
Web Services Core Elements

SOAP, WSDL, UDDI

SOAP

Web Service Alphabet Soup

SOAP - Simple Object Access Protocol

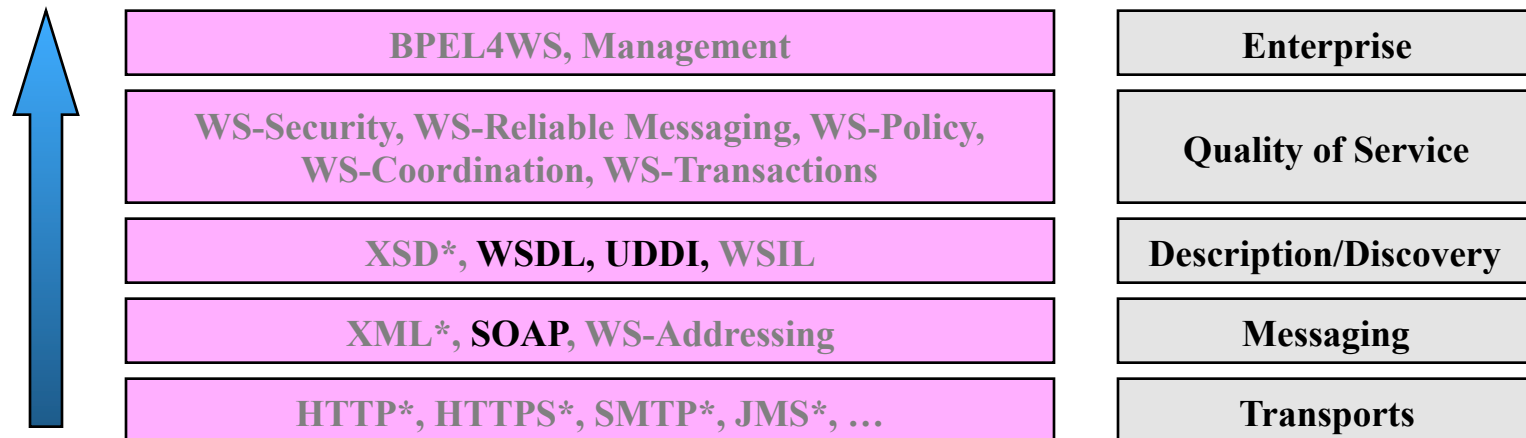
- An XML-based messaging protocol

WSDL - Web Services Description Language

- An XML-based language for describing services
- Considered an IDL for web services, described in XML
- Uses XML Schema (XSD) to describe types for data structures (e.g. PurchaseOrder)

UDDI - Universal Description, Discovery and Integration

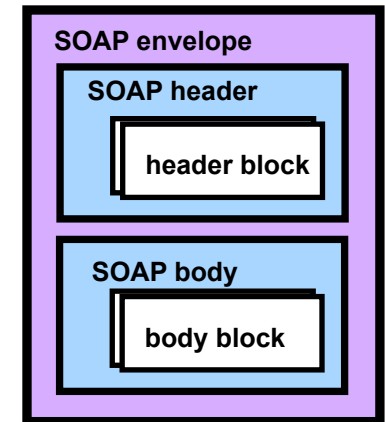
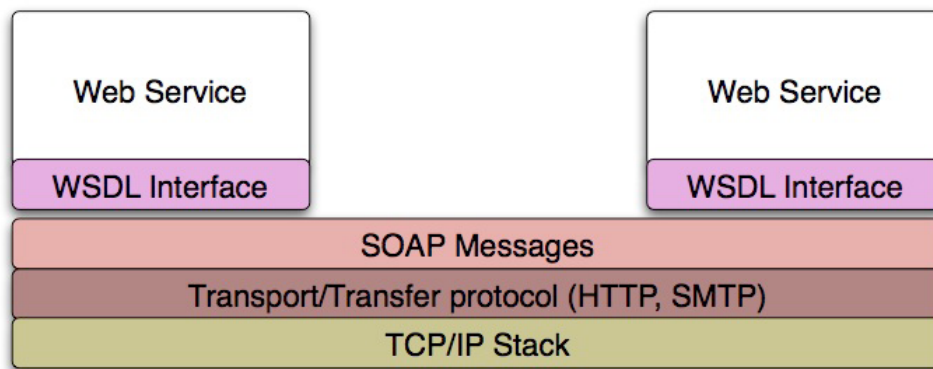
- A registry for clients to discover available services
- Uses SOAP for communication to register and locate



