

---

# SOAPované distribuované systémy

## II.

Róbert Novotný

(UINF/KOPR, 26. november 2013)

# V predošlých dieloch sme videli

---

## Java server

- JAX-WS 2.0
- automaticky vygenerovaný popisovač WSDL



## Java klient

- vygenerovaný z WSDL
- cez SoapUI / wsimport

# WSDL

---

Web  
Services  
Description  
Language

- metadátový popis webovej služby
  - operácie?
  - dátové typy?
  - štýly?
  - bindings?
- XML formát pre čítanie strojom
- strojová a typovo exaktná dokumentácia k službe
- prameň pre **klientov!**

# Návrh webových služieb

---

## Top-Down

*contract-based*

- WSDL => objekty
- bonusovka:
- WSDL => kostra servera + klient

## Bottom-Up

*contract-last*

- objekty + interfejsy s operáciami => WSDL
- WSDL => kostra klienta

# Contract-last! Bottom-up

---

- uvažujeme v objektoch konkrétneho programovacieho jazyka
- WSDL generované automaticky z tried
- expresný bootstrap!
- ale: ohrozený interop!
- ale: nestabilné API => nestabilné WSDL  
=> stále opravovaní klienti

# Contract-first! Top Down!

---

- základ: **ručne** napísané **WSDL**
- navrhujeme operácie
  - vstup: **XML dokument**
  - výstup: **XML dokument**
- prebehnú úvahy nad štruktúrou dát
- zafixuje sa **rozhranie**, menia sa implementácie
- garantovaný **interop**
  - document/literal/wrapped: pre .NET

# XML schéma: XSD

súčasť  
WSDL

- telá správ sú XML
- XML schéma
  - aké elementy/atribúty?
  - v akej hierarchii?
  - aké dátové typy?
- so XSD vieme validovať správy!
- so XSD vieme mapovať schému na triedu!

# Príklad XSD definície

časť schémy!

```
<xs:complexType name="chocolate">
  <xs:sequence>
    <xs:element name="id" type="xs:long" minOccurs="0" />
    <xs:element name="percentage" type="xs:int" />
    <xs:element name="title" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

