Web Services Description Lanugage (WSDL)

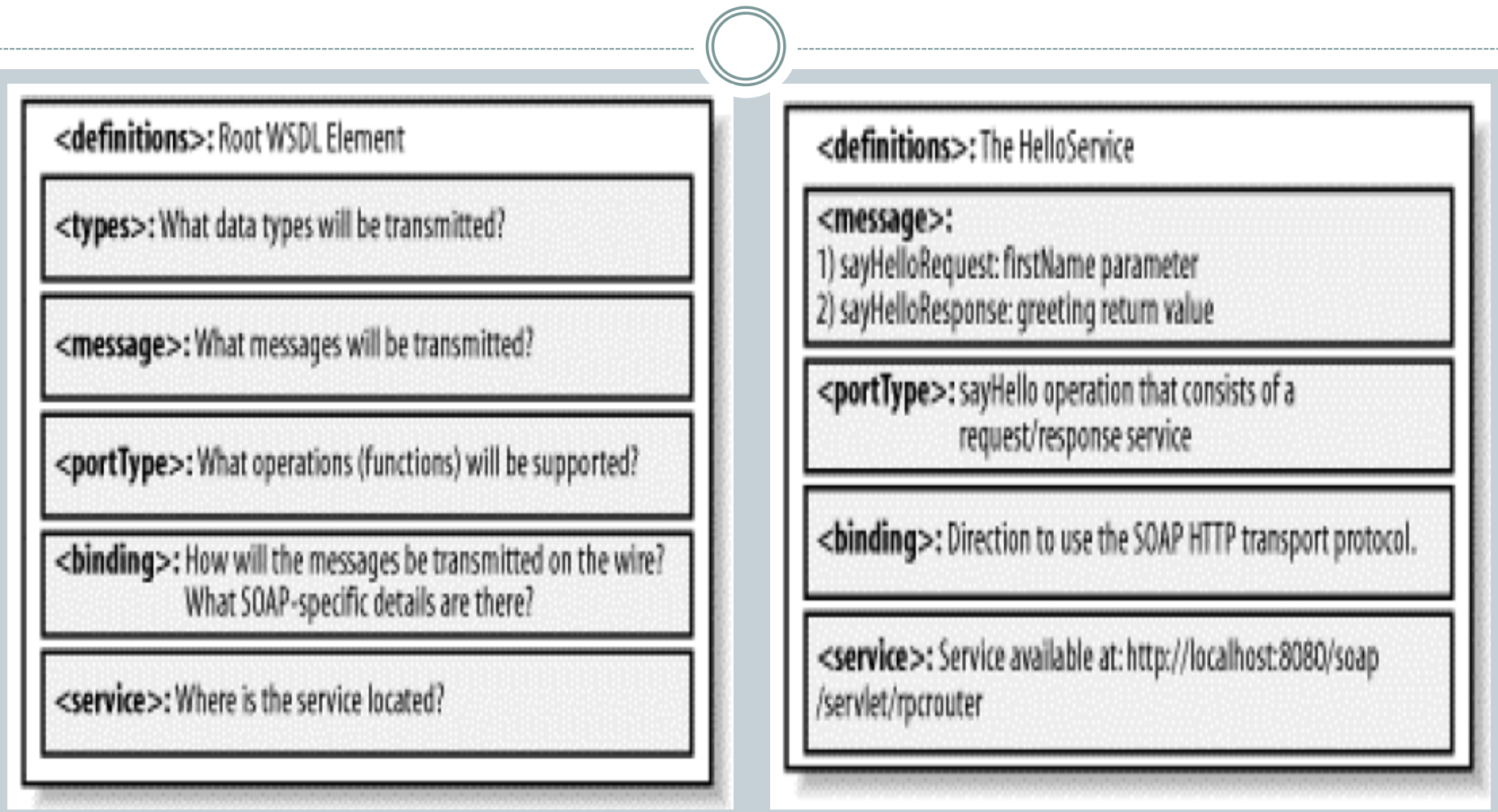
Introduction

- WSDL is an XML based protocol for information exchange in decentralized and distributed environments.
- WSDL is the standard format for describing a web service.
- WSDL definition describes how to access a web service and what operations it will perform.
- WSDL is a language for describing how to interface with XML-based services.
- WSDL is an integral part of UDDI, an XML-based worldwide business registry.
- WSDL was developed jointly by Microsoft and IBM.

