

Računarstvo zasnovano na uslugama

<http://www.fer.hr/predmeti/rznu>

Prof.dr.sc Siniša Srbljić

Dr.sc. Ivan Benc

**Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu**

Predavanje #5

WS-*

izgradnja web usluga

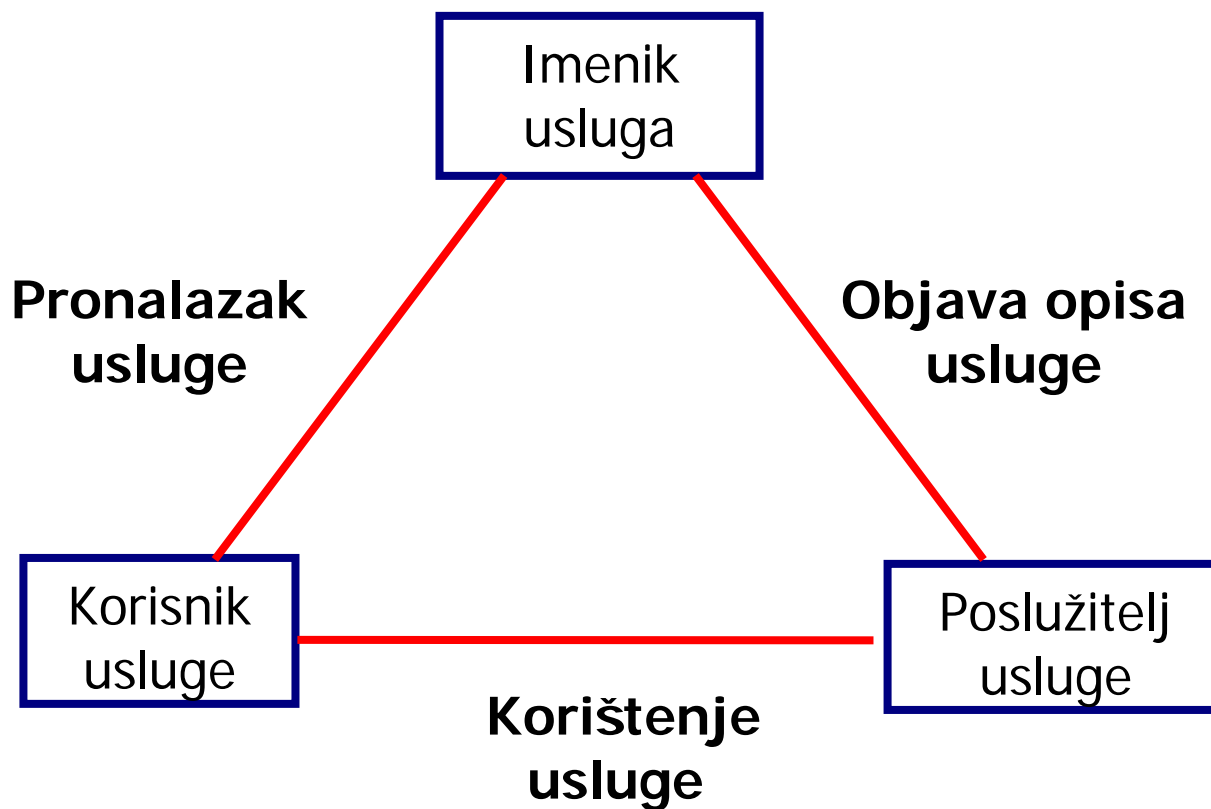


Plan predavanja

- Uvod
- XML Schema
- SOAP
- WSDL
- (WS-Addressing)

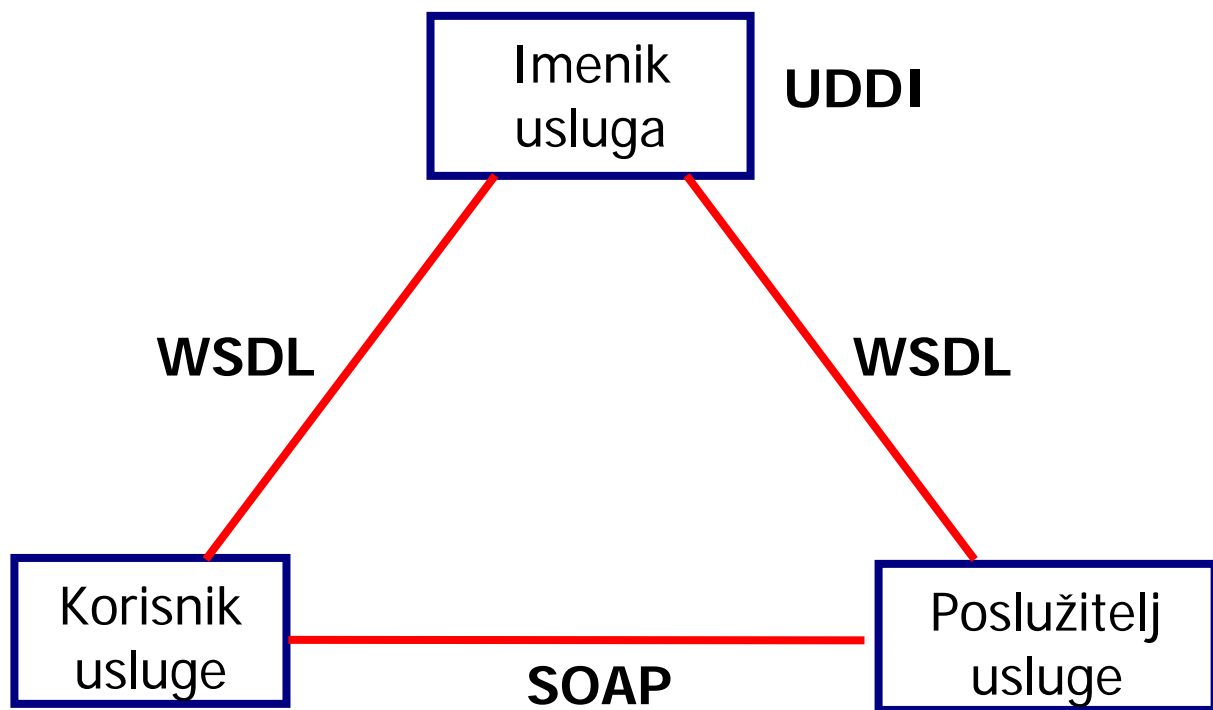
Osnovna arhitektura

(općenito)



