



WSDL

Stop a while to read about me!

Part of the code shown in the following slides is taken from the book "Java Web Services" by D.A. Chappell and T. Jawell, O'Reilly, ISBN 0-596-00269-6



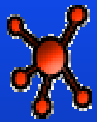
What is WSDL?

Web Services Description Language

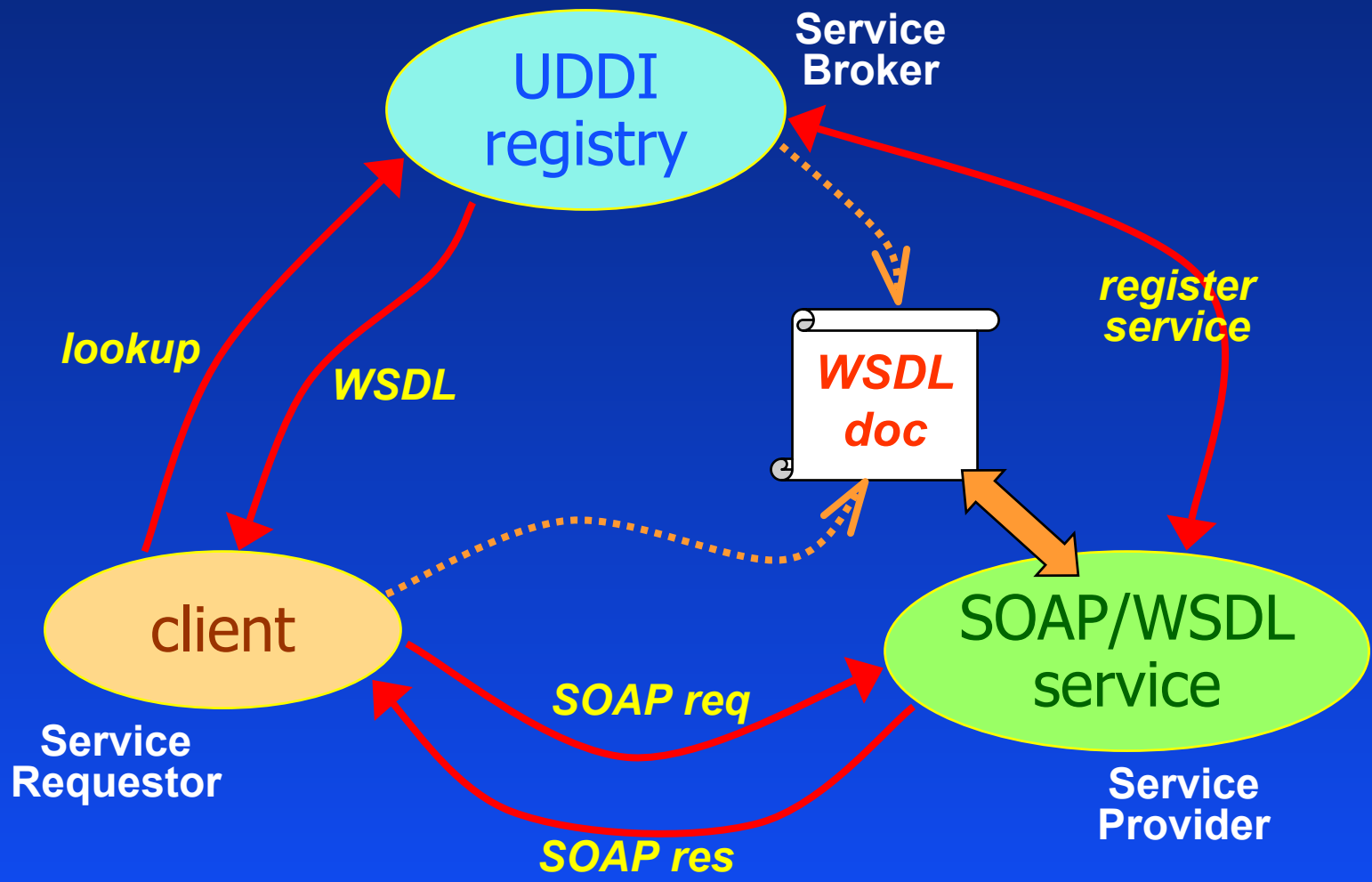
Service Description:

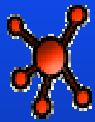
A **contract** between service provider and consumer

- Platform and language independent (unlike XML-RPC)
- Tools automate process of locating, invoking web service functionality



Role of WSDL in WS scene





Aspects of Service Description

WSDL describes four aspects of the service:

- Interface information
 - What methods are publicly available?
- Data type information
 - What info is needed by message requests and sent back by responses?
- Binding information
 - Which “transport protocols” can be used?
- Address information
 - Where is the service located?