Introduction

- WSDL describes four critical pieces of data:
  - Interface information describing all publicly available functions
  - Data type information for all message requests and message responses
  - Binding information about the transport protocol to be used
  - Address information for locating the specified service

The WSDL specification in a nutshell



WSDL Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="HelloService"
  targetNamespace="http://www.ecerami.com/wsdl/HelloService.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.ecerami.com/wsdl/HelloService.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <message name="SayHelloRequest">
    <part name="firstName" type="xsd:string"/>
  </message>
  <message name="SayHelloResponse">
    <part name="greeting" type="xsd:string"/>
  </message>

  <portType name="Hello_PortType">
    <operation name="sayHello">
      <input message="tns:SayHelloRequest"/>
      <output message="tns:SayHelloResponse"/>
    </operation>
  </portType>
```

WSDL Example

```
<binding name="Hello_Binding" type="tns:Hello_PortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="SayHello">
    <soap:operation soapAction="http://tempuri.org/sh/SayHello" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
<service name="Hello_Service">
  <port name="Hello_Port" binding="tns:Hello_Binding">
    <soap:address location="http://localhost:53666/Service1.svc"/>
  </port>
</service>
</definitions>
```

definitions

- Root element of all WSDL documents.
- Defines the name of the web service
- Declares multiple namespaces used throughout the remainder of the document
- targetNamespace is a convention of XML Schema that enables the WSDL document to refer to itself. The namespace specification does not require that the document actually exist at this location; Value specified should be unique and different from all other namespaces that are defined.

```
<definitions name="HelloService"
  targetNamespace="http://www.ecerami.com/wsdl/HelloService.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.ecerami.com/wsdl/HelloService.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">


</definitions>
```

types

- Describes all the data types used between the client and server.
- W3C XML schema specification is the default choice.
- If the service uses only XML Schema built-in simple types, such as strings and integers, the types element is not required.

message

- message
  - Describes a one-way message, whether it is a single message request or a single message response.
  - Defines the name of the message
  - Contains zero or more message part elements, which can refer to message parameters or message return values.
  - For the request, the part specifies the function parameters; For the response, the part specifies the function return values;
  - Part element's type attribute specifies an XML Schema data type.

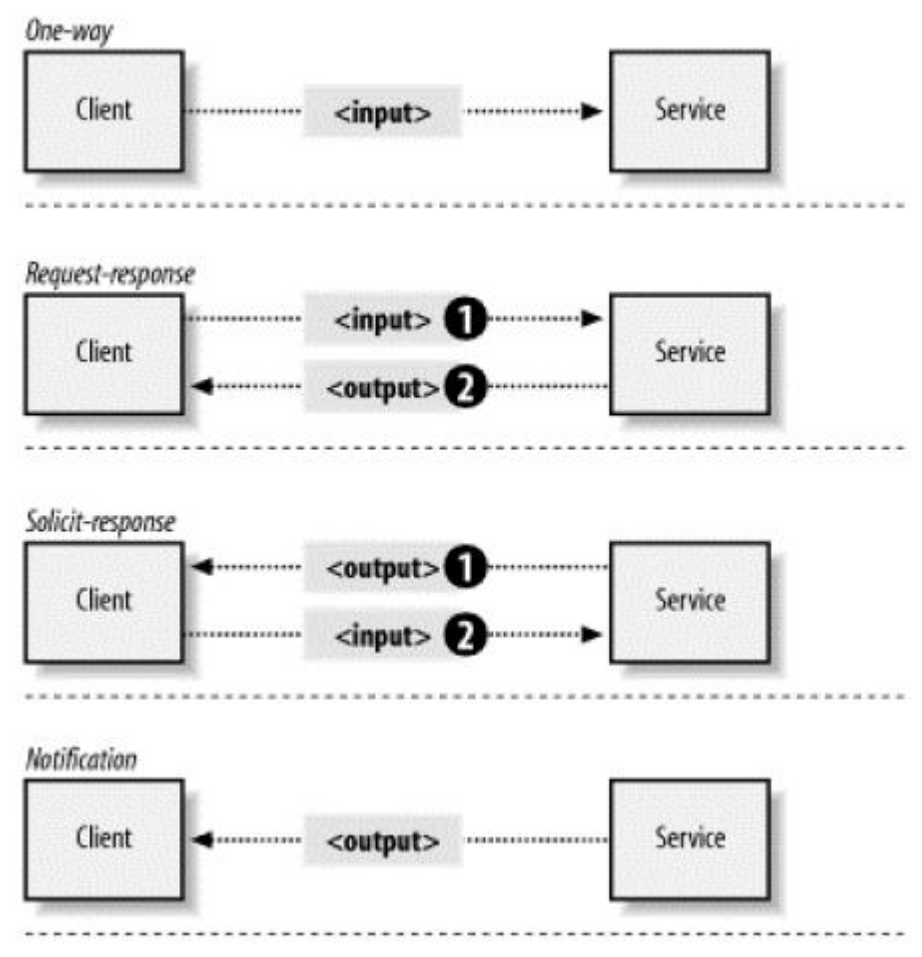
```
<message name="SayHelloRequest">
  <part name="firstName" type="xsd:string"/>
</message>
<message name="SayHelloResponse">
  <part name="greeting" type="xsd:string"/>
</message>
```

portType

- Combines multiple message elements to form a complete one way or round-trip operation.
- For example, a portType can combine one request and one response message into a single request/response operation, most commonly used in SOAP services.

