



SOAP

Simple Object Access Protocol

Bob Brown

t r a n s e n t i a

bob@transentia.com.au

<http://www.transentia.com.au>

- Microsoft says it well:
 - *“SOAP is a lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment.”*
 - Rides on the back of XML’s growing popularity
 - A good fit, actually
 - *“...a messaging protocol that is not limited to Remote Procedure Calls (RPC). It does not require synchronous execution or request/response interaction, and SOAP messages can have multiple parts addressed to different parties”*
 - Messages can be carried via HTTP/SMTP/MSMQ/SMS or carrier pigeon...
 - Useful for RPC-style/messaging/push/... interaction styles

What Is SOAP?...

- Unimpressive in itself, SOAP will serve as the impetus for a number of important activities:
 - Service Oriented Architectures
 - COM/CORBA(IIOP)/RMI/... interoperability
 - An end to those object wars(?)
 - More ‘open’ middleware
 - ?
- SOAP has a past life as ‘XML-RPC’
- SOAP has a future life in the W3C as “XML Protocol”

- SOAP provides both:
 - A better RPC
 - The ubiquity of the (commonly used) underlying HTTP and SMTP protocols, their static nature (no dynamically allocated server ports) and the transparency of firewalls to these protocols means that SOAP is better ‘equipped’ than most competing Remote Procedure Call systems
 - A better NDR
 - SOAP adopts XML as the underlying Network Data Representation. This is *A Good Thing*.
 - There exists a large variety of XML-aware tools and systems
 - XML is (fairly) readable and can be handled in a “low-tech” way if need be, while also making complex systems possible
 - The underlying XML makes it flexible and easily extended to cope with any situation

- **RPC**
 - Operations have input (request) messages that contain the operations' input parameters and output (response) messages that contain the operations' results
 - Typically suited for small-sized data
 - Atomic/simple data structures
 - Data is packaged for immediate/easy processing
- **Document**
 - The input and output messages contain XML documents
 - Can be arbitrarily large
 - Complex data hierarchies possible
 - Require DOM manipulation to process the data

- **Pros**
 - **Functional**
 - Interoperates with diverse, varied systems
 - Possibility of secure exchanges, tunnelled through firewalls
 - **Technical**
 - Simple
 - Based on standards (HTTP/SMTP/XML)
 - Lightweight
 - Clear separation of payload and transport
- **Cons**
 - **Functional**
 - Security perceived as a weak point that needs work
 - **Technical**
 - Too young for full interoperability
 - Still mutating (currently just into version 1.2)
 - Fiddly to use; not transparent (yet)

- *“What is SOAP if not basically a more object-oriented, somewhat buzzword-compliant upgrade to CGI?”*
 - Anon., Internet
- *“...I think that this is what the grown up web was meant to be, combining the best that HTTP has to offer and great server operating systems, and the best that the desktop has to offer in tools and user interfaces.”*
 - Dave Winer, CEO, UserLand
- *“Like the basic web protocols that came before it, SOAP represents a tectonic shift in the way things are done. ...SOAP gives a new breakthrough: reuse of content and services across all traditional boundaries.”*
 - Paul Everitt, CEO, Digital Creations
- *“SOAP will be a key technology for building interoperable applications on the internet.”*
 - Noah Mendelsohn, Distinguished Engineer, Lotus

Quotable Quotes...

- *“Now comes soap...which is better understood as the creation of hackers who happen to work at Microsoft, UserLand, Developmentor and other places, large and small”*
 - Doc Searls, Senior Editor, Linux Journal
- *“While SOAP does have several things going for it, they are more sociological [sic] than technical in my opinion: it is buzzword enabled, backed by Microsoft, builds on well understood Internet mechanisms, you can encode requests by hand. It is also much less ambitious than platforms such as CORBA, and thus easier to learn. It is a sad refelction [sic] on the state of computing that these factors probably overweigh technical criteria.”*
 - Eric Marsden, Internet

- From InfoWorld (17 July 2001)

- <http://iwsun4.infoworld.com/articles/tc/xml/01/07/16/010716tcsoap.xml>

SOAP

This messaging framework and RPC mechanism requires additional integration middleware (brokers, listeners, interpreters) for implementation.

Easy to learn and implement, SOAP can be used by Visual Basic programmers with little training.

SOAP is put into service using HTTP through firewalls and on Web servers, easing interenterprise application integration.

As a text-based protocol, SOAP demands additional bandwidth for transmission and requires parsing on the recipient side.

Because it is XML-based, SOAP might be able to ease platform interoperability issues.

SOAP is not a finalized standard; early adopters may need to update implementations to meet final specifications.

Native security specifications are under consideration but are not yet settled.

CORBA

The CORBA suite of specifications provides a well-integrated interface framework and protocols to support complex interactions, legacy interfacing, fault tolerance, resource control, and use in embedded applications.

CORBA requires training, expertise, and programming to implement, typically limiting its use to large-scale EAI (enterprise application integration) projects.

CORBA requires open network connections, thereby creating potential firewall security issues; and it requires dedicated server architectures for comprehensive implementation.

CORBA's binary protocol provides for quicker transmission and faster processing.

Although also an open-source specification, CORBA requires agreed-on configuration among various object request broker flavors.

CORBA lacks features for open, Internet-centric operation but is being retrofitted for use with SOAP and Web services.

Security service specifications for CORBA are already in place.

- Its the data, dummies!
 - Java lets us have fun with Y.A.P.L. but does nothing to solve the wider issues underlying integration and interoperability
 - Sure, you can write the processing code in Java, but is the *data* intelligible? Can you use the code from COBOL?
- An end to the interoperating object wars
 - Multiple languages, object systems, etc.
 - VB, C/C++, Java, Ada, PERL, COBOL(?),...
 - SOAP's "on-the-wire" format is XML
 - May allow COM/CORBA(IIOP)/RMI interoperability...
- Facilitates dynamic systems based on discovery
 - SOAP acts as the *lingua-franca*
- Low barrier to entry
 - No-brainer concepts; readily available, free toolkits
- Nothing new here
 - SOAP's guiding principle: "*first invent no new technology*"

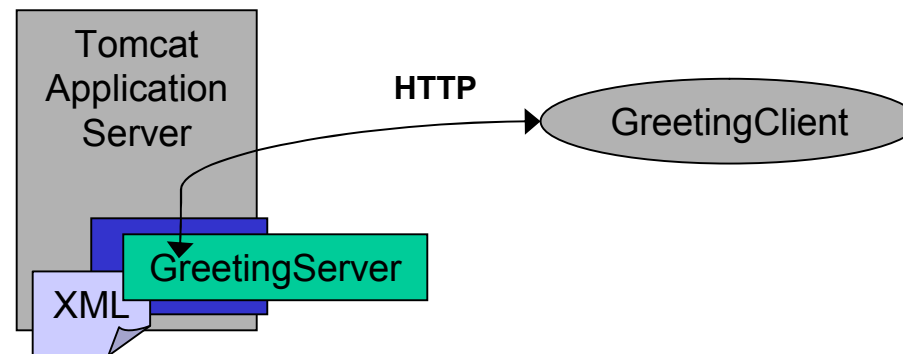
- Overcomes “the tyranny of the firewall”
 - Rides atop standard Internet transports such as HTTP/80; SMTP/25
 - Most firewalls are transparent to these...
 - *“Although HTTP will probably be the primary transport protocol for SOAP messages, the latest revision of the specification allows for sending SOAP messages over almost any conceivable protocol including SMTP, FTP, MQSeries, and MSMQ, or even raw TCP via sockets. This refinement is important because it allows the same SOAP serialization rules to be leveraged in all of these transport protocols.”*
- Strategic across-the-board-support
 - Politics are the primary factor behind SOAP’s acceptance
 - Communities
 - Apache group
 - It’s acceptable to the hacker psyche...
 - W3C
 - SOAP is accepted as a technical note
 - Vendors
 - A creation of IBM/Microsoft/Lotus/UserLand/DevelopMentor
 - Now garnering widespread support

