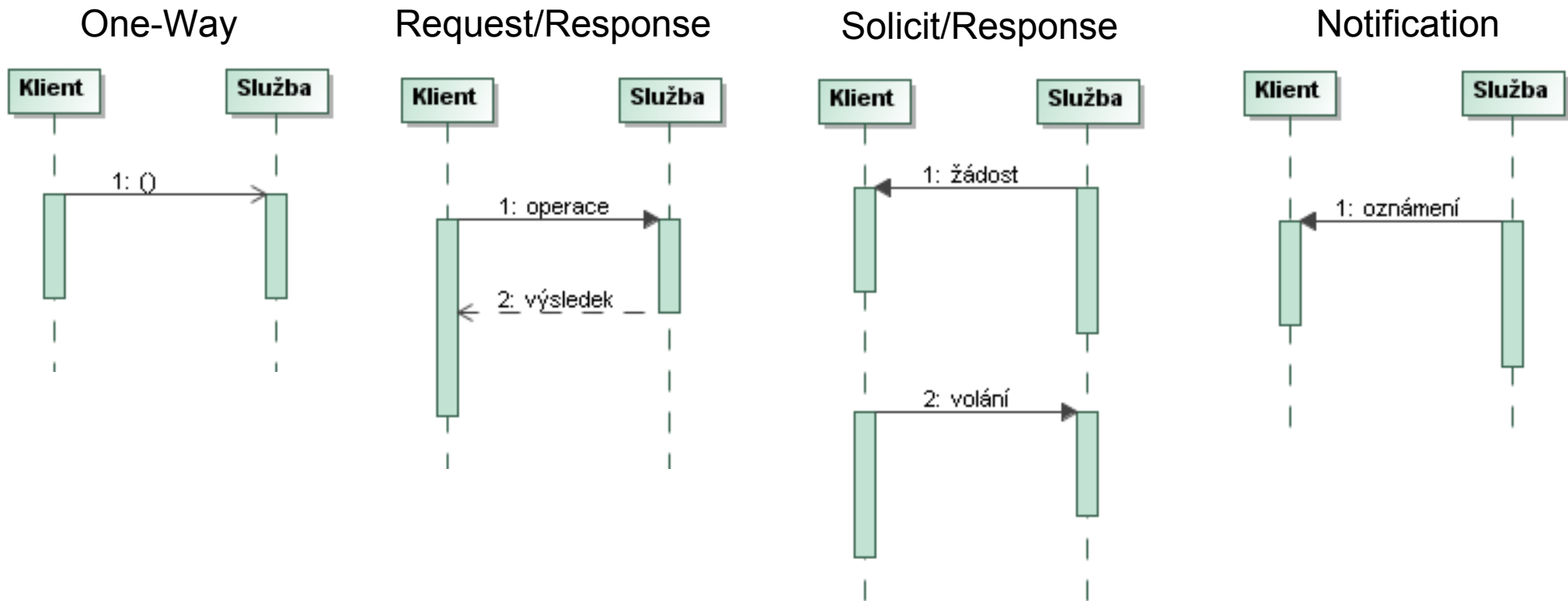
```
<operation name="divide">
  <soap:operation soapAction="divide"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
<operation name="multiply">
  <soap:operation soapAction="multiply"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
<operation name="subtract">
  <soap:operation soapAction="subtract"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
</binding>
<service name="mathservice">
  <port name="MathWSPort" binding="tns:MathWSPortBinding">
    <soap:address location="http://localhost:8080/MathApp/mathservice"/>
  </port>
</service>
</definitions>
```

Scénáře přenosu zpráv (1)

- One-Way
 - Asynchronní od klienta ke službě
- Request/Response
 - Synchronní žádost a odpověď
- Solicit/Response
 - Služba žádá o volání, klient později volá
- Notification
 - Služba jen oznamuje klientovi
- Dáno ve WSDL pořadím zpráv v operaci

Pouze tyto dva scénáře jsou běžně podporovány a povoleny ve WS-I Basic Profile 1.0.

