



Enabling Grids for  
E-science in Europe

[www.eu-egee.org](http://www.eu-egee.org)

17<sup>th</sup> October 2004

**WSDL**

## Which WSDL Style ?



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

# Types of WSDL SOAP binding

- RPC/encoded
- RPC/literal
- Document/encoded
- Document/literal

# Java method example

```
public void myMethod (int x) ;
```

# RPC/Encoded WSDL

```
<message name="myMethodRequest" >  
  <part name="x" type="xsd:int" />  
</message>
```

```
<message name="empty" />
```

```
<portType name="PT" >  
  <operation name="myMethod" >  
    <input message="myMethodRequest" />  
    <output message="empty" />  
  </operation>  
</portType>
```

Binding is RPC/encoded

# SOAP message

```
<soap:envelope>  
  <soap:body>  
    <myMethod>  
      <x xsi:type="xsd:int">value</x>  
    </myMethod>  
  </soap:body>  
</soap:envelope>
```

# Advantages/disadvantages

- Advantages
  - Simple WSDL
  - Operation name appears in the message
  
- Disadvantages
  - Type encoding information overhead
  - SOAP message cannot be validated except against WSDL

# RPC/Literal WSDL

```
<message name="myMethodRequest" >
  <part name="x" type="xsd:int" />
</message>
<message name="empty" />

<portType name="PT" >
  <operation name="myMethod" >
    <input message="myMethodRequest" />
    <output message="empty" />
  </operation>
</portType>
```

Binding is RPC/literal

# SOAP message

```
<soap:envelope>  
  <soap:body>  
    <myMethod>  
      <x>value</x>  
    </myMethod>  
  </soap:body>  
</soap:envelope>
```



# Advantages/Disadvantages

- Advantages
  - WSDL is simple
  - Operation name appears in the message
  - Type encoding information is minimal
  
- Disadvantages
  - Nearly all the definitions in WSDL so not independently validatable

# Document/encoded

- Not implemented !

# Document/literal WSDL

```
<types>
  <schema>
    <element name="xElement" type="xsd:int" />
  </schema>
</types>

<message name="myMethodRequest">
  <part name="x" element="xElement" />
</message>
<message name="empty" />

<portType name="PT">
  <operation name="myMethod">
    <input message="myMethodRequest" />
    <output message="empty" />
  </operation>
</portType>
```

# SOAP message

```
<soap:envelope>  
  <soap:body>  
    <xElement>value</xElement>  
  </soap:body>  
</soap:envelope>
```

# Advantages/Disadvantages

- Advantages
  - No type encoding information
  - The body of the soap message is all defined in a schema and so can be validated independently
- Disadvantages
  - WSDL is more complicated
  - Operation name is lost

# Document/wrapped WSDL

```
<types>
  <schema>
    <element name="myMethod" />
      <complexType>
        <sequence>
          <element name="x" type="xsd:int" />
        </sequence>
      </complexType>
    </element>
  </schema>
</types>
<message name="myMethodRequest">
  <part name="parameters" element="myMethod" />
</message>
<message name="empty" />

<portType name="PT">
  <operation name="myMethod">
    <input message="myMethodRequest" />
    <output message="empty" />
  </operation>
</portType>
```

WSDL schema has a wrapper around the parameters

# SOAP message

```
<soap:envelope>  
  <soap:body>  
    <myMethod>  
      <x>value<x>  
    </myMethod>  
  </soap:body>  
</soap:envelope>
```

# Characteristics

- Input message has a single part
- Part is an element
- Element has the same name as the operation
- Element's complex type has no attributes



# Advantages/disadvantages

- Advantages
  - No type encoding information
  - Soap body is defined in a schema – validation
  - Method name in the soap message
  
- Disadvantages
  - WSDL is complicated
  
- Generally this is the best style to use.

## When not to use document/wrapped

- Document literal wrapped style does not allow for overloading
- Cannot have two elements with the same name in XML (element has to have same name as operation)
- In this case you may wish to use RPC/literal so that the operation name is available.