Spletne tehnologije, UL, FRI (VSP)
ST 8 - Spletne storitve

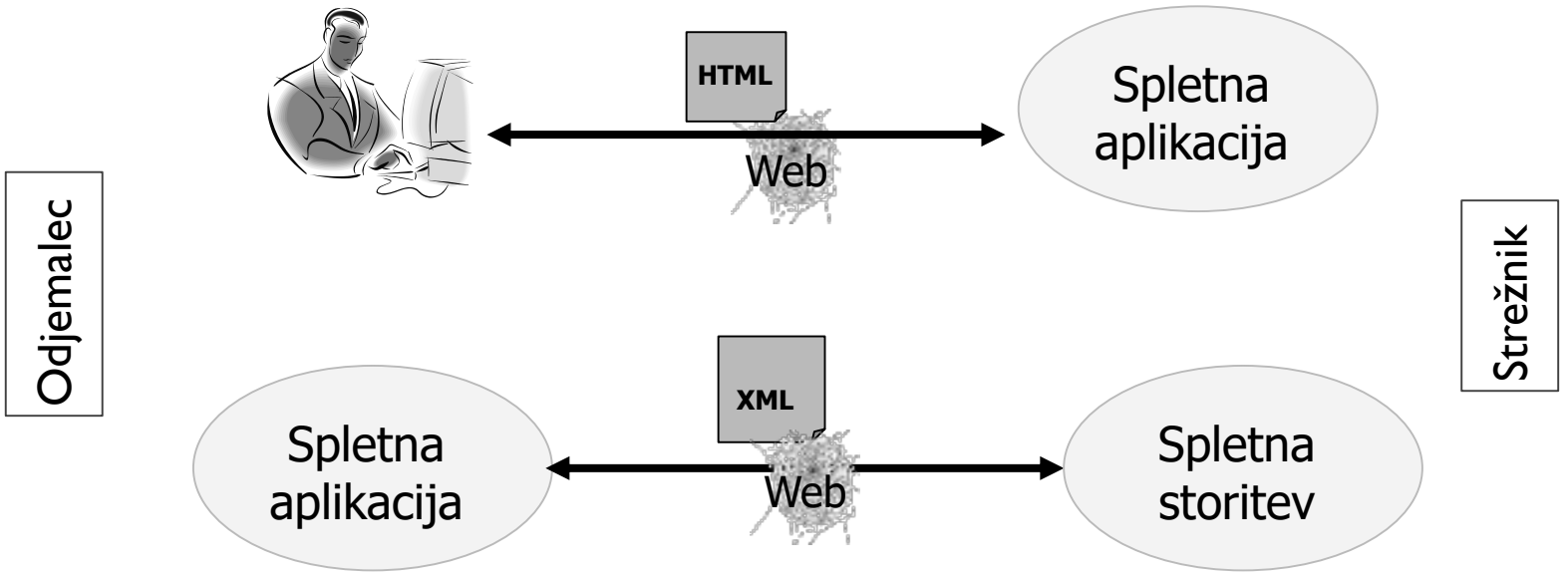
doc.dr. Mira Trebar

Uvod

- Spletne storitve (Web Services)
  - SOAP (Simple Object Access protocol)
  - REST (Representational State Transfer)
- Literatura:
  - http://www.w3schools.com/xml/xml_services.asp
  - http://rest.elkstein.org/2008/02/what-is-rest.html

Spletne storitve (Web Services)

