



X36LOS – 6. seminar

SOAP

(Simple Object Access Protocol)

WSDL

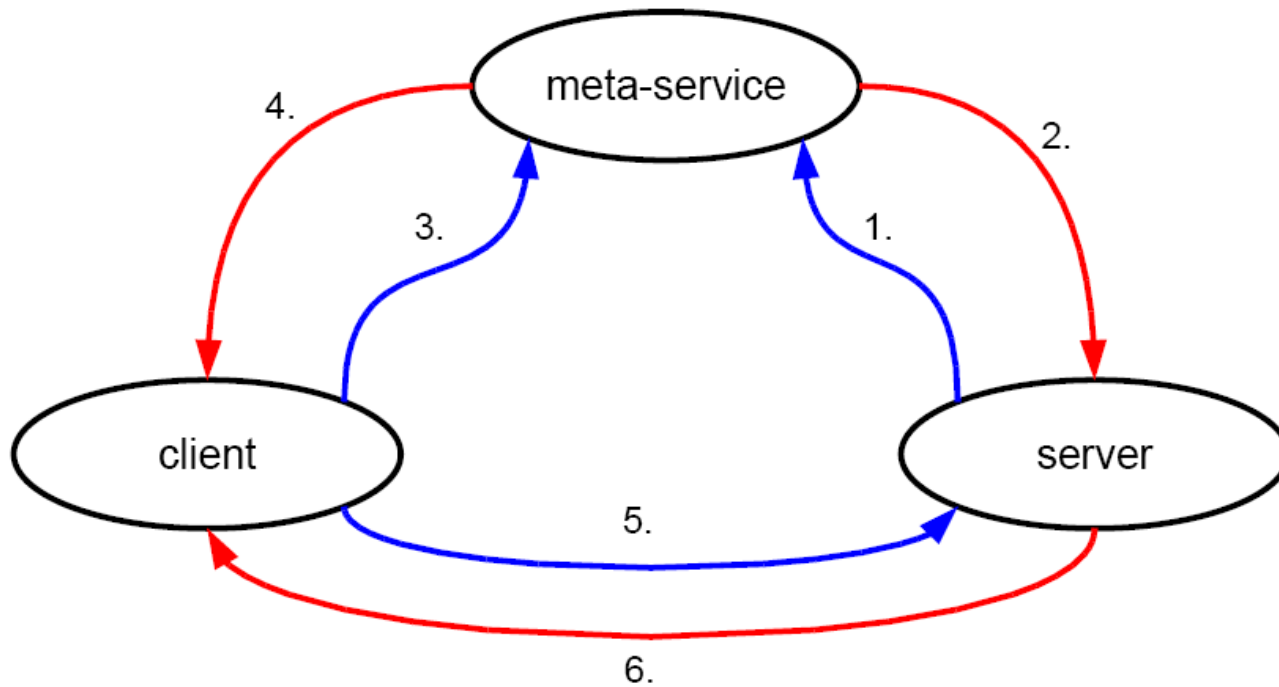
(Web Services Description Language)

UDDI

(Universal Description, Discovery, and Integration)