- There exists ~70 implementations at last count (40 when these slides were started!)
 - Several interoperability issues exist
- Various aims, languages, etc.
 - Apache SOAP
 - Web Services Toolkit
 - DevelopMentor SOAP
 - SoapRMI
 - Soap Toolkit for JBuilder
 - Microsoft SOAP toolkit 2
 - Visual Studio.NET
 - PocketSOAP
 - vbSOAP
 - SOAP::Lite
 - DevelopMentor SOAP
 - SOAP for Ada
 - SOAP Smalltalk
 - PHPSOAP
 - SOAP for BEA WebLogic Server

- Microsoft is firmly behind SOAP
 - *“The purpose of SOAP is to enable rich and automated Web services based on a shared and open Web infrastructure.”*
 - Microsoft talks about “the programmable web”
 - Promotion:
 - *“XML: Data Encoded; SOAP: Data Communicated; .NET: Data Shared”*
 - Underlies much of .NET
 - ASP.NET; Visual Basic.NET, etc.
 - BizTalk server
 - Visual Studio.NET makes it possible to produce SOAP services and clients

*“To be a player
in the .NET
development
world you must
understand XML
and SOAP.”*

- Simple “Hello World” client/server
 - Written for Apache SOAP/Tomcat
 - Originally from IBM
 - Synchronous, RPC-style interaction over HTTP
 - Components:
 - Tomcat server
 - rpcrouter JSP
 - » Examines the HTTP call and determines which server object should be dispatched to handle it
 - XML deployment descriptors



Ex: Client

```
package au.com.transentia;

import java.io.*;
import java.util.*;
import java.net.*;
import org.apache.soap.*;
import org.apache.soap.rpc.*;

public class GreetingClient
{
    private static PrintWriter out = new PrintWriter (System.out, true),
        err = new PrintWriter (System.err, true);

    public static void main (String [] args) throws Exception
    {
        if (args.length != 2)
        {
            err.println ("Usage: java " + GreetingClient.class.getName () +
                " SOAP-router-URL name-to-greet");
            System.exit (1);
        }

        // Build the SOAP RPC call.
        Call call = new Call ();
        call.setTargetObjectURI ("urn:GreetingService");
        call.setEncodingStyleURI (Constants.NS_URI_SOAP_ENC);
        call.setMethodName ("getGreeting");
        Vector params = new Vector ();
        params.addElement (new Parameter("who", String.class, args [1], null));
        call.setParams (params);

        Response resp = call.invoke (new URL (args [0]), "");

        if (resp.generatedFault ())
        {
            Fault fault = resp.getFault ();
            err.println (" Fault Code   = " + fault.getFaultCode ());
            err.println (" Fault String = " + fault.getFaultString ());
            System.exit (2);
        }

        Parameter result = resp.getReturnValue ();
        out.println (result.getValue ());
    }
}
```


- You can't get simpler!
 - Note the total absence of infrastructural 'housekeeping' code!
 - This is excellent...

```
package au.com.transentia;  
  
public class GreetingService  
{  
    public String getGreeting (String who)  
    {  
        return ("[" + new java.util.Date () + "] hello " + who);  
    }  
}
```

- Configures the containing server
 - Advertises a named service
 - Associated with an implementing language
 - Java
 - Associated with an implementation unit
 - Java class
 - Defines the service's interface
 - The getGreeting method
 - Scope
 - Defines the lifetime of the object serving the invocation request

```
<isd:service xmlns:isd="http://xml.apache.org/xml-soap/deployment"
            id="urn:GreetingService">
  <isd:provider type="java"
                scope="Application"
                methods="getGreeting">
    <isd:java class="au.com.transentia.GreetingService"/>
  </isd:provider>
  <isd:faultListener>org.apache.soap.server.DOMFaultListener</isd:faultListener>
</isd:service>
```

- FaultListener
 - Listens for fault events, and adds a DOM Element to the returned XML document which represents the underlying SOAP exception

Ex: Deployment Tool

- Tomcat provides an administration tool
 - So no need to hand-hack deployment descriptors

IBM-SOAP Admin Tool - Microsoft Internet Explorer

Address <http://localhost:8080/soap/admin/index.html>

File Edit View Favorites Tools Back Forward Stop Reload

XML-SOAP Admin

List

Deploy

Un-deploy

Deploy a Service

Service Deployment Descriptor Template

Property	Details										
ID	<input type="text"/>										
Scope	<input type="text" value="Request"/>										
Methods	<input type="text"/> (Whitespace separated list of method names)										
Provider Type	<input type="text" value="Java"/>										
For User-Defined Provider Type, Enter FULL Class Name: <input type="text"/>											
Number of Options: <input type="text"/>											
	<table border="1"><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td><input type="text"/></td><td><input type="text"/></td></tr><tr><td><input type="text"/></td><td><input type="text"/></td></tr><tr><td><input type="text"/></td><td><input type="text"/></td></tr><tr><td><input type="text"/></td><td><input type="text"/></td></tr></tbody></table>	Key	Value	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Key	Value										
<input type="text"/>	<input type="text"/>										
<input type="text"/>	<input type="text"/>										
<input type="text"/>	<input type="text"/>										
<input type="text"/>	<input type="text"/>										
Java Provider	<table border="1"><tr><td>Provider Class</td><td><input type="text"/></td></tr><tr><td>Static?</td><td><input type="text" value="No"/></td></tr><tr><td>Script Language</td><td><input type="text" value="BML"/></td></tr><tr><td>Script Filename, or</td><td><input type="text"/></td></tr></table>	Provider Class	<input type="text"/>	Static?	<input type="text" value="No"/>	Script Language	<input type="text" value="BML"/>	Script Filename, or	<input type="text"/>		
Provider Class	<input type="text"/>										
Static?	<input type="text" value="No"/>										
Script Language	<input type="text" value="BML"/>										
Script Filename, or	<input type="text"/>										

Done Local intranet

- Controlling Tomcat
 - Simple script to deploy to Tomcat; list the active deployments; invoke the deployed service; undeploy the service; list deployments

```
@echo off
```

```
set CLASSPATH=...
```

```
set SRV=http://localhost:8080
```

```
set RPCR=%SRV%/soap/servlet/rpcrouter
```

```
Set MGR=java org.apache.soap.server.ServiceManagerClient %RPCR%
```

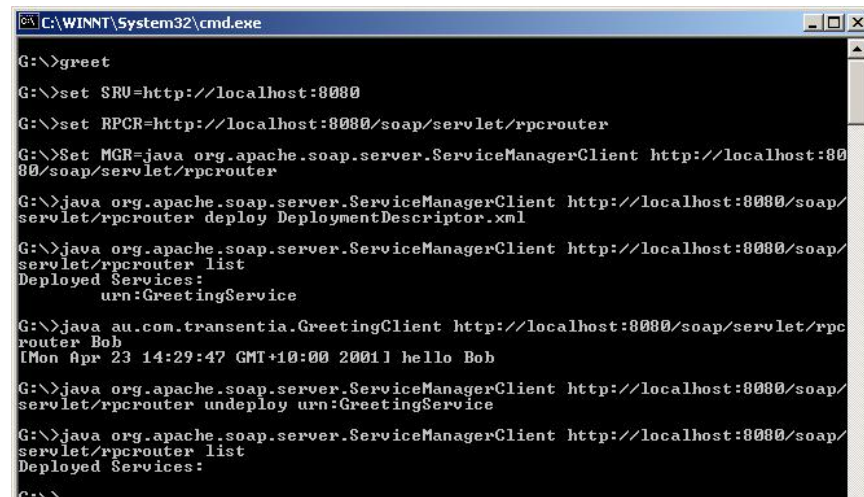
```
%MGR% deploy DeploymentDescriptor.xml
```

```
%MGR% list
```

```
java au.com.transentia.GreetingClient %RPCR% Bob
```

```
%MGR% undeploy urn:GreetingService
```