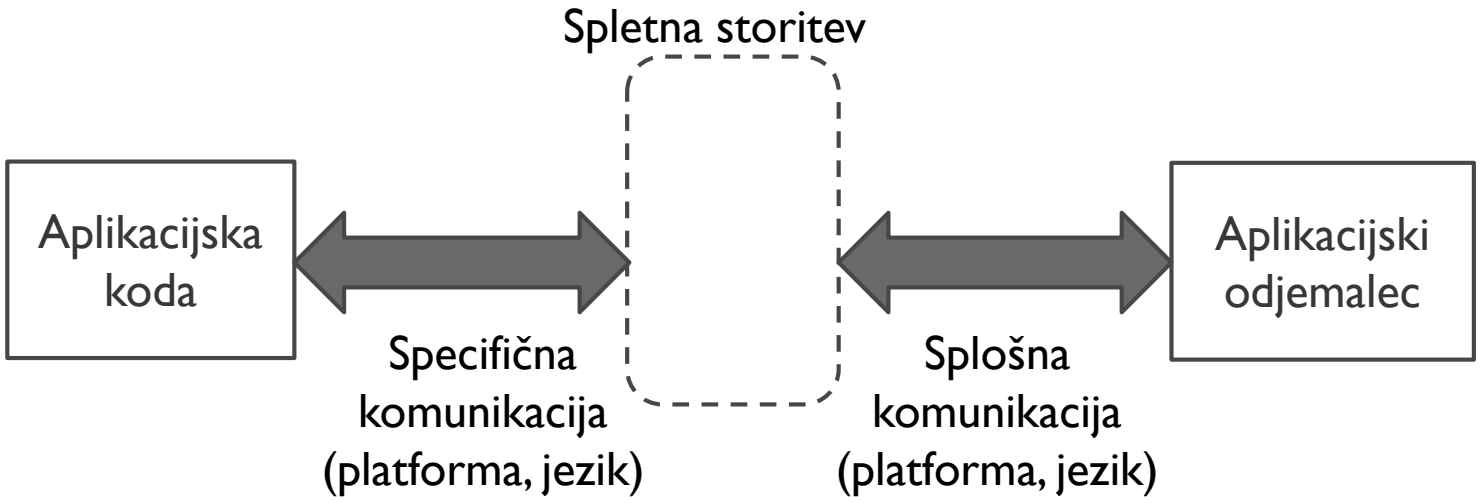
- Uporabnik - Spletne aplikacije – Spletne storitve
- Spletna storitev – programski sistem razvit za podporo spletne interakcije:
  - stroj-stroj;
  - proces-proces;
  - računalnik-računalnik



Uvod

- Spletne aplikacijske komponente
- Samostojne in samo-opisljive komponente
- Na voljo drugim aplikacijam, ki niso del našega sistema
- Se uporabljajo za funkcije, ki jih pogosto kličejo različne aplikacije
- Komunicirajo preko odprtokodnih protokolov
- HTTP in XML predstavljata osnovo spletnih storitev
- Interoperabilnost - Povezujejo različne aplikacije in platforme
- Primeri:
  - pretvorba valut,
  - vremenska poročila,
  - ...
- Primer: Borza postavi spletno storitev za branje cene posamezne delnice. Druge aplikacije (spletne strani, namizne aplikacije, Android/iOS aplik.) uporabljajo to storitev za pridobitev podatkov, ki jih potem prikažejo.

- Spletna storitev dovoli dostop do aplikacijske kode s standardnimi spletnimi tehnologijami (HTTP, XML, ...)
- Vmesnik med aplikacijsko kodo in uporabnikom te kode (aplikacija).
- Abstraktni nivo – ločevanje med podrobnostmi okolja in programskega jezika pri dejanskem klicu aplikacijske kode.
- Brskalnik – spletna storitev – izvedba naloge



Spletne storitve

- Spletne storitve zasnovane na protokolu SOAP
  - Opis informacije

    XML – Extended Markup Language
  - Opis spletnih storitev

    WSDL – Web Service Description Language
  - Dostop do spletnih storitev

    SOAP – Simple Object Access Protocol
  - Iskanje spletnih storitev

    UDDI – Universal Description, Discovery, Integration
- REST (REpresentational State Transfer)