```
public class Chocolate {
    private int id;
    private int percentage;
    private String title;
}
```



# Encoding rules

---

- ako zareprezentovať objekt v XML?
- aby to bolo interoperabilné?
- aby to dodržiavalo rozumné dátové typy?

5+ štýlov

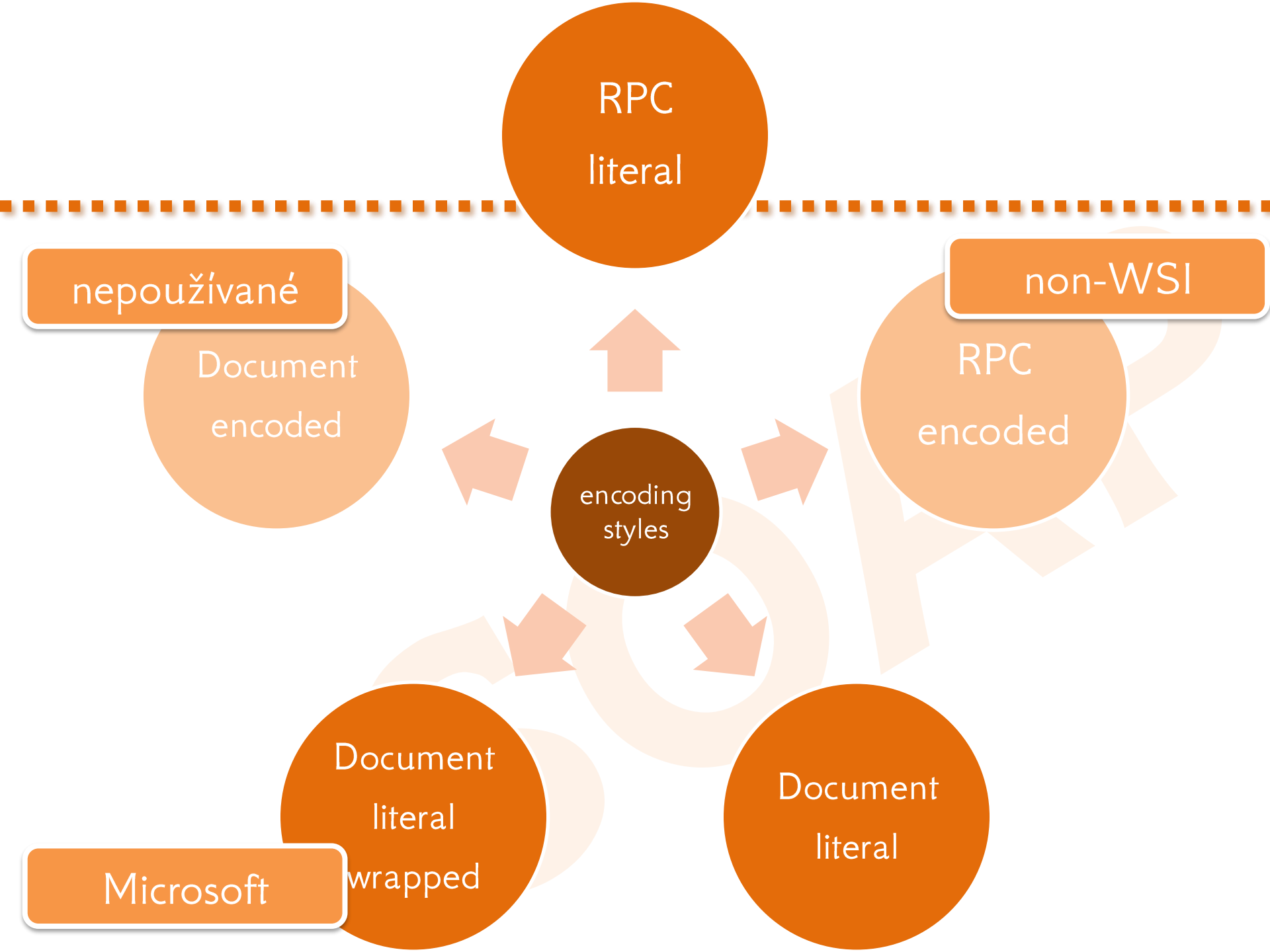
Je jeden svet a predsa sú dva svety...

---

RPC/encoded

tie ostatné

rozdielna reprezentácia  
tiel



RPC  
literal

nepoužívané

Document  
encoded

non-WSI

RPC  
encoded

encoding  
styles

Document  
literal  
wrapped

Microsoft

Document  
literal

# RPC/literal

??? myMethod(??? x, ??? y)

```
<soap:envelope>  
  <soap:body>  
    <myMethod>  
      <x>5</x>  
      <y>5.0</y>  
    </myMethod>  
  </soap:body>  
</soap:envelope>
```

vidím  
názov  
metódy!

zložitejšia  
validácia

??? ???(??? x, ??? y)

# Document/literal

