

Web servisi i SOAP protokol

Pojam web servisa

- puno raznih definicija, ali...
- web servisi predstavljaju programe koji su dostupni putem javno objavljenih interfejsa i putem standardnih komunikacionih protokola.
- ono što se danas najčešće podrazumeva pod web servisima su programi
 - dostupni putem SOAP protokola,
 - sa interfejsom za pristup opisanim pomoću WSDL jezika i
 - (potencijalno) registrovani u UDDI servisu.

Pojam web servisa

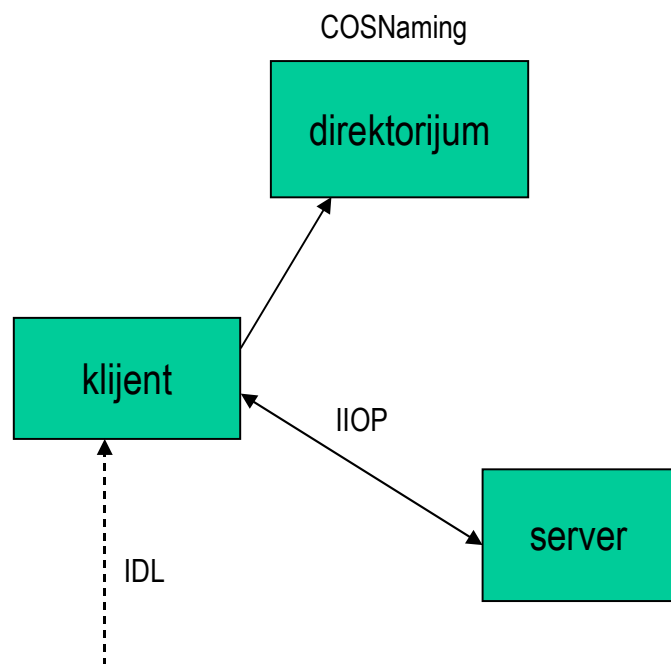
- skraćenice:
 - SOAP = Simple Object Access Protocol
<http://www.w3.org/>
 - WSDL = Web Services Description Language
<http://www.w3.org/>
 - UDDI = Universal Description, Discovery, and Integration
<http://www.uddi.org>

Web servisi i tehnologije distribuiranih sistema

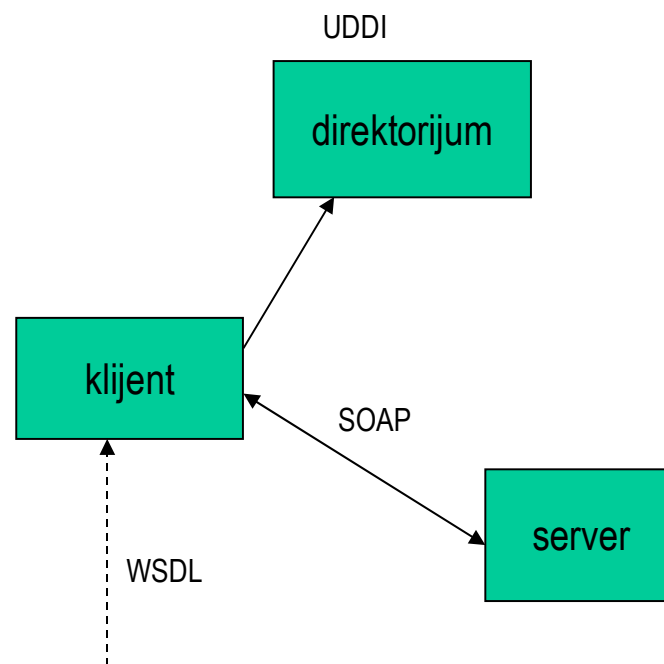
- DCOM
 - tehnologija distribuiranih objekata specifična za Windows platformu
- Java RMI
 - tehnologija distribuiranih objekata specifična za Java platformu
- CORBA
 - tehnologija distribuiranih objekata nezavisna od platforme i programskog jezika
- EJB
 - tehnologija distribuiranih objekata koja se oslanja na RMI i CORBA
- .NET
 - tehnologija distribuiranih objekata specifična za Windows platformu

Web servisi i tehnologije distribuiranih sistema

- slični koncepti



CORBA



web servisi

Web servisi – zašto?