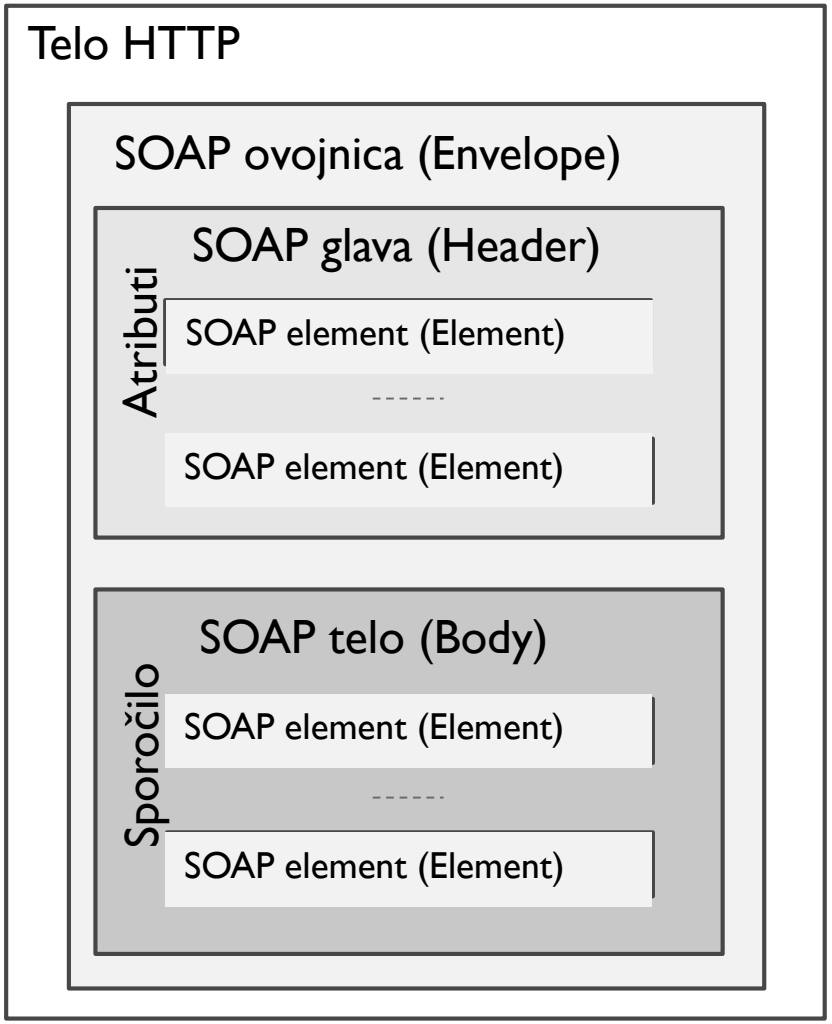
  Prenos podatkov z uporabo protokola HTTP
- Uporaba:
  - Aplikacijske komponente (pogosta uporaba)
  - Povezovanje obstoječih rešitev - interoperabilnost

SOAP (Simple Object Access Protocol)

- aplikacijski komunikacijski protokol – deluje med aplikacijami
- format za pošiljanje in sprejemanje sporočil
- neodvisen od ogrodja in od programskega jezika
- zasnovan na opisnem jeziku XML
- enostaven in razširljiv
- je W3C priporočilo

- Sintaksa sporočila SOAP - XML dokument, ki vsebuje:
  - "Envelope element " – identificira XML dokument kot sporočilo SOAP
  - "Header element " - vsebuje informacijo glave
  - "Body element " – vsebuje informacijo klica in odgovora
  - "Fault element " – vsebuje napake in statusno informacijo

Sporočilo SOAP




```
<?xml version="1.0"?>
<soap:Envelope
            xmlns:soap=…
            soap:encodingStyle>

<soap:Header>
…
</soap:Header>

<soap:Body>
…
    <soap:Fault>
  …
    </soap:Fault>
</soap:Body>
</soap:Envelope>
```

Ovojnica SOAP (soap:Envelope)

- Korenski element: sporočila: <soap:Envelope> </soap:Envelope>
- Definira dokument xml kot SOAP sporočilo
- Primer:

```
<?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">
  ...
      Sporočilo
  ...
</soap:Envelope>
```

- xmlns:soap je imensko področje (http://www.w3.org/2001/12/soap-envelope)
- soap:encodingStyle določa tip podatkov (soap:encodingStyle="URI")

  Pojavi se lahko v vsakem elementu SOAP in se uporabi na tem in na vseh elementih, ki so njegovi otroci.

Glava SOAP (soap:Header)

- Opcijski element
- Ni predpisana vsebina:
  - Informacija, ki jo bo uporabil prejemnik sporočila (avtentikacija, plačilo).
  - Pošiljatelj in sprejemnik se dogovorita o informaciji, ki se tu nahaja (številka transakacije).
- Uporabljajo jo ebXML (electronic business XML)

- Sintaksa: soap:mustUnderstand="0|1" – označi obvezno uporabo glave
- Določa kako naj prejemnik obdela sporočilo SOAP.

```
<soap:Header>
<m:Trans xmlns:m="http://www.w3schools.com/transaction/"
soap:mustUnderstand="1">234      /* ="1"  prejemnik mora razpoznati element
</m:Trans>
</soap:Header>
```

Telo SOAP (soap:Body)

- Telo SOAP vključuje podatke za izmenjavo

- V sporočilu je zahteva za ceno jabolk

```
<soap:Body>
  <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
    <m:Item>Apples</m:Item>
  </m:GetPrice>
</soap:Body>
```

