

# **Core WS standards**

Antonio Brogi

Department of Computer Science  
University of Pisa

Short recap of XML

# EXtensible Markup Language

- Markup languages use “tags” to mark text

This *is* a **word**

can be coded (in HTML) as

```
This <i>is</i> a <b>word</b>
```

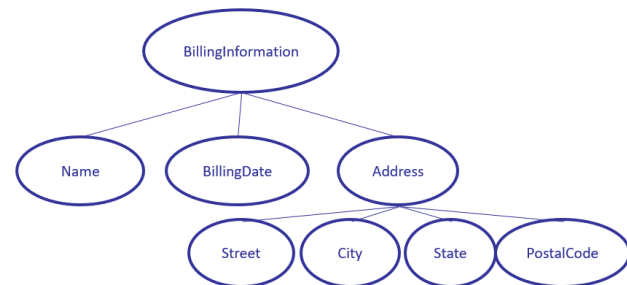
- XML portability: machine/platform/OS/language independence
- XML notion of *document type*
- XML documents consist of named constructors (declarations, elements, attributes), and of the data values therein contained
- An XML document is also called *instance* or *XML document instance* to recall it represents one possible set of data

```
<?xml version="1.0" encoding="UTF-8"?>
<BillingInformation customer-type="manufacturer">
  <Name> Right Plastic Products </Name>
  <BillingDate> 2002-09-15 </BillingDate>
  <Address>
    <Street> 158 Edward st. </Street>
    <City> Brisbane </City>
    <State> QLD </State>
    <PostalCode> 4000 </PostalCode>
  </Address>
</BillingInformation>
```

XML declaration (version, type of encoding)

Elements contain data, nested elements, or both

Nested elements define tree structure



Name-value **attributes** can be used to specify content of elements

# XML namespaces

- Designers can choose any names for tags
- Elements/attributes can be associated with *namespaces*

```
<?xml version="1.0" encoding="UTF-8"?>
<BillingInformation customer-type="manufacturer"
  xmlns="http://www.plastics_supply.com/BillInfo">
  <Name> Right Plastic Products </Name>
  <Address xmlns="http://www.plastics_supply.com/Addr">
    <Street> 158 Edward st. </Street>
    <City> Brisbane </City>
    <State> QLD </State>
    <PostalCode> 4000 </PostalCode>
  </Address>
  <BillingDate> 2002-09-15 </BillingDate>
</BillingInformation>
```

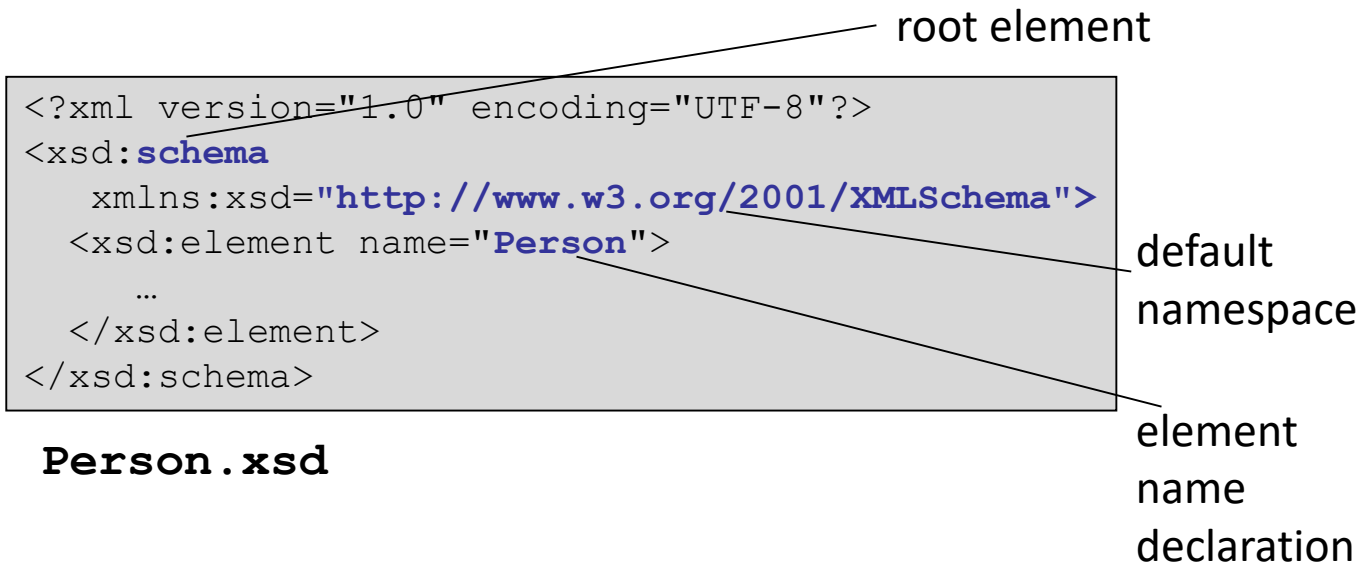
namespaces (URI)

- Scope of namespace: element in which it is declared and all its descendants
- Shorthand notation: *qualified names*

```
xmlns:bi="http://www.plastics_supply.com/BillInfo"
...
<bi:BillingDate> 2002-09-15 </bi:BillingDate>
```

# XML schemas

- Define content and structure of a class of XML documents
- Permit to specify structure relationships, cardinality, valid values, data types
- Very useful to **validate** XML documents
- XML Schema Definition Language (XSD)