Osnovna arhitektura

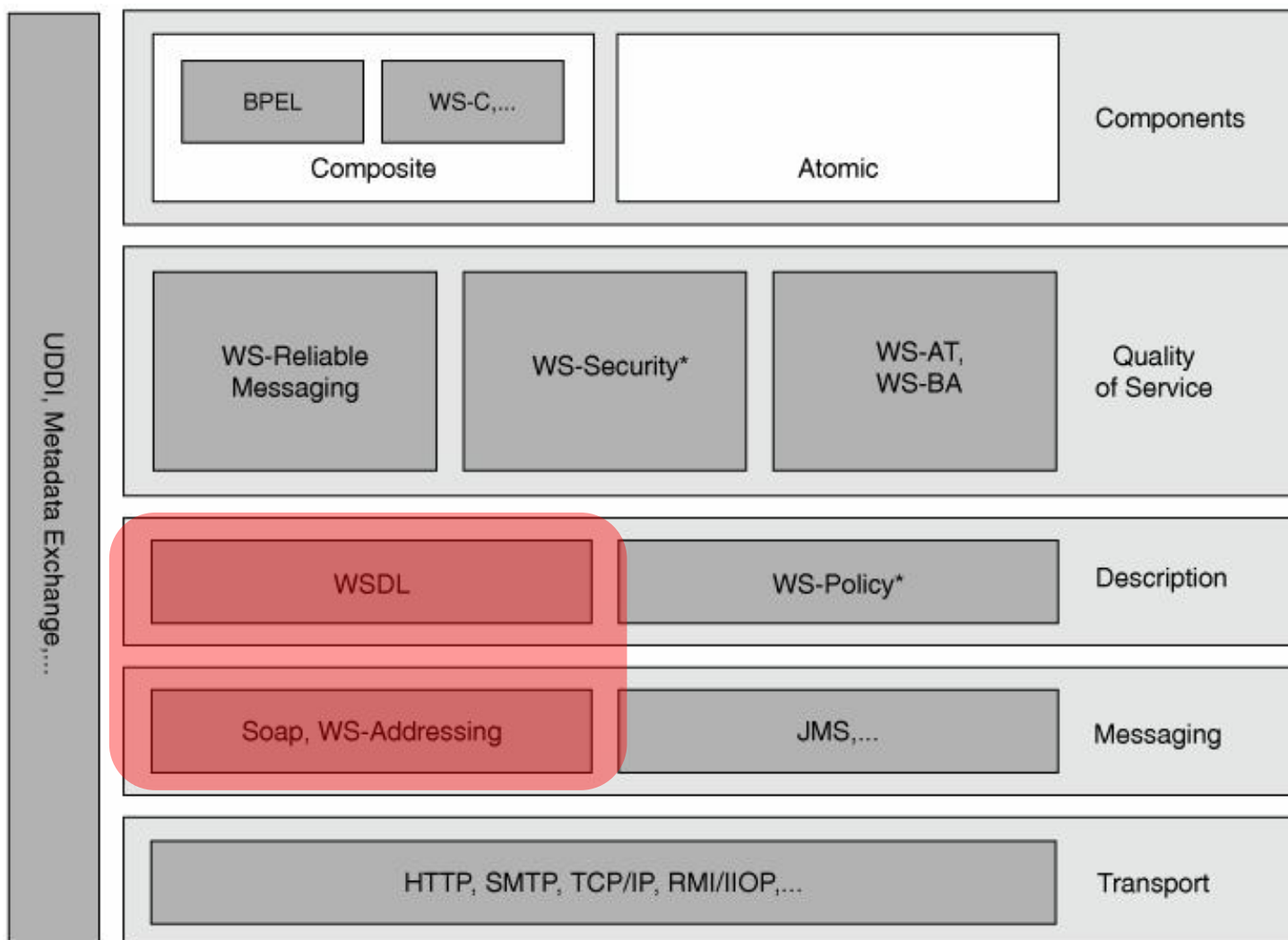
(Web Services, WS-*)



Što je WS-*

- Skup specifikacija koje se odnose na web usluge
- Ime započinje s “**WS-**”
 - WS-Addressing, WS-Transfer, WS-Notification, WS-Trust ...
- Velika lista specifikacija
 - nekompletna
 - specifikacije komplementarne
 - neformalna
- http://en.wikipedia.org/wiki/List_of_Web_service_specifications

Web Services stog



Radni okvir za razmjenu poruka

- Najniži dijelovi Web Services stoga
- Osnovni standardi i tehnologije za razmjenu poruka
 - XML
 - SOAP
 - WS-Addressing

Radni okvir za razmjenu poruka

- XML & XML Schema
 - metajezik za definiranje drugih jezika
 - neovisan o platformi i zasnovan na Unicode standardu
 - podržava ga gotovo svaki proizvođač programske opreme
 - de facto format za razmjenu podataka
 - XML Schema za definiranje strukture i sadržaja XML dokumenata

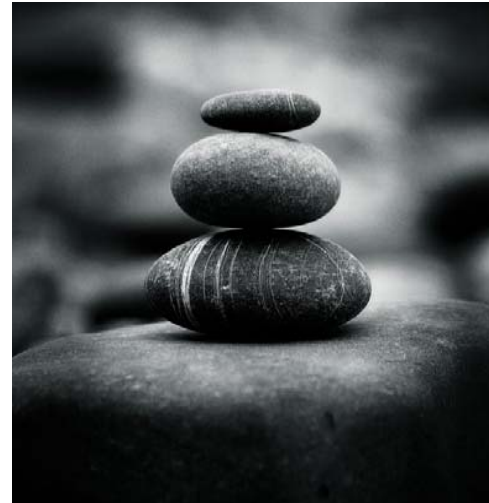
Radni okvir za razmjenu poruka

- SOAP
 - standardan način *zapisa* poruka
 - razmjena strukturiranih i tipiziranih informacija između usluga
 - neovisna o pozadinskoj tehnologiji razmjene podataka - SOAP poruka je XML dokument
 - posljednja verzija: SOAP 1.2 (W3C)

Radni okvir za razmjenu poruka

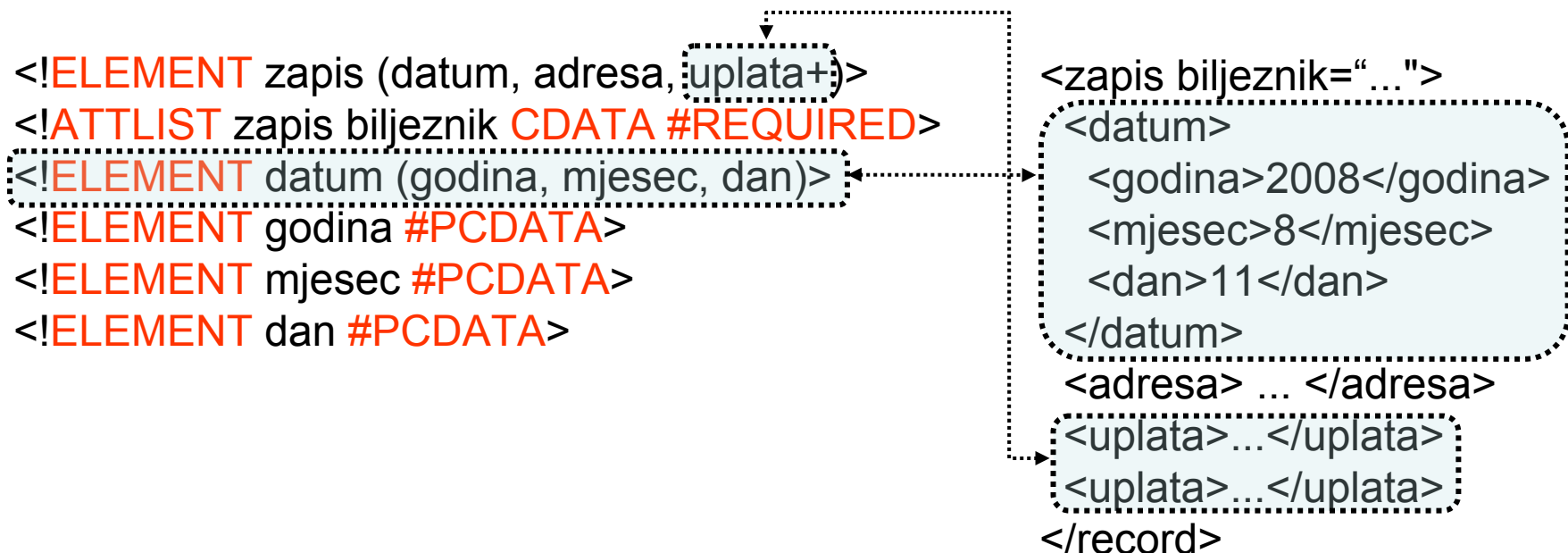
- WS-Addressing
 - ostvaruje identifikaciju pošiljatelja i primatelja uključenih u razmjenu poruka
 - osigurava identifikaciju krajnjih točaka usluga
 - interoperabilna tehnologija neovisna o prijenosnom sloju

1 XML Schema



XML Schema

- **DTD: Document Type Definition**
 - definiranje strukture XML dokumenta



XML Schema

- Definiranje strukture
 - jednostavni primjer XML Schema dokumenta

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:element name="visina"/>  
</xs:schema>
```

- odgovarajuća *XML instance*