- elementa <m:GetPrice > in <m:Item> sta specifična za aplikacijo

- Odgovor SOAP

```
<soap:Body>
  <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices">
    <m:Price>1.90</m:Price>
  </m:GetPriceResponse>
</soap:Body>
```

Napaka SOAP (soap:Fault)

- Fault SOAP je opcijski, se pojavi enkrat in označuje napake sporočila.
- Podelementi:
  - <faultcode> - koda za določanje napake
  - <faultstring> - človeku berljiva razlaga napake
  - <faultactor> - kdo je povzročil napako
  - <detail> - informacija v povezavi s specifično aplikacijo
- Kode napak:
  - VersionMismatch – neveljaven imenski prostor
  - MustUnderstand – otrok elementa Header z atributom mustUnderstand postavljenim na "1“ ni bil razumljen
  - Client – sporočilo je bilo napačno
  - Server – strežnik ni mogel obdelati sporočila

Primer: Zahteva za ceno delnic

- 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>
```

Primer: Odgovor cene delnic

- 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>
```

WSDL- Web Services Description Language

- Uporabljen za opis spletnih storitev
- Zapisan je v XML, je XML dokument
- Uporabljen je za lociranje spletnih storitev
- Je W3C priporočilo od 2007 dalje
- Ob definiciji WSDL dokumenta se:
  - avtomatsko generira klic spletne storitve in razčlenitev odgovora
  - preostane le uporaba pridobljenih podatkov.
- Dokument WSDL določa lokacijo in metode
- Struktura - elementi
  - <types> - podatkovni tip, ki ga uporablja spletna storitev
  - <message> - sporočilo, ki ga uporablja spletna storitev
  - <portType> - operacija, ki jo uporablja spletna storitev
  - <binding> - komunikacijski protokol, ki ga uporablja spletna storitev

Struktura WSDL

- <definitions> - korenski element

  <types> - podatkovni tipi
    definition of types.....
  </types>

  <message> - sporočilo
    definition of a message....
  </message>

  <portType> - nabor podprtih operacij
    definition of a port.....
  </portType>

  <binding> - definira transakcijski protokol
    definition of a binding.... prenos rpc ali dokument (operation)
  </binding>