```
<soap:envelope>  
  <soap:body>  
    <xElement>5 </xElement>  
    <yElement>5.0</yElement>  
  </soap:body>  
</soap:envelope>
```

telo je  
dokument

nevidím  
názov  
metódy!

zložité  
WSDL

# RPC vs Document

---

## RPC

- uvažujeme v metódach a ich parametroch
- jednoduchší popis metadát WSDL

## Document

- posielame hore-dole XML dokumenty
- jednoduchšia validácia tiel správ

# Štýly sťahujú interop

---

**Axis**<sup>[Java]</sup>, **SoapServer**<sup>[PHP]</sup>: RPC/encoded

VS

**JAX-WS 2.0**<sup>[Java]</sup>: žiadny RPC/encoded

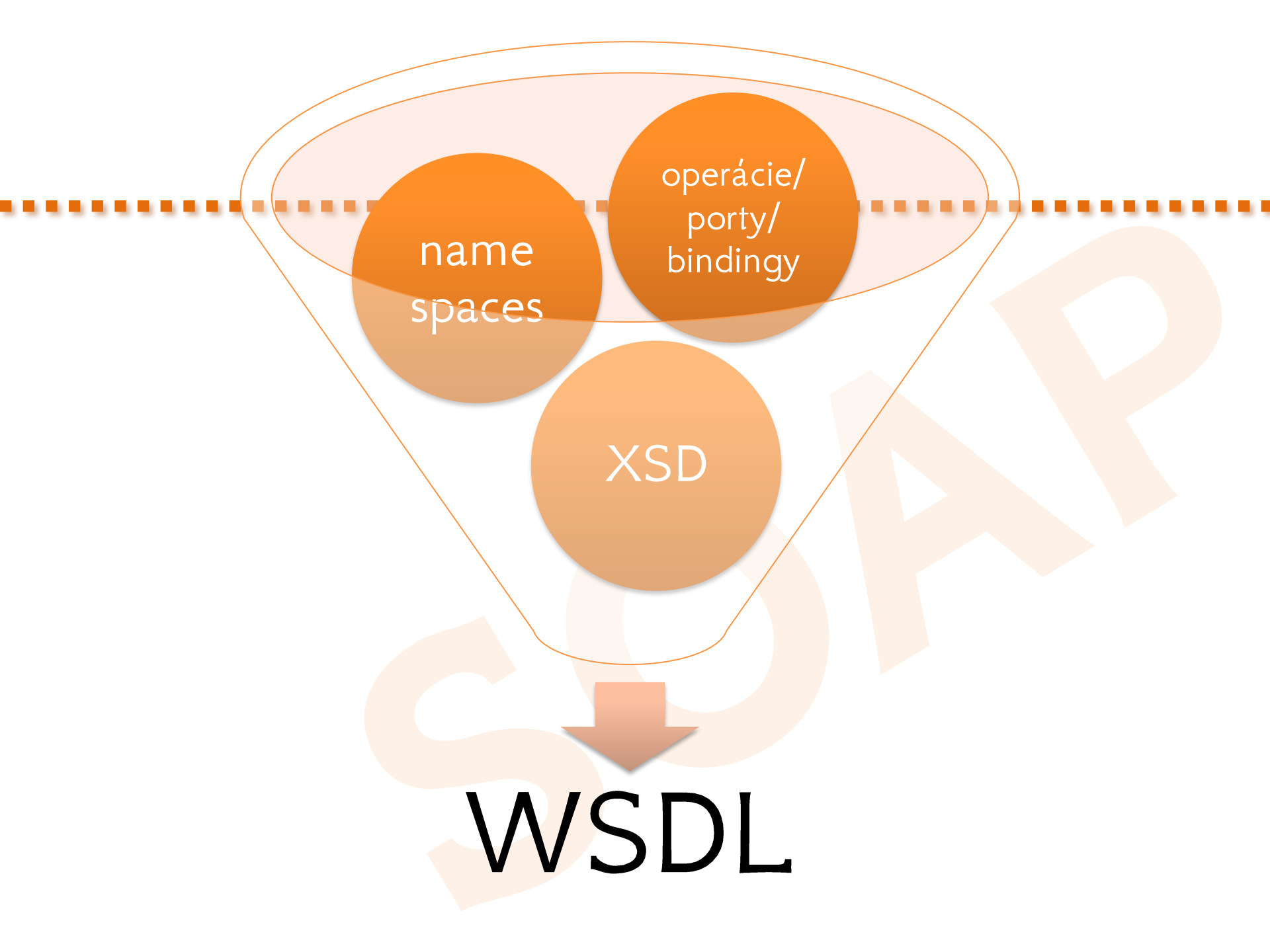
VS