- čemu još jedna tehnologija?
 - svi elementi arhitekture, uključujući i komunikacioni protokol, su zasnovani na XML-u
 - jednostavnija implementacija nego kod binarnih formata
 - veća interoperabilnost među različitim implementacijama
 - upotreba standardnih Internet transportnih mehanizama
 - firewall-friendly
 - globalna dostupnost web servisa

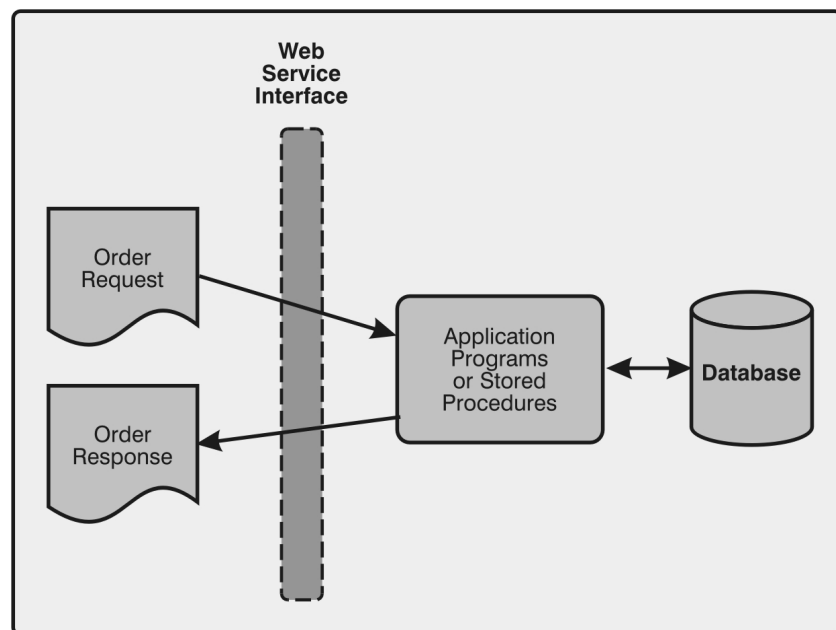
Web servisi – zašto?

- fokus je na interoperabilnosti
- konverzija podataka u/iz XML formata
 - nisu pogodni za sisteme gde su performanse komunikacije od posebnog značaja
- homogeni sistemi u celosti implementirani na jednoj razvojnoj platformi
 - nepotrebno komplikovanje implementacije sistema

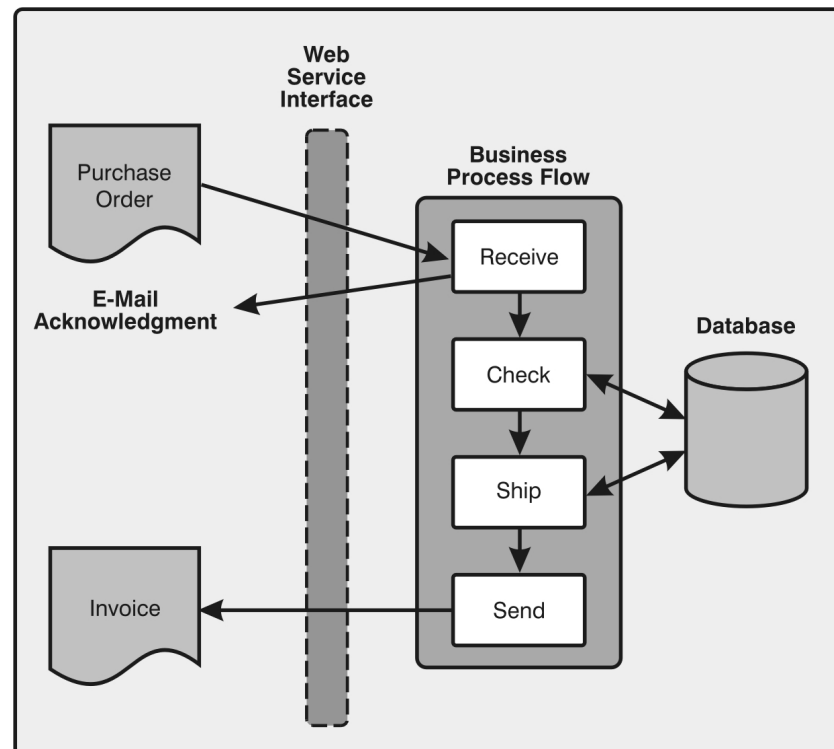
Dve paradigme distribuiranih sistema

- RPC = Remote Procedure Calls
 - podražavaju sintaksu i semantiku pozivanja funkcija/metoda
 - jednostavno za učenje
 - efikasno za kodiranje
 - tipično za sinhronu komunikaciju
- slanje poruka (message passing, document-style)
 - komunikacija između sistema pomoću slanja (strukturiranih) poruka
 - veza između sistema je data formatima poruka
 - bolje razdvajanje delova sistema (loosely coupled)
 - veća međusobna nezavisnost pojedinih delova
 - tipično za asinhronu komunikaciju