# Simple and complex types

```
<xsd:element name="First" type="xsd:string"/>
```

```
<xsd:simpleType>  
  <xsd:restriction base="xsd:decimal">  
    <xsd:fractionDigits value="2"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsd:schema  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
  <xsd:element name="Person">  
    <xsd:complexType>  
      <xsd:sequence>  
        <xsd:element name="First"  
          type="xsd:string"/>  
        <xsd:element name="Middle"  
          type="xsd:string"  
          minOccurs="0"/>  
        <xsd:element name="Last"  
          type="xsd:string"/>  
        <xsd:element name="Age"  
          type="xsd:integer"/>  
      </xsd:sequence>  
    </xsd:complexType>  
  </xsd:element>  
</xsd:schema>
```

Type compositor

- **sequence**
- **all** (order not relevant)
- **choice** (only one)

If minOccurs/maxOccurs  
not present then default is 1

# XML Path Language

- Permits to create expressions to find specific pieces of information in a XML document
- XPath data model views document as a tree  
(attributes “associated” to node by dashed lines, not children of node)
- Example of XPath query

**Xpath Query#1: /PurchaseOrder/Order[2]/child::\***

1. Locate this element in tree

2. Locate second **Order** element within **PurchaseOrder**

3. Select all child elements of second **Order** element  
(located at the previous step)

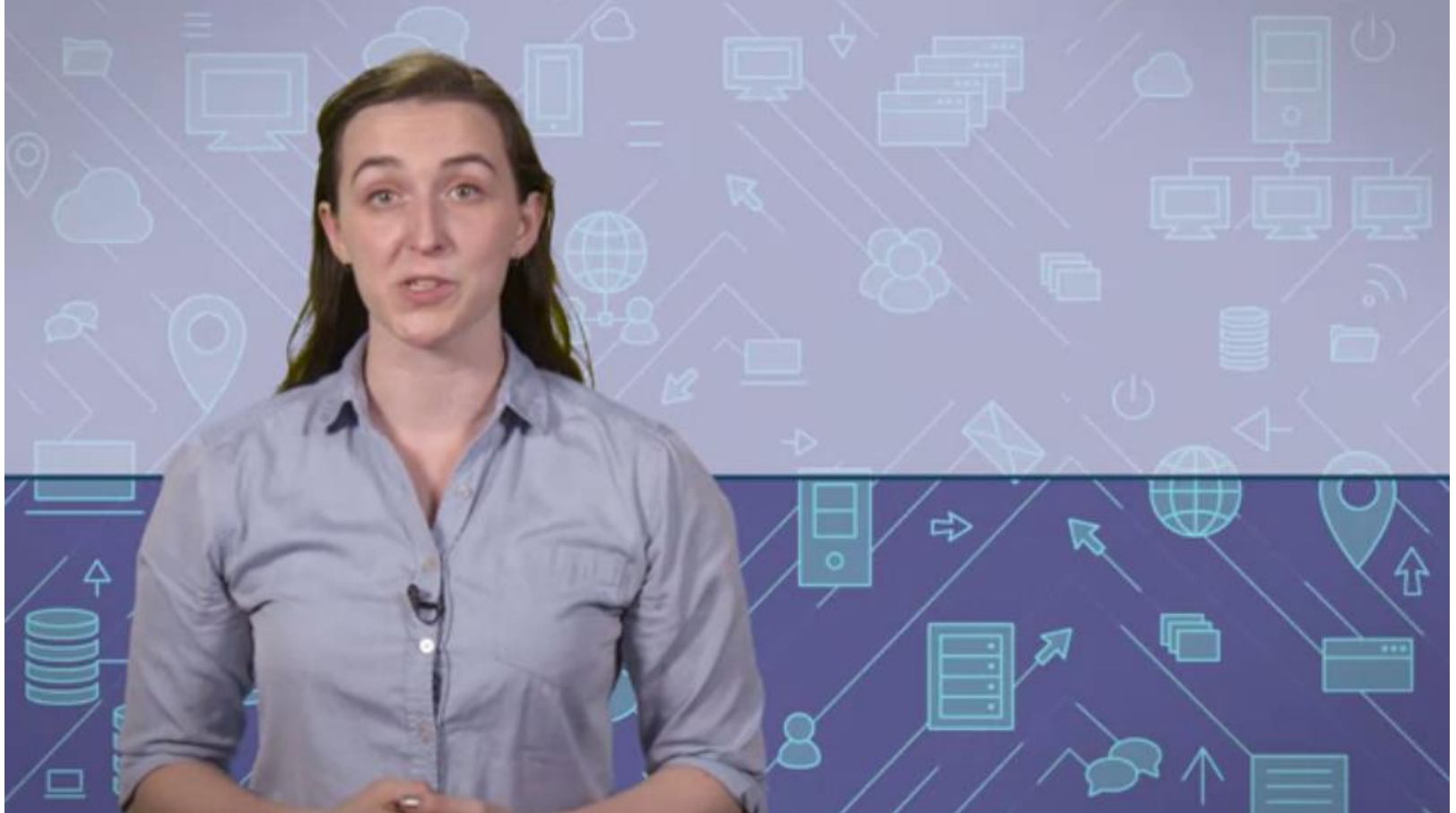


</xml>

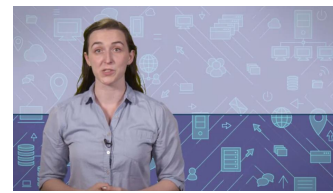
<http://www.w3.org/standards/xml/>

Short recap of XML

SOAP



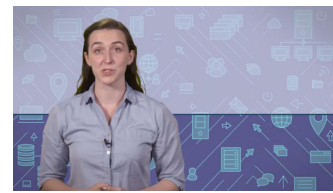
# Summary



- SOAP allows service requesters to invoke services
  - based on XML, transported over HTTP
- SOAP envelope
  - header (control info, optional)
  - body (payload)
- Types of communication
  - RPC vs. document style
  - synchronous vs. asynchronous communication
  - messaging patterns
    - request-response, solicit-response
    - one way, notification
- Conclusions
  - + machine readable
  - + interoperability
  - XML overhead