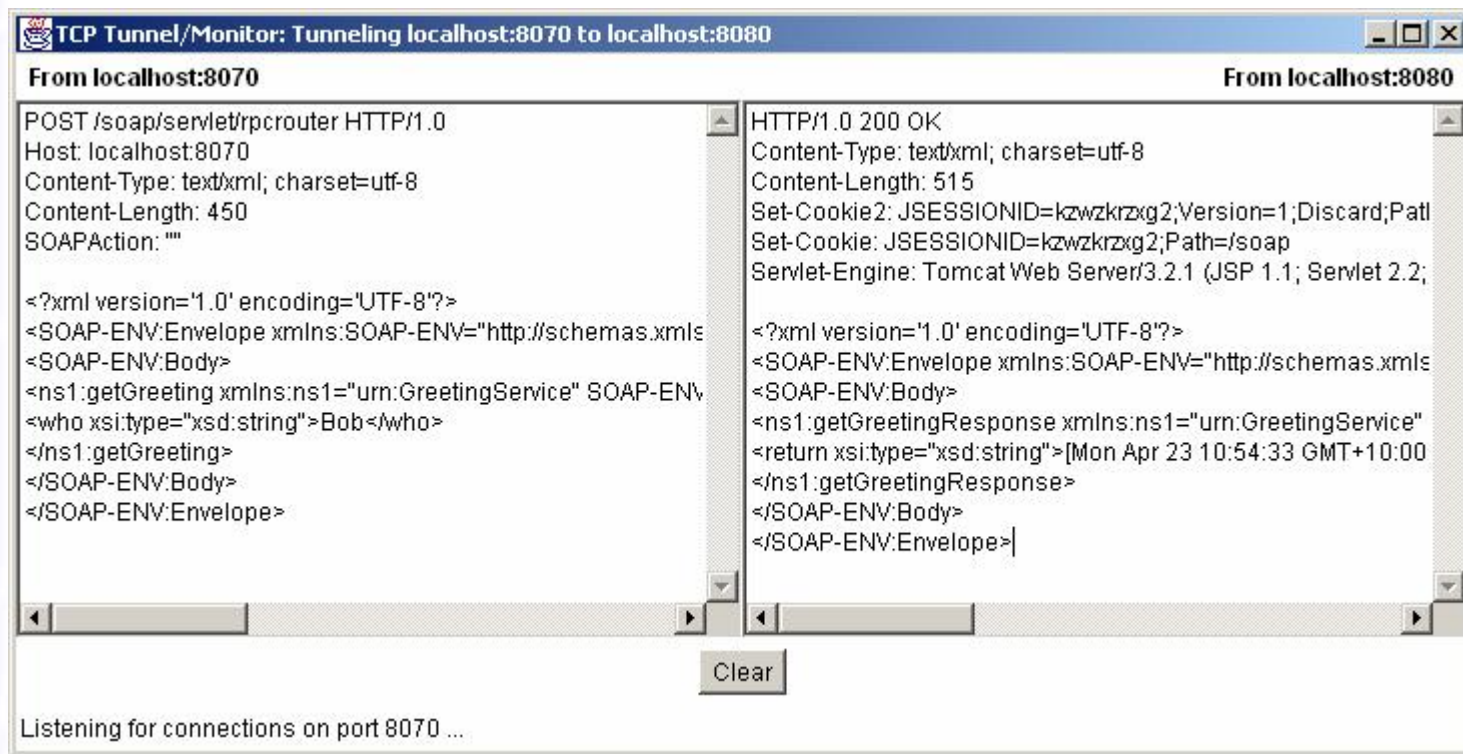
```
%MGR% list
```



```
C:\WINNT\System32\cmd.exe

G:\>greet
G:\>set SRV=http://localhost:8080
G:\>set RPCR=http://localhost:8080/soap/servlet/rpcrouter
G:\>Set MGR=java org.apache.soap.server.ServiceManagerClient http://localhost:8080/soap/servlet/rpcrouter
G:\>java org.apache.soap.server.ServiceManagerClient http://localhost:8080/soap/servlet/rpcrouter deploy DeploymentDescriptor.xml
G:\>java org.apache.soap.server.ServiceManagerClient http://localhost:8080/soap/servlet/rpcrouter list
Deployed Services:
    urn:GreetingService
G:\>java au.com.transentia.GreetingClient http://localhost:8080/soap/servlet/rpcrouter Bob
[Mon Apr 23 14:29:47 GMT+10:00 2001] hello Bob
G:\>java org.apache.soap.server.ServiceManagerClient http://localhost:8080/soap/servlet/rpcrouter undeploy urn:GreetingService
G:\>java org.apache.soap.server.ServiceManagerClient http://localhost:8080/soap/servlet/rpcrouter list
Deployed Services:
```

- The TCP Tunneller tool
 - A simple aid for debugging/visualisation of the SOAP protocol
 - A *much* more difficult proposition for most of the other NDRs out there...
 - A great strength(?)

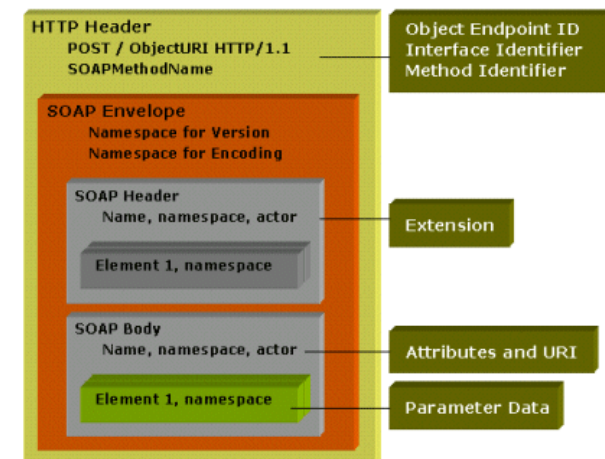


- HTTP header
 - SOAPAction field may indicate what the message is for
 - Firewalls, etc. may filter on this
- SOAP message has 3 parts
 - Envelope: defines namespaces, etc.
 - Header: optional for carrying auxiliary information (authentication, transaction, payment, etc.)
 - Note mustUnderstand field
 - “allows for robust evolution” of the protocol
 - Body: payload data
 - Uses XML Schema data types for tagging

```
POST /EventManager HTTP/1.1
Host: www.techmetrix.com
Content-Type: text/xml;
charset="utf-8"
```

```
Content-Length: 60
SOAPAction="http://www.techmetrix.com/Event#New Customer"
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xml.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xml.org/soap/encoding/" />
  <SOAP-ENV:Header>
    <t:Name xmlns:t="http://www.techmetrix.com/EventManager"
      SOAP-ENV:actor="http://schemas.xml.org/soap/actor/next/"
      SOAP-ENV:mustUnderstand="1">Dumser</t:Name>
  </SOAP-ENV:Header>
  <SOAP:Body>
    <m:NewCustomer xmlns:m="http://www.techmetrix.com/Event">
      <Enterprise>SQLI</Enterprise>
      <Address>Paris</Address>
    </m:NewCustomer>
  </SOAP:Body>
</SOAP:Envelope>
```



- Allows asynchronous request/reply

To: <soap@example.org>
 From: <soap@client.com>
 Reply-To: <soap@client.com>
 Date: Tue, 15 Nov 2001 23:27:00 -0700
 Message-Id: <1F75D4D515C3EC3F34FEAB51237675B5@client.com>
 MIME-Version: 1.0
 Content-Type: text/xml; charset=utf-8
 Content-Transfer-Encoding: QUOTED-PRINTABLE