Dve paradigme distribuiranih sistema



RPC-style



document-style

Dve paradigme distribuiranih sistema

	RPC	MP
DCOM	x	
RMI	x	
CORBA	x	
EJB	x	x ¹
.NET	x	?
web servisi	x	x

¹Podrška za komunikaciju putem poruka kroz Message-Driven Beans i Java Message Service (JMS)

API-level i wire-level specifikacije

- API-level specifikacije
 - predstavljaju definicije klase, interfejsa, metoda, itd. koje su načinjene za date potrebe
 - primeri:
 - JDBC – pristup relacionim bazama podataka
 - JDO – pristup objektnim bazama podataka
 - JNDI – pristup direktorijumskim sevisima
- wire-level specifikacije
 - predstavljaju definicije formata poruka koji se razmenjuju između učesnika u komunikaciji i postupka razmene poruka
 - primeri:
 - HTTP
 - SMTP

API-level i wire-level specifikacije

- API-level specifikacije omogućavaju pojednostavljenu zamenu programskih modula koji su apstrahovani datim API-jem
 - zamena baze podataka => zamena JDBC drajvera, neznatna promena programa
 - zamena direktorijumskog servera => zamena JNDI provajdera, neznatna promena programa
 - uspešna komunikacija JDBC drajvera jednog proizvođača sa bazom podataka drugog nije garantovana
- wire-level specifikacije omogućavaju komunikaciju heterogenih delova sistema
 - zamena jednog web servera drugim => klijenti će i dalje moći da komuniciraju

API-level i wire-level specifikacije

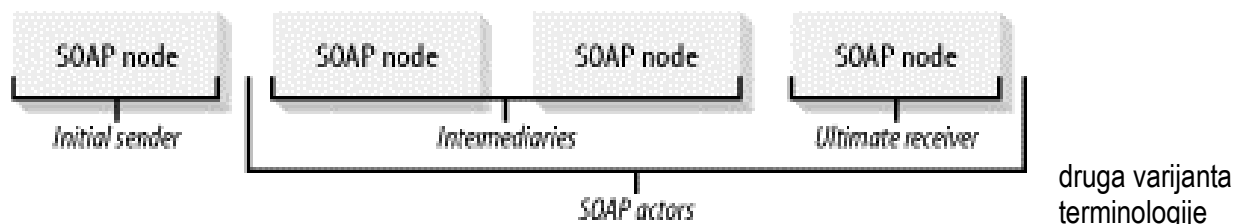
- API-level
 - konkretna implementacija predstavlja programsku biblioteku koja podržava API
 - preuzima se od nekog proizvođača
 - pišemo je sami
- wire-level
 - konkretna implementacija podrazumeva biblioteku koja implementira dati protokol
 - način pristupa biblioteci nije unapred propisan
 - programski kod zavisi od upotrebljene biblioteke

API-level i wire-level specifikacije

- API-level specifikacija za pristup wire-level protokolima
 - najbolje rešenje u smislu prenosivosti i interoperabilnosti programa
 - komplikovano za upotrebu sa stanovišta programera (više isprepletanih standarda)
 - primeri:
 - JAXM – Java API for XML Messaging
 - JAXB – Java API for XML Binding
 - JAX-RPC – Java API for XML-based RPC
 - JAXR – Java API for XML Registries

SOAP – Simple Object Access Protocol

- protokol za komunikaciju sa web servisima
- definiše format poruka koje razmenjuju učesnici
- oslanja se na neki transportni mehanizam za prenos SOAP poruka
 - najčešće HTTP, ali nije obavezno, može i npr. SMTP
- verzije
 - SOAP verzija 1.2 je standard koga propisuje W3C
- u komunikaciji između krajnjih učesnika (endpoints) može biti posrednika (SOAP proxies)



SOAP i HTTP

- upotreba HTTP protokola za prenos SOAP poruka – “HTTP binding”
- za slanje zahteva koristi se HTTP POST komanda