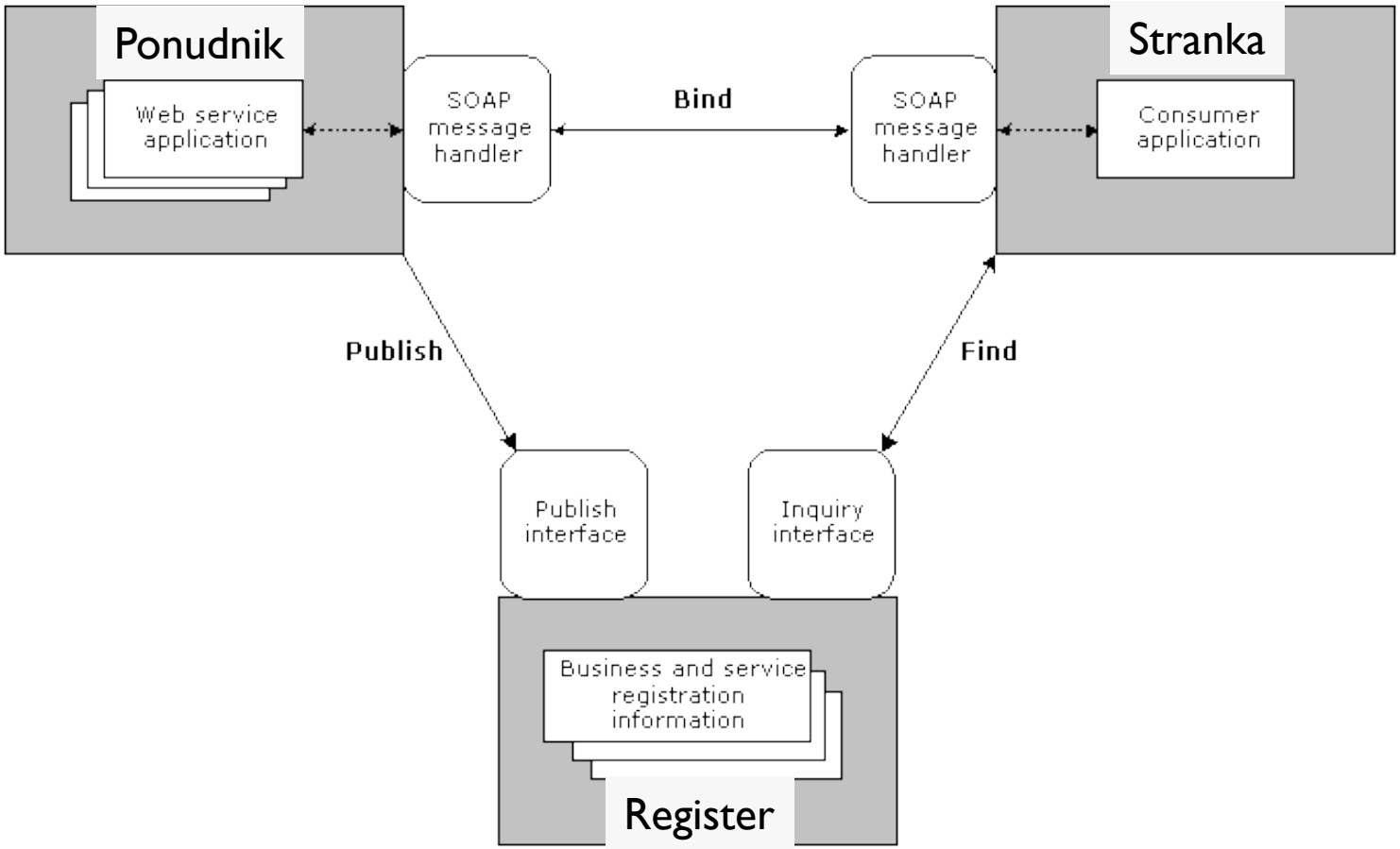
  </definitions>

UDDI-Universal Description, Discovery and Integration

- Je standard, ki je zasnovan na XML (opis, objava in iskanje spletnih storitev).
- Je specifikacija za porazdeljen register spletnih storitev.
- UDDI je od platforme neodvisno ogrodje za
  - opis storitev,
  - odkrivanje posla in
  - integracije poslovnih storitev z uporabo interneta.
- Mapa za shranjevanje informacij o spletnih storitvah
- Mapa vmesnikov spletnih storitev opisanih z WSDL
- Komunicira preko SOAP
- UDDI ima dve sekciji:
  - Register vseh metapodatkov spletnih storitev, vključno s kazalcem na njihove opise WSDL
  - Niz definicij za WSDL <portType> za upravljanje in iskanje registra

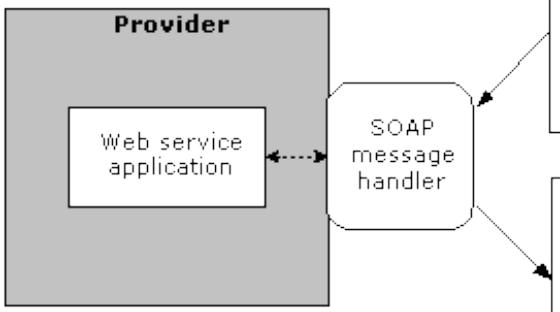
Spletne storitve - opis

- https://www.novell.com/documentation/extend52/Docs/help/Director/books/utoolsUnderstandingServices.html



Spletne storitve – ponudnik

- Kreiranje komponent



HTTP request from consumer

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml;
charset="utf-8"
Content-Length: nnnn
SOAPAction: "www.stockquoteserver.com/services/getquote.htm"

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <SOAP-ENV:Body>
            <m:GetLastTradePrice
            xmlns:m="www.stockquoteserver.com/services/getquote.htm">
                  <symbol>DIS</symbol>
            </m:GetLastTradePrice>
      </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml;
charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      <SOAP-ENV:Body>
            <m:GetLastTradePriceResponse
            xmlns:m="www.stockquoteserver.com/services/getquote.htm">
                  <Price>34.5</Price>
            </m:GetLastTradePriceResponse>
      </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