Scénáře přenosu zpráv (2)



Styly SOAP zpráv

- Document
 - V SOAP Body jde nějaký XML dokument, který je popsán schématem
- RPC
 - V SOAP Body jde jméno vzdálené procedury a uvnitř parametry volání
 - Historický přežitek, pro nové WS nepoužívat
- Je určeno ve WSDL u `<soap:binding/>` uvnitř `<wsdl:binding/>`

```
<soap:binding
  transport="http://schemas.xmlsoap.org/soap/http"
  style="document"/>
```


Kódování přenášených dat

- Literal
 - Formát dat dán schématem, například XSD
- Encoded
 - Použito SOAP kódování
- Je určeno ve WSDL u `<soap:body/>` každého vstupu/výstupu každé operace

```
<operation name="add">  
  <soap:operation soapAction="add"/>  
  <input><soap:body use="literal"/></input>  
  <output><soap:body use="literal"/></output>  
</operation>
```

Kombinace stylů a kódování

- RPC/encoded

- RPC/literal

- document/encoded

- document/literal

Zakázáno ve WS-I
Basic Profile 1.0.

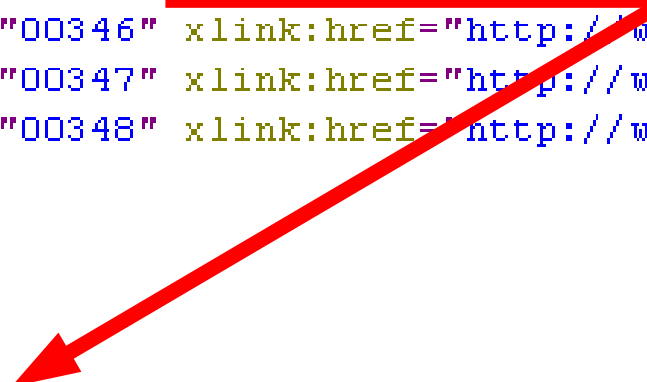
Podporováno WS-I
Basic Profile 1.0.

REST služby

- Alternativní styl k SOAPu
- Popsal Roy Fielding ve své dizertaci
- Menší zprávy, není SOAP obálka apod.
- Lepší výkonnost – kešování apod.
- Nad HTTP hned chodí XML
 - Co typ zdroje, to jiný typ XML
 - Nutno nějak „ručně“ zpracovat každý typ XML, neexistuje něco jako SOAP Engine
 - Lze použít odkazy mezi zdroji – XLink, Xpointer → síť objektů se vzájemnými odkazy

Příklady RESTových zpráv

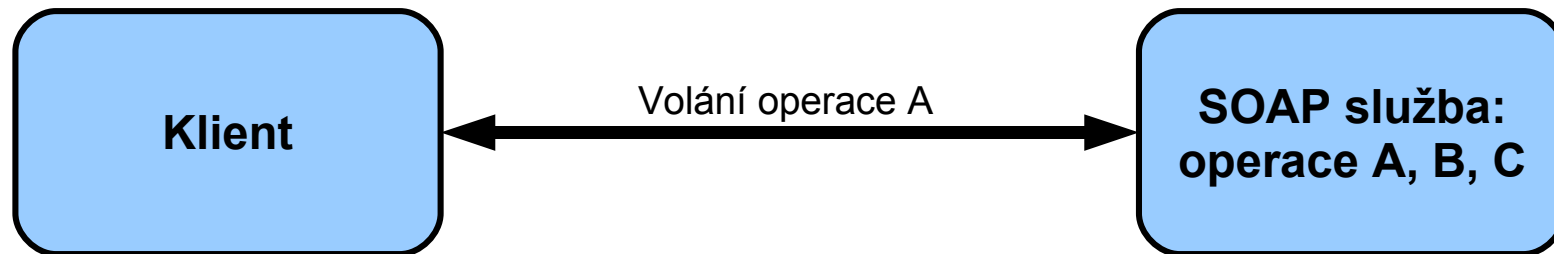
```
<?xml version="1.0"?>
<p:Parts xmlns:p="http://www.parts-depot.com"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <Part id="00345" xlink:href="http://www.parts-depot.com/parts/00345"/>
  <Part id="00346" xlink:href="http://www.parts-depot.com/parts/00346"/>
  <Part id="00347" xlink:href="http://www.parts-depot.com/parts/00347"/>
  <Part id="00348" xlink:href="http://www.parts-depot.com/parts/00348"/>
</p:Parts>
```