SOAP i HTTP

- primer SOAP+HTTP zahteva i odgovora

POST /item HTTP/1.1

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

Content-Length: 250

```
<?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>
```

```
    <m:GetPrice xmlns:m="http://www.nebitno.com/prices">
```

```
      <m:Item>Apples</m:Item>
```

```
    </m:GetPrice>
```

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

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

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>
```

```
    <m:GetPriceResponse xmlns:m="http://www.nebitno.com/prices">
```

```
      <m:Price>1.90</m:Price>
```

```
    </m:GetPriceResponse>
```

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

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

Struktura SOAP poruke

- XML dokument sa
 - korenskim elementom Envelope
 - opcionim podelementom Header
 - obaveznim podelementom Body
 - sa opcionim podelementom Fault

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

```
<soap:Envelope>
```

```
  <soap:Header>
```

```
    ...
```

```
    ...
```

```
  </soap:Header>
```

```
  <soap:Body>
```

```
    ...
```

```
    ...
```

```
    <soap:Fault>
```

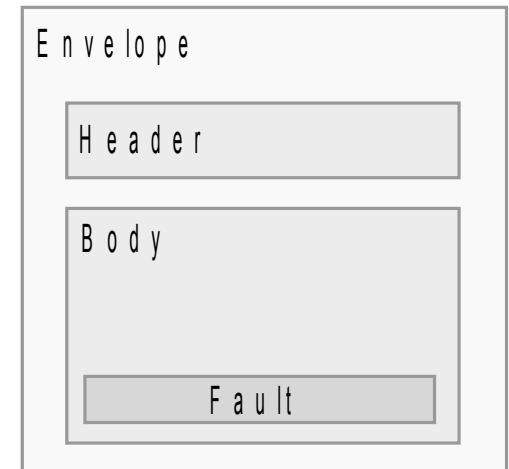
```
      ...
```

```
      ...
```

```
    </soap:Fault>
```

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

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



soap:Envelope

- obuhvata celu SOAP poruku
- definisan je u prostoru imena:
`http://www.w3.org/2001/12/soap-envelope`
- atribut **encodingStyle** definiše namespace sa tipovima podataka koji se koriste u dokumentu
 - odnosi se na element u kome je definisan i sve njegove podelemente
 - formalno nema podrazumevanu vrednost
 - u praksi se koristi
`http://www.w3.org/2001/12/soap-encoding`
 - zapravo definiše način serijalizacije podataka iz aplikacije u XML

soap:Envelope

- izgled elementa Envelope

```
<?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:Envelope>
```

soap:Header

- sadrži podatke koji opisuju kontekst u kome se šalje poruka ili uputstva za posrednike u komunikaciji između krajnjih učesnika
- ovi podaci ne predstavljaju samu poruku, već pomoćne podatke koji utiču na način obrade poruke
- na primer:
 - podaci za autentifikaciju
 - podaci za praćenje sesije
 - podaci za upravljanje transakcijama
- Header pripada istom namespace-u kao i Envelope
- svi Header podelementi moraju biti kvalifikovani u odgovarajući namespace

soap:Header

- primer: identifikacija korisnika

```
<soap:Header>
  <usr:user xmlns:usr="http://www.nebitno.com/userauth">
    <usr:username>mbranko</usr:username>
    <usr:password>*****</usr:password>
  </usr:user>
</soap:Header>
```

soap:Header@actor

- atribut **soap:actor** označava onaj čvor u komunikaciji (proxy ili endpoint) za koga je namenjen dati podatak u zaglavlju

```
<soap:Header>
  <usr:user xmlns:usr="http://www.nebitno.com/userauth"
            soap:actor="http://www.nebitno.com/AppServer">
    <usr:username>mbranko</usr:username>
    <usr:password>*****</usr:password>
  </usr:user>
</soap:Header>
```

soap:Header

- svaki SOAP procesor može da dodaje elemente u zaglavlje na putu poruke od pošiljaoca do konačnog primaoca
- svaki SOAP procesor je dužan da ukloni one elemente iz zaglavlja koji su namenjeni njemu
 - ono što je namenjeno njemu može ponovo dodati u zaglavlje prilikom daljeg slanja

soap:Header@mustUnderstand

- Koncept opcionih i obaveznih elemenata u zaglavlju
 - ne u smislu pojavljivanja u SOAP dokumentu, već
 - u smislu da je primalac dužan da razume i na pravilan način upotrebi dati podatak u zaglavlju.
- Efekat: primalac poruke ne može da obradi poruku jer ne ume da interpretira obavezni podatak u zaglavlju
- ako je element zaglavlja obavezan, atribut **soap:mustUnderstand** ima vrednost 1