# Summary



- SOAP allows service requesters to invoke services
  - based on XML, transported over HTTP
- SOAP envelope
  - header (control info, optional)
  - body (payload) (or fault message)
- Types of communication
  - RPC vs. document style
  - synchronous vs. asynchronous communication
  - messaging patterns
    - request-response, solicit-response
    - one way, notification
- Conclusions
  - + machine readable
  - + interoperability
  - XML overhead

|                     |          | encoding style |         |
|---------------------|----------|----------------|---------|
|                     |          | Literal        | Encoded |
| communication style | RPC      |                |         |
|                     | Document |                |         |

# Example of RPC-style request-response

```
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m="http://www.plastics_supply.com/product-prices">
  <env:Header>
    <tx:Transaction-id
      xmlns:tx="http://www.transaction.com/transaction"
      env:mustUnderstand='1'>
      512
    </tx:Transaction-id>
  </env:Header>
  <env:Body>
    <m:GetProductPrice>
      <product-id> 450R60P </product-id>
    </m:GetProductPrice >
  </env:Body>
</env:Envelope>
```

```
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m="http://www.plastics_supply.com/product-prices">
  <env:Header>
    ...
  </env:Header>
  <env:Body>
    <m:GetProductPriceResponse>
      <product-price> 134.32 </product-price>
    </m:GetProductPriceResponse>
  </env:Body>
</env:Envelope>
```

# Examples of document-style message

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    ...
  </env:Header>
  <env:Body>
    <po:PurchaseOrder orderDate="2004-12-02" xmlns:po="http://www.plastics_supply.com/POs">
      <po:from>
        <po:accountName>    RightPlastics    </po:accountName>
        <po:accountNumber>  PSC-0343-02    </po:accountNumber>
      </po:from>
      <po:to>
        <po:supplierName>  Plastic Supplies Inc.  </po:supplierName>
        <po:supplierAddress> Yara Valley Melbourne  </po:supplierAddress>
      </po:to>
      <po:product>
        <po:product-name> injection molder </po:product-name>
        <po:product-model> G-100T </po:product-model>
        <po:quantity> 2 </po:quantity>
      </po:product>
    </ po:PurchaseOrder >
  </env:Body>
</env:Envelope>
```

# SOAP request over HTTP: example

**POST /Purchase Order HTTP/1.1**

**Host: http://www.plastics\_supply.com**

**Content-Type: application/soap+xml;**

**charset = "utf-8"**

**Content-Length: ...**

```
<?xml version="1.0" ?>
```

```
<env:Envelope
```

```
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
```

```
  xmlns:m="http://www.plastics_supply.com/product-prices">
```

```
    <env:Header>
```

```
      <tx:Transaction-id
```

```
        xmlns:tx="http://www.transaction.com/transactions"
```

```
        env:mustUnderstand='1'>
```

```
          512
```

```
      </tx:Transaction-id>
```

```
    </env:Header>
```

```
    <env:Body>
```

```
      <m:GetProductPrice>
```

```
        <product-id> 450R60P </product-id >
```

```
      </m:GetProductPrice >
```

```
    </env:Body>
```

```
</env:Envelope>
```



# SOAP response over HTTP: example

```
HTTP/1.1 200 OK
```

```
Content-Type: application/soap+xml;
```

```
charset = "utf-8"
```

```
Content-Length: ...
```

```
<?xml version="1.0" ?>
```

```
<env:Envelope
```

```
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
```

```
  xmlns:m="http://www.plastics_supply.com/product-prices">
```

```
  <env:Header>
```

```
    <!--! - Optional context information -->
```

```
  </env:Header>
```

```
  <env:Body>
```

```
    <m:GetProductPriceResponse>
```

```
      <product-price> 134.32 </product-price>
```

```
    </m:GetProductPriceResponse>
```

```
  </env:Body>
```

```
</env:Envelope>
```

</env:Envelope>

[http://www.w3.org/standards/techs/soap#w3c\\_all](http://www.w3.org/standards/techs/soap#w3c_all)

Short recap of XML

SOAP

WSDL

# Motivation

- SOAP defines how to format and to process communication messages
- To send a message to a service S, sender must know
  - **names** and **parameters** of operations featured by S,

```
<env:Envelope ... >
  ...
  <env:Body>
    <m:GetProductPrice>
      <product-id> 450R60P </product-id>
    </m:GetProductPrice >
  </env:Body>
</env:Envelope>
```

- and the “**address**” of S,
- and which **transport protocol** to use

→ need of a (machine-understandable) description of services

# WSDL

- WSDL: platform- and language-independent, XML-based language for describing *public interface* of a service
- Public interface kind of “contract” between provider and consumer, describes only info relevant to both parties – no implementation details
- WSDL used to describe
  - *What* a service does
  - *Where* it resides
  - *How* to invoke it
- WSDL describes a service as a set of endpoints operating on messages
- SOAP+WSDL: enough to allow interoperation of heterogeneous applications!

# WSDL

- WSDL description consists of two parts
  - Abstract interface
    - operations, parameters, abstract data types
  - Concrete interface
    - binding to network address, specific protocol, and concrete data structures
- Service can support different bindings of same abstract interface, each accessible at a different address (endpoint)
- WSDL description contains all\* the info requester needs to be able to communicate with provider

Short recap of XML

SOAP

WSDL

abstract interfaces