```
<portType name="Hello_PortType">
  <operation name="sayHello">
    <input message="tns:SayHelloRequest"/>
    <output message="tns:SayHelloResponse"/>
  </operation>
</portType>
```

Operation patterns supported by WSDL 1.1



binding

- Describes the concrete specifics of how the service will be implemented on the wire.
- Bindings can be made available via multiple transports, including HTTP GET, HTTP POST, or SOAP
- The type attribute references the portType defined earlier in the document.

```
<binding name="Hello_Binding" type="tns:Hello_PortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="SayHello">
    <soap:operation soapAction="http://tempuri.org/sh/SayHello" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

binding (contd.)

- SOAP binding
  - Indicates that the binding will be made available via SOAP.
  - The style attribute indicates the overall style of the SOAP message format.
    - A style value of rpc specifies an RPC format. This means that the body of the SOAP request will include a wrapper XML element indicating the function name. Function parameters are then embedded inside the wrapper element. Likewise, the body of the SOAP response will include a wrapper XML element that mirrors the function request. Return values are then embedded inside the response wrapper element.
    - A style value of document specifies an XML document call format. This means that the request and response messages will consist simply of XML documents. The document style is flatter than the rpc style and does not require the use of wrapper elements.
  - The transport attribute indicates the transport of the SOAP messages. The value http://schemas.xmlsoap.org/soap/http indicates the SOAP HTTP transport, whereas http://schemas.xmlsoap.org/soap/smtp indicates the SOAP SMTP transport.

binding (contd.)

- soap:operation
  - Indicates the binding of a specific operation to a specific SOAP implementation.
  - The soapAction attribute specifies that the SOAPAction HTTP header be used for identifying the service.
- soap:body
  - This element enables you to specify the details of the input and output messages. In the case of HelloWorld, the body element specifies the SOAP encoding style and the namespace URN associated with the specified service.

service

- Defines the address for invoking the specified service. Most commonly, this includes a URL for invoking the SOAP service.
- documentation element provides human-readable documentation.

```
<service name="Hello_Service">
  <documentation>WSDL File for HelloService</documentation>
  <port binding="tns:Hello_Binding" name="Hello_Port">
    <soap:address
      location="http://localhost:8080/soap/servlet/rpcrouter"/>
  </port>
</service>
```

Soap Headers

If you want to pass username and password through the soap header then following changes are required.

Add the following message and modify the binding.

```
<message name="SayHelloRequest_Headers">
  <part name="Password" element="tns:Password"/>
  <part name="UserName" element="tns:UserName"/>
</message>

<binding name="Hello_Binding" type="tns:Hello_PortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="SayHello">
    <soap:operation soapAction="http://tempuri.org/sh/SayHello" style="document"/>
    <input name="SayHelloRequest">
      <soap:header message="tns:SayHelloRequest_Headers" part="Password" use="literal"/>
      <soap:header message="tns:SayHelloRequest_Headers" part="UserName" use="literal"/>
      <soap:body use="literal"/>
    </input>
    <output name="SayHelloResponse">
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```