* Outside WS

SOAP (Simple Object Access Protocol)

Standard messaging protocol used by web services
Supports inter application communication

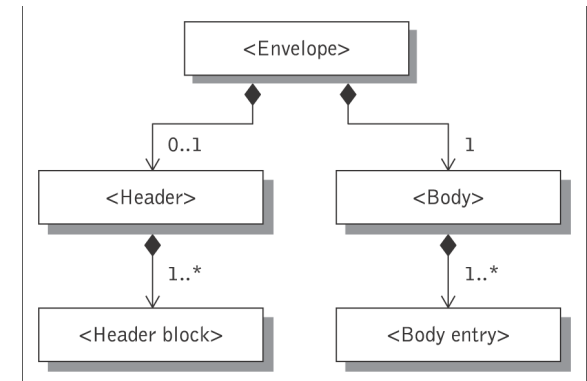


SOAP messages

- Application envelopes or encloses data to be sent

Components of a SOAP message

- <Envelope> element
- An optional <Header>
- Mandatory <Body> element



SOAP

An XML-based, stateless communication protocol

- Envelope & body required, header and fault optional

Specified by World Wide Web Consortium (w3c)

```
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  soap:encodingStyle=
    "http://www.w3.org/2003/05/soap-encoding">
  <soap:Header> ... </soap:Header>
  <soap:Body>
    ...
    <soap:Fault> ... </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Example:

HTTP
request/
response

```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap+xml; charset=utf-8
<?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.stock.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

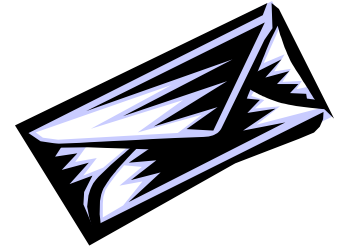
Application specific
content

```
HTTP/1.1 200 OK
Content-Type: application/soap; charset=utf-8
Content-Length: nnn<?xml version="1.0"?>
<soap:Envelope ...>
  <soap:Body xmlns:m="http://www.stock.org/stock">
    <m:GetStockPriceResponse><m:Price>81.5</m:Price>
  </m:GetStockPriceResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP Elements

Envelope

- The root document for all SOAP messages
- Defines the SOAP namespace and encoding



Header

- Optional section contains information not part of the payload
- Allows SOAP to be extended (composable services)
 - Extensions are commonly ‘control’ information (routing, encryption, etc.)
- Endpoints along message path may inspect and augment header

Body

- Mandatory element containing message’s content
- Format is application specific, but root must be namespace-qualified
- May contains a fault for requests that generate exceptions

WSDL & UDDI

WSDL

XML language for describing Web services & how to access

- What are methods? Method arguments? Service location (URL)?
- References other WSDL and XSD documents describing datatypes

Two sections

- Abstract specification section
 - Describes service, ports (operation sets), & data passed via messages
- Concrete binding section

```
<?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/">
```

Specify all
referenced
namespaces

WSDL Example for StockQuote Service

```
<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/1999/XMLSchema">
    <element name="GetLastTradePrice">
      <complexType>
        <all><element name="symbol" type="string"/></all>
      </complexType>
    </element>
    <element name="GetLastTradePriceResponse">
      <complexType>
        <all><element name="Price" type="float"/></all>
      </complexType>
    </element>
  </schema>
</types>
<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:GetLastTradePrice"/>
</message>
<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:GetLastTradePriceResponse"/>
</message>
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
```

Define data passed into and out of messages in a tech independent manner

Define messages and their data (bodies)

Port types are a set of operations the web service will provide

Operations define input and output messages

WSDL Example for StockQuote Service

Describe a binding to
the port type

```
<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/stockquote.xsd"/>
    <input>
      <soap:body use="encoded" namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="encoded" namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>
```

Specify concrete service, its
binding, and its location

```
<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://www.stockquoteserver.com/StockQuote"/>
  </port>
</service>
</definitions>
```

Universal Description, Discovery, & Integration (UDDI)

Registry for listing services

- Uses SOAP and WSDL for communication and descriptions
- Standard mechanism to advertise web services

Information distributed over three ‘pages’

- White pages – Company name and contact information, etc.
- Yellow – Company fields of businesss
- Green – technical means to access Company’ s services

Known limitations

- Only provides static, structural information on services, no expected message protocol, QoS, dynamic information, etc.
- Designed to be processed by a person and has limited meta-information for programmatic decisions

