MONASH University

Information Technology

FIT5183: Mobile and Distributed Computing Systems (MDCS)

Lecture 2C
SOAP Web Services

Outline

❑SOAP Basics

❑SOAP Message Format

❑SOAP Implementation

❑SOAP binding with http

Slides are adopted from these sources:

- W3school SOAP http://www.w3schools.com/xml/xml_soap.asp
- Gustavo Alonso, https://www.vs.inf.ethz.ch/edu/WS0405/VS/VS-050124.pdf

SOAP Web Service Definitions

- ❑ World Wide Web consortium (W3C):

  “a *software application* identified by *a URI*, whose *interfaces* and *bindings* are capable of being defined, described, and discovered as *XML artifacts*.

  “A web service supports direct interactions with other software agents using XML-based messages exchanged via *Internet-based protocols*.”

- ❑ UDDI consortium:

  “self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces”

W3C is an international standards organization for the WWW
UDDI (Universal Description, Discovery and Integration)

What is SOAP ?

- ❑ **SOAP** stands for **Simple Object Access Protocol**
- ❑ SOAP is an application communication protocol
- ❑ SOAP is a format for sending and receiving messages
- ❑ SOAP is designed to communicate via Internet
- ❑ SOAP is platform independent
- ❑ SOAP is language independent
- ❑ SOAP is based on XML
- ❑ SOAP is simple and extensible
- ❑ SOAP allows you to get around firewalls

What do we need ?

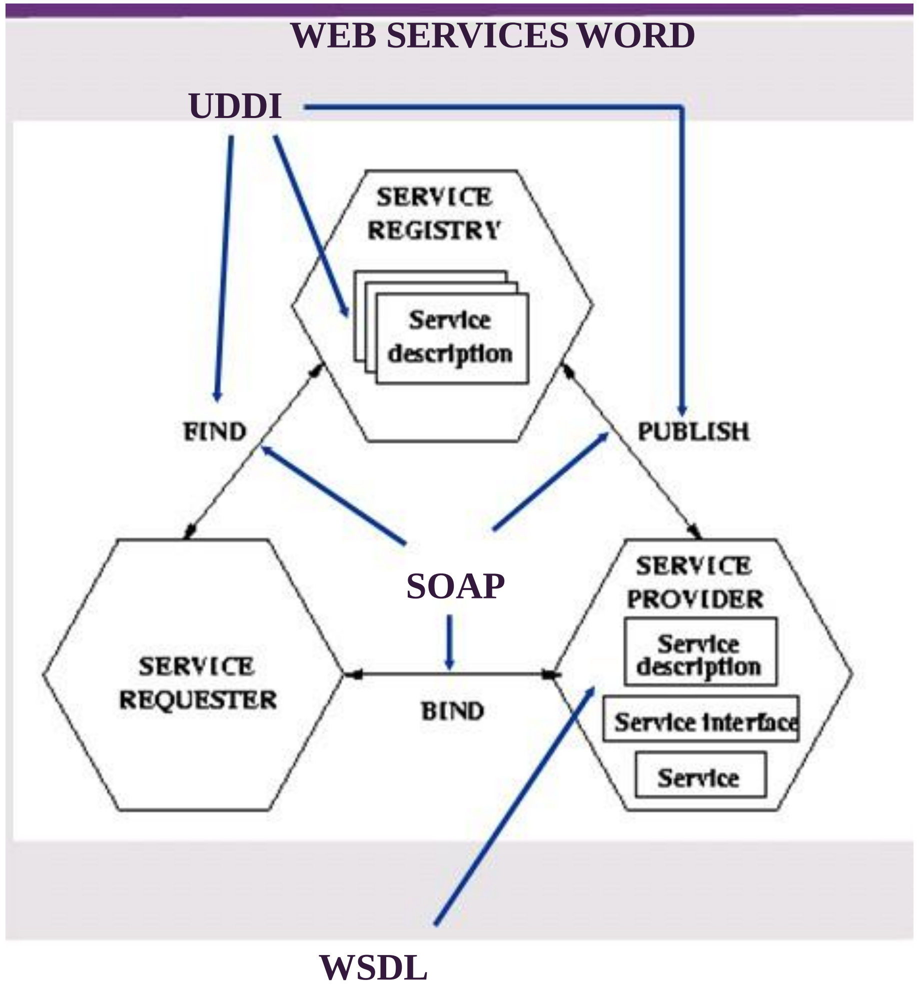
W3C: "…XML-based messages exchanged via Internet-based protocols."