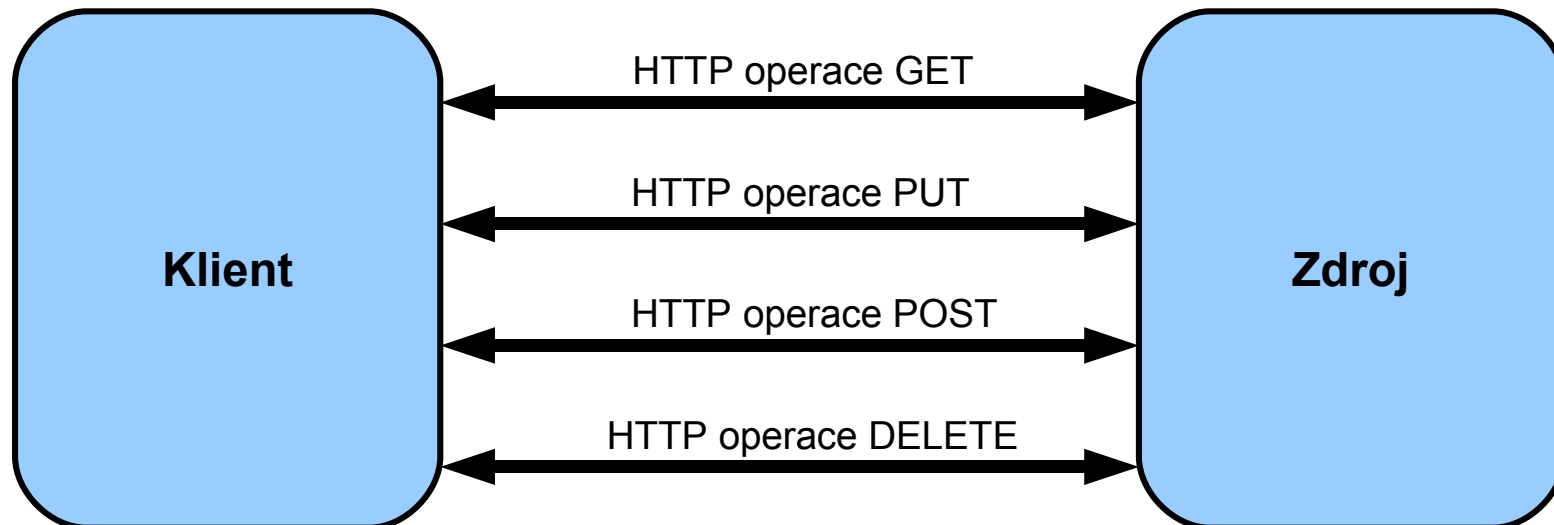
```
<?xml version="1.0"?>
<p:Part xmlns:p="http://www.parts-depot.com"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <Part-ID>00345</Part-ID>
  <Name>Widget-A</Name>
  <Description>This part is used within the frap assembly</Description>
  <Specification xlink:href="http://www.parts-depot.com/parts/00345/specification"/>
  <UnitCost currency="USD">0.10</UnitCost>
  <Quantity>10</Quantity>
</p:Part>
```

SOAP versus REST

SOAP



REST



Co je nutné znát pro vývoj

- XML, včetně jmenných prostorů
- XML Schema – definice XML typů
- WSDL
- SOAP
- Zvolený implementační jazyk
 - Včetně implementace SOAPu, pokud existuje
 - Případně „ruční“ práci se XML, (de)serializaci XML
 - Některé platformy (např. Java, .NET) mají přímo podporu pro WS

Programovací jazyky a WS

- Java
 - Starší API JAX-RPC pro J2EE 1.4
 - Novější API JAX-WS pro Java EE 5
 - Lze použít i samostatné SOAP enginey: [Apache Axis](#)
- .NET – součást platformy
- C/C++
 - Není standardizováno, lze použít např. [gSOAP](#)
- [PHP – ?](#)
- ...

