

# SOAP

- **Simple Object Access Protocol.**
- SOAP is a protocol for communication between applications – it is a format for sending messages.
- Since SOAP is based on XML, it is:
  - Platform independent.
  - Can be transported over HTTP.
  - Can get around firewalls.

1

# Why?

- Current distributed applications communicate between them using **Remote Procedure Calls** (RPCs).
- Common protocols behind these RPCs are **DCOM** and **CORBA**.
- Since these protocols are not transported over HTTP they will be blocked by firewalls.
- Also DCOM and CORBA are platform dependent.

2

## Syntax (1)

- A SOAP message is an ordinary XML document:
  - It requires a special **Envelope** element surrounding the content so the XML document is recognised as SOAP message.
  - Has an optional **Header**.
  - Contains the **Body** of the message.
  - Contains an optional **Fault** section that describes errors that occurred when processing.

3

## Syntax (2)

```
<?xml version="1.0"?>
  <soap:Envelope
    xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

    <soap:Header>...</soap:Header>

    <soap:Body>...
      <soap:Fault>...</soap:Fault>
    </soap:Body>

  </soap:Envelope>
```

4

## The Envelope

- The root element of a soap message must be the Envelope.
- The XML **namespace** should always be <http://www.w3.org/2001/12/soap-envelope>
- The **encodingStyle** attribute defines the data types that will be present in the document.

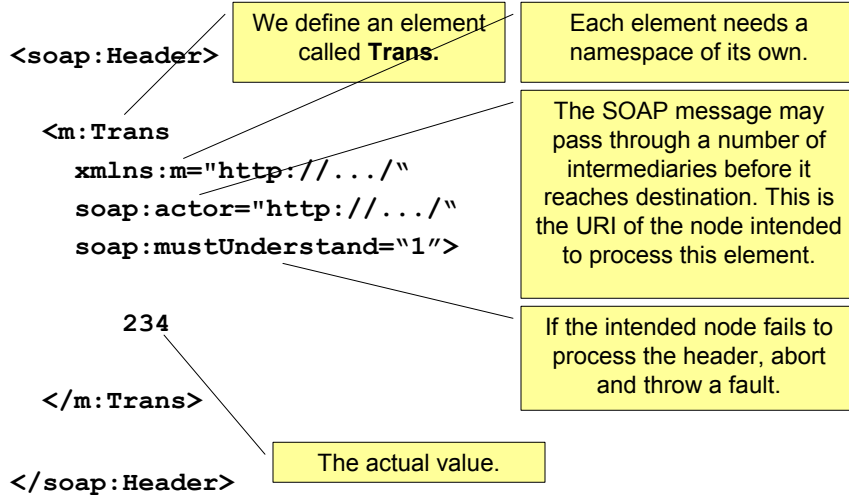
5

## The Header

- When a SOAP message is transferred between a sender and receiver, a number of **SOAP Intermediaries** may be involved. These would typically provide some sort of value-added services.
- The Header provides a space for SOAP nodes to exchange information.

6

# Anatomy of a Header



7

# The Body

- Contains the actual SOAP message intended for the final endpoint node.

```
<soap:Body>  
  <m:GetPrice xmlns:m="http://...">  
    <m:Item>Apples</m:Item>  
  </m:GetPrice>  
</soap:Body>
```

- Response:

```
<soap:Body>  
  <m:GetPriceResponse xmlns:m="http://...">  
    <m:Price>1.90</m:Price>  
  </m:GetPriceResponse>  
</soap:Body>
```

8

# The Fault

- Error messages in SOAP are carried inside a Fault element. The following sub elements describe it:

<code>&lt;faultcode&gt;</code>	A code.
<code>&lt;faultstring&gt;</code>	Description.
<code>&lt;faultactor&gt;</code>	Which node caused the fault.
<code>&lt;detail&gt;</code>	Error information.

9

# Hello World - Request

```
<s:Envelope ... ..>

  <s:Body>
    <m:sayHello xmlns:m='urn:Example1'>
      <name xsi:type='xsd:string'>James</name>
    </m:sayHello>
  </s:Body>

</s:Envelope>
```

10

# Hello World - Response

```
<s:Envelope ... ..>

  <s:Body>
    <n:sayHelloResponse xmlns:n="urn:Example1">
      <return xsi:type="xsd:string">
        Hello James
      </return>
    </n:sayHelloResponse>
  </s:Body>

</s:Envelope>
```

11

## Data Types (1)

- Scalar data types:

Type	Desc	Example
xsd:int	32-bit Unsigned	1234
xsd:boolean	1/0	1
xsd:string	String	Hello World
xsd:float/xsd:double	Floating-Point	3.1415
xsd:dateTime	Date/Time	2001-03-27T00:00:01-08:00
SOAP-ENC:base64	Binary Data	fjdf=dsafjlds

12

## Data Types (2)

- Structures/Records:

```
<param>
  <lowerBound xsi:type="xsd:int">18</lowerBound>
  <upperBound xsi:type="xsd:int">139</upperBound>
</param>
```

13

## Data Types (3)

- Arrays:

```
<param SOAP-ENC:arrayType="xsd:ur-type[4]"
      xsi:type="SOAP-ENC:Array">

  <item xsi:type="xsd:int">12</item>
  <item xsi:type="xsd:string">Egypt</item>
  <item xsi:type="xsd:boolean">0</item>
  <item xsi:type="xsd:int">-31</item>
</param>
```

14

## How Web Services Use SOAP

- A client wants to call a function and format the call in SOAP.
- The SOAP messages is communicated to the server over HTTP.
- The server receives the XML SOAP data, parses it and maps it to the 'real' business logic function calls.
- The result of the function call is wrapped inside a SOAP response which is sent back to the client.
- The Client parses the response and extracts the data.

15

## UDDI

- Universal Description, Discovery, and Integration.
- Like a large 'phone book' where businesses register their web services.
- Clients can query UDDI to find services that are required.
- Check out
  - <http://www.uddi.org/find.html>.
  - <http://uddi.microsoft.com/search/frames.aspx>

16



# WSDL

- **Web Services Description Language.**
- WSDL is (yet another) XML format for describing services in a structured way:
  - Types – the types defined by some type system (e.g. XSD).
  - Message – a definition of the data being communicated.
  - etc...
  - See <http://www.w3.org/TR/wsdl>