```
<visina>180</visina>
```

- kako odrediti sadržaj elementa?
 - 180, 50.897, 90/60/90, veliko, srednje ...

XML Schema

- Ugrađeni tipovi podataka
 - jednostavni tipovi
 - xs:type
 - xs:string, xs:integer, xs:float, xs:decimal, xs:boolean, xs:int (32bit integer)
 - xs:time, xs:date, xs:dateTime
 - xs:token, xs:ID, xs:IDREF, xs:IDREFS

XML Schema

- Ugrađeni tipovi podataka
 - xs:string
 - nisu dozvoljeni znakovi bitni za XML jezik
 - koristi se poseban zapis za specijalne znakove
 - *character references*
 - “ - £quot
 - ‘ - £apos
 - & - £amp
 - < - £lt
 - > - £gt
 - CDATA (<![CDATA[...]]>)

XML Schema

- simpleType
 - tip *simpleType* nasljeđuje od osnovnih tipova (*baseType*)
 - *baseType*
 - string, integer, date ...
 - moguće definirati ograničenja
 - length, minLength, maxLength
 - pattern, enumeration, whitespace
 - minInclusive, maxInclusive, minExclusive, maxExclusive

XML Schema

- simpleType
 - primjer

```
<simpleType name="Postotak">  
  <restriction base="integer">  
    <minInclusive value="0">  
    <maxInclusive value="100">  
  </restriction>  
</simpleType>
```

```
<simpleType name="VipTel">  
  <restriction base="string">  
    <pattern value="091-dddd-ddd">  
  </restriction>  
</simpleType>
```

XML Schema

- simpleType

- lista

```
<simpleType name="ListaPrimjer">  
  <list base="xs:integer">  
</simpleType>
```

- primjer

```
<xs:element name="primjer" type="ListaPrimjer">  
<primjer>1 2 3</primjer>
```

XML Schema

- Složeni tipovi
 - *complexType* element
 - sadrži deklaracije svih elemenata i atributa koji se mogu pojaviti unutar definiranog elementa

```
<xs:element name="Narudzba">
  <xs:complexType>
    <xs:element name="Artikl" type="tns:ArtiklID"/>
    <xs:element name="Kolicina" type="int"/>
  </xs:complexType>
</xs:element>
```

XML Schema

- Složeni tipovi
 - *complexType* nasljeđivanje
 - *restriction*

```
<complexType name="Obitelj">  
  <element name="Roditelj" minOccurs="1" maxOccurs="2"/>  
  <element name="Dijete" type="string" minOccurs="0"/>  
</complexType>
```

```
<complexType name="SamohraniRoditelj">
```

```
  <complexContent>
```

```
    <restriction base="Obitelj">
```

```
      <element name="Roditelj" type="String" minOccurs="1" maxOccurs="1"/>
```

```
      <element name="Dijete" type="string" minOccurs="0"/>
```

```
    </restriction>
```

```
  </complexContent>
```

```
</complexType>
```

XML Schema

- Složeni tipovi

- *complexType* nasljeđivanje

- *extension*

```
<complexType name="cijeliBroj">  
  <simpleContent>  
    <extension base="int">  
      <attribute name="id" type="ID"/>  
      <attribute name="href" type="uri:reference"/>  
    </extension>  
  </simpleContent>  
</complexType>
```

- *simpleContent* element

- ne sadržava druge elemente

XML Schema

- Složeni tipovi
 - referenciranje

```
<element name="datum">
```

```
  <complexType>
```

```
    <all>
```

```
      <element ref="godina"/>
```

```
      <element ref="mjesec"/>
```

```
      <element ref="dan"/>
```

```
    </all>
```

```
  </complexType>
```

```
</element>
```

```
<element name="godina" type="Integer"/>
```

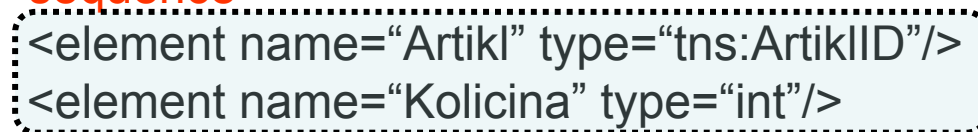
```
<element name="mjesec" type="Integer"/>
```

```
<element name="dan" type="Integer"/>
```

XML Schema

- Složeni tipovi
 - osiguravanje ispravnog redoslijeda elemenata
 - *sequence* element

```
<element name="Narudzba">  
  <complexType>  
    <sequence>  
      <element name="Artikl" type="tns:ArtiklID"/>  
      <element name="Kolicina" type="int"/>  
    </sequence>  
  </complexType>  
</element>
```



– *choice*, *all* elementi

XML Schema

- Složeni tipovi
 - grupe atributa
 - *attributeGroup* element

```
<attributeGroup name="commonAttributes">  
  <attribute name="id" type="ID"/>  
  <attribute name="href" type="uri:Reference"/>  
</attributeGroup>
```

```
<element name="stringId">  
  <complexType>  
    <extension base="string">  
      <attributeGroup ref="tns:commonAttributes"/>  
    </extension>  
  </complexType>  
</element>
```

XML Schema

- Prostori imena (namespaces)
 - logičke granice entiteta definiranih shemama
 - *targetNamespace* atribut definira identifikator prostora imena
 - vrijednost atributa jest URI
 - URL (<http://primjer.org>)
 - URN (<urn:WebService:primjer>)
 - entiteti koji mogu pripadati prostoru imena su tipovi podataka, elementi i atributi

XML Schema

- Prostori imena (namespaces)
 - primjer

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:Primjer">
  <xs:element name="visina"/>
  ...
</xs:schema>
```

- svi elementi definirani unutar sheme pripadaju prostoru imena Primjer

XML Schema

- Prostori imena (namespaces)

- *xmlns* atribut

- koristi se za referenciranje shema iz dokumenata instanci

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:element name="visina"/>  
</xs:schema>
```

- moguće referenciranje više shema korištenjem više xmlns atributa
 - referenciranje entiteta definiranih unutar istog dokumenta

```
<element name="Envelope" type="tns:Envelope"/>
```

XML Schema

- Prostori imena (namespaces)
 - *schemaLocation* atribut
 - referenca na prostor imena (URI) nije valjana kao URL
 - potreban mehanizam pronalaska stvarnog shema dokumenta skrivenog iza identifikatora prostora imena

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
xs:schemaLocation="http://www.w3.org/2001/XMLSchema.xsd">  
  ...  
</xs:schema>
```

2SOAP



SOAP specifikacija

- Simple Object Access Protocol
- SOAP 1.1 – travanj 2000
 - Microsoft, Developmentor, Userland & IBM
- SOAP 1.2 (Recommendation) - lipanj 2003
 - W3C
- Rasprave oko imena (SOA Protocol ...)
 - Rezultat: SOAP više nije kratica

SOAP specifikacija

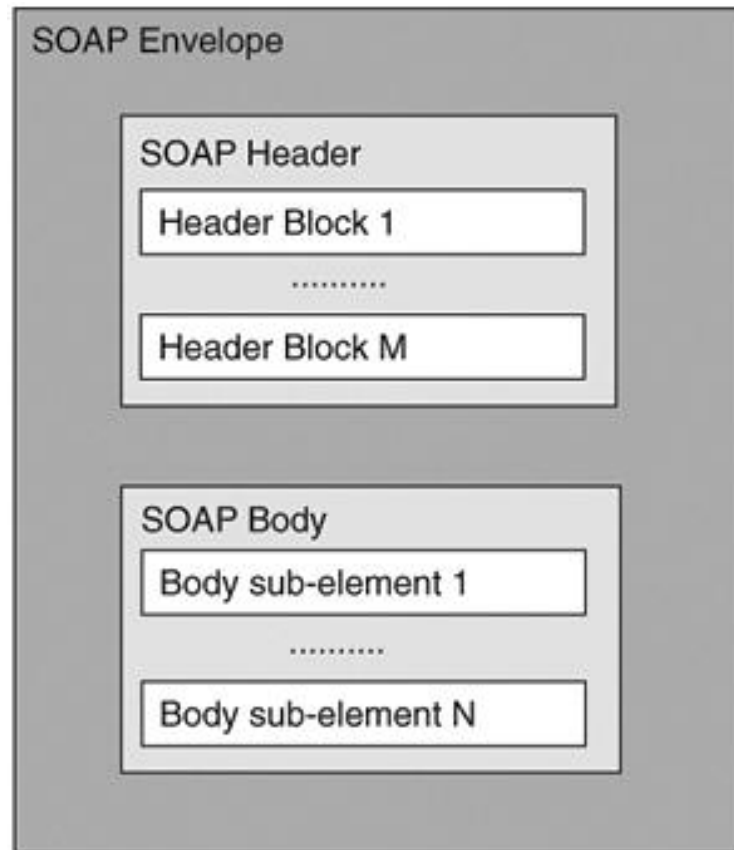
- Što je SOAP
 - komunikacijski protokol
 - format za razmjenu poruka
 - zasnovan na jeziku XML
 - neovisan o jeziku i platformi