- ❑ **Messages**: You want to send **XML documents**, not make procedure or method calls, so no RPC or RMI
- ❑ **Internet: Across firewalls**, so HTTP or SMTP instead of RPC or RMI
  - sometimes you don't want to have to wait for an ACK
  - sometimes you want an answer back
  - sometimes you want to simulate RPC (or RMI)
  - sometimes you want to use over existing transport protocols, e.g., SMTP, HTTP, FTP, even TCP
- ❑ **SOAP** is Simple Object Access Protocol.
  'protocol': not a language, not an implementation

Where Does SOAP Fit into Web Services?

IBM's Web service architecture

- Service requester: The potential user of a Service
- Service provider: The entity that implements the service and offers to carry it out on behalf of the requester
- Service registry: A place where available services are listed and which allows providers to advertise their services and requesters to query for services



What is SOAP Protocol?

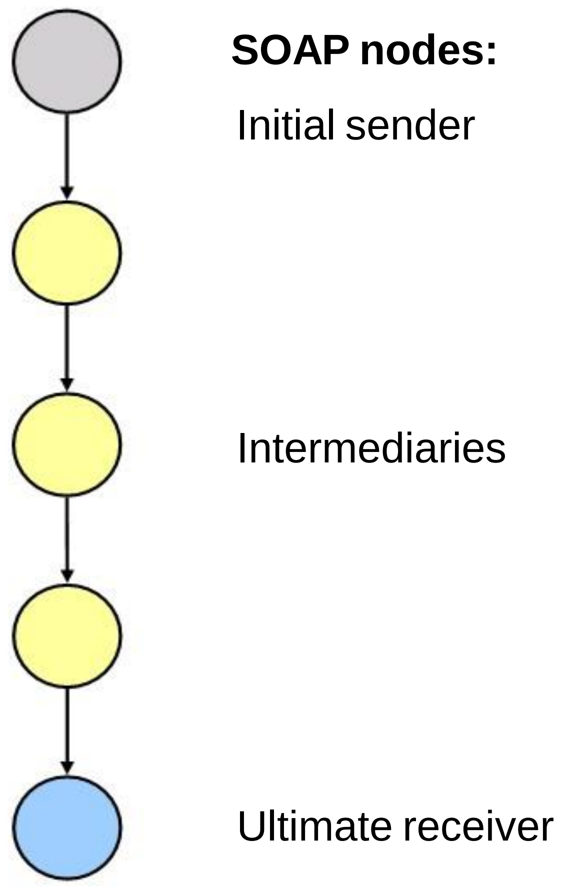
The W3C started working on SOAP in 1999. The current W3C recommendation is Version 1.2

SOAP covers the following main areas:

- ❑ **Message construct**: provides a message format describing how a message can be packed into an XML document
- ❑ **Processing model**: rules for processing a SOAP message and a simple classification of the entities involved in processing a SOAP message. Which parts of the messages should be read by whom and how to react in case the content is not understood
- ❑ **Extensibility Model**: How the basic message construct can be extended with application specific constructs
- ❑ **Protocol binding framework**: Allows SOAP messages to be transported using different protocols (HTTP, SMTP, …) A concrete **binding for HTTP**

The SOAP message path

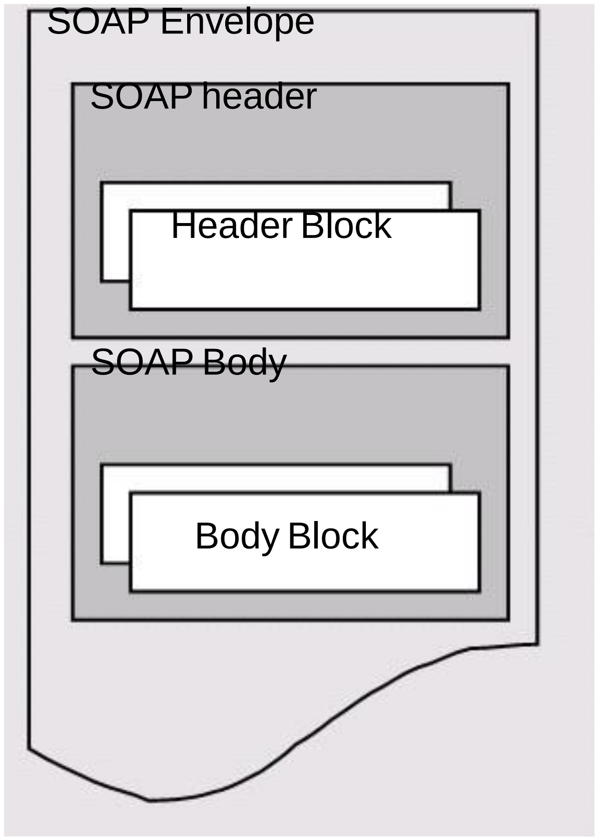
- A SOAP message can pass through **multiple hops** on the way from the initial sender to the ultimate receiver
- The entities involved in transporting the message are called SOAP **nodes**
- SOAP **intermediaries** forward the message and may manipulate it
- Every SOAP node assumes a certain **role** which influences the message processing at the node.