```
<soap:Header>
  <usr:user xmlns:usr="http://www.nebitno.com/userauth"
            soap:actor="http://www.nebitno.com/AppServer"
            soap:mustUnderstand="1">
    <usr:username>mbranko</usr:username>
    <usr:password>*****</usr:username>
  </usr:user>
</soap:Header>
```

soap:Body

- sadrži konkretan SOAP zahtev ili odgovor
- pripada istom namespace-u kao i Envelope i Header
- koristi se kod svih vrsta web servisa (RPC-style, document-style)
- primer:

```
<soap:Body>  
  <m:getCurrentTemperature xmlns:m="WeatherStation"  
    <m:scale>Celsius</m:scale>  
  </m:getCurrentTemperature>  
</soap:Body>
```

soap:Fault

- opcioni podelement elementa Body
- sadrži podatke o nastalim greškama namenjene klijentu
- ima četiri podelementa:
 - faultcode
 - faultstring
 - faultactor
 - detail

soap:Fault/faultcode

- indikacija greške namenjena programskoj obradi
- obavezan
- moguće vrednosti:
 - **VersionMismatch**
element Envelope pripada pogrešnom namespace-u
 - **MustUnderstand**
neposredni Header podelement, sa atributom `mustUnderstand="1"`, nije interpretiran
 - **Client**
poruka sa zahtevom je nepravilno formirana ili sadrži neispravne podatke. Klijent ne bi trebalo da istu poruku ponovo šalje
 - **Server**
poruka nije mogla biti obrađena zbog problema u radu servera; sama poruka ima ispravan format i sadržaj. Klijent može pokušati sa istom porukom kasnije.
- Predefinisani kodovi greške su proširivi, na primer:
 - **Server.DatabaseFailure**
 - **Server.DatabaseFailure.MaxUsersConnected**

soap:Fault/faultstring

- tekstualni opis greške namenjen človeku (human-readable)
- obavezan

soap:Fault/faultactor

- indikacija koji čvor u komunikaciji je uzrok greške (u paru sa **actor** atributom)
- ako se ne navede, podrazumeva se da je greška nastala na krajnjem čvoru komunikacije (endpoint)

soap:Fault/detail

- opisuje greške koje su posledica neispravnog sadržaja Body elementa u zahtevu
- ako sadržaj (ili nedostatak sadržaja) u Body elementu sprečavaju obradu poruke, element **detail** sadrži opis greške
- ako greška nije nastala usled sadržaja Body elementa u zahtevu, element **detail** se ne sme pojaviti

SOAP greške: primer 1

- nije bilo greške

```
<?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>
    <m:getCurrentTemperature xmlns:m="http://www.nebitno.com/WeatherStation">
      <m:scale>Celsius</m:scale>
    </m:getCurrentTemperature>
  </soap:Body>
</soap:Envelope>
```

```
<?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>
    <m:getCurrentTemperatureResponse
      xmlns:m="http://www.nebitno.com/WeatherStation">
      <m:temperature>1.90</m:temperature>
    </m:getCurrentTemperatureResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP greške: primer 2

- neispravni podaci u Body elementu

```
<?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>
    <m:getCurrentTemperature xmlns:m="http://www.nebitno.com/WeatherStation">
      <m:scale>Celzijus</m:scale>
    </m:getCurrentTemperature>
  </soap:Body>
</soap:Envelope>
<?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>
    <soap:Fault>
      <soap:faultcode>soap:Client</soap:faultcode>
      <soap:faultstring>Illegal temperature scale</soap:faultstring>
      <soap:detail>
        <m:weatherfaultdetails xmlns:m="http://www.nebitno.com/WeatherStation">
          <m:message>No such temperature scale: Celzijus</m:message>
          <m:errorCode>1234</m:errorCode>
        </m:weatherfaultdetails>
      </soap:detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

SOAP greške: primer 3

- obavezni element zaglavlja nije interpretiran

```
<?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:Header>
  <usr:user xmlns:usr="http://www.nebitno.com/userauth"
            soap:actor="http://www.nebitno.com/AppServer"
                soap:mustUnderstand="1">
    <usr:username>mbranko</usr:username>
    <usr:password>*****</usr:username>
  </usr:user>
</soap:Header>
</soap:Envelope>
```