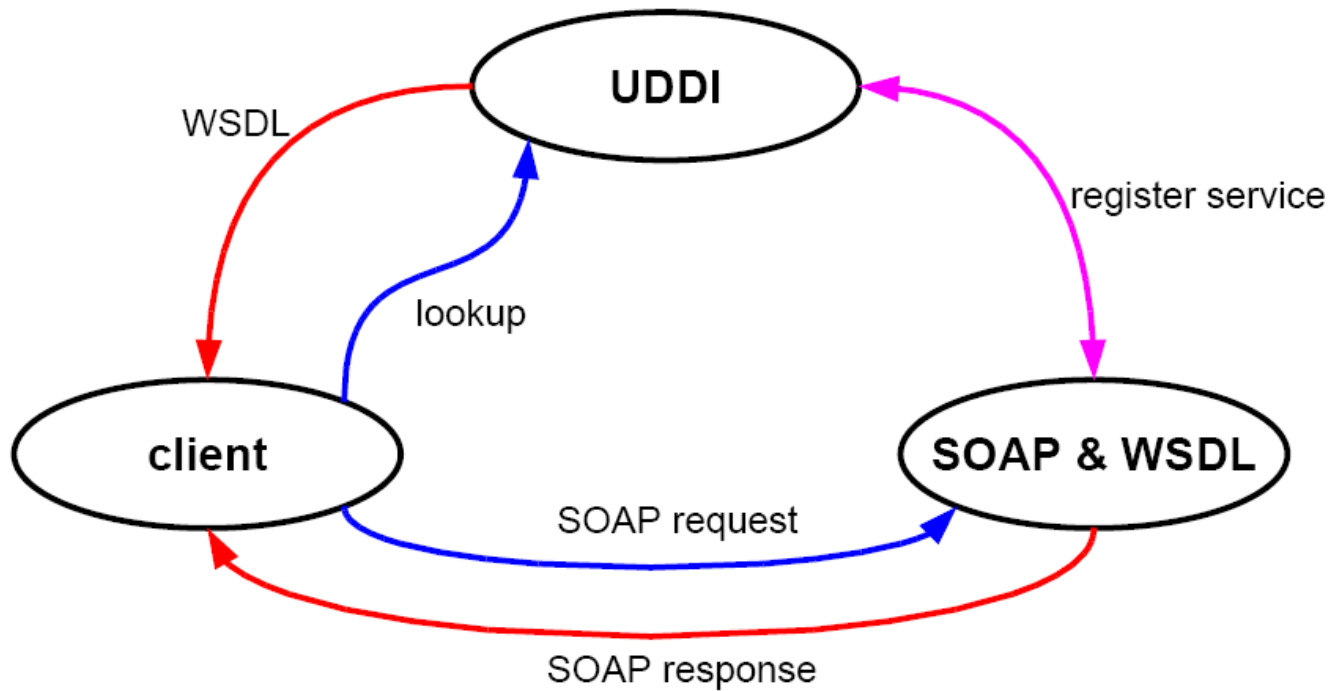


Web Services





SOAP, WSDL, UDDI





SOAP

- message format: XML
- transportation protocol: HTTP, SMTP

Example – request for information about product # 827653

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.com/ws">
      <productID>827635</productID>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```



SOAP

Example – response to further request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.com/ws">
      <getProductDetailsResult>
        <productName>Toptimate 3-Piece Set</productName>
        <productID>827635</productID>
        <description>
          3-Piece luggage set. Black Polyester.
        </description>
        <price>96.50</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```



WSDL

- service interface definitions

Example – WSDL file structure

```
<?xml version="1.0" encoding="utf-8">
<definitions>
  <types> describes SOAP parameters' types
  <\types>
  <message> describes structure of SOAP messages
  <\message>
  <portType> defines method invocations
  <\portType> (bind requests and responses)
  <binding>
  <\binding>
  <service>
  <\service>
<\definitions>
```



Axis2

- framework pro Web Services from Apache
- AXIOM (AXIs Object Model)
- Structure
 - Information model
 - XML processing model (StAX – Streaming API for XML)
 - SOAP processing model
 - Deployment module
 - WSDL and code generation module
 - Client API
 - Transports (HTTP, SMTP, TCP)



Usage

- 2 main approaches
 - Contract first – begins from WSDL
 - then You use utility `WSDL2java`, `WSDL2code` ...
 - appropriate for WSDL understanding developers
 - Code first – begins with code
 - then You use class `org.apache.ws.java2wsdl.Java2WSDL`
 - appropriate for beginners



Axis2 – web service

- ZIP/JAR archive s AAR extension (Axis Archive)
- Structure
 - ./<package>/<class>.class
 - ./META-INF/services.xml
 - ./META-INF/services.wsdl
 - ./lib/<used_libraries>.jar



Server – implementation

- implementation of simple class using API

Example:

```
package x36dsv.cv6;

import javax.xml.stream.XMLStreamException;
import org.apache.axiom.om.OMElement;

public class TestServer1 {
    /* echo service */
    public OMElement echo(OMElement element) throws
XMLStreamException {
    /* vytvoreni 'DOM' stromu z cele prichozi zpravy */
    element.build();
    /* 'osamostatneni' prijate struktury */
    element.detach();
    return element;
    }
}
```



Server - services.*

- Information about published service + interface
 - services.xml
 - http://ws.apache.org/axis2/0_94/axis2config.html#Service_Configuration
 - implementing class
 - published methods
 - methods parameters
 - services.wsdl
 - <http://www.w3.org/TR/wsdl>
 - definition of communication interface and messages



SOAP - sources

- SOAP
 - <http://www.itu.dk/courses/IWJX/E2003/>
- WSDL
 - <http://www.oreilly.com/catalog/webservess/chapter/ch06.html>
 - <http://api.google.com/GoogleSearch.wsdl>
 - <http://code.google.com/apis/soapsearch/>
- Axis2
 - <http://ws.apache.org/axis2/index.html>