SOAP specifikacija

- SOAP poruka
 - osnovna komunikacijska jedinica među SOAP *čvorovima*
 - sastoji se od SOAP *omotnice* koja može imati više blokova SOAP *zaglavlja*
 - SOAP omotnica *mora* imati SOAP *tijelo* koje sadrži skup korisničkih informacija

SOAP specifikacija

- SOAP poruka

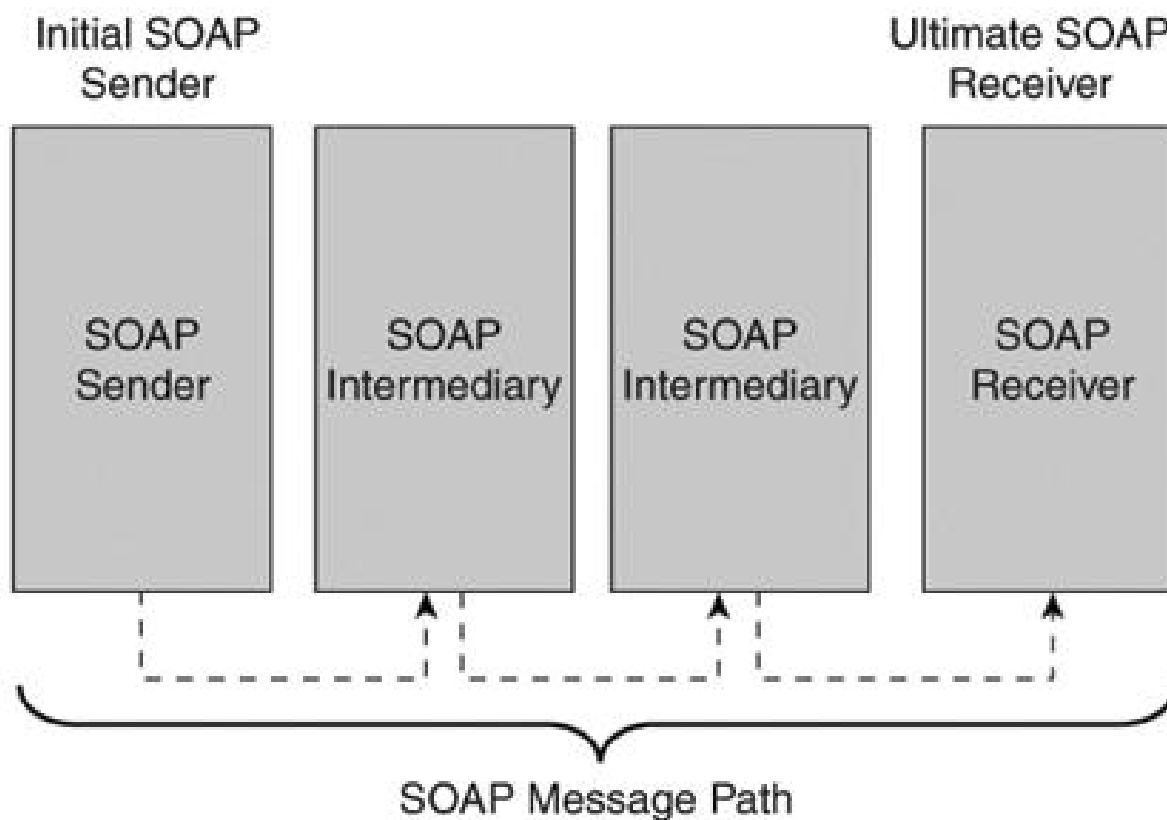


SOAP specifikacija

- SOAP čvorovi
 - šalju i primaju SOAP poruke
 - SOAP poruke prolaze kroz više čvorova
 - SOAP *Pošiljatelji* - šalju poruke
 - SOAP *Primatelji* - primaju poruke
 - čvorovi koji šalju i primaju poruke nazivaju se SOAP *Posrednici*

SOAP specifikacija

- SOAP čvorovi



SOAP specifikacija

- SOAP model obrade poruka
 - definiranje više blokova zaglavlja
 - blok zaglavlja predstavlja specifikaciju obrade
 - npr. kriptiranje sadržaja poruka, pouzdanu razmjenu poruka, sažimanje poruka, itd.
 - sadržaj tijela SOAP omotnice je obično namijenjen samo krajnjem primatelju

SOAP specifikacija

- SOAP usmjeravanje (WS-Routing)
 - SOAP standard ne specificira način usmjeravanja poruka
 - rješenje: dodati standardni blok za usmjeravanje unutar zaglavlja
 - definira točan slijed čvorova kroz koje prolazi poruka

SOAP specifikacija

- SOAP uloge
 - odnosi se na uloge čvorova tijekom obradbe poruke
 - **role** atribut
 - označene s URI
 - Npr. <http://primjer.org/QosUloga>
 - ili definirane SOAP specifikacijom
 - **none** – zaglavlje se ne obrađuje
 - **next** – svaki čvor mora biti u mogućnosti obraditi
 - **ultimateReceiver**

SOAP specifikacija

- SOAP uloge
 - obavezni blok zaglavlja
 - **mustUnderstand**="true"
 - pretpostavljeno ponašanje čvora jest uklanjanje bloka zaglavlja nakon obradbe
 - ako ne želimo takvo ponašanje, onda je potrebno postaviti atribut: **relay**="true"
 - mustUnderstand nadjačava relay

SOAP specifikacija

- SOAP dojava greške
 - nakon pojave greške tijekom obradbe
 - dojava greške nalazi se u tijelu SOAP poruke unutar *fault* elementa
 - *fault* element
 - *code* element
 - VersionMismatch, MustUnderstand, DataEncodingUnknown, Sender, Receiver
 - *reason* element - opis
 - proizvoljno: *detail*, *node* i *role* elementi

SOAP specifikacija

- SOAP prilozi
 - za slanje većih količina binarnih podataka
 - Za manje količine koristi se Base64
 - rješenje: XML-binary Optimized Packaging (**XOP**)
 - MIME paket - dokumenti s više dijelova
 - Base64 enkodirani podaci zamijenjeni su s **xop:Include** elementima koji pokazuju na dijelove MIME paketa sa binarnim podacima

SOAP specifikacija

- SOAP programski modeli: **EDI**
 - **E**lectronic **D**ocument **I**nterchange
 - zasnovan na dokumentima
 - osnova za automatiziranje poslovnih transakcija
 - SOAP poruka sadržava poslovni dokument
 - rezervacija
 - narudžba
 - potvrda

SOAP specifikacija

- SOAP programski modeli: **RPC**
 - Remote **P**rocedure **C**all
 - osnova raspodijeljenog računarstva
 - zahtjev sadržava ime procedure i skup parametara
 - odgovor sadržava povratnu vrijednost pozvane udaljene procedure

SOAP specifikacija

- SOAP programski modeli: **RPC**
 - koriste se procedure napisane u nekom jeziku

