Enabling Grids for E-science in Europe

www.eu-egee.org

3 – 4 June 2004

SOAP
Simple Object Access Protocol

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

Objectives

- Understand the use of SOAP messages
- Become familiar with the syntax of SOAP
- SOAP in the context of Web Services

SOAP messages

- SOAP provides a standard ‘envelope’ within which a message can be delivered.

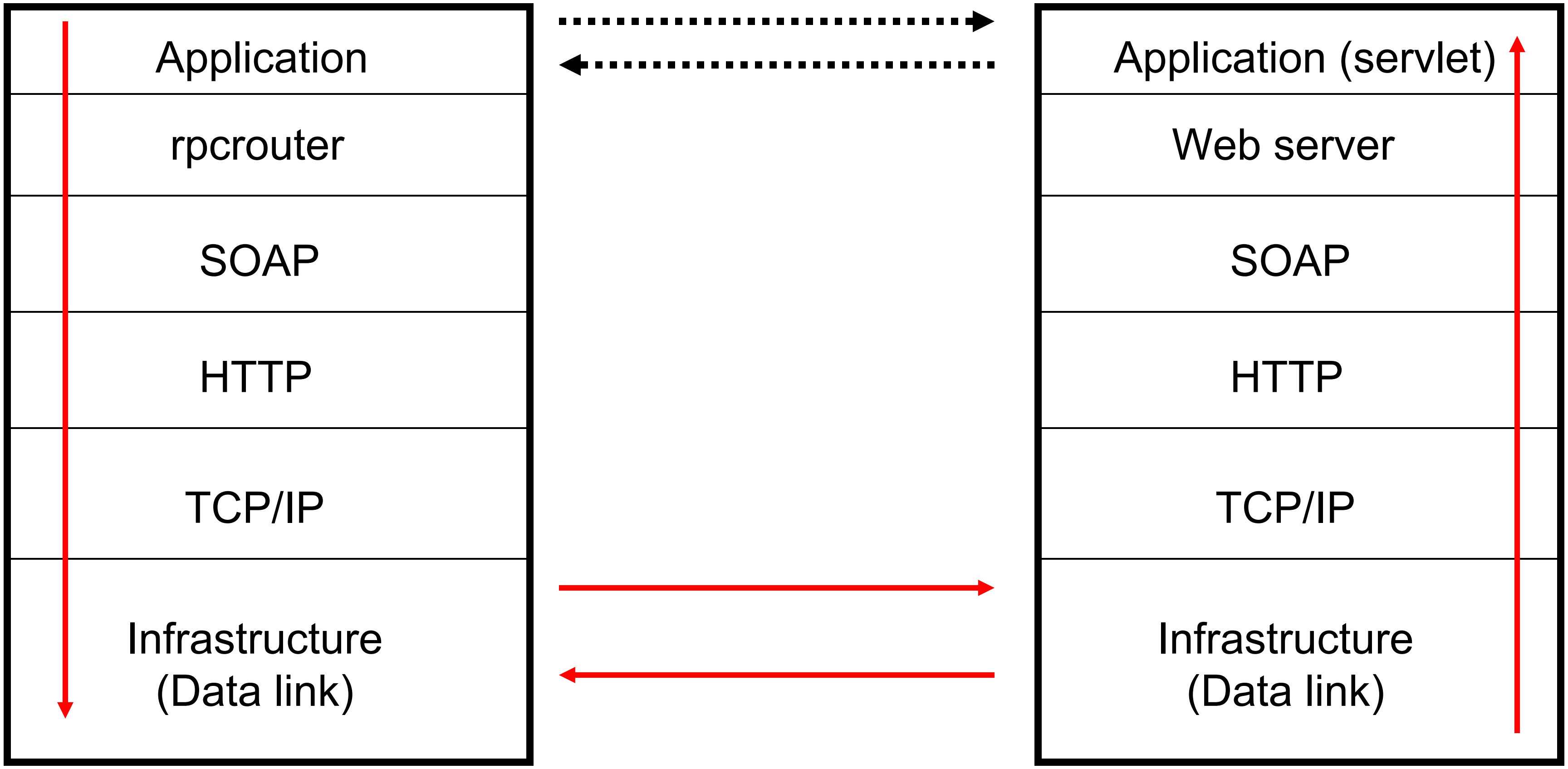
- SOAP is mechanism (protocol) for transferring information (messages) between applications which may be widely distributed.

- SOAP says nothing about the content of the message – the sender and the receiver must understand the message for themselves.

- SOAP is part of a communication stack.

Web services stack

Enabling Grids for E-science in Europe

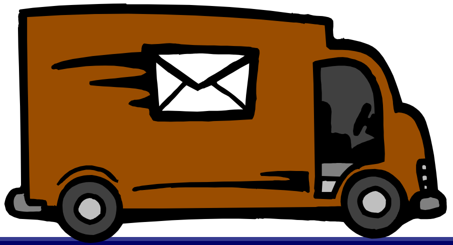


Business mail analogy

- The SOAP envelope is analogous to a business letter with an address within a distant organisation. This gives the information needed to get it from the sender’s building to the recipient’s building.