```
<?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>
  <soap:Fault>
    <soap:faultcode>soap:MustUnderstand</soap:faultcode>
    <soap:faultstring>Did not understand the element user</soap:faultstring>
    <soap:faultactor>http://www.nebitno.com/AppServer</soap:faultactor>
  </soap:Fault>
</soap:Body>
</soap:Envelope>
```

SOAP greške i HTTP greške

- HTTP kodovi 2xx – poruka je primljena i uspešno obrađena
- HTTP kodovi 3xx – redirekcija: traženi servis je premešten na drugu adresu
- HTTP kodovi 4xx – greška u zahtevu
- HTTP kodovi 5xx – greška je na strani servera, u toku obrade poruke

SOAP reprezentacija podataka

- dve mogućnosti:
 - *SOAP encoding*: mapiranje podataka iz Jave/C++/... na XML u skladu sa SOAP specifikacijom
 - *XML Schema*: podaci koji se prenose definisani su XML šemom

SOAP encoding

- podaci koji se nalaze u okviru Body elementa nastali su serijalizacijom podataka iz aplikacije u XML format u skladu sa nekom XML šemom
- SOAP ne definiše podrazumevanu šemu
- može se koristiti više različitih šema u jednoj SOAP poruci
- izbor aktivne šeme određuje se atributom **encodingStyle**
- jedina unapred definisana šema pripada namespace-u `http://www.w3.org/2001/12/soap-encoding`
- ona sadrži sve predefinisane tipove podataka iz specifikacije *XML Schema Part 2: Datatypes*
- npr: **xsd:string**, **xsd:int**, **xsd:boolean**, itd.
- svi podaci u telu poruke predstavljeni su kao sadržaj elementa

SOAP tipovi podataka

- jednostavni (simple)
 - atomičke vrednosti
 - ne sadrži podelemente niti attribute
 - primeri

```
int a = 10;
```

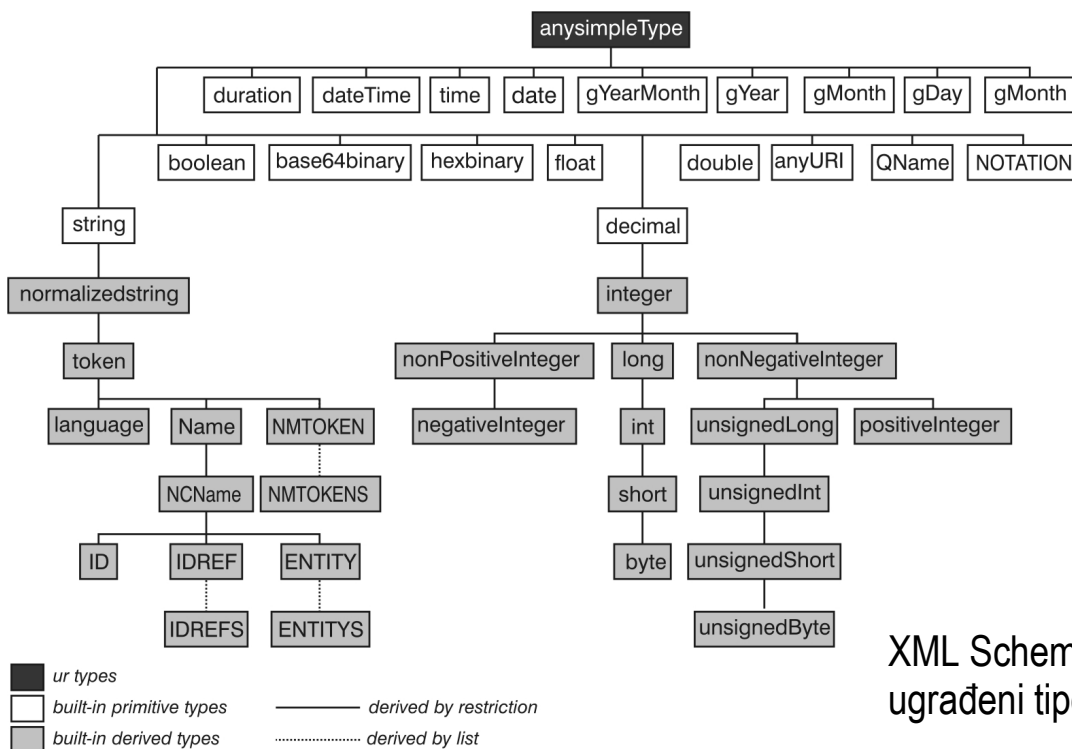
```
float x = 3.14159;
```