SOAP Message Format

SOAP Messages

- ❑ Envelop encloses the data to be sent
- ❑ Two parts:
  - - header
    - > optional
  - - **body**
    - **> mandatory**
  - - Fault
    - > optional



SOAP Building Blocks

- **A SOAP message is an ordinary XML document containing the following elements:**
  - A required **Envelope element** that identifies the XML document as a SOAP message
  - An optional **Header element** that contains header information
  - A required **Body element** that contains call and response information
  - An optional **Fault element** that provides information about errors that occurred while processing the message

- **All the elements above are declared in the default namespace for the SOAP envelope:**

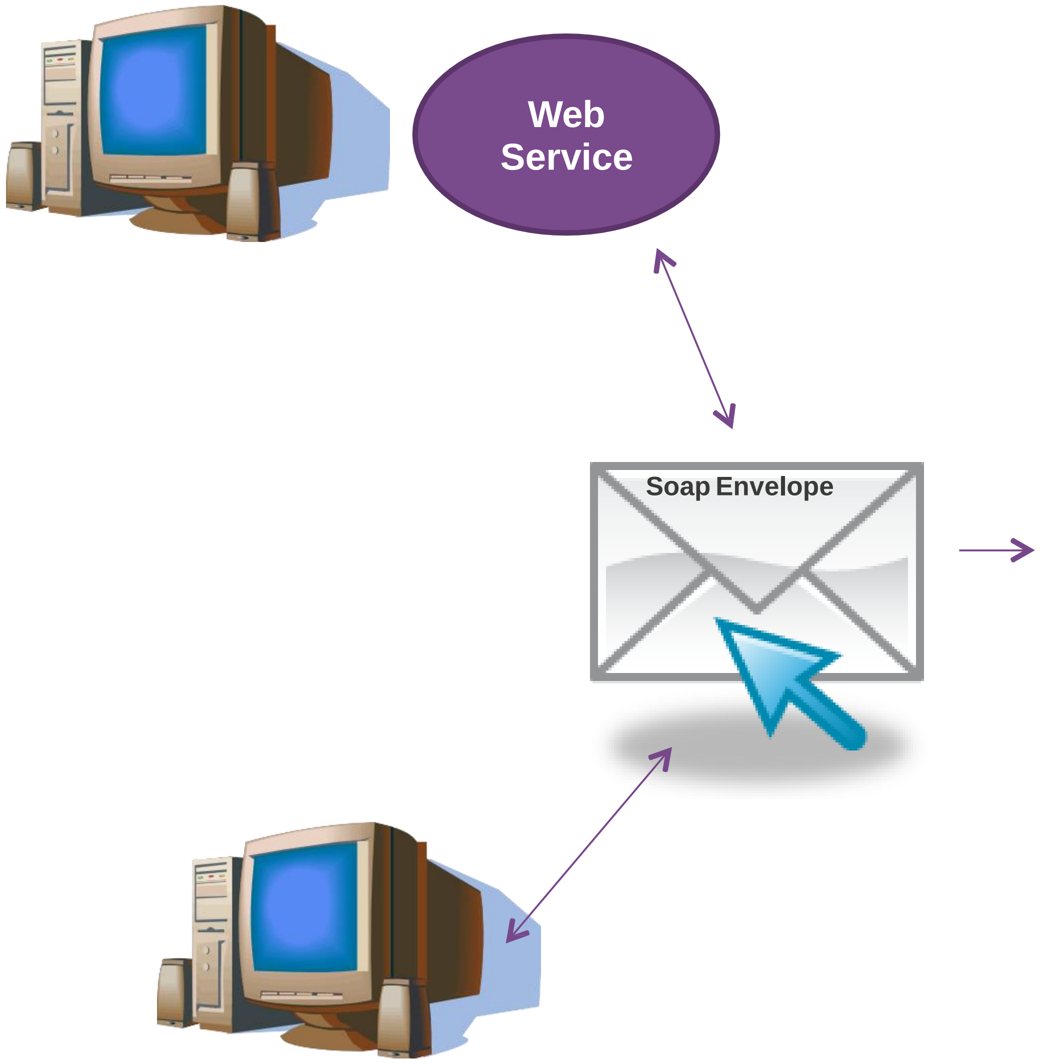
  http://www.w3.org/2001/12/soap-envelope