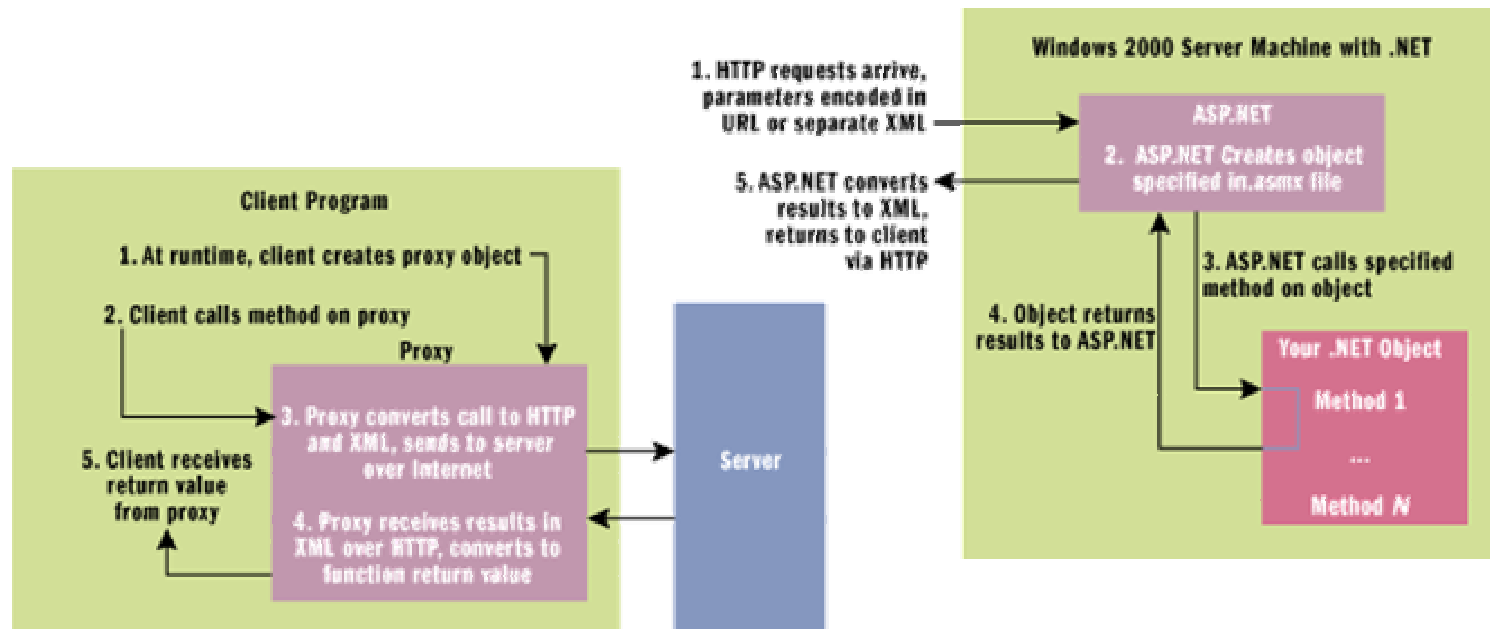
```
<?xml version=3D"1.0" encoding=3D"UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle=3D"http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENC=3D"http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV=3D"http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd=3D"http://www.w3.org/2001/XMLSchema"
  xmlns:xsi=3D"http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <m:echoString xmlns:m=3D"http://soapinterop.org/">
      <inputString>get your SOAP over SMTP here !</inputString>
    </m:echoString>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

To: <soap@client.com>
 From: <soap@example.org>
 Date: Tue, 13 Nov 2001 23:27:00 -0700
 In-Reply-To: <1F75D4D515C3EC3F34FEAB51237675B5@client.com>
 Message-Id: <FF75D4D515C3EC3F34FEAB51237675B5@soap.example.org>
 MIME-Version: 1.0
 Content-Type: TEXT/XML; charset=utf-8
 Content-Transfer-Encoding: QUOTED-PRINTABLE



```
<?xml version=3D"1.0" encoding=3D"UTF-8"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle=3D"http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENC=3D"http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV=3D"http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd=3D"http://www.w3.org/2001/XMLSchema"
  xmlns:xsi=3D"http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <m:echoStringResponse xmlns:m=3D"http://soapinterop.org/">
      <return>get your SOAP over SMTP here !</return>
    </m:echoStringResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


- A complex implementation
 - ‘High’ (usage/tool) and ‘Low’ (developer) APIs
 - Provides client-side proxy capabilities
 - Allows a client to access a web service as if it were a COM object
 - Also allows for callbacks
 - Not easy in many other toolkits



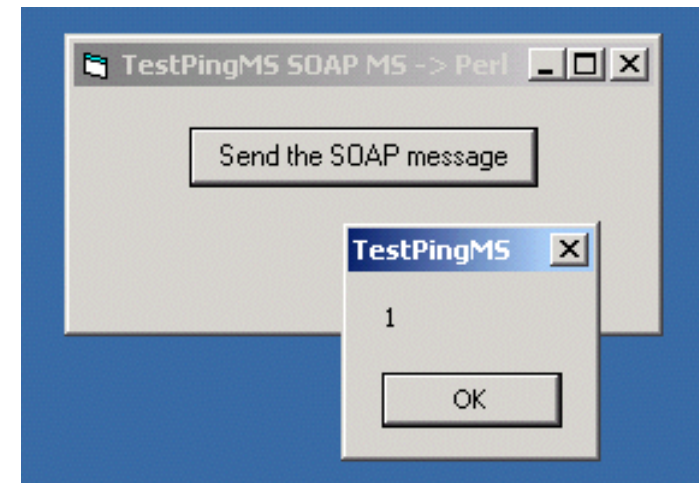
- Send and receive SOAP information without having to write code
 - Uses a WSDL file as a method to reach the requested service
 - Able to map an existing COM object by making it SOAP-enabled

```
<%@ LANGUAGE = VBScript %>  
<% Response.ContentType = "text/xml" %>  
<%  
set soapserver = CreateObject("MSSOAP.SoapServer")  
wsdl = Server.MapPath("Sample.wsdl")  
wsml = Server.MapPath("Sample.wsml")  
call soapserver.init(wsdl, wsml)  
call soapserver.SoapInvoke(request, response)  
%>
```

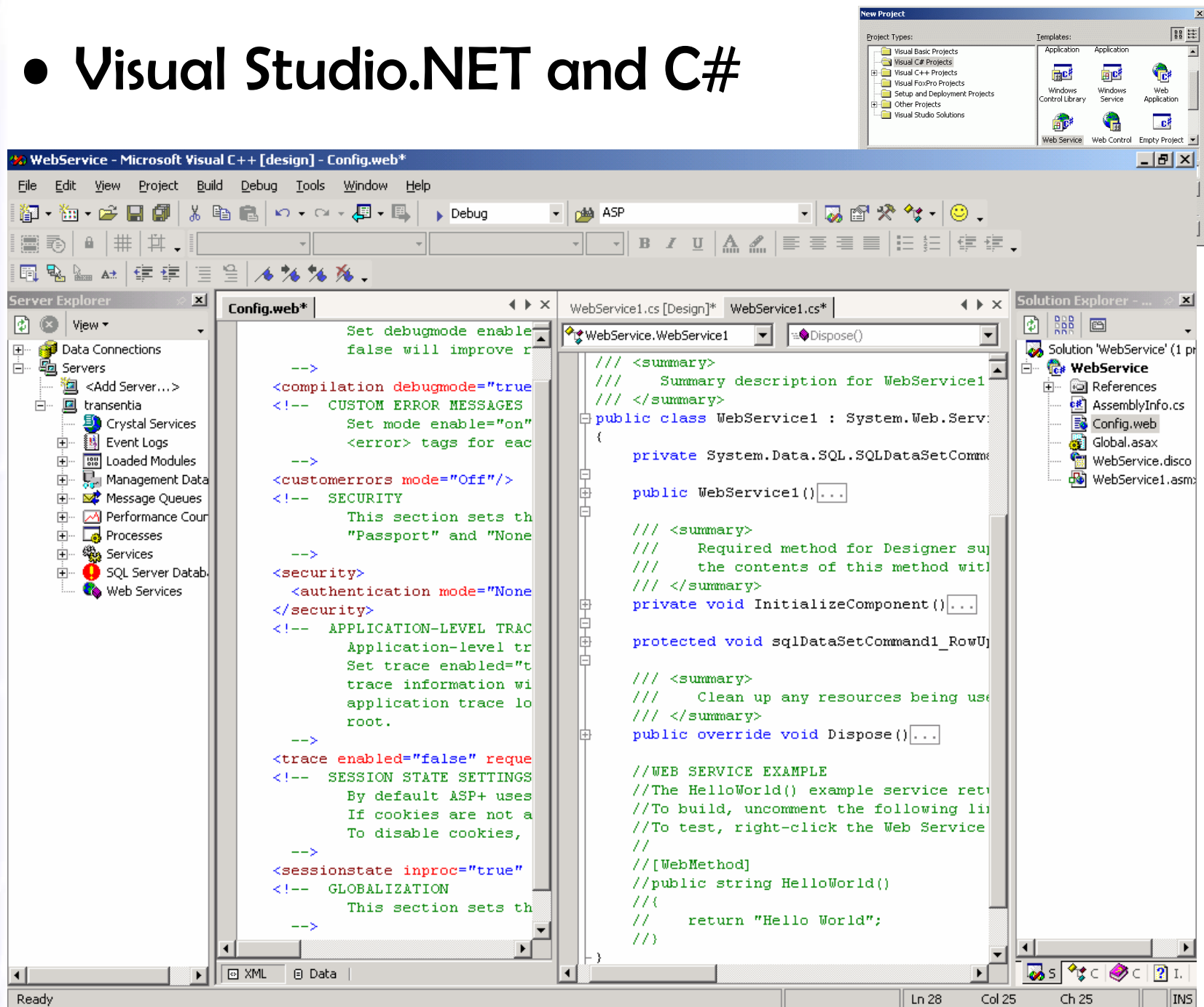
```
set soapclient = CreateObject("MSSOAP.SoapClient")  
Call soapclient.mssoapinit("http://www.xmethods.net/sd/PingService.wsdl",  
                             "PingService", "PingPort")  
wscript.echo soapclient.pingHost("www.yahoo.fr")
```