**.NET**: len document/literal/wrapped

---

Vytvorte WSDL pre službu na  
preklad zo slovenčiny do  
angličtiny.





# Ručná tvorba WSDL

---

<http://ics.upjs.sk/~novotnyr/wiki/Java/OdWSDLKWebovejSluzbe>

# Nástroje

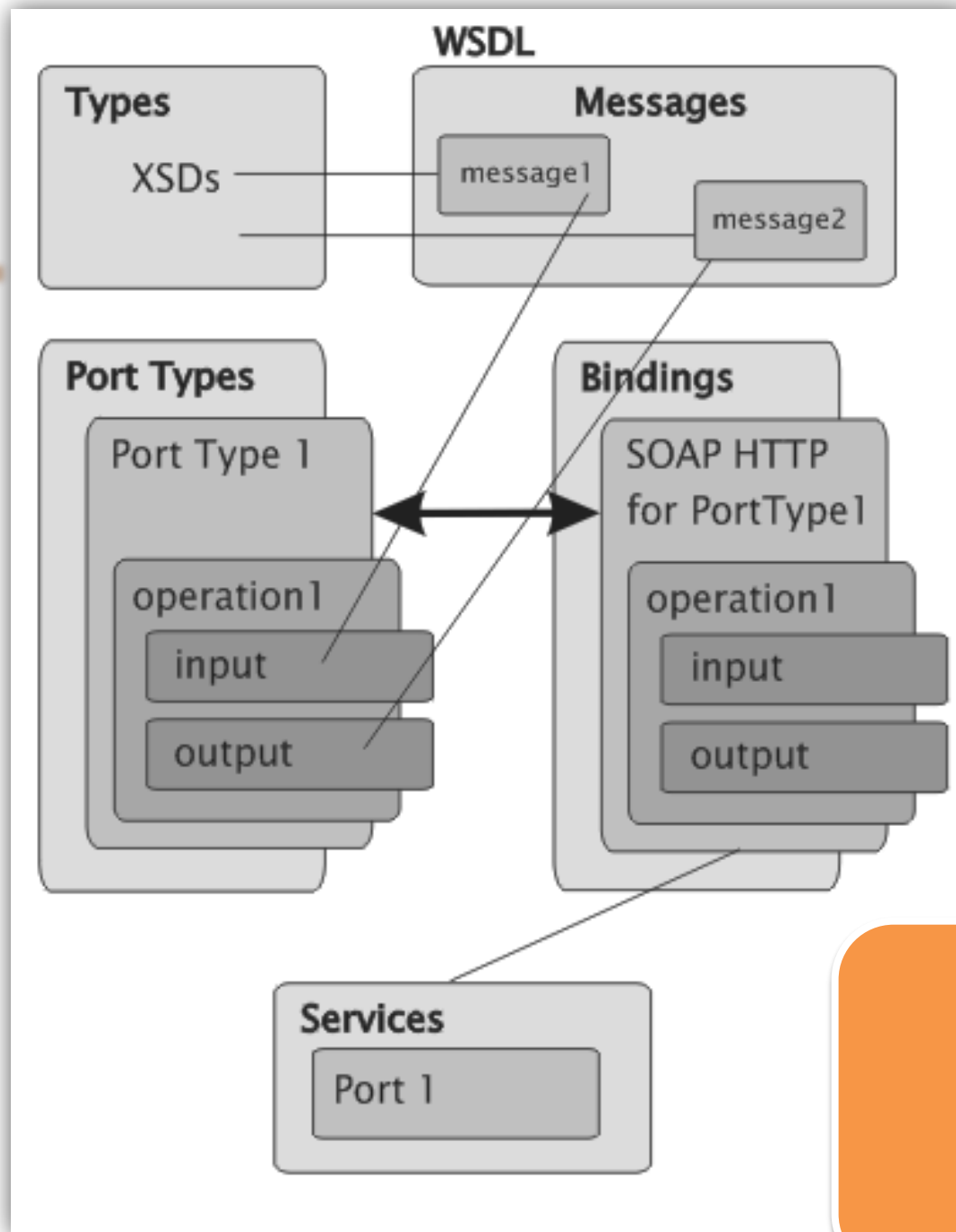
---

- väčšina platená
- našťastie: Eclipse Java Web Developer Tools
- doinštalovať pluginy:

Eclipse Java  
Web Developer  
Tools

JAX-WS

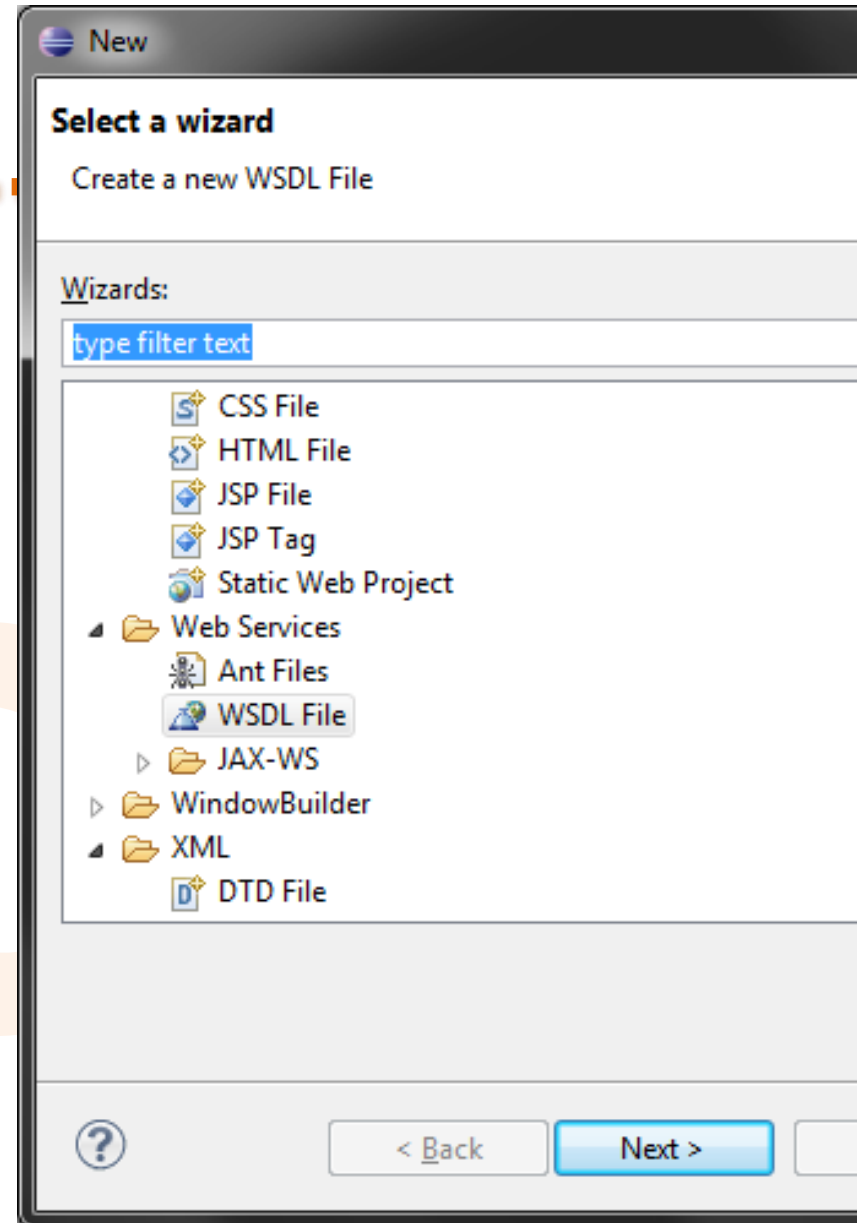


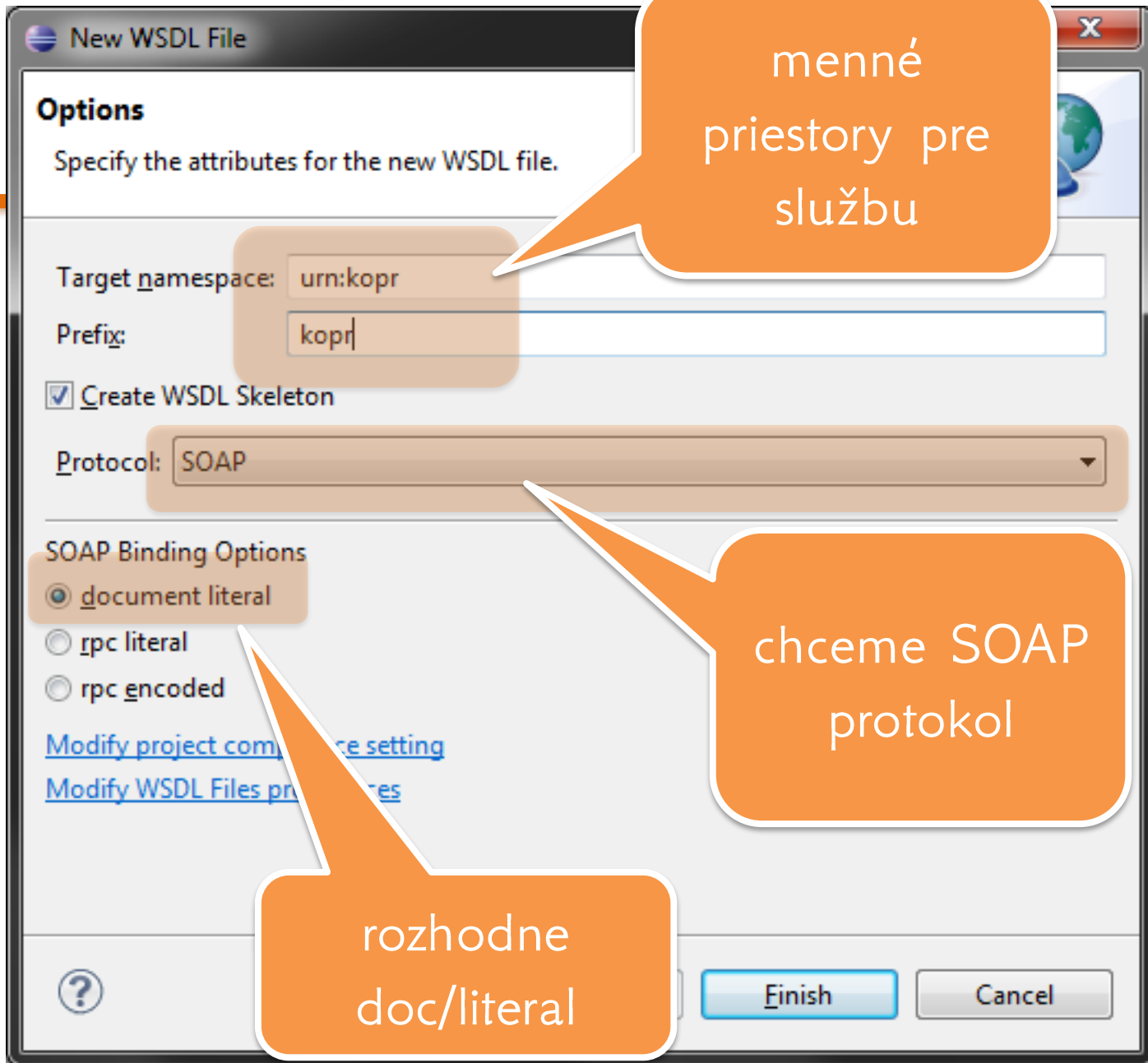


štruktúra  
WSDL

# Postup

- k dispozici je klikací editor + XML editor
- zároveň aj editor XSD schémy
- **File | New | Web-Services | WSDL File**