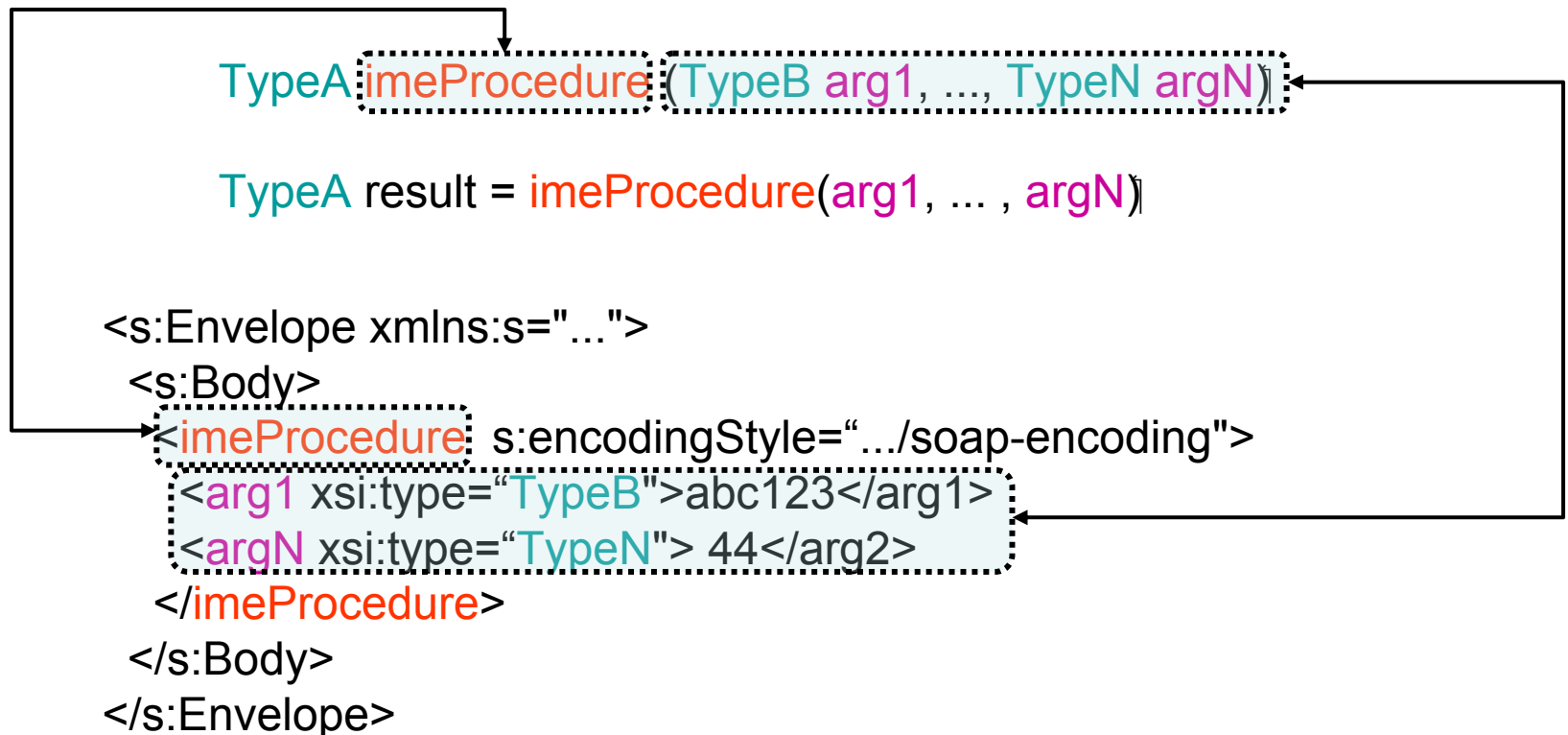
TypeA imeProcedure(TypeB arg1, ..., TypeN argN)

TypeA result = imeProcedure(arg1, ... , argN)

- poziv procedure zapisuje se u SOAP zahtjev
 - svaki parametar je zapisan kao XML element
 - imena i poredak parametara moraju biti isti u potpisu procedure i XML zapisu

SOAP specifikacija

- SOAP programski modeli: **RPC**
 - Primjer: RPC SOAP zahtjev



SOAP specifikacija

- SOAP programski modeli: **RPC**
 - Primjer: RPC SOAP odgovor

```
<s:Envelope xmlns:s="...">  
  <s:Body>  
    <imeProcedureResponse s:encodingStyle=".../soap-encoding">  
      <return xsi:type="xsd:TypeA">new</return>  
    </imeProcedureResponse>  
  </SOAP:Body>  
</SOAP:Envelope>
```

SOAP specifikacija

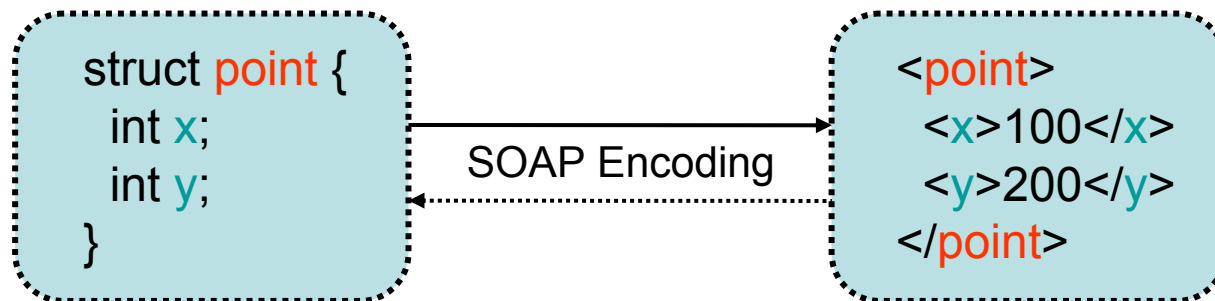
- Zapisivanje aplikacijskih podataka u XML
 - **SOAP Encoding** – serijalizacija podataka
 - jednostavan sustav tipova
 - generalizacija sustava tipova u programskim jezicima
 - preporuka – nije obavezno korištenje

SOAP specifikacija

- SOAP Encoding
 - jednostavni tipovi podataka
 - string, integer, boolean, date/time itd.
 - prema *XML Schema* specifikaciji
 - npr. <tezina>70</tezina>
 - složeni tipovi podataka
 - strukture, polja itd.

SOAP specifikacija

- SOAP Encoding
 - složeni tipovi podataka
 - strukture
 - logički grupirani podaci



SOAP specifikacija

- SOAP Encoding
 - složeni tipovi podataka
 - Polja
 - soap-enc:arrayType atribut

```
<soap:Body>  
  <someArray>  
    <a soap-enc:arrayType="xsi:int[3]">  
      <int>1</int>  
      <int>2</int>  
      <int>3</int>  
    </a>  
  </someArray>  
</soap:Body>
```

A diagram illustrating the mapping from the XML `soap-enc:arrayType="xsi:int[3]"` attribute to a table structure. A dashed arrow points from the attribute value to the top of a table. Another dashed arrow points from the first row of the table back to the first integer element in the XML snippet.

Array
1
2
3

SOAP specifikacija

- SOAP Encoding
 - složeni tipovi podataka
 - Polja
 - polja koja sadrže bilo kakve tipove

```
<soap:Body>  
  <someArray>  
    <a soap-enc:arrayType="xsi:ur-type[3]">  
      <object>10.12</object>  
      <object>niz znakova</object>  
      <object>20</object>  
    </a>  
  </someArray>  
</soap:Body>
```



SOAP specifikacija

- SOAP Encoding
 - složeni tipovi podataka
 - Polja
 - višedimenzionalna polja

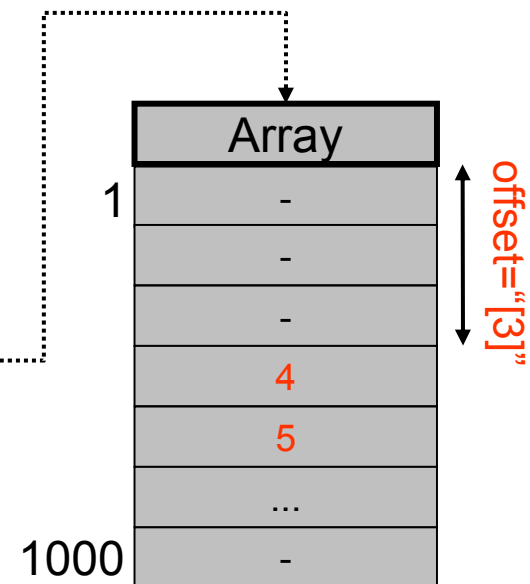
```
<soap:Body>  
  <someArray>  
    <a soap-enc:arrayType="xsi:string[2,3]">  
      <string>red 1, stupac 1</string>  
      <string>red 1, stupac 2</string>  
      <string>red 2, stupac 1</string>  
      <string>red 2, stupac 2</string>  
    </a>  
  </someArray>  
</soap:Body>
```