- Exposes the technical details of the SOAP message

```
Connector.Property("EndPointURL") = "http://services.xmethods.net:80/perl/soaplite.cgi"  
Connector.Connect Nothing  
Connector.Property("SoapAction") = "urn:xmethodsSoapPing#pingHost"  
  
Connector.BeginMessage Nothing  
  
Serializer.Init Connector.InputStream  
Serializer.startEnvelope  
Serializer.startBody  
Serializer.startElement "pingHost", "urn:xmethodsSoapPing", , "namesp01"  
Serializer.startElement "hostname"  
Serializer.writeString CStr("www.yahoo.com")  
Serializer.endElement  
Serializer.endElement  
Serializer.endBody  
Serializer.end Envelope  
  
Connector.EndMessage  
  
Reader.Load Connector.OutputStream  
MsgBox Reader.RPCResult.Text
```



- Visual Studio.NET and C#



- Windows Scripting Host Example
 - Client is JavaScript hosted by WSH; server is VB

```
var WSDL_URL = "http://MSSoapSampleServer/MSSoapSamples/Calc/Service/Rpc"

WScript.echo("Connecting: " + WSDL_URL)

var Calc = WScript.CreateObject("MSSOAP.SoapClient")

Calc.mssoapinit(WSDL_URL, "", "", "")

var Answer

Answer = Calc.add(14,28)
WScript.Echo("14+28=" + Answer)

Answer = Calc.subtract(73,31)
WScript.Echo("73-31=" + Answer)

Answer = Calc.multiply(14,3)
WScript.Echo("14* 3=" + Answer)

Answer = Calc.divide(126,3)
WScript.Echo("126/3=" + Answer)
```

```
VERSION 1.0 CLASS
BEGIN
    MultiUse = -1    'True
    Persistable = 0  'NotPersistable
    DataBindingBehavior = 0    'vbNone
    DataSourceBehavior  = 0    'vbNone
    MTSTransactionMode  = 0    'NotAnMTSObject
END
Attribute VB_Name = "Calc"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Option Explicit

Public Function Add(ByVal A As Double, ByVal B As Double)
As Double
    Add = A + B
End Function

ETC...
```

- SOAP can be used in ASP/Servlet/JSP/EJBs as well as standalone applications
 - Silverstream's 'eXtend' SOAP container for EJBs
 - BEA/Orion/etc.
 - Allowing an EJB to service SOAP requests requires a simple Deployment Descriptor change
 - ASP

- More usual server-side example...

```
<%@ Language=VBScript %>
<%
Option Explicit
%>
<!--#include file="ROPEConstants.inc"-->
<!--#include file="ServiceURI.inc"-->
<%
Dim oSOAP, oWire, oInfo, oMethod
Dim hResult, sState, sTaxable, sParam, sRequestPayload, sResponsePayload
sState = Request.Form("state2")
sTaxable = Request.Form("amount")
Response.Write "<HTML><HEAD><TITLE>GetStateTaxAmt</TITLE></HEAD><BODY>"
Set oSOAP = Server.CreateObject("ROPE.SOAPPackager")
hResult = oSOAP.LoadServicesDescription(icURI, URI_SDL)
If hResult <> False Then
    Set oInfo = Server.CreateObject("ROPE.ServiceDescriptors")
    Set oMethod = Server.CreateObject("ROPE.SDMethodInfo")
    Set oInfo = oSOAP.GetServiceDescriptors(icMETHODINFO)
    Set oMethod = oInfo.Item("GetStateTaxAmt")
    Set oWire = Server.CreateObject("ROPE.WireTransfer")
    With oSOAP

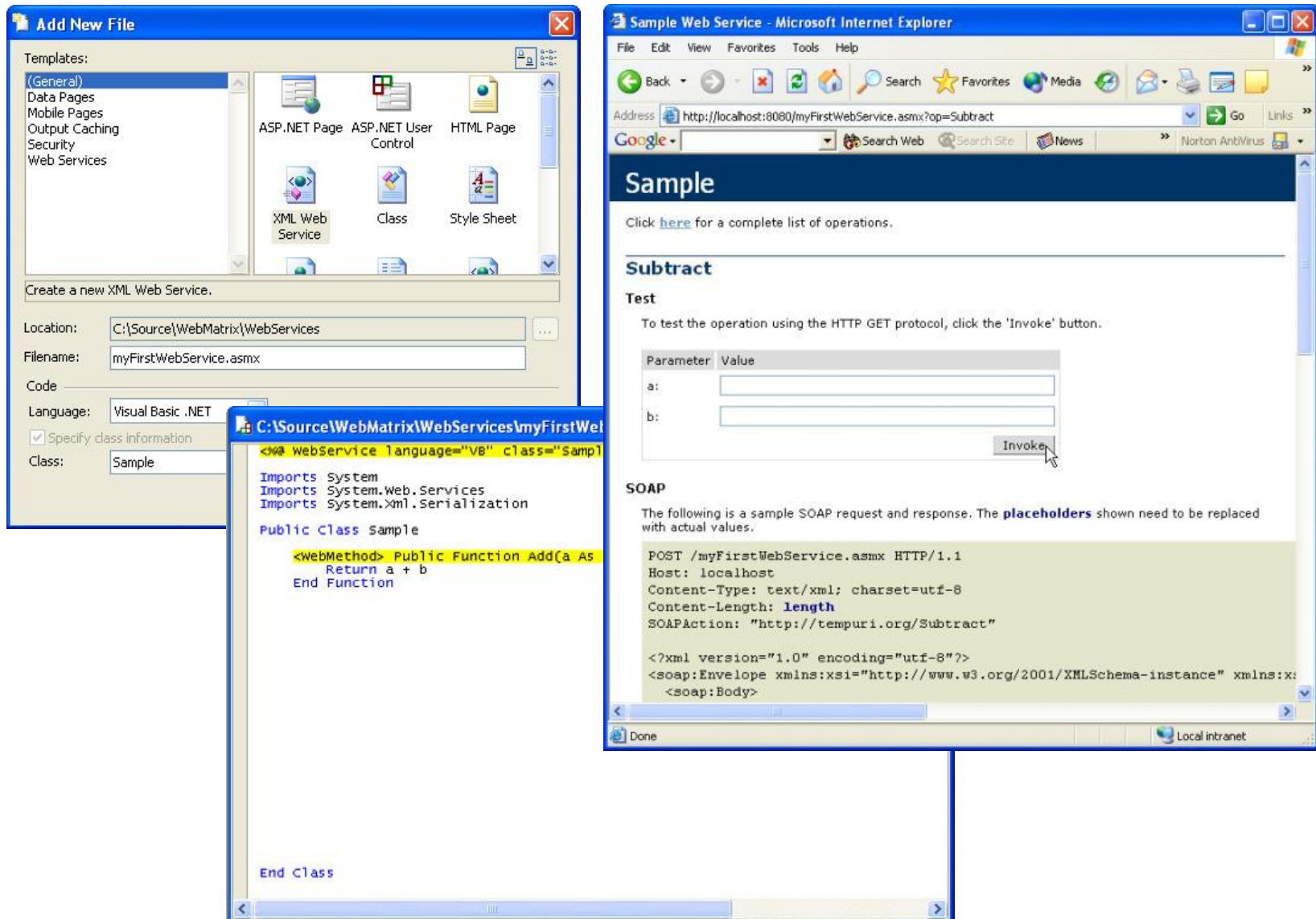
        .SetPayloadData icREQUEST, "", "GetStateTaxAmt", oMethod.InputStructure
        .SetParameter icREQUEST, "StateAbbrev", CStr(sState)
        .SetParameter icREQUEST, "TaxableTotal", CStr(sTaxable)
        sRequestPayload = .GetPayload(icREQUEST)

        oWire.AddStdSOAPHeaders URI_SDL, "GetStateTaxAmt", Len(sRequestPayload)
        sResponsePayload = oWire.PostDataToURI(URI_Endpoint, sRequestPayload)
        Set oWire = Nothing

        .SetPayload icRESPONSE, sResponsePayload

    End With
    sParam = oSOAP.GetParameter(icRESPONSE, "return")
    Response.Write "Sales tax in " & sState & " for " & sTaxable & " is $" & sParam
    Set oWire = Nothing
    Set oMethod = Nothing
    Set oInfo = Nothing
Else
    Response.Write "Failed to load service description."
End If
Set oSOAP = Nothing
Response.Write "</BODY></HTML>"
%>
```