```
String s = "abcd";
```

```
<a xsi:type="xsd:int">10</a>
```

```
<x xsi:type="xsd:float">3.14159</x>
```

```
<s xsi:type="xsd:string">abcd</s>
```



XML Schema
ugrađeni tipovi

SOAP tipovi podataka

- agregirani (compound)
 - sadrži više atomičkih podataka organizovanih u neku strukturu
 - primeri

```
int array[3] = {1, 2, 3};  
<array xsi:type="soapenc:Array" soapenc:arrayType="xsd:int[3]">  
  <val>1</val>  
  <val>2</val>  
  <val>3</val>  
</array>
```

```
class Sample {  
  public int iVal = 10;  
  public String sVal = "Ten";  
}  
<sample>  
  <iVal xsi:type="xsd:int">10</iVal>  
  <sVal xsi:type="xsd:string">Ten</sVal>  
</sample>
```

SOAP tipovi podataka

- navođenje tipa podatka
 - atributom xsi:type
 - elementima niza je tip definisan za ceo niz
 - tipovi su
 - predefinisani tipovi
 - novi tipovi definisani pomoću XML Schema jezika

```
<complexType name="TAutomobile">  
  <sequence>  
    <element name="make" type="xsd:string"/>  
    <element name="model" type="xsd:string"/>  
    <element name="year" type="xsd:string"/>  
  </sequence>  
</complexType>
```

```
<car xsi:type="TAutomobile">  
  <model>Corvette</model>  
  <make>Chevrolet</make>  
  <year>1999</year>  
</car>
```


SOAP tipovi podataka

- često korišćeni prostori imena
 - XML Schema
`xmlns:xsd="http://www.w3.org/2001/XMLSchema"`
 - XML Schema instances
`xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`
 - SOAP Encoding
`xmlns:soapenc="http://www.w3.org/2001/12/soap-encoding"`

SOAP tipovi podataka

- prethodni primer, uz uvažavanje prostora imena

```
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsd:targetNamespace="http://www.nebitno.com/cars">
  <xsd:complexType name="TAutomobile">
    <xsd:sequence>
      <xsd:element name="make" type="xsd:string"/>
      <xsd:element name="model" type="xsd:string"/>
      <xsd:element name="year" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

<car
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.nebitno.com/cars"
  xsi:type="TAutomobile">
  <model>Corvette</model>
  <make>Chevrolet</make>
  <year>1999</year>
</car>
```

SOAP tipovi podataka

- reference
 - element na koji ukazuje referenca mora sadržati atribut `id` tipa `ID`
 - elementi koji predstavljaju referencu su prazni i imaju atribut `href` čiji sadržaj je identifikator

```
<employee id="Bob">
  <firstName>Bob</firstName>
  <lastName>Englander</lastName>
</employee>
<employee>
  <firstName>Ben</firstName>
  <lastName>Jones</lastName>
  <manager href="#Bob"/>
</employee>
<employee>
  <firstName>Andrew</firstName>
  <lastName>Smith</lastName>
  <manager href="#Bob"/>
</employee>
<employee>
  <firstName>Lorraine</firstName>
  <lastName>White</lastName>
  <manager href="#Bob"/>
</employee>
```

korisno za serijalizaciju grafova objekata!

SOAP tipovi podataka

- stringovi
 - `xsd:string`
 - `soapenc:string` – dodata podrška za `id` i `href`

- nabrojivi tipovi

```
<simpleType name="TTrafficSignal" base="xsd:string">  
  <enumeration value="Red"/>  
  <enumeration value="Yellow"/>  
  <enumeration value="Green"/>  
</simpleType>
```

```
<trafficSignal xsi:type="TTrafficSignal">Green</trafficSignal>
```

- binarni podaci
 - `xsd:hexBinary`
 - `xsd:base64Binary`
 - `soapenc:base64`

SOAP tipovi podataka

- null vrednosti
`<name xsi:null="1"/>`

SOAP tipovi podataka

- nizovi
 - soapenc:Array
 - tip niza je definisan pomoću soapenc:arrayType

```
int array[3] = {1, 2, 3};
```

```
<array xsi:type="soapenc:Array" soapenc:arrayType="xsd:int[3]">  
  <val>1</val>  
  <val>2</val>  
  <val>3</val>  
</array>
```

- null vrednosti u nizovima

soapenc:root

- specijalni atribut koji označava

- početak liste
- koren stabla
- itd.