Array	
red 1, stupac 1	red 1, stupac 2
red 2, stupac 1	red 2, stupac 2

SOAP specifikacija

- SOAP Encoding
 - složeni tipovi podataka
 - Polja
 - parcijalna polja

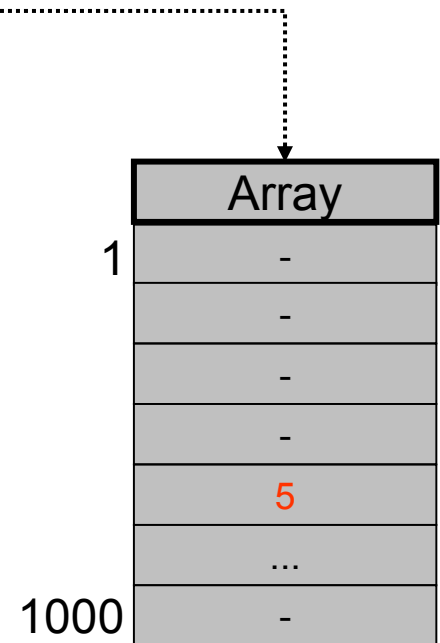
```
<soap:Body>  
  <someArray>  
    <a soap-enc:arrayType="xsi:string[1000]"  
      soap-enc:offset="[3]">  
      <string>4</string>  
      <string>5</string>  
    </a>  
  </someArray>  
</soap:Body>
```



SOAP specifikacija

- SOAP Encoding
 - složeni tipovi podataka
 - Polja
 - rijetka polja

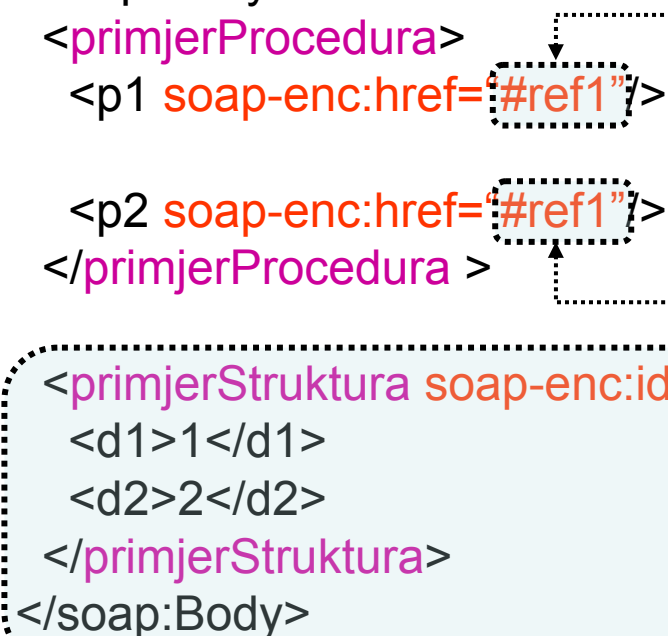
```
<soap:Body>  
  <someArray>  
    <a soap-enc:arrayType="xsi:string[1000]">  
      <string soap-enc:position="[5]">pet</string>  
    </a>  
  </someArray>  
</soap:Body>
```



SOAP specifikacija

- SOAP Encoding
 - poziv preko reference
 - .NET primjer

```
<soap:Body>  
  <primjerProcedura>  
    <p1 soap-enc:href="#ref1"/>  
  
    <p2 soap-enc:href="#ref1"/>  
  </primjerProcedura>  
  
  <primjerStruktura soap-enc:id="ref1">  
    <d1>1</d1>  
    <d2>2</d2>  
  </primjerStruktura>  
</soap:Body>
```



SOAP specifikacija

- Veza na prijenosni sloj (**SOAP binding**)
 - serijalizacija SOAP poruke u prijenosni sloj
 - rekonstrukcija bez gubitka informacija
 - moguće korištenje više mrežnih protokola
 - HTTP, SMTP, vlasnički protokoli, itd.

SOAP specifikacija

- Veza na prijenosni sloj (SOAP binding)
 - **HTTP** najpopularniji protokol
 - pogodan zbog sigurnosne stijene (firewall)
 - port 80
 - razvijena robusna infrastruktura
 - internet
 - komunikacija bez čuvanja stanja (stateless)
 - prirodan protokol za RPC
 - jednostavan
 - otvoren

SOAP specifikacija

- Veza na prijenosni sloj (SOAP binding)
 - primjer: SOAP preko HTTP zahtjev

POST /InStock **HTTP/1.1**

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

```
<?xml version="1.0"?>
```

```
<soap:Envelope
```

```
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
```

```
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

```
  <soap:Body xmlns:m="http://www.example.org/stock">
```

```
    <m:GetStockPrice>
```

```
      <m:StockName>IBM</m:StockName>
```

```
    </m:GetStockPrice>
```

```
  </soap:Body>
```

```
</soap:Envelope>
```

SOAP specifikacija

- Veza na prijenosni sloj (SOAP binding)
 - primjer: SOAP preko HTTP odgovor

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

```
<?xml version="1.0"?>
```

```
<soap:Envelope
```

```
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
```

```
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

```
  <soap:Body xmlns:m="http://www.example.org/stock">
```

```
    <m:GetStockPriceResponse>
```

```
      <m:Price>34.5</m:Price>
```

```
    </m:GetStockPriceResponse>
```

```
  </soap:Body>
```