```

<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"      xmlns:xsd="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"   xmlns="http://schemas.xmlsoap.org/wsdl/">
  <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>
  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd:TradePriceRequest"/>
  </message>
  <message name="GetLastTradePriceOutput">
    <part name="body" element="xsd:TradePrice"/>
  </message>
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>
  <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 name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>

```

types



```

<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"      xmlns:xsd="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"    xmlns="http://schemas.xmlsoap.org/wsdl/">
  <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>
  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd:TradePriceRequest"/>
  </message>
  <message name="GetLastTradePriceOutput">
    <part name="body" element="xsd:TradePrice"/>
  </message>
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>
  <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 name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>

```

Messages (possibly structured into parts)

```

<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"      xmlns:xsd="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"   xmlns="http://schemas.xmlsoap.org/wsdl/">
<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>
<message name="GetLastTradePriceInput">
    <part name="body" element="xsd:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd:TradePrice"/>
</message>
<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>
<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 name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
        <soap:address location="http://example.com/stockquote"/>
    </port>
</service>
</definitions>

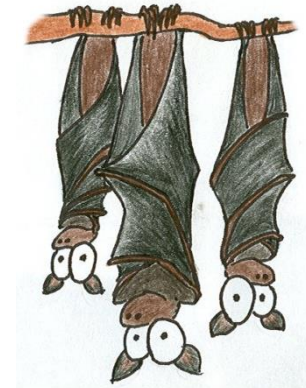
```

featured operation(s), grouped in portType(s)

```

<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"      xmlns:xsd="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <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>
  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd:TradePriceRequest"/>
  </message>
  <message name="GetLastTradePriceOutput">
    <part name="body" element="xsd:TradePrice"/>
  </message>
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>
  <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 name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>

```



Abstract interface

Short recap of XML

SOAP

WSDL

abstract interfaces

concrete interfaces

# WSDL concrete interface specifies

a **binding** that defines how **portType** and **operations** (of abstract interface) are mapped into concrete representations via a combination of

- concrete protocols (e.g., SOAP over HTTP, SOAP over SMTP)
- messaging styles (e.g., RPC- or document-style)
- encoding styles (e.g., literal or SOAP encoding)

```

<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"      xmlns:xsd="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  xmlns="http://schemas.xmlsoap.org/wsdl/">
<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>
<message name="GetLastTradePriceInput">
    <part name="body" element="xsd:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd:TradePrice"/>
</message>
<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>
<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 name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
        <soap:address location="http://example.com/stockquote"/>
    </port>
</service>
</definitions>

```

binding

# WSDL concrete interface

Finally **service** binds service to a specific net location by specifying one (or more) **ports**

- ❑ **port** associates an *endpoint* with a **binding**
- ❑ same service interface can have multiple implementations, all with different addresses (and possibly with different bindings)

```

<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"      xmlns:xsd="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"   xmlns="http://schemas.xmlsoap.org/wsdl/">
<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>
<message name="GetLastTradePriceInput">
    <part name="body" element="xsd:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd:TradePrice"/>
</message>
<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>
<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 name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
        <soap:address location="http://example.com/stockquote"/>
    </port>
</service>
</definitions>

```

service



```

<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"      xmlns:xsd="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"   xmlns="http://schemas.xmlsoap.org/wsdl/">
  <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>
  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd:TradePriceRequest"/>
  </message>
  <message name="GetLastTradePriceOutput">
    <part name="body" element="xsd:TradePrice"/>
  </message>
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>
  <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap" />
    <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 name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>

```

Abstract interface



Concrete interface

```
<?xml version="1.0"?>
<schema targetNamespace="http://example.com/stockquote/schemas"
  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>
```

<http://example.com/stockquote/stockquote.xsd>

```
<?xml version="1.0"?>
<definitions name="StockQuote"

targetNamespace="http://example.com/stockquote/definitions"
  xmlns:tns="http://example.com/stockquote/definitions"
  xmlns:xsd="http://example.com/stockquote/schemas"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <import namespace="http://example.com/stockquote/schemas"
    location="http://example.com/stockquote/stockquote.xsd"/>

  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd:TradePriceRequest"/>
  </message>

  <message name="GetLastTradePriceOutput">
    <part name="body" element="xsd:TradePrice"/>
  </message>

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

<http://example.com/stockquote/stockquote.wsdl>

```
<?xml version="1.0"?>
<definitions name="StockQuote"

targetNamespace="http://example.com/stockquote/service"
  xmlns:tns="http://example.com/stockquote/service"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:defs="http://example.com/stockquote/definitions"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <import namespace="http://example.com/stockquote/definitions"
    location="http://example.com/stockquote/stockquote.wsdl"/>

  <binding name="StockQuoteSoapBinding" type="defs: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 name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>
```

<http://example.com/stockquote/stockquoteservice.wsdl>

Short recap of XML

SOAP

WSDL

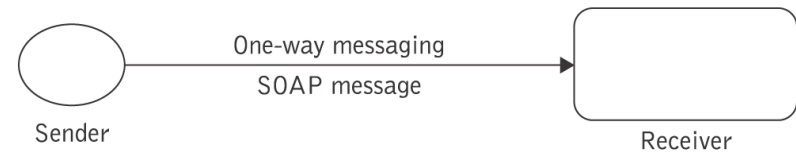
- abstract interfaces
- concrete interfaces
- message exchange patterns