HTTP response to consumer

Spletne storitve - stranka

- Uporaba

HTTP request to provider

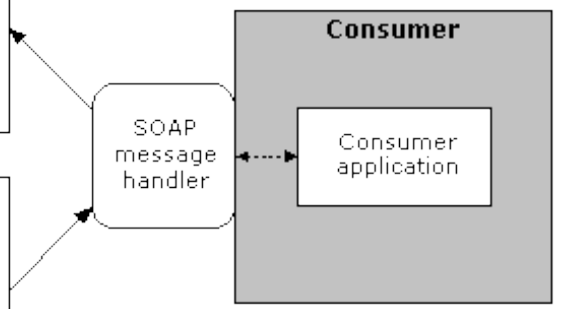
```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml;
charset="utf-8"
Content-Length: nnnn
SOAPAction: "www.stockquoteserver.com/services/getquote.htm"

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <SOAP-ENV:Body>
            <m:GetLastTradePrice
            xmlns:m="www.stockquoteserver.com/services/getquote.htm">
                  <symbol>DIS</symbol>
            </m:GetLastTradePrice>
      </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml;
charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      <SOAP-ENV:Body>
            <m:GetLastTradePriceResponse
            xmlns:m="www.stockquoteserver.com/services/getquote.htm">
                  <Price>34.5</Price>
            </m:GetLastTradePriceResponse>
      </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

HTTP response from provider

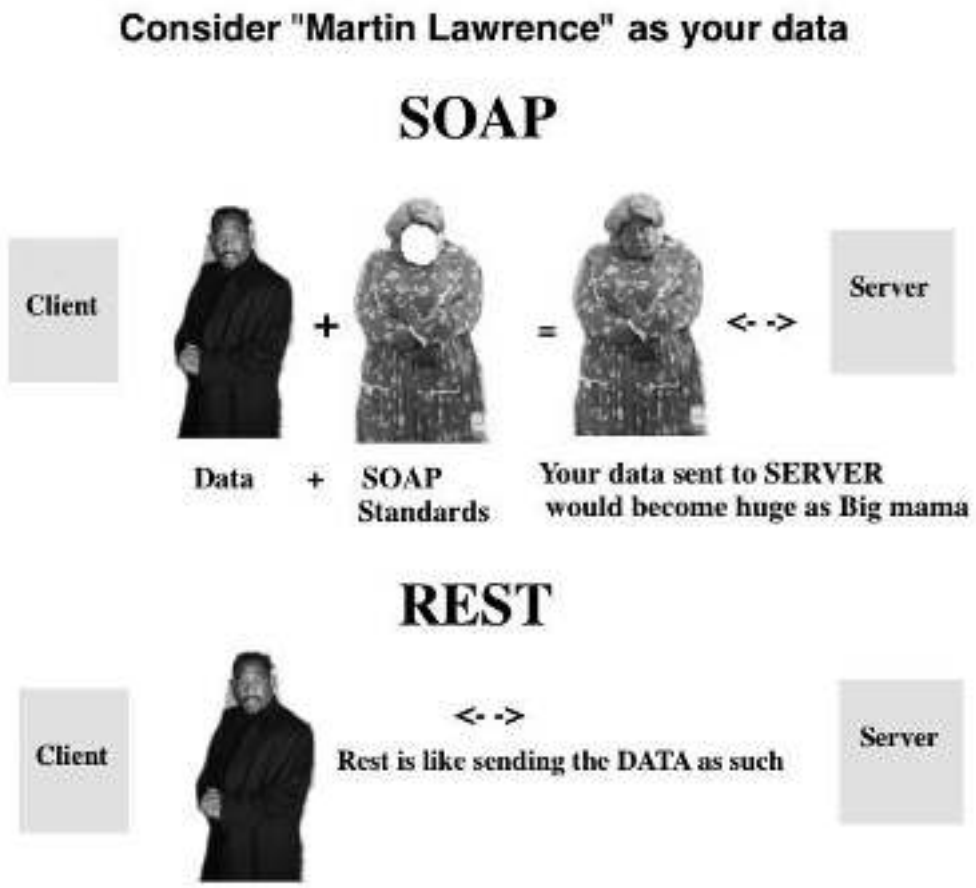


Primer 1: Spletna storitev - SOAP

- Ustvarimo spletno storitev s PHP:
  - SOAP strežnik (SimpleServer.php)
  - SOAP odjemalec (SimpleClient.php)
  - Interakcija z uporabnikom (SimpleView.php)
  - Knjižnica nusoap.php (PHP classes - SOAP 1.1, WSDL 1.1, HTTP 1.0/1.1)
- Vir: http://www.phpgang.com/create-a-web-service-with-php_43.html

REST - **RE**presentational **S**tate **T**ransfer

- Protokol HTTP je uporabljen za izvedbo klicev med aplikacijami/stroji



http://stackoverflow.com/questions/209905/representational-state-transfer-rest-and-simple-object-access-protocol-soap

Primer: SOAP in REST

- Branje podatkov v aplikaciji ‘phonebook‘ za parameter UserID= 12345
  - **SOAP** (zahteva):