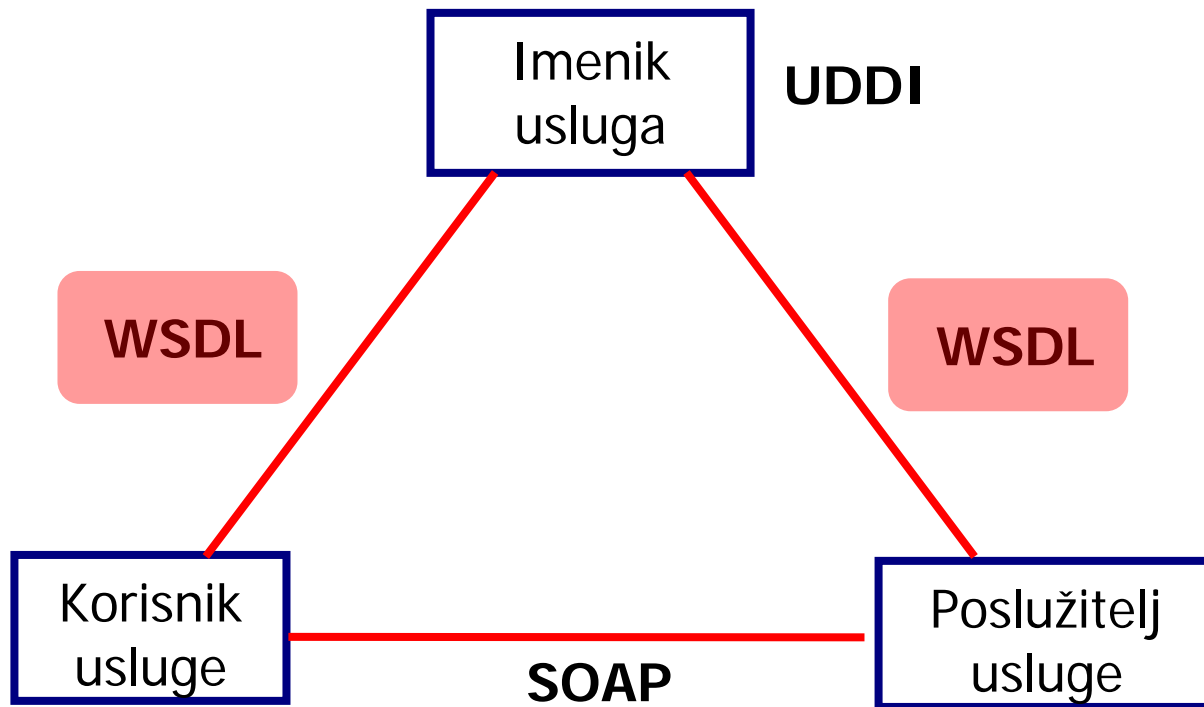
```
</soap:Envelope>
```

3WSDL

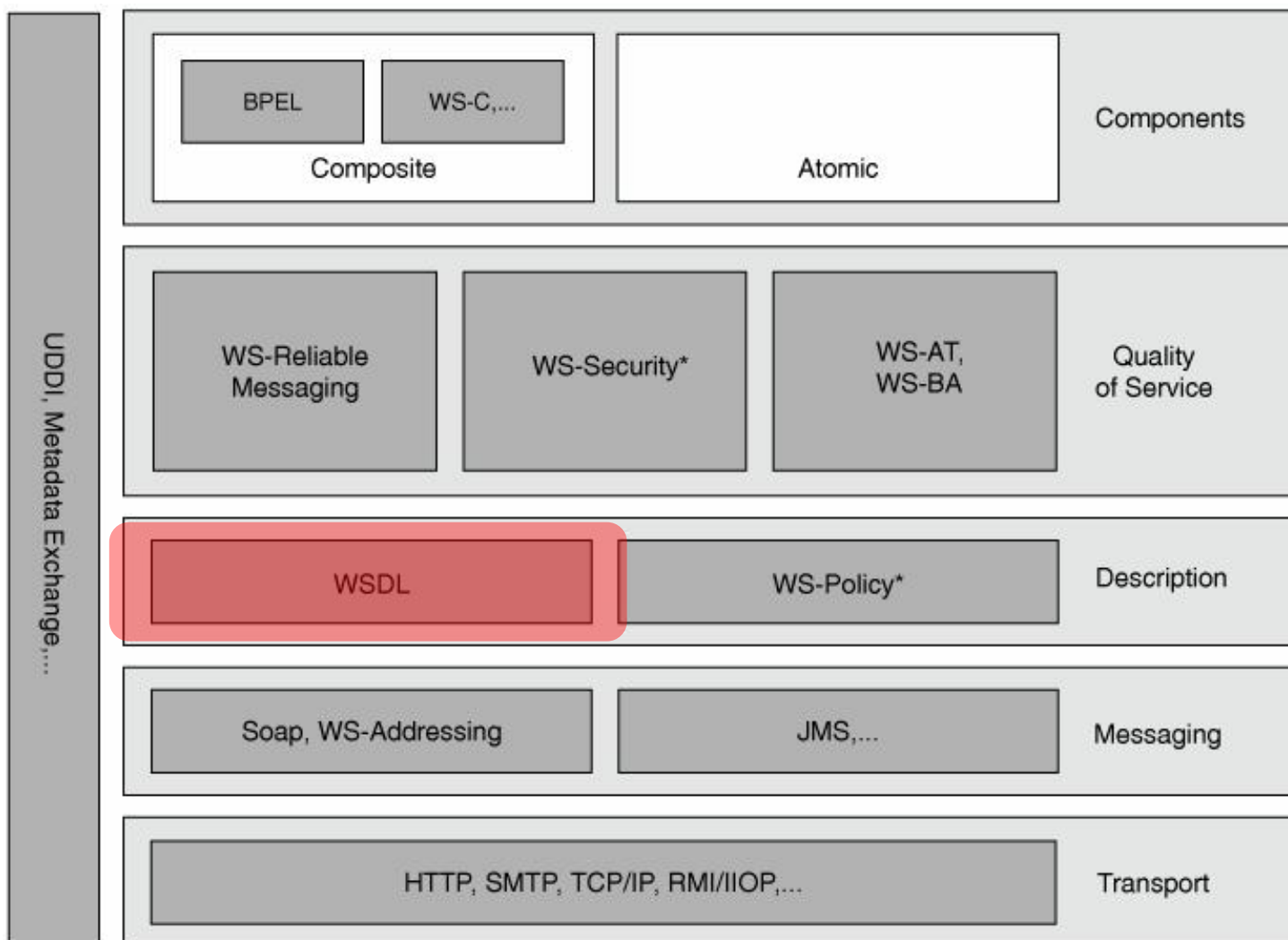


Osnovna arhitektura

(Web Services, WS-*)



Web Services stog



WSDL

- **W**eb **S**ervices **D**escription **L**anguage
 - opis programskog sučelja Web usluge
 - što radi (operacije ...)
 - način pristupa (protokoli ...)
 - lokacija
 - zasnovan na jeziku XML
 - W3C standard

WSDL

- WSDL 1.0
 - IBM, Microsoft i Ariba (2000)
- WSDL 1.1
 - formalizacija WSDL-a 1.0 (2001)
- WSDL 1.2 (2.0)
 - *W3C recommendation* (2007)

WSDL

- Osnovni elementi WSDL jezika
 - operation
 - metode, funkcije
 - message
 - tipizirani podaci koji su ulazi ili izlazi operacija
 - types
 - XML Schema tipovi podataka

WSDL

- WSDL primjer (operacija)

```
<wsdl:operation name="GetAge">  
  <wsdl:input message="GetAgeRequest"/>  
  <wsdl:output message="GetAgeResponse"/>  
  <wsdl:fault message="GetAgeFault"/>  
</wsdl:operation>
```

```
<wsdl:message name="GetAgeRequest">  
  <wsdl:part name="body" element="xsd:String"/>  
</wsdl:message>
```

```
<wsdl:message name="GetAgeResponse">  
  <wsdl:part name="body" element="xsd:GetAgeResponse"/>  
</wsdl:message>
```

```
<xsd:element name="GetAgeResponse">  
  ...  
</xsd:element>
```

WSDL

- Osnovni elementi WSDL jezika
 - portType
 - skup operacija (sučelje)
 - binding
 - povezuje portType s protokolom
 - port
 - krajnja točka
 - service
 - skup portova

WSDL

- WSDL primjer (nastavak)

```
<wsdl:portType name="GetAgePortType">  
  <wsdl:operation name="GetAge">...</wsdl:operation>  
</wsdl:portType>
```

```
<wsdl:binding name="TempSvcSOAPBinding" type="GetAgePortType">  
  <wsdl:operation name="GetAge">  
    <soap:operation soapAction="http://www.test.com/TempSvc"/>  
    ...  
  </wsdl:operation>  
</wsdl:binding>
```

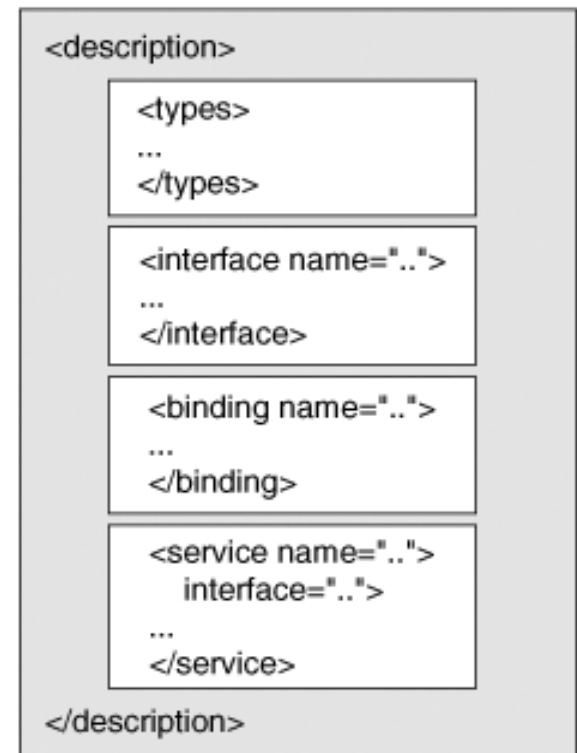
```
<wsdl:service name="AgeService">  
  <wsdl:port name="GetAgePort" binding="TempSvcSOAPBinding">  
    <soap:address location="http://www.test.com/TempSvc">  
  </wsdl:port>  
</wsdl:service>
```