# WSDL Message Exchange Patterns

WSDL interfaces support four types of operations

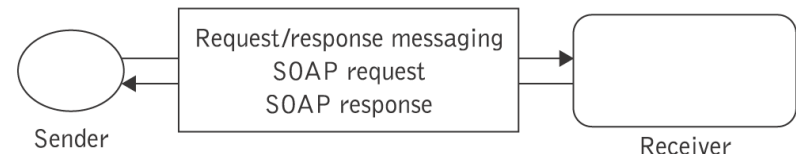
- One-way operations

- Service endpoint receives message but does not send response
- Typical asynchronous messaging
- One-way operation declared with an input element and no output elements



- Request-response operations

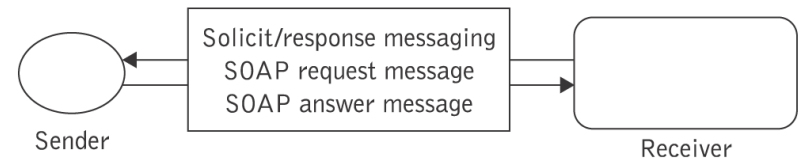
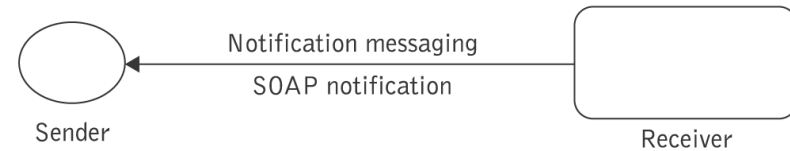
- Service receives request and sends response
- Typical remote procedure call
- operation declared with an input element *followed by* an output element



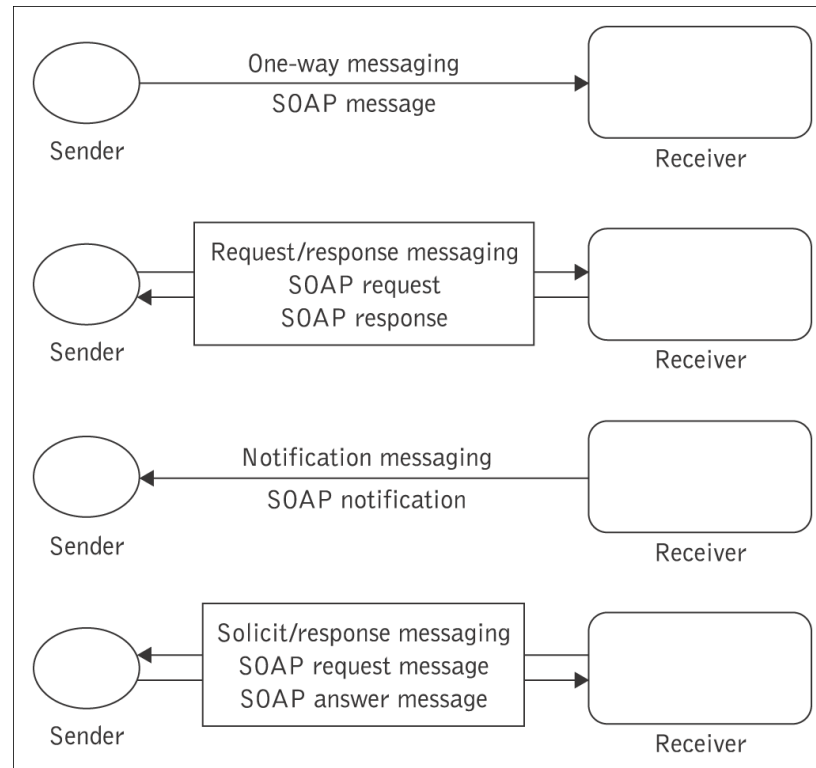
```
<operation name="GetLastTradePrice">  
  <input message="tns:GetLastTradePriceInput"/>  
  <output message="tns:GetLastTradePriceOutput"/>  
</operation>
```

# WSDL Message Exchange Patterns

- Notification operations
  - Service sends message to client but does not expect response
  - Typical use: event notification (“push”) to (subscribed) clients
  - `operation` contains only `output` element
- Solicit-response
  - Dual of request-response: here server sends request and waits for response
  - `operation` declares an `output` element *followed by* an `input` element



# WSDL Message Exchange Patterns



*Any combination of the above types of operations can be included in a single WSDL interface*

Short recap of XML

SOAP

WSDL

- abstract interfaces

- concrete interfaces

- message exchange patterns

- from WSDL to SOAP: an example



Invoking the *ChangeMetricWeightUnit* operation

featured by the  WebserviceX.NET

*ChangeMetricWeight* service to ask converting 2 kilograms into grams.