- **The namespace for SOAP encoding and data types:**

  http://www.w3.org/2001/12/soap-encoding

SOAP Message - Skeleton

```
<?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>
```



Soap message in xml

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

<soap:Envelope>
  - namespaces

  <soap:Header>

  - *Destination/Roles/Actors*
  - *How to get there*

  </soap:Header>

  <soap:Body>

  - *Data/message/Payload*

   <soap:Fault>
   …
  </ soap:Fault>

</soap:Body>
</soap:Envelope>

SOAP Envelope Element

- The required SOAP Envelope element is the root element of a SOAP message. It defines the XML document as a SOAP message.
- Note the use of the **xmlns:soap namespace**. It should always have the value of: http://www.w3.org/2001/12/soap-envelope
- It defines the Envelope as a SOAP Envelope:

```
<?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">
  Message information goes here
</soap:Envelope>
```

- SOAP message must always have an Envelope element associated with the "http://www.w3.org/2001/12/soap-envelope" namespace.
- If a different namespace is used, the application must generate an error and discard the message

SOAP Header Element

- Optional
- SOAP Header element contains application specific information which can be processed by *intermediaries*
  - authentication, payment, are more obvious examples
- If the Header element is present, it must be the first child element of the Envelope element.

(Note: All immediate child elements of the Header element must be namespace qualified.)

SOAP Header Element

```
<?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>
 <m:Trans
 xmlns:m="http://www.w3schools.com/transaction/"
 soap:mustUnderstand="1">234</m:Trans>
 </soap:Header>
 …
 </soap:Envelope>
```

- The example above contains a header with a "Trans" element, a "mustUnderstand" attribute value of "1", and a value of 234.
- The attributes defined in the SOAP Header defines how a recipient should process the SOAP message.
  - http://www.w3.org/2001/12/soap-envelope defines three attributes of **actor**, **mustUnderstand**, and **encodingStyle**
  - http://www.w3.org/2003/05/soap-envelope defines three attributes: **role**, **mustUnderstand** and **Relay** (whether or not to forward unprocessed header block.)

SOAP Body Element

- ❑ The required SOAP Body element contains the actual SOAP message intended for the ultimate endpoint of the message

(Immediate child elements of the SOAP Body element may be namespace-qualified.)

- ❑ SOAP defines one element inside the Body element in the default namespace (i.e. Fault)
- ❑ This is the SOAP Fault element, which is used to indicate error messages.

SOAP Body Element

```
<?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:Body>
   <m:GetPrice
  xmlns:m="http://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
   </m:GetPrice>
</soap:Body>
</soap:Envelope>
```

- ❑ The example above requests the price of apples.
- ❑ Note that the m:GetPrice and the Item elements above are application-specific elements. They are not a part of the SOAP standard.

SOAP Body Element

- Example SOAP Response

```
<?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:Body>
   <m:GetPriceResponse
  xmlns:m="http://www.w3schools.com/prices">
      <m:Price>1.90</m:Price>
   </m:GetPriceResponse>
</soap:Body>
</soap:Envelope>
```

SOAP Fault Element

- Optional
- Used to hold error/status information for a SOAP message.
- If a Fault element is present, it must appear as a child element of the Body element.
- A Fault element **can only appear once** in a SOAP message.
- The SOAP Fault element has the following sub elements:
  - <**faultcode**> A code for identifying the fault
  - <**faultstring**> A human readable explanation of the fault
  - <**faultactor**> Information about who caused the fault to happen
  - <**detail**> Holds application specific error information related to the Body element