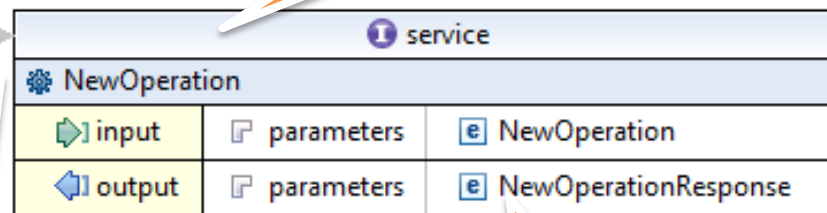
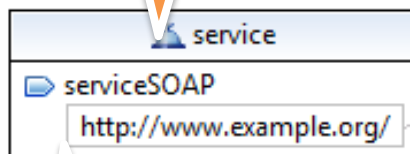


service = sada portov

binding:  
definícia nasadenia

definícia portu

service.wsdl



port ~ Java interface

operácia ~ metóda

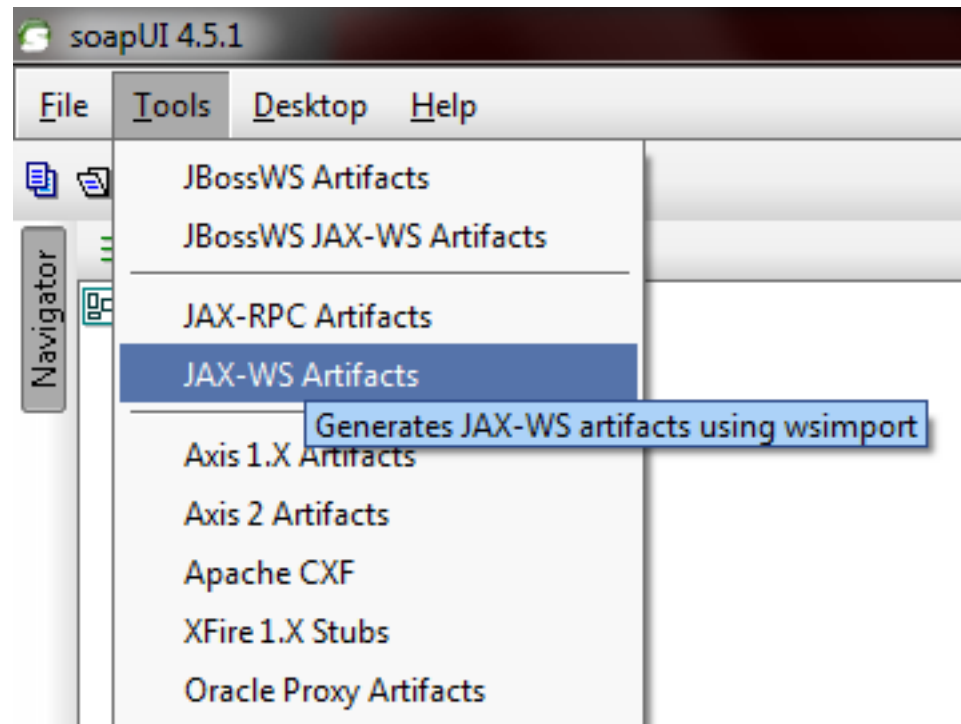
parametre +  
návratové  
hodnoty

---

Vytvorte JAX-WS klienta a server  
na základe WSDL



# SoapUI



# SoapUI Preferences

Preferences

Preferences

SoapUI settings

HTTP Settings	JBossWS wstools:	<input type="text"/>	Browse...
Proxy Settings	JAX-RPC WSCompile:	<input type="text"/>	Browse...
SSL Settings	JAX-WS WSImport:	c:\java\jdk7\bin	Browse...