Recall

types, messages, operations, portTypes  
bindings, ports, services



```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.webserviceX.NET/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://www.webserviceX.NET/" xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/ ... >
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
    <s:element name="ChangeMetricWeightUnit">
      <s:complexType>
        <s:sequence>
          <s:element name="MetricWeightValue" type="s:double" />
          <s:element name="fromMetricWeightUnit" type="tns:MetricWeights" />
          <s:element name="toMetricWeightUnit" type="tns:MetricWeights" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:simpleType name="MetricWeights">
      <s:restriction base="s:string">
        <s:enumeration value="milligram"/> ... <s:enumeration value="hectogram"/> <s:enumeration value="kilogram"/>
      </s:restriction>
    </s:simpleType>
    <s:element name="ChangeMetricWeightUnitResponse">
      <s:complexType> <s:sequence> <s:element name="ChangeMetricWeightUnitResult" type="s:double"/> </s:sequence> </s:complexType>
    </s:element>
    <s:element name="double" type="s:double" />
  </s:schema>
</wsdl:types>
<wsdl:message name="ChangeMetricWeightUnitSoapIn">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnit" />
</wsdl:message>
<wsdl:message name="ChangeMetricWeightUnitSoapOut">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnitResponse" />
</wsdl:message>
...
<wsdl:portType name="MetricWeightUnitSoap">
  <wsdl:operation name="ChangeMetricWeightUnit">
    <wsdl:input message="tns:ChangeMetricWeightUnitSoapIn" /> <wsdl:output message="tns:ChangeMetricWeightUnitSoapOut" />
  </wsdl:operation>
</wsdl:portType>
...
<wsdl:binding name="MetricWeightUnitSoap" type="tns:MetricWeightUnitSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ChangeMetricWeightUnit">
    <soap:operation style="document" soapAction="http://www.webserviceX.NET/ChangeMetricWeightUnit" />
    <wsdl:input> <soap:body use="literal" /> </wsdl:input>
    <wsdl:output> <soap:body use="literal" /> </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
...
<wsdl:service name="MetricWeightUnit">
  <wsdl:port name="MetricWeightUnitSoap" binding="tns:MetricWeightUnitSoap">
    <soap:address location="http://www.webservicex.net/convertMetricWeight.asmx" />
  </wsdl:port>
  ...
</wsdl:service>
</wsdl:definitions>

```

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>

```

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.webserviceX.NET/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://www.webserviceX.NET/" xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/ ... >
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
    <s:element name="ChangeMetricWeightUnit">
      <s:complexType>
        <s:sequence>
          <s:element name="MetricWeightValue" type="s:double" />
          <s:element name="fromMetricWeightUnit" type="tns:MetricWeights" />
          <s:element name="toMetricWeightUnit" type="tns:MetricWeights" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:simpleType name="MetricWeights">
      <s:restriction base="s:string">
        <s:enumeration value="milligram"/> ... <s:enumeration value="hectogram"/> <s:enumeration value="kilogram"/>
      </s:restriction>
    </s:simpleType>
    <s:element name="ChangeMetricWeightUnitResponse">
      <s:complexType> <s:sequence> <s:element name="ChangeMetricWeightUnitResult" type="s:double"/> </s:sequence> </s:complexType>
    </s:element>
    <s:element name="double" type="s:double" />
  </s:schema>
</wsdl:types>
<wsdl:message name="ChangeMetricWeightUnitSoapIn">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnit" />
</wsdl:message>
<wsdl:message name="ChangeMetricWeightUnitSoapOut">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnitResponse" />
</wsdl:message>
...
<wsdl:portType name="MetricWeightUnitSoap">
  <wsdl:operation name="ChangeMetricWeightUnit">
    <wsdl:input message="tns:ChangeMetricWeightUnitSoapIn" /> <wsdl:output message="tns:ChangeMetricWeightUnitSoapOut" />
  </wsdl:operation>
</wsdl:portType>
...
<wsdl:binding name="MetricWeightUnitSoap" type="tns:MetricWeightUnitSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ChangeMetricWeightUnit">
    <soap:operation style="document" soapAction="http://www.webserviceX.NET/ChangeMetricWeightUnit" />
    <wsdl:input> <soap:body use="literal" /> </wsdl:input>
    <wsdl:output> <soap:body use="literal" /> </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
...
<wsdl:service name="MetricWeightUnit">
  <wsdl:port name="MetricWeightUnitSoap" binding="tns:MetricWeightUnitSoap">
    <soap:address location="http://www.webserviceX.NET/convertMetricWeight.aspx" />
  </wsdl:port>
  ...
</wsdl:service>
</wsdl:definitions>

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ChangeMetricWeightUnit xmlns="http://www.webserviceX.NET/">
```

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.webserviceX.NET/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://www.webserviceX.NET/" xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/ ... >
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
    <s:element name="ChangeMetricWeightUnit">
      <s:complexType>
        <s:sequence>
          <s:element name="MetricWeightValue" type="s:double" />
          <s:element name="fromMetricWeightUnit" type="tns:MetricWeights" />
          <s:element name="toMetricWeightUnit" type="tns:MetricWeights" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:simpleType name="MetricWeights">
      <s:restriction base="s:string">
        <s:enumeration value="milligram"/> ... <s:enumeration value="hectogram"/> <s:enumeration value="kilogram"/>
      </s:restriction>
    </s:simpleType>
    <s:element name="ChangeMetricWeightUnitResponse">
      <s:complexType> <s:sequence> <s:element name="ChangeMetricWeightUnitResult" type="s:double"/> </s:sequence> </s:complexType>
    </s:element>
    <s:element name="double" type="s:double" />
  </s:schema>
</wsdl:types>
<wsdl:message name="ChangeMetricWeightUnitSoapIn">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnit" />
</wsdl:message>
<wsdl:message name="ChangeMetricWeightUnitSoapOut">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnitResponse" />
</wsdl:message>
...
<wsdl:portType name="MetricWeightUnitSoap">
  <wsdl:operation name="ChangeMetricWeightUnit">
    <wsdl:input message="tns:ChangeMetricWeightUnitSoapIn" /> <wsdl:output message="tns:ChangeMetricWeightUnitSoapOut" />
  </wsdl:operation>
</wsdl:portType>
...
<wsdl:binding name="MetricWeightUnitSoap" type="tns:MetricWeightUnitSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ChangeMetricWeightUnit">
    <soap:operation style="document" soapAction="http://www.webserviceX.NET/ChangeMetricWeightUnit" />
    <wsdl:input> <soap:body use="literal" /> </wsdl:input>
    <wsdl:output> <soap:body use="literal" /> </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
...
<wsdl:service name="MetricWeightUnit">
  <wsdl:port name="MetricWeightUnitSoap" binding="tns:MetricWeightUnitSoap">
    <soap:address location="http://www.webservicex.net/convertMetricWeight.aspx" />
  </wsdl:port>
  ...
</wsdl:service>
</wsdl:definitions>

```

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <ChangeMetricWeightUnit xmlns="http://www.webserviceX.NET/">
    <MetricWeightValue>2</MetricWeightValue>
  </ChangeMetricWeightUnit>
</soap:Body>