SOAP Fault Element

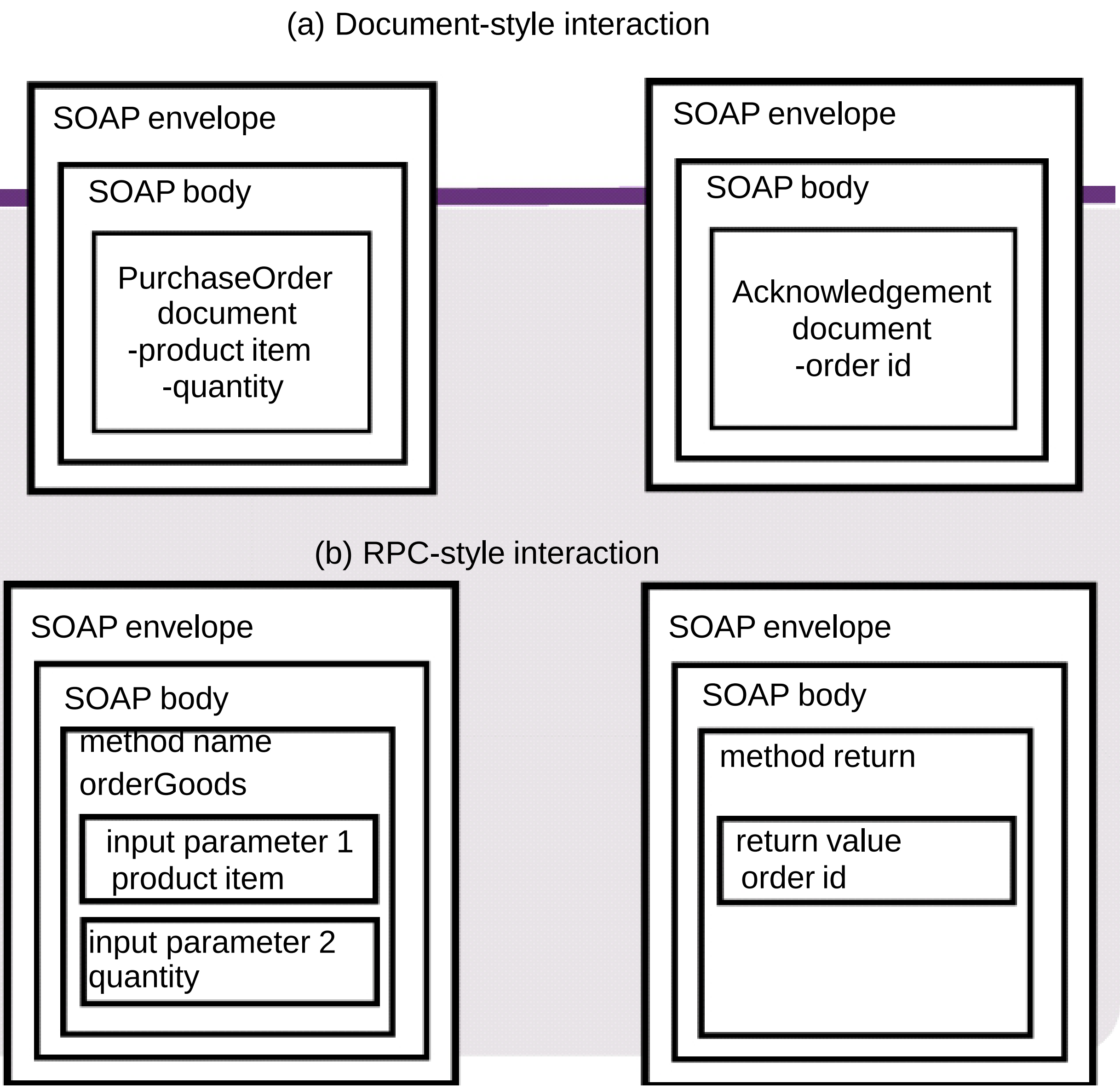
The faultcode values:

- **VersionMismatch**
  - Found an invalid namespace for the SOAP Envelope Element
- **MustUnderstand**
  - An immediate child element of the Header element, with the mustUnderstand attribute set to "1“ was not understood
- **Client**
  - The message was incorrectly formed or contained incorrect information
- **Server**
  - There was a problem with the server so the message could not proceed