- Free IDE promoted by Microsoft
 - Cut-down Visual Studio.Net...sort of



• Ada Web Server

– <http://libre.act-europe.fr/>

```
with Ada.Text_IO;
with SOAP.Types;
with SOAP.Parameters;
with SOAP.Client;
with SOAP.Message.Payload;
with SOAP.Message.XML;
with SOAP.Message.Response.Error;
procedure SOAP_Client is
  use Ada;
  use SOAP;
  use SOAP.Types;
  use type SOAP.Parameters.List;
  procedure Good is
    P_Set : Parameters.List := +I (10, "p1") & I (32, "p2") & F (12.4, "p3");
    P      : Message.Payload.Object;
  begin
    P := Message.Payload.Build ("This_Proc", P_Set);
    declare
      R : constant Message.Response.Object'Class
        := SOAP.Client.Call ("http://localhost:8080/soapdemo", P);
      P : constant Parameters.List
        := SOAP.Message.Parameters (R);
      Myres : constant Integer := SOAP.Parameters.Get (P, "myres");
    begin
      Text_IO.Put_Line ("----- XML Response -----");
      Text_IO.Put_Line (Message.XML.Image (R));
      Text_IO.New_Line;
      Text_IO.Put_Line ("Myres = " & Integer'Image (Myres));
      Text_IO.New_Line;
    end;
  end Good;
begin
  Text_IO.Put_Line ("==> Call GOOD"); Good;
end SOAP_Client;
```

- SOAPLite
 - <http://www.soaplite.com>

Client

```
use SOAP::Lite;

print SOAP::Lite
  -> service('http://services.xmethods.net/soap/urn:xmethods-delayed-quotes.wsdl')
  -> getQuote('MSFT');
```

CGI-based Server

```
use SOAP::Transport::HTTP;

SOAP::Transport::HTTP::CGI
  -> dispatch_to('/Path/To/Deployed/Modules', 'Module::Name', 'Module::method')
  -> handle;
```

- Very important aspect
- Two major efforts to improve interoperability currently underway
 - SOAPBuilders
 - Community effort; mainly SOAP stack implementors getting together for 'interopathons' and living in newsgroups; some face-to-face meetings
 - *"...it's really hard to be mean to a person you've had a few beers with."*
 - WS-I
 - Political/formal entity aimed at giving 'users' a voice
 - SOAPBuilders and WS-I 'cross-fertilise'

Issue: Interoperability...

- Initially not so good
 - Getting better now
- WS-I helping things along
 - Industry forum
- SoapBuilders Interoperability Lab
 - Round 1 testing
 - SOAP interoperability
 - <http://www.xmethods.net/ilab/>
 - Round 2 testing
 - SOAP + WSDL interoperability
 - <http://www.whitemesa.com/interop.htm>
- Message
 - We're not there yet...

● Test Matrices

Results for: MS SOAP Toolkit 3.0

echoHdrString (Understandable, MU="0", Actor: "next")	PASS
echoHdrStruct (Understandable, MU="0", Actor: "next")	PASS
echoHdrString (Understandable, MU="1", Actor: "next")	PASS
echoHdrStruct (Understandable, MU="1", Actor: "next")	PASS
echoHdrString (Not Understandable, MU="0", Actor: "next")	PASS
echoHdrStruct (Not Understandable, MU="0", Actor: "next")	PASS
echoHdrString (Not Understandable, MU="1", Actor: "next")	PASS
echoHdrStruct (Not Understandable, MU="1", Actor: "next")	PASS
echoHdrString (Understandable, MU="0", Actor: "other")	FAIL
echoHdrStruct (Understandable, MU="0", Actor: "other")	FAIL
echoHdrString (Understandable, MU="1", Actor: "other")	????

	echoStructAs Simple Types	echoSimpleTypes AsStruct	echo2DStringArray	echoNestedStr
4s4c	PASS	PASS	FAIL	PASS
4s4c 2.0	FAIL	PASS	PASS	FAIL
Apache Axis	PASS	PASS	PASS	PASS
Apache Axis (Map)	1	1	1	1
ASP.NET Web Services	PASS	PASS	FAIL	PASS
Delphi SOAP	FAIL	PASS	PASS	PASS
EasySoap++	PASS	PASS	PASS	PASS
gSOAP	2	2	2	2
GLUE	FAIL	FAIL	FAIL	FAIL
HP SOAP	FAIL	PASS	PASS	PASS
IONA XMLBus	FAIL	40	40	40
Kafka XSLT	FAIL	FAIL	FAIL	PASS
MS.NET Remoting	PASS	PASS	PASS	PASS
MS STK v2.0 (typed)	FAIL	PASS	PASS	PASS
MS STK v3.0	PASS	PASS	PASS	PASS
MS STK v3.0 (typed)	FAIL	PASS	PASS	FAIL
NuSOAP	PASS	PASS	PASS	PASS
OpenLink Virtuoso	4	4	4	4
PEAR SOAP	PASS	PASS	FAIL	PASS
Phalanx (typed)	5	5	5	5
Phalanx (untyped)	5	5	5	5
SIM	FAIL	PASS	N/A	FAIL
SOAP4R	PASS	PASS	PASS	PASS
SOAP:Lite	8	8	8	8
Spheon JSOAP	6	6	6	6
Spray 2001	FAIL	PASS	PASS	FAIL
SQLData SOAP Server	PASS	PASS	PASS	PASS
Sun	PASS	PASS	PASS	PASS
WASP Advanced 4.0	PASS	PASS	PASS	PASS
White Mesa SOAP Server	PASS	PASS	PASS	PASS
Wingfoot	PASS	PASS	34	34

Interop results for: ASP.NET - Microsoft Internet Explorer

Address: <http://mssoapinterop.org/results/res.aspx>

Tests against server: Delphi

Endpoint: <http://soap-server.borland.com/WebServices/Interop/cgi-bin/InteropService.exe/soap/InteropTestPortType>

Name	Result	Request trace	Response trace
1. echoVoid	PASSED	Request trace	Response trace
2. echoString	PASSED	Request trace	Response trace
3. echoStringArray	PASSED	Request trace	Response trace
4. echoInteger	PASSED	Request trace	Response trace
5. echoIntegerArray	PASSED	Request trace	Response trace
6. echoFloat	PASSED	Request trace	Response trace
7. echoFloatArray	PASSED	Request trace	Response trace
8. echoSOAPStruct	PASSED	Request trace	Response trace
9. echoStructArray	FAILED	Request trace	Response trace
10. echoDate	FAILED	Request trace	Response trace
11. echoDecimal	PASSED	Request trace	Response trace
12. echoBoolean	PASSED	Request trace	Response trace

Tests against server: 4s4c

Endpoint: <http://soap.4s4c.com/llab/soap.asp>

Name	Result	Request trace	Response trace
1. echoVoid	FAILED	Request trace	Response trace
2. echoString	FAILED	Request trace	Response trace
3. echoStringArray	FAILED	Request trace	Response trace
4. echoInteger	FAILED	Request trace	Response trace
5. echoIntegerArray	FAILED	Request trace	Response trace
6. echoFloat	FAILED	Request trace	Response trace
7. echoFloatArray	FAILED	Request trace	Response trace
8. echoStruct	FAILED	Request trace	Response trace