WSDL Settings	Axis 1.X:	C:\java\axis	Browse...
UI Settings	Axis 2:	<input type="text"/>	Browse...
Monitor Settings	.NET 2.0 wsdl.exe:	<input type="text"/>	Browse...
Tools	XFire 1.X:	<input type="text"/>	Browse...
WS-I Settings	CXF 2.X:	<input type="text"/>	Browse...
Global Properties	ANT 1.6+:	<input type="text"/>	Browse...
Security Settings	GSoap:	<input type="text"/>	Browse...
WS-A Settings	JAXB xjc:	<input type="text"/>	Browse...
UI Settings			
Recording Settings			
Save Information Tokens			
Update Settings			



### JAX-WS Artifacts



Specify arguments for JWSDP/JAX-WS wsimport

Basic

Custom Args

WSDL:

Target Directory:

Package:

Source Directory:

HTTP Proxy:

Catalog:

Binding files:

wsdlLocation:

Keep:  (keep generated files)



Generate

Close

Tools

# Výsledok generovania

- vypadnú triedy pre klienta
- + interfejs pre webovú službu
- stačí doimplementovať!

vygenerovaná  
interfejs s  
metódami

```
@WebService(  
    endpointInterface="sk.novotnyr.movie.wsdl.MovieReservationPortType",  
)  
public class DefaultMovieReservationService {  
    ...  
}
```

# Komplexný návod pre server i klienta

---

<http://ics.upjs.sk/~novotnyr/wiki/Java/OdWSDLKWebovejSluzbe-Metro>