```
<?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 pb="http://www.acme.com/phonebook">
   <pb:GetUserDetails>
   <pb:UserID>12345</pb:UserID> </pb:GetUserDetails>
    </soap:Body>
</soap:Envelope>
```

  - **REST klic** bo z metodo GET na naslov URL:

    http://www.acme.com/phonebook/UserDetails/12345

REST

- REST uporabi protokol HTTP za izvedbo operacij (CRUD):
  - Create
  - Read
  - Update
  - Delete
- Delovanje REST:
  - Neodvisno od okolja (odjemalec je Mac - Strežnik je Unix, ...)
  - Neodvisno od jezika (C# in Java, ..)
  - Standardizirana zasnova (deluje nad HTTP)
  - Enostavna uporaba ob prisotnosti požarnih zidov
- Nima vgrajenih varnostnih mehanizmov - Potrebno jih je dodati (HTTPs, ...)

Kompleksnejša spletna storitev REST

- Vključitev večjega števila parametrov:
  http://www.acme.com/phonebook/UserDetails?firstName=John&lastName=Doe
- REST odgovor:
  - XML zapis podatkov, ki omogoča preverjanje veljavnosti podatkov
  - JSON (Java Script Object Notation)
  - CMS (Comma-Separated Values)

```
<parts-list>
 <part id="3322">
  <name>ACME Boomerang</name>
  <desc>
   Used by Coyote in <i>Zoom at the Top</i>, 1962
  </desc>
  <price currency="usd" quantity="1">17.32</price>
  <uri>http://www.acme.com/parts/3322</uri>
 </part>
 <part id="783">
  <name>ACME Dehydrated Boulders</name>
  <desc>
   Used by Coyote in <i>Scrambled Aches</i>, 1957
  </desc>
  <price currency="usd" quantity="pack">19.95</price>
  <uri>http://www.acme.com/parts/783</uri>
 </part>
</parts-list>
```

- Realni primeri:
  - Twitter ima REST API
  - Flickr
  - Atom

REST in AJAX

- Zahteva je poslana z objektom XMLHttpRequest

```
function createRequest() {
  var result = null;
  if (window.XMLHttpRequest) {
    // FireFox, Safari, etc.
    result = new XMLHttpRequest();
    if (typeof xmlhttp.overrideMimeType != 'undefined') {
      result.overrideMimeType('text/xml'); // Or anything else
    }
  }
  else if (window.ActiveXObject) {
    // MSIE
    result = new ActiveXObject("Microsoft.XMLHTTP");
  }
  else {
    // No known mechanism -- consider aborting the application
  }
  return result;
}
```

- Ne vrne takoj odgovora (pogosto je ta v JSON, nima povezave z XML)
- Vključiti je potrebno *callback function*

REST in AJAX

```
var req = createRequest(); // defined above
// Create the callback:
req.onreadystatechange = function() {
  if (req.readyState != 4) return; // Not there yet
  if (req.status != 200) {
    // Handle request failure here...
    return;
  }
  // Request successful, read the response
  var resp = req.responseText;
  // ... and use it as needed by your app.
}
```

- Pošljemo zahtevo

```
req.open("GET", url, true);
req.send();
```

For POST requests, use:

```
req.open("POST", url, true);
req.setRequestHeader("Content-Type",
                     "application/x-www-form-urlencoded");
req.send(form-encoded request body);
```

Primer 2: Spletna storitev -RESTFul

- Kreiranje uporabnika, pridobivanje podatkov o uporabniku, ažuriranje podatkov uporabnika, brisanje uporabnika
- http://www.scalsys.com/blog/build-rest-web-service-with-php-json-and-mysql/
- Ustvarimo datoteke s PHP:

  db_config.php

  create.php

  select.php

  update.php

  delete.php
- Podatkovna baza: test, tabela

```
`id` int(11) NOT NULL AUTO_INCREMENT,
`firstname` varchar(255) NOT NULL,
`lastname` varchar(255) NOT NULL,
`email` varchar(255) NOT NULL,
`password` varchar(255) NOT NULL,
`status` enum('1','0') NOT NULL,
```

Dodamo uporabnika

Pridobimo podatke o uporabniku