```

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.webserviceX.NET/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://www.webserviceX.NET/" xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/ ... >
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
    <s:element name="ChangeMetricWeightUnit">
      <s:complexType>
        <s:sequence>
          <s:element name="MetricWeightValue" type="s:double" />
          <s:element name="fromMetricWeightUnit" type="tns:MetricWeights" />
          <s:element name="toMetricWeightUnit" type="tns:MetricWeights" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:simpleType name="MetricWeights">
      <s:restriction base="s:string">
        <s:enumeration value="milligram"/> ... <s:enumeration value="hectogram"/> <s:enumeration value="kilogram"/>
      </s:restriction>
    </s:simpleType>
    <s:element name="ChangeMetricWeightUnitResponse">
      <s:complexType> <s:sequence> <s:element name="ChangeMetricWeightUnitResult" type="s:double"/> </s:sequence> </s:complexType>
    </s:element>
    <s:element name="double" type="s:double" />
  </s:schema>
</wsdl:types>
<wsdl:message name="ChangeMetricWeightUnitSoapIn">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnit" />
</wsdl:message>
<wsdl:message name="ChangeMetricWeightUnitSoapOut">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnitResponse" />
</wsdl:message>
...
<wsdl:portType name="MetricWeightUnitSoap">
  <wsdl:operation name="ChangeMetricWeightUnit">
    <wsdl:input message="tns:ChangeMetricWeightUnitSoapIn" /> <wsdl:output message="tns:ChangeMetricWeightUnitSoapOut" />
  </wsdl:operation>
</wsdl:portType>
...
<wsdl:binding name="MetricWeightUnitSoap" type="tns:MetricWeightUnitSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ChangeMetricWeightUnit">
    <soap:operation style="document" soapAction="http://www.webserviceX.NET/ChangeMetricWeightUnit" />
    <wsdl:input> <soap:body use="literal" /> </wsdl:input>
    <wsdl:output> <soap:body use="literal" /> </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
...
<wsdl:service name="MetricWeightUnit">
  <wsdl:port name="MetricWeightUnitSoap" binding="tns:MetricWeightUnitSoap">
    <soap:address location="http://www.webservicex.net/convertMetricWeight.aspx" />
  </wsdl:port>
  ...
</wsdl:service>
</wsdl:definitions>

```

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ChangeMetricWeightUnit xmlns="http://www.webserviceX.NET/">
      <MetricWeightValue>2</MetricWeightValue>
      <fromMetricWeightUnit>kilogram</fromMetricWeightUnit>
    </ChangeMetricWeightUnit>
  </soap:Body>
</soap:Envelope>

