Enabling Grids for E-science in Europe

www.eu-egee.org

*3 – 4 June 2004*

# WSDL

## Web Service Description Language

EGEE is a project funded by the European Union under contract IST-2003-508833

# Objectives

- The role of WSDL
- The structure of a WSDL document
  - types
  - message
  - portType
  - binding
  - service

# The function of WSDL

Enabling Grids for E-science in Europe

- WSDL describes a service's exposed interface
- It is what a client sees of your service
- WSDL includes information about
  - The data types it uses
  - Parameters it requires and returns
  - Groupings of functionality
  - The protocol to be used to access the service
  - The location or address of the service

# WSDL Structure

- A WSDL document is an XML document

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions>
   <types>
        <!- define the types here using XML Schema →
   </types>
   <message>
        <!- XML messages the web service uses are defined here →
   </message>
   <portType>
        <!- define the input and output parameters here -→
   </portType>
   <binding>
        <!- define the network protocol here →
   </binding>
   <service>
        <!- location of the service →
   </service>
</definitions>
```

# <import> element



```
<definitions
    targetNamespace="urn:3950"
    xmlns= "http://schema.xmlsoap.org/wsdl/"
    xmlns:xsd= "http://www.w3c.org/2001/XMLSchema"
    xmlns:soap= "http://schemas.xmlsoap.org/wsdl/soap/"
    xmlnssoapenc= "http://schemas.xmlsoap.org/soap/emcoding/"
    xmlns:tns= "urn:3950">

    <import namespace= "http://nesc.ac.uk" location= "http://nesc.ac.uk/ez.xsd"/>
```

**Acts like C/C++ #include , or Java import.**
**Incorporates external namespaces**

# Namespaces

Enabling Grids for E-science in Europe

- WSDL uses a number of different namespaces including
- XML Schema Namespaces
  - http://www.w3.org/2000/10/XMLSchema
  - http://www.w3c.org/2001/XML-Schema-instance
- WSDL Namespaces
  - http://schemas.xmlsoap.org/wsdl/soap/
  - http://schemas.xmlsoap.org/wsdl/
- SOAP Namespaces
  - http://schemas.xmlsoap.org/soap/encoding
  - http://schemas.xmlsoap.org/soap/envelope

# The <types>

- The types element contains XML Schemas defining the datatypes that are to be passed to and from the web service

```
<types>
 <schema targetNamespace="http://example.com/stockquote.xsd"
         xmlns="http://www.w3.org/2000/10/XMLSchema">
  <element name="TradePriceRequest">
   <complexType>
     <all><element name="tickerSymbol" type="string"/></all>
   </complexType>
  </element>
  <element name="TradePrice">
    <complexType>
     <all><element name="price" type="float"/></all>
    </complexType>
  </element>
 </schema>
</types>
```

# The <message>

- The **<message>** element is used to define the messages that will be exchanged between the client and the service
- These message elements contain **<part>** elements, which will be using types defined in the types element

```
<message name="GetLastTradePriceInput">
   <part name="body" element="xsd1:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
   <part name="body" element="xsd1:TradePrice"/>
</message>
```

- All the parts are namespace qualified

# The <portType>

- The types and messages have been defined, but they have not been defined in terms of where they fit in the functionality of the web service
- This is done within **<portType>** and **<operation>** elements

```
<portType name="StockQuotePortType">
   <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
   </operation>
</portType>
```

- A portType is analogous to a class
- An operation is analogous to a method in that class

# Types of <operation>

- There are four distinct types of operation
- Synchronous
  - **Request-response** - The service receives a message and sends a reply
  - **Solicit-response** - The service sends a message and receives a reply message
- Asynchronous
  - **One-way** - The service receives a message
  - **Notification** - The service sends a message
- All of these can be defined in WSDL

# Defining the type of operation

Enabling Grids for E-science in Europe

- Presence and order of input/output elements defines the type of operation.
- Request-response `<input><output>`
- Solicit-response `<output><input>`
- One-way `<input>` only
- Notification `<output>` only

# The <binding> element

- This element is used to define the mechanism that the client will actually use to interact with the web service
- There are three possibilities
  1. SOAP
  2. HTTP
  3. MIME
- The most common choice is currently SOAP
- The binding element defines the protocol specific information for the portTypes previously defined

# The binding tag

```
<binding name="ez3950SOAPBinding" type="tns:ez3950PortTypes">
```

The `<binding>` tag indicates that we will map a `<Port Type>` to a protocol

```
<soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http/">
```

Indicates we will be using the SOAP binding extensions to map the operations.
The alternative to “rpc” is “document”.

*( to use GET/POST use* `<http:binding…>`
*to use MIME use* `<mime:binding…..>` *)*

# <binding> Example

- Below is an example of a binding element for SOAP

```
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
   <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
   <operation name="GetLastTradePrice">
     <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
    </operation>
</binding>
```

# <service>

- The final component of a WSDL file is the **<service>** element
- The <service> element defines **<port>** elements that specify where requests should be sent

```
<service name="StockQuoteService">
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
        <soap:address location="http://example.com/stockquote"/>
    </port>
</service>
```

- The <soap:address> subelement identifies the URL of the service

- The precise content of <port> elements will be dependent upon the mechanism, i.e. SOAP, HTTP or MIME