Two Key Concepts for WSDL

- **Services** are a collections of network endpoints, or **ports**
- Separation between the **abstract definition of an endpoint**, and its concrete **network deployment**



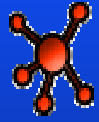
WSDL Terminology

- Messages
 - Description of data being exchanged
- Port types
 - Collections of operations
- Binding
 - Concrete protocol and data format specifications for a particular port type
- Port
 - Binding + network address
- Service
 - Collection of ports



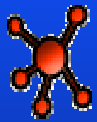
WSDL Elements (I)

- `<definitions> ... </definitions>`
 - **Root element** of WSDL specification
 - Defines name of the web service, multiple namespaces
- `<types>...</types>`
 - Describes **types used between the client and server**
 - Default: XML Schema specification
- `<message>...</message>`
 - Describes data exchanged in WS usage
 - A one-way message, containing zero or more `<part>...</part>` elements



WSDL Elements (II)

- `<portType>...</portType>`
 - Combines multiple messages to form **operation(s)**
- `<binding>...</binding>`
 - Specifics of how a service (portType) is actually implemented in terms of a protocol, SOAP specifics, ...
- `<service>...</service>`
 - One or more ports, each with its own network address



<definitions> </definitions>

- Can specify a *name* for the service
- Also defines *additional namespaces* used in the rest of the document

```
<definitions
  name="HertzReserveService"

  targetNamespace="http://oreilly.jaws.book/chapter03/hertz_reserve"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://oreilly.jaws.book/chapter03/hertz_reserve"
  xmlns:defs="http://oreilly.jaws.book/chapter03/definitions"
  xmlns:hrs="http://oreilly.jaws.book/chapter03/schemas"
>
... ["body" of the WSDL document]
</definitions>
```

service name

ns defined by this doc

Other used ns



<types> </types>

- Defines the types used in messages
- WSDL itself is not tied to any specific typing system, but it uses the XML
- **Schema specification** (XSD) is the default choice
- Built-in types,
simple and complex types
(as possible through XML Schema)

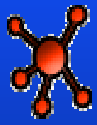


<message> </message>

- “name” attribute specifies the message name
- <part> ... </part> specifies one data item (type + name)
- **By convention**, SOAP RPC msgs have only one <part></part> element: a structure containing the actual parameters

```
<message name="soapHeader">
  <part type="xsd:string" name="id"/>
  <part type="xsd:string" name="timeout"/>
</message>
```

```
<message name="initRequest">
  <part type="xer:initRequest" name="initRequest"/>
</message>
```



<portType> </portType>

- An abstract collection of *operation(s)*
- Each operation consists of a pattern of messages
- Message names must be namespace-qualified
- Four basic patterns
 - One-way
 - Request-response
 - Solicit-response
 - Notification



PortType operations examples

```
<operation name="search">  
  <input message="searchRequest"/>  
  <output message="searchResponse"/>  
</operation>
```



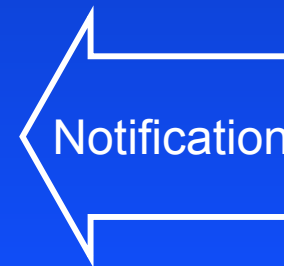
```
<operation name="resourceControl">  
  <output message="resourceControlResponse"/>  
  <input message="resourceControlRequest"/>  
</operation>
```



```
<operation name="triggerResourceControl">  
  <input message="triggerResourceControlRequest"/>  
</operation>
```



```
<operation name="segment">  
  <output message="segmentRequest"/>  
</operation>
```

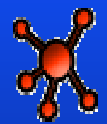


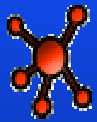
<binding></binding>



```
<binding name="MainClassSoap" type="s0:MainClassSoap">  
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"  
    style="document" />  
  <operation name="Compute">  
    <soap:operation  
      soapAction="http://www..." style="document" />  
    <input> <soap:body use="literal" /> </input>  
    <output> <soap:body use="literal" /> </output>  
  </operation>  
</binding>
```

- Provides information about how “Compute” operation messages are transported over the Internet





SOAP Binding

- **Built-in extensions to express SOAP-specific details**
 - SOAP headers, encoding styles
 - SOAPAction HTTP header (identifies the service)
- **<soap:binding>**
 - Indicates binding will be made available via SOAP
 - **style** attribute indicates message format
 - ☞ document: simple XML documents (.NET preference)
 - ☞ rpc: additional wrapper element indicating the function name
- **<soap:operation>**
 - Indicates binding of a specific operation to a specific SOAP implementation (SOAPAction header)
- **<soap:body>**
 - For each operation, specifies details of the input/output messages
 - ☞ Encoding, header blocks, headerfault, fault, ...



`<service> </service>`

Web Services

- Service element specifies **location** of one or more ports
 - **binding** denotes portType (hence, operations, messages, types)
 - **location** provides info about where service is “physically” accessible
 - ☞ soap:address value goes into HTTP POST header
 - ☞ Used by the SOAP-enabled web server to route SOAP requests
- Can optionally contain human-readable documentation describing the service



<service> example

```
<service name="Oxford University Libraries">
```

```
  <documentation>
```

```
    Z39.50 Server for Oxford University Libraries
```

```
  </documentation>
```

```
  <port name="OLIS" binding="ez:ez3950SOAPBinding">
```

```
    <soap:address location="http://jafer.las.ox.ac.uk/ez3950"/>
```

```
  </port>
```

```
</service>
```



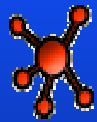
comments



Binding name



location



<import>

- Its purpose is similar to that of C/C++ #include: allows the elements of a service definition to be organized upon separate independent files
- A WSDL document can have none or multiple <import> elements
- The <import> scope is the element the tag is used within
- The import action operates **through namespaces**

```
<import namespace="http://oreilly.jaws.book/ch03/definitions"  
  location="file:///d:/jaws/ch03/wsd1/hr/hertz_reserve.wsd1"  
>
```



```
<import namespace="http://asf.gils.net/xer"  
  location=http://asf.gils.net/xer/ez.xsd  
>
```