```
class Node {  
    int value;  
    Node next;  
}
```

```
<list>  
  <node>  
    <value id="node-3" xsi:type="xsd:int">70</value>  
    <next xsi:null="1"/>  
  </node>  
  <node>  
    <value id="node-2" xsi:type="xsd:int">60</value>  
    <next href="#node-3"/>  
  </node>  
  <node>  
    <value soapenc:root="1" id="node-1" xsi:type="xsd:int">50</value>  
    <next href="#node-2"/>  
  </node>  
</list>
```

SOAP i "polimorfizam"

- recimo da je element `quantity` u nekoj šemi definisan kao tip `xsd:int`
- "polimorfizam" predstavlja dinamičku promenu tipa u telu poruke
- primer

```
<quantity>37</quantity>
```

```
<quantity xsi:type="xsd:float">37</quantity>
```


Web servisi RPC tipa

- pozivanje metoda udaljenog objekta
 - SOAP zahtev = poziv metode
 - SOAP odgovor = rezultat metode
- za poziv je potrebno
 - identifikator objekta kome se upućuje poruka
 - naziv metode
 - parametri metode
 - podaci za SOAP zaglavlje

Web servisi RPC tipa

- identifikacija objekta kome se upućuje poruka
 - naziv objekta se ugrađuje u POST komandu HTTP zahteva

POST /PriceService HTTP/1.1

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

Content-Length: 250

```
<?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>
```

```
    <m:GetPrice xmlns:m="http://www.nebitno.com/PriceService">
```

```
      <m:Item>Apples</m:Item>
```

```
    </m:GetPrice>
```

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

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

Web servisi RPC tipa

- naziv metode
 - određen je podelementom Body elementa
 - servis mora prepoznati dati element i pozvati odgovarajuću metodu stvarnog objekta

```
POST /PriceService HTTP/1.1
```

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

```
Content-Length: 250
```

```
<?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>
```

```
    <m:GetPrice xmlns:m="http://www.nebitno.com/PriceService">
```

```
      <m:Item>Apples</m:Item>
```

```
    </m:GetPrice>
```

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

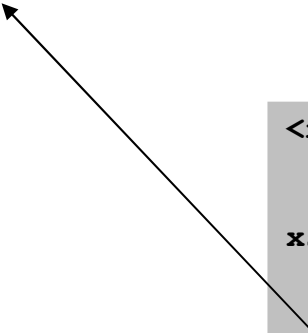
```
</soap:Envelope>
```

Web servisi RPC tipa

- parametri metode
 - dobijeni serijalizacijom podataka iz aplikacije

```
POST /PriceService HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 250
```

```
<?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>
    <m:GetPrice xmlns:m="http://www.nebitno.com/PriceService">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>
</soap:Envelope>
```



```
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsd:targetNamespace="http://www.nebitno.com/PriceService">
  ...
  <xsd:element name="Item" type="TItem"/>
  <xsd:simpleType name="TItem" base="xsd:string">
    <enumeration value="Apples"/>
    <enumeration value="Oranges"/>
    <enumeration value="Mushrooms"/>
  </xsd:simpleType>
</xsd:schema>
```

Web servisi RPC tipa

- rezultat metode
 - strukturiran prema datoj šemi

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>
```

```
    <m:GetPriceResponse xmlns:m="http://www.nebitno.com/prices">
```

```
      <m:Price>1.90</m:Price>
```

```
    </m:GetPriceResponse>
```

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

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

Document-style web servisi

- ne postoji koncept metode i parametara
- servisu se šalje XML dokument
- metoda servisa koja će obraditi dokument se ipak mora nekako navesti
 - naziv prvog Body podelementa mora naznačavati metodu koja će se pozvati za obradu datog dokumenta
 - namespace tog elementa identifikuje servis koji će obraditi zahtev

```
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope">
  <soap:Body>
    <recordTemperature xmlns="urn:WeatherDiary"/>
    <temperature>75.5</temperature>
    <zipcode>50328</zipcode>
  </soap:Body>
</soap:Envelope>
```

Document-style web servisi

- komunikacija nije obavezno po modelu zahtev/odgovor