WSDL

- WSDL <-> Java
 - portType = interface
 - operation = public method
 - types = java classes
 - service = stub interface & class

WSDL 2.0

- Jednostavniji, ali manje moćan
 - izbačen *message* element
 - *portType* preimenovano u *interface*
 - *port* preimenovano u *endpoint*



4WS Addressing

WS-Addressing

- Motivacija
 - zašto ne koristiti postojeće mehanizme?
 - npr. HTTP model adresiranja (URI)
 - zahtjevi
 - potpora za više protokola
 - potpora za asinkrone poruke i kvalitetu usluge
 - potpora za razmjenu poruka sa čuvanjem stanja (stateful)
 - HTTP nije oblikovan za bogatu i fleksibilnu komunikacijsku infrastrukturu Web usluga

WS-Addressing

- Uvod
 - opisuje mehanizme koji osiguravaju ispravnu isporuku poruka prema odgovarajućim krajnjim točkama usluga
 - osnovni koncepti
 - krajnje točke usluga
 - WS-Addressing zaglavlja

WS-Addressing

- Krajnje točke usluga
 - **referenca** na krajnju točku
 - različito od WSDL port strukture!
 - krajnje točke mogu biti dinamički generirane
 - struktura podataka koja sadrži sve informacije o krajnjoj točki usluge tijekom rada sustava
 - dvije podgrupe podataka
 - podaci o interakciji s krajnjim točkama
 - metapodaci

WS-Addressing

- Krajnje točke usluga
 - podaci o interakciji s krajnji točkama
 - **adresa**
 - jedino obavezno polje
 - vrijednost jest URI gdje se nalazi krajnja točka usluge
 - **skup značajki** krajnje točke
 - dodatni XML podaci i adrese
 - **parametri** krajnje točke
 - potrebni za interakciju sa krajnjom točkom

WS-Addressing

- Krajnje točke usluga
 - metapodaci
 - WSDL port
 - WSDL opis usluge
 - skup WS-Policy elemenata
 - usporedba krajnjih točaka
 - dvije krajnje točke su jednake ako imaju istu *adresu* i *skup značajki*

WS-Addressing

- Krajnje točke usluga
 - primjer reference krajnje točke

```
<wsa:EndpointReference>  
  <wsa:Address>http://www.example.com/stq</wsa:Address>  
  <wsa:ReferenceProperties>  
    <exp:customer-id>xgtopsnlk-0001</exp:customer-id>  
  </wsa:ReferenceProperties>  
  <wsa:ReferenceParameters>  
    <exp:model-id>llp-11</exp:model-id>  
  </wsa:ReferenceParameters>  
  <wsa:PortType>tns:StockQuotePortType</wsa:PortType>  
  <wsa:ServiceName PortName="StockQuotePort"> tns:StockQuoteService  
  </wsa:ServiceName>  
</wsa:EndpointReference>
```

WS-Addressing

- Krajnje točke usluga
 - WS-Addressing zaglavlja
 - podrška za složenu asinkronu interakciju među uslugama
 - dva obavezna zaglavlja
 - “*To*”: URI koji predstavlja krajnju točku destinacije
 - “*Action*”: URI koji predstavlja semantiku poruke
 - » definira način na koji je potrebno obraditi poruku
 - » preslikavanje na WSDL operacije

WS-Addressing

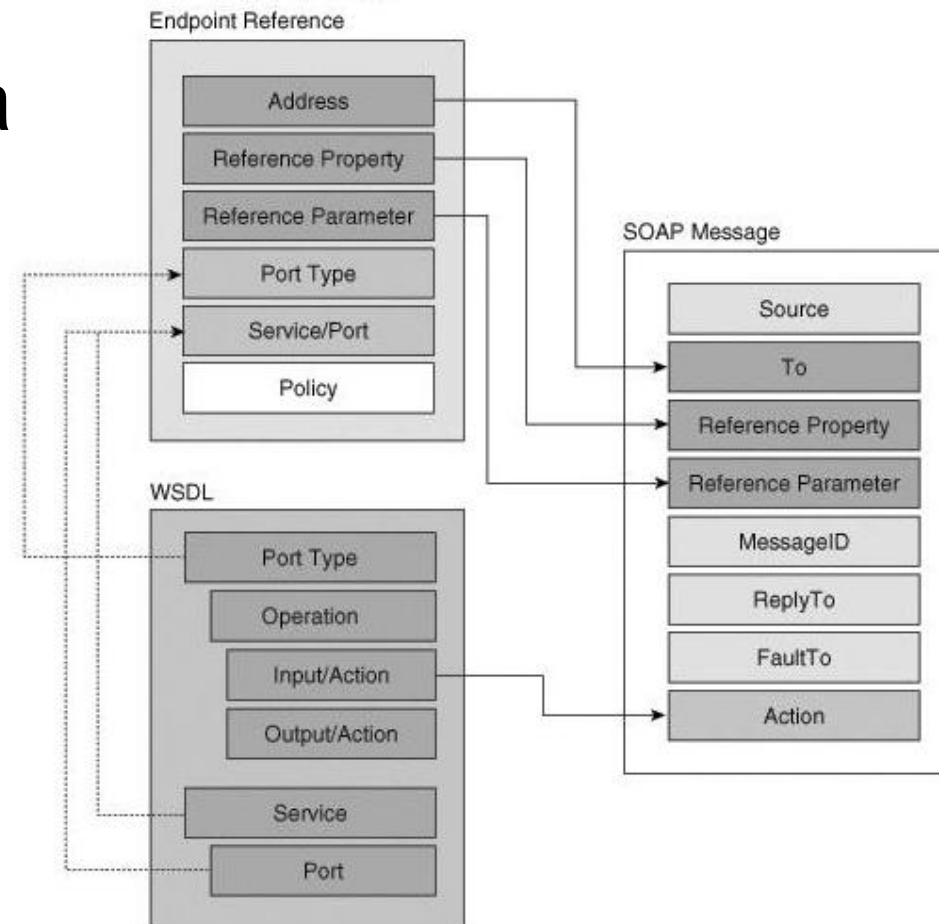
- Krajnje točke usluga
 - WS-Addressing zaglavlja
 - ostala zaglavlja
 - “*ReplyTo*”: krajnja točka za poruke odgovora
 - » obavezno ako je zahtjevana *zahtjev-odgovor* razmjena
 - “*FaultTo*”: krajnja točka za poruke greške
 - “*Source*”: krajnja točka koja je inicirala razmjenu poruka
 - “*MessageId*”: URI koji jedinstveno označava poruku
 - “*RelatesTo*”: URI prethodne poruke
 - » atribut *RelationshipType* označava tip korelacije među dvjema porukama, npr: *wsa:Reply*

WS-Addressing

- Krajnje točke usluga
 - veza na SOAP poruke
 - preslikavanje podataka
 - URI adrese u referenci krajnje točke se preslikava u vrijednost *To* zaglavlja SOAP poruke
 - svaki element skupa značajki krajnje točke se preslikava kao individualno SOAP zaglavlje
 - svaki element parametara krajnje točke se preslikava kao individualno SOAP zaglavlje

WS-Addressing

- Krajnje točke usluga
 - preslikavanje
 - krajnja točka
 - WSDL
 - SOAP



WS-Addressing

- *Zahtjev-odgovor* u WS-Addressing
 - zahtjev
 - obavezna *ReplyTo* i *MessageID* zaglavlja
 - odgovor
 - adresa krajnje točke iz *ReplyTo* zaglavlja zahtjeva kopira se u *To* zaglavlje odgovora
 - *RelatesTo* zaglavlje odgovora sadrži *MessageID* zahtjeva
 - *RelationshipType* se postavlja na vrijednost "*wsa:Reply*"

WS-Addressing

- *Zahtjev-odgovor* u WS-Addressing