	String	String Array	Integer	Integer Array	Float	Float Array	Struct	Struct Array	Void	Base64	Hex Binary	Date	Decimal	Boolean	Map	Any Array
4s4c	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
4s4c 2.0	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Apache Axis	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Apache SOAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
ASP .Net	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
CapeConnect	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Delphi	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
EasySoap++	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
gSOAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Frontier	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Obix	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
sSOAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
HP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
IONA XMLBus	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Kafka XSLT	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MSOAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MS STK v3 Untyped	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MS STK v3 Typed	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MS Net Remoting	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
NuSOAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
NuWaves	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
OpenLink	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
PEAR	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SIM	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Soap4R	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Soap4R	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Soap:Lite	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Spheon JSOAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Spray 2001	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SQLData	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Sun JAX-RPC	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
ThruSOAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
VFW Openlink	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
WASP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
WASP C++	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

- Useful references
 - Web Service Interoperability Organization at <http://www.ws-i.org/>
 - White Mesa Software at <http://www.whitemesa.com/>
 - Microsoft SOAP Interop Server at <http://www.mssoapinterop.org/>

- Support for marshalling primitive and common types “built in”
 - org.apache.soap.encoding.SOAPMappingRegistry
 - A map of all known types and their associated registered handler classes
 - The built-in encoders/decoders are simply classes implementing these interfaces that are preregistered
 - ShortDeserializer, MimePartSerializer, DateSerializer, etc.
- For the rest:
 - org.apache.soap.util.xml.(De)Serializer interfaces
 - Specify (un)marshall method handling transformation from/to XML
- Undergoing great change
 - Support for multi-dimensional arrays is *continually* being re-written, for example

- An issue of some controversy:
 - *“Why SOAP doesn’t lack security while it does...The answer is simple: it’s not its job.”*
 - *“While the SOAP specification does seemingly ignore all security issues, the infrastructure to deploy SOAP on HTTP(S) in a secure way exists **now**. By using all features of SSL (including client-side certificates), authentication, integrity and privacy can be guaranteed for all traffic.”*
 - *“...HTTP passes through firewalls more easily than the CORBA and DCOM protocols. ...Given that HTTP streams may now contain complex client-server interactions, firewall vendors will need to examine the semantics of the traffic at the encapsulated level. It is arguably more difficult to do this for SOAP traffic than for CORBA or DCOM requests.”*

- *“New verbs have been introduced in an HTTP protocol extension proposal. The verbs, such as M-POST, are designed to simplify administration of firewalls and proxies.”*

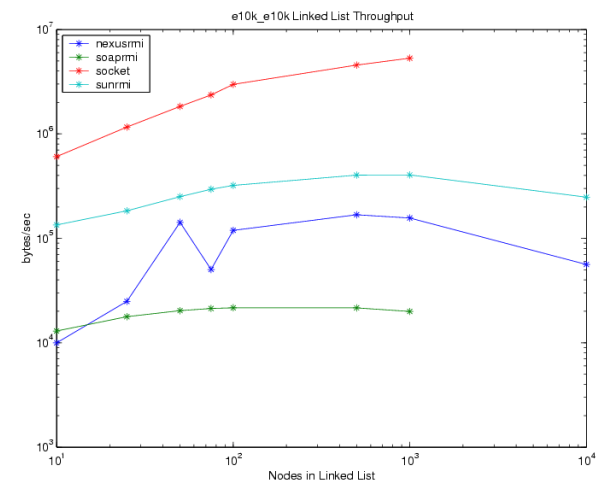
```
M-POST /foobar HTTP/1.1
Host: 209.110.197.2
Man: "urn:schemas-xmlsoap-org:soap.v1; ns=42"
42-SOAPMethodName: urn:bobnsid:IFoo#DoIt
```

- *Unease exists: people worry about permitting arbitrary communication across a firewall*
 - *First we close all ports except 80, then we allow arbitrary objects to pass through port 80...*
 - *We are not, however, passing the actual executable code. This may help us salvage our security...text/xml is easier to grok than binary executables*

- Firewalls are being extended
 - Check Point Software's new "Application Intelligence Technology" provides:
 - Stateful Inspection of Web services traffic from the network layer to the application layer
 - QoS of Web services traffic

- Experiments at Indiana University
 - <http://www.extreme.indiana.edu/soap/scOO/paper/index.html>
<double> 3.141592653589793E+000 </double>
 - Data representation size
 - In general is about 10 times the size of binary representations
 - E.g. sending this 8-byte double in XML, requires 40 bytes of data
 - » If Unicode is used, this doubles!
 - IBM estimates an *average* message size of 60K!
 - Speed
 - Serializing Java objects into SOAP-encoded XML data takes approximately ten times more memory than the binary representation....
Serialization and deserialization speeds...are approximately 100 times slower and their throughputs are also a 100 times lower

"In my experiments with Apache, I normally get about 30 round trip messages per second between Java programs on the same machine. I have been evaluating another SOAP implementation that gets around 700 round trip messages per second in the same configuration, so obviously there is a lot of room for improvement"



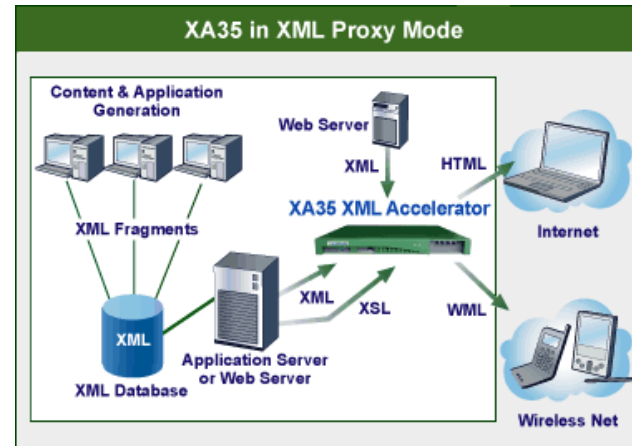
Issue: Performance...

- Do not despair!
 - Is the message coming from the SOAP community...
- There may be opportunities for optimisations
 - Compression; dictionaries, etc....
 - Could postulate the use of Header information to handshake an optimisation protocol
 - Turn feature 'X' on, turn 'Y' off, etc.
- Marshalling time may represent a fraction of network I/O time
 - The network may be the limiting factor...
 - Hmm...for an Amazon-level service, perhaps but for many (sporadically-used) services will this be true?
- There are 'always' bigger pipes/servers/memory chips, etc.
 - Vendors are just going to *love* SOAP ☺

- W3C work
 - SOAP Optimized Serialization Requirements
 - Gobbledegook!
- Poss. partial remedy
 - HTTP1.1 Chunking with persistent connection

```
HTTP/1.1 411 Size Required
Date: Thu, 13 Sep 2001 15:53:37 GMT
Server: Apache/1.3.21-dev (Unix) PHP/3.0.18
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
    [empty delimiting line]
165    [size of chunk that follows]
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>411 Size Required</TITLE>
</HEAD><BODY>
<H1>Size Required</H1>
A request of the requested method GET requires a valid Content-size.<P>
chunked Transfer-Encoding forbidden: /cgi-bin/showenv<P>
<HR>
<ADDRESS>Apache/1.3.21-dev Server at localhost Port 10000</ADDRESS>
</BODY></HTML>
    [empty delimiting line]
0    [zero-size chunk; end of message-body]
    [empty delimiting line]
```

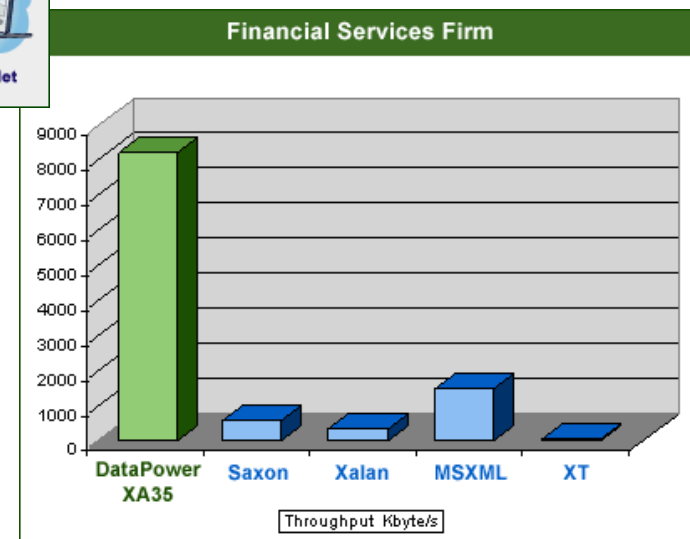

- Hardware acceleration
 - <http://www.datapower.com/xa35accelerator.html>



"WEB SERVICES WILL INFLATE XML TRAFFIC ON THE NET. AS A RESULT, FIRMS WILL TURN TO SPECIALIZED HARDWARE FROM NEW PLAYERS LIKE **DATAPOWER** TO OFFLOAD XML PROCESSING FROM SERVERS TO DEDICATED NETWORK DEVICES."