Lze použít v OS/2

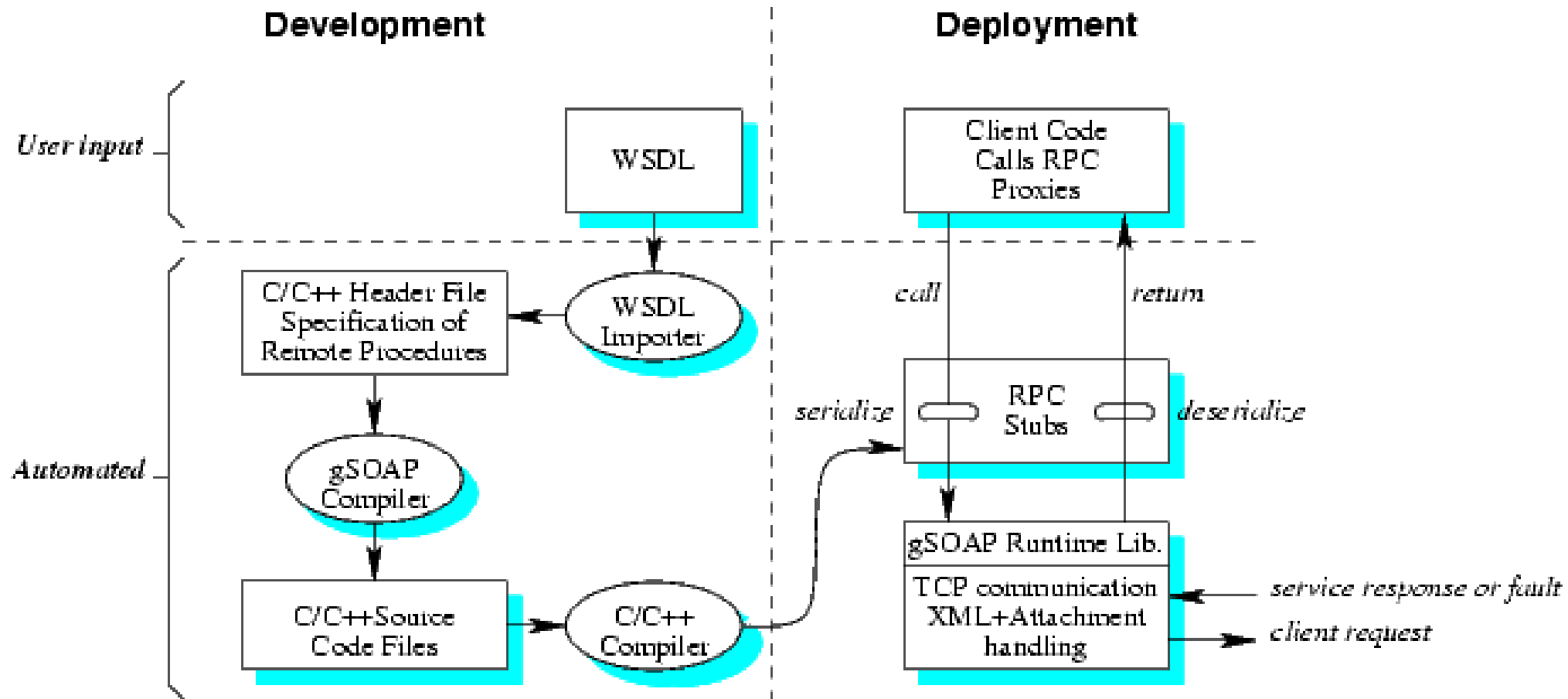
OS/2 + Java

- Nejnovější dostupná verze 1.4 – Golden Code nebo InnoTek
- Webová aplikace běžící v servlet kontejneru:
Apache Tomcat 5.0 nebo 5.5
 - Verze 5.0 je poslední pro Javu 1.4
 - Verze 5.5 vyžaduje pro Javu 1.4 balíček pro zpětnou kompatibilitu
- Buď JAX-RPC anebo Apache Axis

gSOAP – klient (1)

- `wSDL2h -o MojeSluzba.h MojeSluzba.wsdl`
 - Z popisovače vygeneruje hlavičkový soubor
- `soapcpp2 MojeSluzba.h`
 - Vygeneruje stuby (lokální spojky), které se starají o volání vzdálené WS
- Zkompilovat aplikaci s vygenerovanými stuby
- Slinkovat s knihovnou gSOAPu
- Všude lze rozlišovat mezi C a C++
 - Funkce vs. třídy