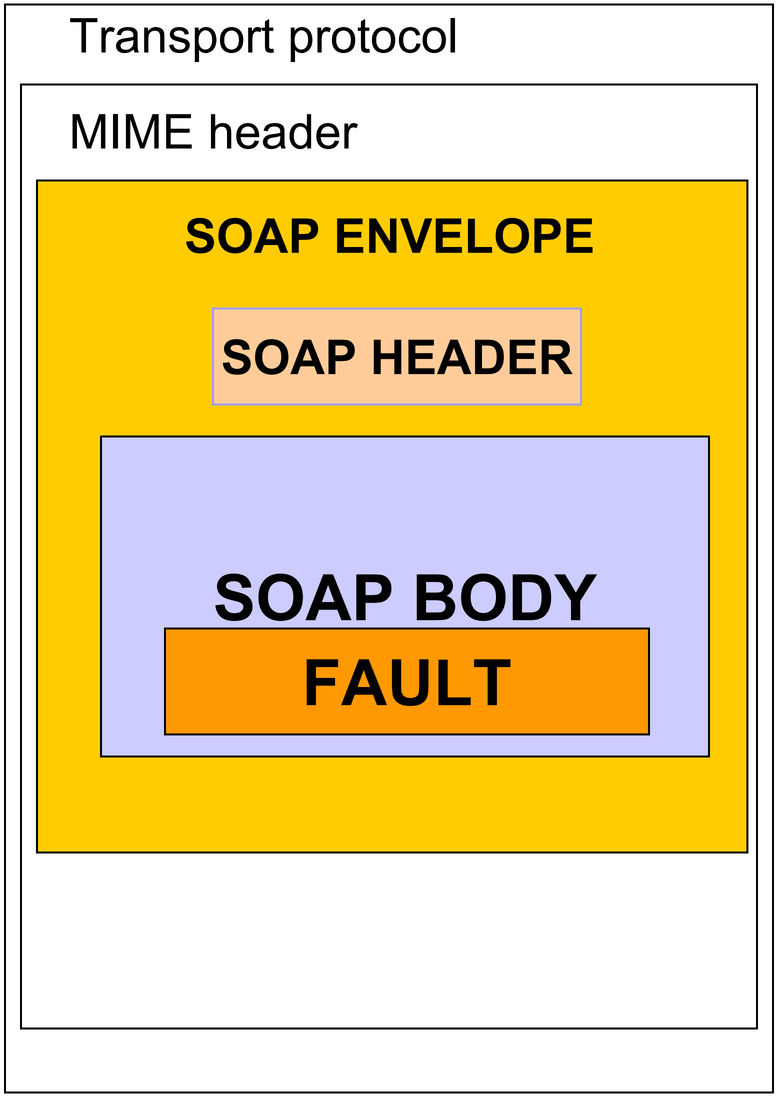
- The transport protocol is analogous to the carrier used for transport between buildings. (eg. FedEx.)

- The web server and container act like the local services for the recipient which place the message in his/her pigeon-hole.



SOAP Structure(1)

- Each SOAP message will have:
  - An `Envelope`
  - A `Header` (optional)
  - A `Body`
  - The `Body` may contain a `Fault` element



SOAP Structure(2)

- The envelope wraps the entire soap document
- The header contains allows additional information to be passed as well as the body of the document – e.g. authentication
- The body element contains the core of the SOAP document – this will contain either the RPC call or the XML message itself
- The fault information will contain any exception information

Anatomy of a SOAP message

```
<?xml version=‘1.0’ encoding=‘UTF-8’?>

<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
         xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
         xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>
         <SOAP-ENV:Header>

         </SOAP-ENV:Header


         <SOAP_ENV:Body>


         </SOAP-ENV:Body>


</SOAP-ENV:Envelope>
```

SOAP protocol binding

```
SOAPAction = “urn:soaphttpclient-action-uri”
Host = localhost
Content-Type = text/xml; charset=utf-8
Content-Length = 701
```

```
<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
        xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
        xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>



</SOAP-ENV:Envelope>
```

SOAP RPC

- SOAP RPC messages contain XML that represents a method call or method response

- The SOAP XML will be converted into a method call on the server and the response will be encoded into SOAP XML to be returned to the client

SOAP Faults

- SOAP errors are handled using a specialised envelope known as a Fault Envelope
- A SOAP Fault is a special element which must appear as an immediate child of the body element
- <faultcode> and <faultstring> are required.