SOAP Styles

- ❑ Two interaction styles: Document and RPC styles
- ❑ Document style:
  - – the body simply contains an XML document
  - – two interacting applications agree on the structure/format of documents exchanged between them
    - E.g. A client ordering goods from one supplier creates a PurchaseOrder document as a SOAP message (i.e. items and their quantities)
    - Supplier sends an Acknowledgement document containing order Id for confirmation
- ❑ RPC style:
  - – two sides agreeing on the RPC method signature instead of document structure
  - – The request message contains the actual call including the name of the procedure and input parameters
  - – The response message contains the results and output parameters

SOAP Interaction Styles


(a) Document-style interaction
SOAP envelope
SOAP body
PurchaseOrder document -product item -quantity
SOAP envelope
SOAP body
Acknowledgement document -order id
(b) RPC-style interaction
SOAP envelope
SOAP body
method name orderGoods
input parameter 1 product item
input parameter 2 quantity
SOAP envelope
SOAP body
method return
return value order id

SOAP simulates RPC

Another example:

```
<m:chargeReservation>
 <m:reservation>
  <m:code>FT35ZBQ</m:code>
 </m:reservation>
 <o:creditCard>
  <n:name> John Citizen </n:name>
  <o:number>123456789099999</o:number>
 <o:expiration>2008/08</o:expiration>
 </o:creditCard>
</m:chargeReservation>
```

RPC method signature was something like

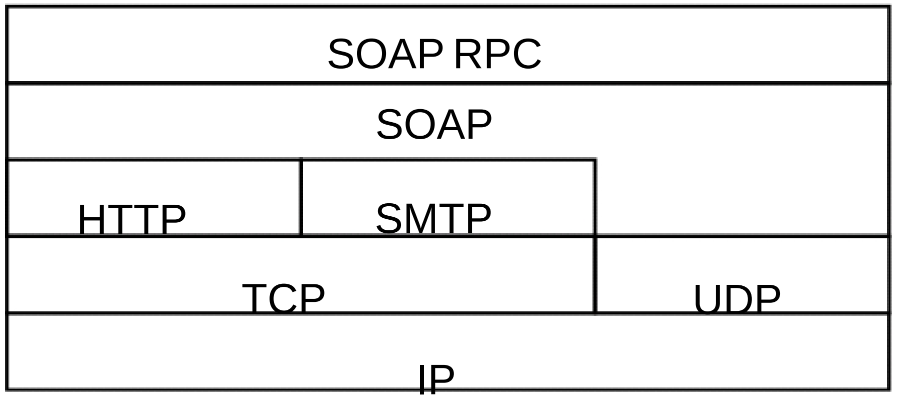
```
chargeReservation(string code, struct creditCard);
```

Source: http://www.w3.org/TR/soap12-part0/#Example2

SOAP Binding with HTTP

SOAP Protocol Binding Framework

- SOAP messages **can be transferred using any protocol**
- A binding of SOAP to a transport protocol is a description of how a SOAP message is to be sent using that transport protocol
- A binding specifies how response and request messages are correlated
- The SOAP binding framework expresses guidelines for specifying a binding to a particular protocol



https://www.vs.inf.ethz.ch/edu/WS0405/VS/VS-050124.pdf

SOAP HTTP Binding

- ❑ SOAP messages are enclosed in the payload of an HTTP request or response
- ❑ **HTTP + XML = SOAP** (at least by established convention)
- ❑ A SOAP request must be an HTTP POST in version 1.0, but can be either HTTP GET or POST in version 1.2
- ❑ The HTTP POST request specifies at least two HTTP headers: Content-Type and Content-Length.

SOAP HTTP Binding

- **Content-Type** header for a SOAP request and response defines
  - The MIME type for the message
  - The character encoding (optional) used for the XML body of the request or response.
- **Syntax**
  - `Content-Type: MIMEType;`
  - `charset=character-encoding`
- **Example**
  - `POST /item HTTP/1.1`
  - **`Content-Type: application/soap+xml; charset=utf-8`**

SOAP HTTP Binding

- ❑ **Content-Length** header for a SOAP request and response
  - specifies the number of bytes in the body of the request or response.