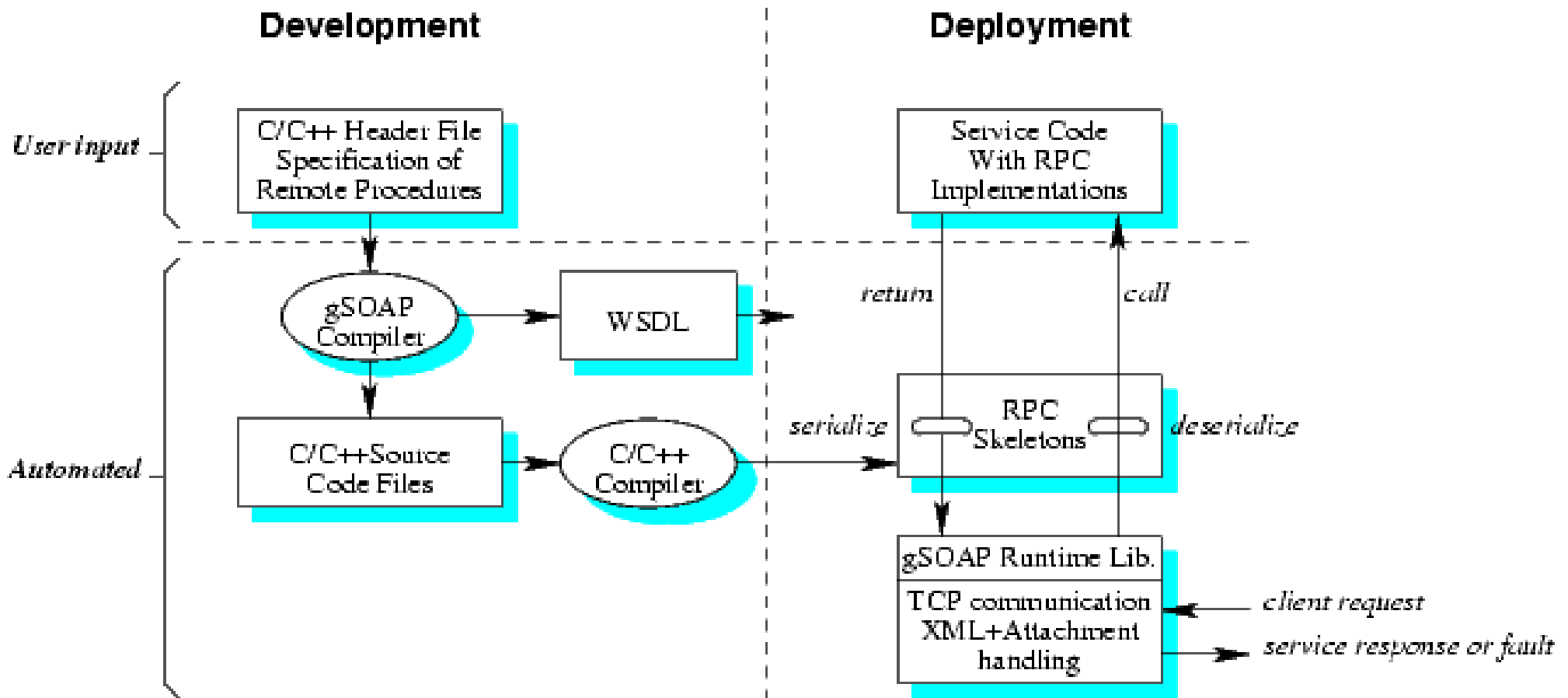
gSOAP – klient (2)



gSOAP – server (1)

- Opět wsdl2h (pokud máme WSDL) anebo ruční vytvoření .h souboru (nemáme WSDL)
- soapcpp2 MojeSluzba.h generuje skeletony
- Implementovat skeletony
- Zkompilovat, slinkovat
- Deployment
 - CGI, nativní aplikace
 - Samostatný HTTP server – gSOAP Server

gSOAP – server (2)



Další užitečné nástroje

- SoapUI – prostředí pro testování SOAP WS
 - Samo umí vytvořit SOAP requesty
 - Validace oproti XSD
 - Podporuje HTTPS – certifikáty apod.
 - Umí i zátěžové testy
 - Zdarma

The screenshot displays the SoapUI interface for a project named 'MathWSPortBinding'. The left sidebar shows a tree view with folders for Schemas, Messages, PortTypes, Bindings, Services, Complex Types, and Global Elements. The main area shows the WSDL content for 'mathservice?WSDL' at 'http://localhost:8080/MathApp/mathservice?WSDL'. The WSDL content includes a header, a generated comment, and a schema definition for 'math:add'. Below the WSDL, a 'Request 1' tab is active, showing a SOAP request in XML format. The request is a 'soapenv:Envelope' with a 'math:add' body containing two numbers, 1 and 2. At the bottom, there are input fields for Username, Password, and Domain, and a button labeled 'Aut'.

```
<!--Published by JAX-WS RI at http://jax-ws.dev.java.n...-->
<!--Generated by JAX-WS RI at http://jax-ws.dev.java.n...-->
<definitions targetNamespace="http://math.demos.dws/...>
  <types>
    <xsd:schema>
      <xsd:import namespace="http://math.demos.dws/...>
    </xsd:schema>
  </types>
```

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:math="http://math.demos.dws/">
  <soapenv:Header/>
  <soapenv:Body>
    <math:add>
      <num1>1</num1>
      <num2>2</num2>
    </math:add>
  </soapenv:Body>
```

Další témata

- Zabezpečení WS, protokol HTTPS
- Standardy WS-*
- Service Oriented Architecture (SOA)
- Business Processes, BPEL
- Enterprise Service Bus (ESB)
- UDDI, ebXML