A SOAP fault

```
<?xml version=‘1.0’ encoding=‘UTF-8’?>
<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
   xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
   xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>
   <SOAP_ENV:Body>


            <SOAP-ENV:Fault>
                    <faultcode>SOAP-ENV:Server</faultcode>
                    <faultstring>Test fault</faultstring>
                    <faultactor>/soap/servlet/rpcrouter</faultactor>
                    <detail>
                            ..
                    </detail>
            </SOAP-ENV:Fault>

   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

mustUnderstand attribute

- The mustUnderstand attribute can be placed in any top level header element.

- The presence of this attribute with a value of 1 or true means that the SOAP processor must recognise this element.

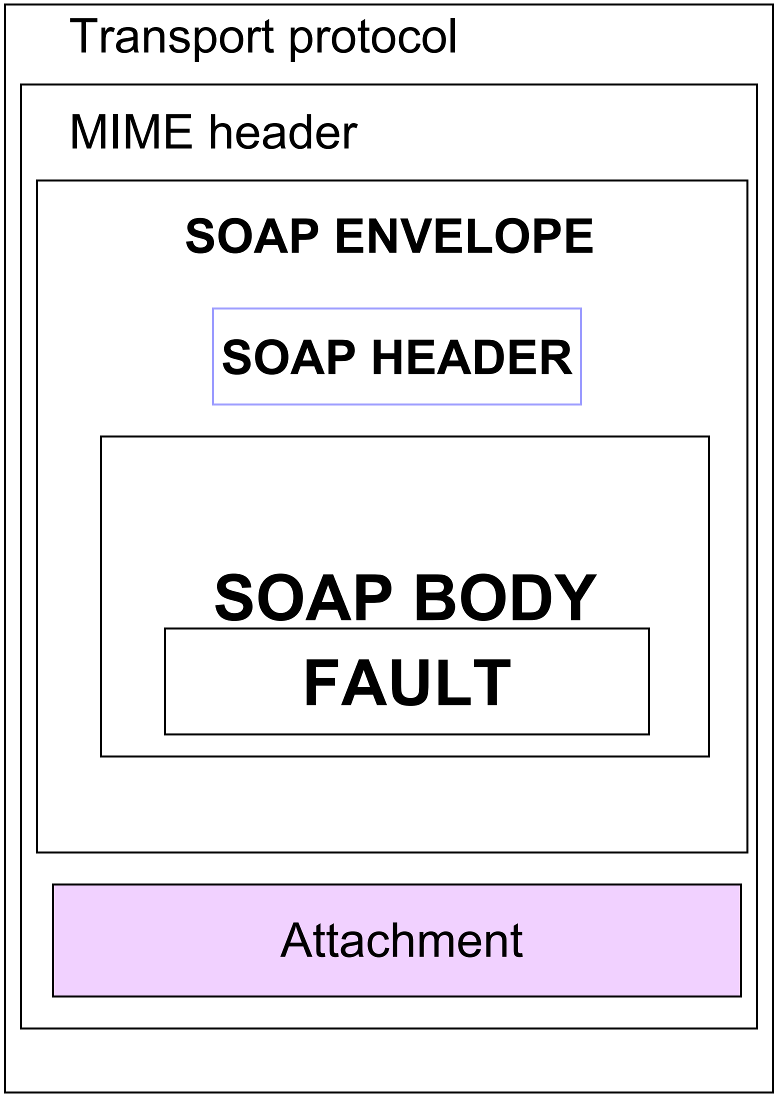
- If it does not recognise it, a fault must be generated.

mustUnderstand example

```
<?xml version=‘1.0’ encoding=‘UTF-8’?>
<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
   xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
   xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>


        <SOAP-ENV:Header>

                <jaws:MessageHeader xmlns:jaws=“http://...”
                        SOAP-ENV:mustUnderstand=“1”>
                </jaws:MessageHeader>

        <SOAP-ENV:Header>



   <SOAP_ENV:Body>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Attachment

- Large quantities or binary data may not fit well into a XML SOAP message.
- In which case it can be sent ‘out of band’ by attaching it to a SOAP message
- *Analogy : email attachments.*



Attaching a file to a SOAP message

- To add a file to a SOAP message a tag is added within the body of the message.

```
<?xml version=‘1.0’ encoding=‘UTF-8’?>
<SOAP-ENV:Envelope
xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
          xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
          xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>
          <SOAP_ENV:Body>


                    <attachment href=“{URL}”/>


          </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Deployment:
Making the container aware of a servlet

- The web server has to be aware of the interface and exposed methods of a servlet in order to use it.
- Using Tomcat as an example this can be done in a number of ways.

1. We will entry the values manually into the SOAP admin page from a Deployment descriptor in the practical.
2. You can use the SOAP manager application from the command line
3. You can manually edit Tomcat's WEB-INFO/web.xml file
4. You can create a WAR file and place it in Tomcat's webapps folder
5. You can use ANT

Using a WAR file

- A WAR file is basically an archive description of a servlet installation

  (*JAR and WAR naming derives from UNIX TAR – java archive, web archive, tape archive*).

- Example: placed in Tomcat’s webapps folder it can be interpreted by the container.

Deployment Descriptor

A SOAP manager file

```
<isd:service xmlns:isd=“http://xml.apache.org/xml-soap/deployment” id=“urn:stock-onhand”>
   <isd:provider type=“java” scope=“Application” methods=“getQty”>
        <isd:java class=“StockQuantity”/>
   </isd:provider>
   <isd:faultListener>org.apache.soap.sever.DOMFaultListener</isd:faultListener>
</isd:service>
```

Some containers (Tomcat) provide GUIs for deployment

SOAP Implementations

- There are several implementations of the SOAP Specification
    - Apache Axis
    - GLUE

- Most J2EE application servers contain a SOAP implementation

- .NET has a SOAP implementation