```

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.webserviceX.NET/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://www.webserviceX.NET/" xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/ ... >
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
    <s:element name="ChangeMetricWeightUnit">
      <s:complexType>
        <s:sequence>
          <s:element name="MetricWeightValue" type="s:double" />
          <s:element name="fromMetricWeightUnit" type="tns:MetricWeights" />
          <s:element name="toMetricWeightUnit" type="tns:MetricWeights" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:simpleType name="MetricWeights">
      <s:restriction base="s:string">
        <s:enumeration value="milligram"/> ... <s:enumeration value="hectogram"/> <s:enumeration value="kilogram"/>
      </s:restriction>
    </s:simpleType>
    <s:element name="ChangeMetricWeightUnitResponse">
      <s:complexType> <s:sequence> <s:element name="ChangeMetricWeightUnitResult" type="s:double"/> </s:sequence> </s:complexType>
    </s:element>
    <s:element name="double" type="s:double" />
  </s:schema>
</wsdl:types>
<wsdl:message name="ChangeMetricWeightUnitSoapIn">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnit" />
</wsdl:message>
<wsdl:message name="ChangeMetricWeightUnitSoapOut">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnitResponse" />
</wsdl:message>
...
<wsdl:portType name="MetricWeightUnitSoap">
  <wsdl:operation name="ChangeMetricWeightUnit">
    <wsdl:input message="tns:ChangeMetricWeightUnitSoapIn" /> <wsdl:output message="tns:ChangeMetricWeightUnitSoapOut" />
  </wsdl:operation>
</wsdl:portType>
...
<wsdl:binding name="MetricWeightUnitSoap" type="tns:MetricWeightUnitSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ChangeMetricWeightUnit">
    <soap:operation style="document" soapAction="http://www.webserviceX.NET/ChangeMetricWeightUnit" />
    <wsdl:input> <soap:body use="literal" /> </wsdl:input>
    <wsdl:output> <soap:body use="literal" /> </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
...
<wsdl:service name="MetricWeightUnit">
  <wsdl:port name="MetricWeightUnitSoap" binding="tns:MetricWeightUnitSoap">
    <soap:address location="http://www.webservicex.net/convertMetricWeight.aspx" />
  </wsdl:port>
  ...
</wsdl:service>
</wsdl:definitions>
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ChangeMetricWeightUnit xmlns="http://www.webserviceX.NET/">
      <MetricWeightValue>2</MetricWeightValue>
      <fromMetricWeightUnit>kilogram</fromMetricWeightUnit>
      <toMetricWeightUnit>gram</toMetricWeightUnit>
    </ChangeMetricWeightUnit>
  </soap:Body>
</soap:Envelope>
```

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.webserviceX.NET/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://www.webserviceX.NET/" xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/ ... >
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
    <s:element name="ChangeMetricWeightUnit">
      <s:complexType>
        <s:sequence>
          <s:element name="MetricWeightValue" type="s:double" />
          <s:element name="fromMetricWeightUnit" type="tns:MetricWeights" />
          <s:element name="toMetricWeightUnit" type="tns:MetricWeights" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:simpleType name="MetricWeights">
      <s:restriction base="s:string">
        <s:enumeration value="milligram"/> ... <s:enumeration value="hectogram"/> <s:enumeration value="kilogram"/>
      </s:restriction>
    </s:simpleType>
    <s:element name="ChangeMetricWeightUnitResponse">
      <s:complexType> <s:sequence> <s:element name="ChangeMetricWeightUnitResult" type="tns:ChangeMetricWeightUnitResponseResult" />
    </s:element>
    <s:element name="double" type="s:double" />
  </s:schema>
</wsdl:types>
<wsdl:message name="ChangeMetricWeightUnitSoapIn">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnit" />
</wsdl:message>
<wsdl:message name="ChangeMetricWeightUnitSoapOut">
  <wsdl:part name="parameters" element="tns:ChangeMetricWeightUnitResponse" />
</wsdl:message>
...
<wsdl:portType name="MetricWeightUnitSoap">
  <wsdl:operation name="ChangeMetricWeightUnit">
    <wsdl:input message="tns:ChangeMetricWeightUnitSoapIn" /> <wsdl:output message="tns:ChangeMetricWeightUnitSoapOut" />
  </wsdl:operation>
</wsdl:portType>
...
<wsdl:binding name="MetricWeightUnitSoap" type="tns:MetricWeightUnitSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ChangeMetricWeightUnit">
    <soap:operation style="document" soapAction="http://www.webserviceX.NET/ChangeMetricWeightUnit" />
    <wsdl:input> <soap:body use="literal" /> </wsdl:input>
    <wsdl:output> <soap:body use="literal" /> </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
...
<wsdl:service name="MetricWeightUnit">
  <wsdl:port name="MetricWeightUnitSoap" binding="tns:MetricWeightUnitSoap">
    <soap:address location="http://www.webservicex.net/convertMetricWeight.aspx" />
  </wsdl:port>
  ...
</wsdl:service>
</wsdl:definitions>

```

```

POST /convertMetricWeight.aspx HTTP/1.1
Host: www.webservicex.net
Content-Type: text/xml; charset=utf-8
Content-Length: ...

```

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <ChangeMetricWeightUnit xmlns="http://www.webserviceX.NET/">
    <MetricWeightValue>2</MetricWeightValue>
    <fromMetricWeightUnit>kilogram</fromMetricWeightUnit>
    <toMetricWeightUnit>gram</toMetricWeightUnit>
  </ChangeMetricWeightUnit>
</soap:Body>
</soap:Envelope>

```

# Short recap of XML

## SOAP

## WSDL

- abstract interfaces

- concrete interfaces

- message exchange patterns

- from WSDL to SOAP: an example

- concluding remarks

# Concluding remarks 1/2

- WSDL 1.1 is the *de facto* standard to describe Web services
- WSDL public interface of service describes
  - ❑ Available operations
  - ❑ XML message protocols supported
  - ❑ Data types info for messages
  - ❑ Binding info on transport protocol to be used
  - ❑ Address info to locate service
- Note that WSDL bindings do not contain any programming language / service-specific detail
- WSDL 2.0 simpler and improved, its adoption may however be delayed by the wide spread and support of WSDL 1.1



# Concluding remarks 2/2

- Many **tools** available that automate the mapping between WSDL and programming languages, both for clients and providers
  - ❑ eg, AXIS Java2WSDL generates WSDL interface from Java code
- Example: using WSDL to generate client stubs
  - ❑ Proxy classes –also called *stub classes*- widely used by developers to reference remote Web services and use their functionalities as if the data returned by the service were generated locally
    - Proxy classes are client-side images of (remote) object classes that implement the Web service
    - Client-side application invokes proxy, proxy forwards invocation to remote instance (skeleton)
    - Proxy encapsulates marshalling of arguments and unmarshalling of results
  - ❑ Eg, AXIS WSDL2Java implements JAX-RPC specification to generate Java RMI interface and net *proxy* from WSDL
- Non-functional properties can be described by means of a separate language (WS-Policy, WS-Policy Attachment,...)

</wsdl>

[http://www.w3.org/standards/techs/wsdl#w3c\\_all](http://www.w3.org/standards/techs/wsdl#w3c_all)