- ❑ **Syntax**
  - **`Content-Length: bytes`**

- ❑ **Example**

```
POST /item HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 250
```

SOAP HTTP Example - Request

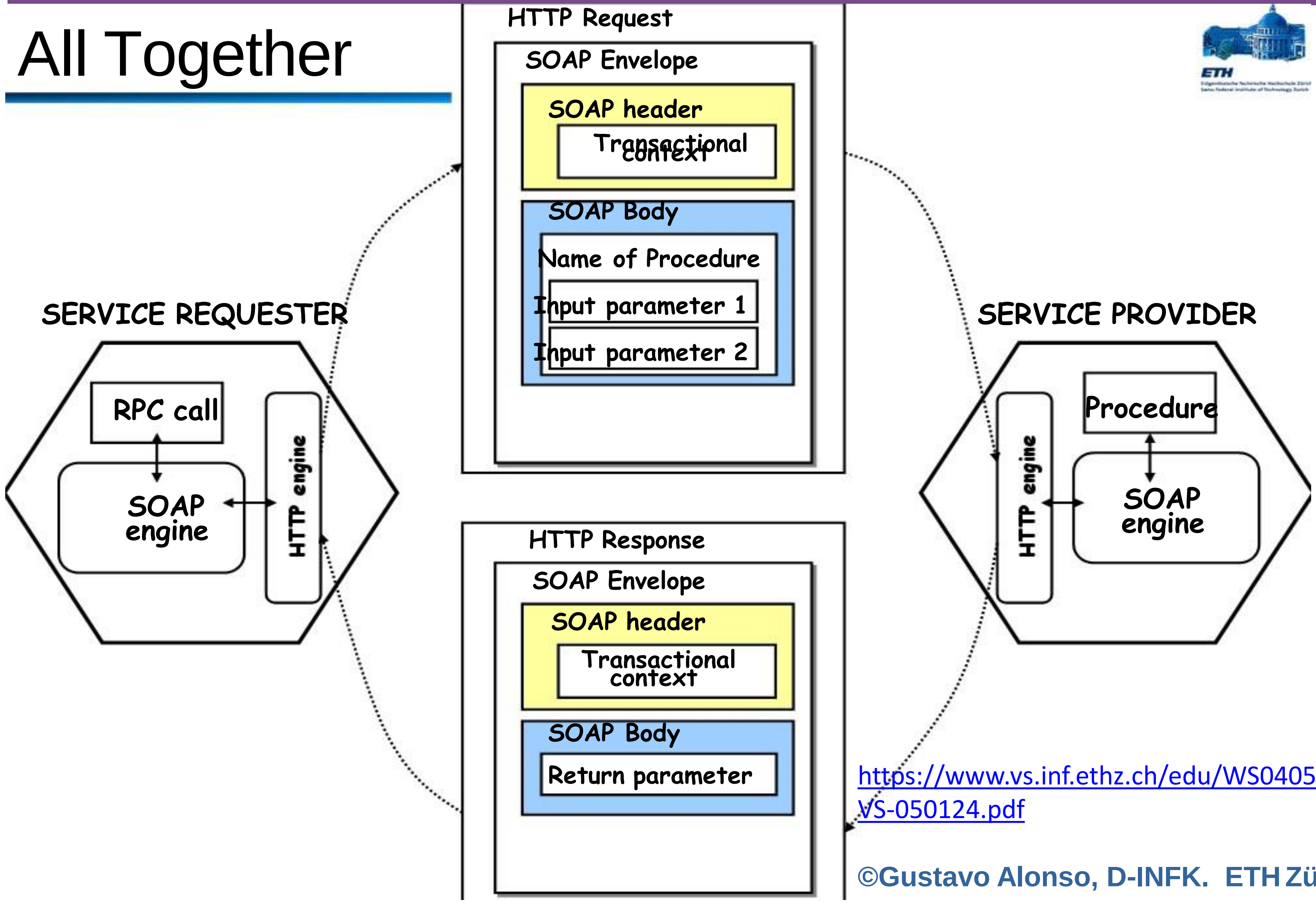
```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 200

<?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:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

.

SOAP HTTP Example - Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 250

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soa
  p-encoding">
  <soap:Body
  xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>
</soap:Envelope>
```

All Together



https://www.vs.inf.ethz.ch/edu/WS0405/VS/VS-050124.pdf

©Gustavo Alonso, D-INFK. ETH Zürich

References

- Gustavo Alonso, et al. Web Services Concepts, Architectures and Applications, 2004
- https://www.vs.inf.ethz.ch/edu/WS0405/VS/VS-050124.pdf
- http://www.w3schools.com/soap/default.asp
- http://docs.oracle.com/cd/E19651-01/817-2151-10/wsgoverview.html

An Example

- http://opendap.co-ops.nos.noaa.gov/axis/