DAVID TRUOG
PRINCIPAL ANALYST
FORRESTER RESEARCH
"HARDWARE HELP FOR
THE LOOMING XML BLITZ"

FORRESTER



- BEEP
 - Blocks Extensible Exchange Protocol
 - Multiplexing of connections to a server for efficiency
- Useful info
 - *Performance testing SOAP-based applications*,
<http://www-106.ibm.com/developerworks/webservices/library/ws-testsoap/>
 - Latency Performance of SOAP Implementations,
<http://www.caip.rutgers.edu/TASSL/Papers/p2p-p2pws02-soap.pdf>

- How difficult can this be?
- An extension to SOAP 1.1
 - Taking a number of iterations

```
MIME-Version: 1.0
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
    start="<claim061400a.xml@claiming-it.com>"
Content-Description: This is the optional message description.
```

```
--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <claim061400a.xml@claiming-it.com>

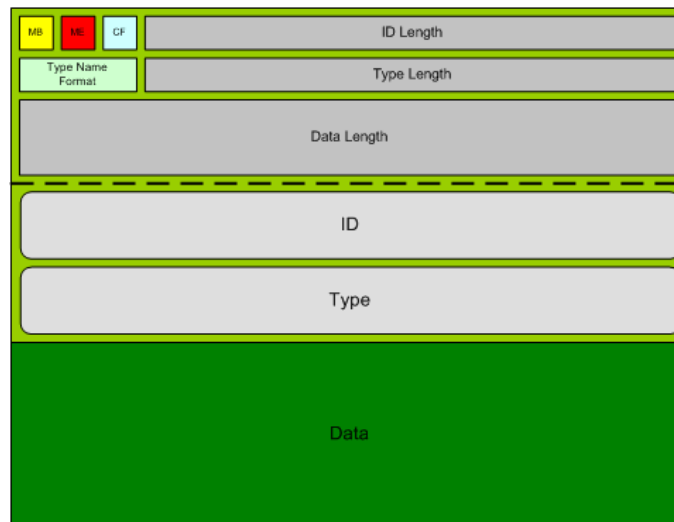
<?xml version='1.0' ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
..
<theSignedForm href="cid:claim061400a.tiff@claiming-it.com"/>
..
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
--MIME_boundary
Content-Type: image/tiff
Content-Transfer-Encoding: binary
Content-ID: <claim061400a.tiff@claiming-it.com>
```

```
...binary TIFF image...
--MIME_boundary--
```

Issue: Attachments...

- **Microsoft proposes DIME**
 - **Direct Internet Message Encapsulation**
 - **a mechanism for packaging binary data with SOAP messages**
 - **data not in XML format, applications need not be constrained by the SOAP specification**
 - **Being proposed as WS-Attachments**



```

1 0 0 0000000000000000
010 0000000101001
000000000000000000000000010110110
http://schemas.xmlsoap.org/soap/envelope/
<envelope>
  <body>
    <ConvertImage>
      <ConversionType>JPEGtoGIF</ConversionType>
      <image href="Image1" />
    </ConvertImage>
  </body>
</envelope>

```

```

0 0 1 00000000000110
001 0000000001010
0000000000000000000111111111111111
Image1
image/jpeg
  <64K of binary data>

```

```

0 1 0 00000000000000
000 00000000000000
00000000000000000000011000111110000
  <12784 bytes of binary data>

```

- Reflects changes in XML Schema/datatypes specs.
 - adopts the XML Infoset
- Better support for arrays
 - or at least, different...
- Finer-grained exception reporting
- Apache-Axis implementation
 - *very* early days

"The issue is whether or not Web services, as about to be described by the SOAP 1.2 protocol specification, has anything whatsoever to do with the Web. There are fundamental architectural differences between frameworks for remote procedure call, such as CORBA, DCOM [Distributed Component Object Model], and SOAP 1.1, and how the Web was designed"

"SOAP Version 1.2 brings a refined processing model, which reduces ambiguities created by various interpretations of the SOAP/1.1 Specification. SOAP Version 1.2 includes strong recommendations for explicit error messages for mandatory extensions, giving developers better information, and helping them to develop better applications. This provides a solid first step forward in ensuring better interoperability and extensibility in SOAP Version 1.2."

- 1.2 now allows the exchange of SOAP messages either as payload of a HTTP POST request and response, or as a SOAP message in the response to a HTTP GET

```
GET /travelcompany.example.org/reservations?code=FT35ZBQ HTTP/1.1
Host: travelcompany.example.org
Accept: text/html, application/soap+xml
```

- Important facility for idempotent resources
- Good overview at
 - http://www.idealliance.org/papers/xml02/dx_xml02/papers/02-02-02/02-02-02.html

“Debate Foams Over SOAP

...two vendors, webMethods and Epicentric, have stated they may have possible patents pertaining to SOAP 1.2, with webMethods stating as of Sept. 11 it is not willing to waive its patent rights and Epicentric, recently acquired by Vignette, saying it has not been given permission to make a public statement...”

- Currently in version 1.1
 - “We still have a **lot** of open bugs on the books”
- The “third generation of Apache SOAP”
- Contains
 - a simple stand-alone server
 - allows ‘.JWS’ files for instant deployment
 - a server which plugs into servlet engines such as Tomcat
 - support for the Web Service Description Language (WSDL)
 - emitter tooling that generates Java classes from WSDL and WSDL from Java interfaces
 - a tool for monitoring TCP/IP packets
- Provides
 - preliminary support for SOAP with attachments
 - a reengineered framework
 - chains of handlers



- Claims
 - more speed
 - better interoperability
 - but...see chart
- Slightly different API

```
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import javax.xml.namespace.QName;

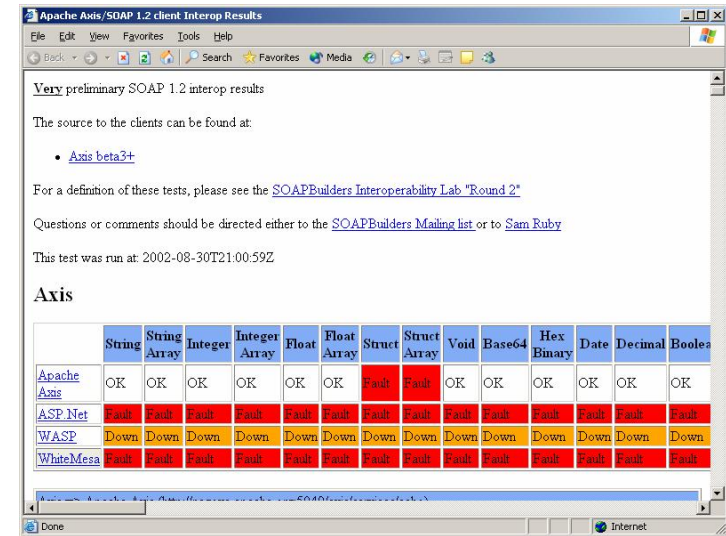
public class TestClient
{
    public static void main(String [] args) {
        try {
            String endpoint =
                "http://nagoya.apache.org:5049/axis/servlet/AxisServlet";

            Service service = new Service();
            Call call = (Call) service.createCall();

            call.setTargetEndpointAddress( new java.net.URL(endpoint) );
            call.setOperationName(new QName("http://soapinterop.org/", "echoString") );
            call.addParameter("testParam", org.apache.axis.Constants.XSD_STRING,
                javax.xml.rpc.ParameterMode.IN);
            call.setReturnType(org.apache.axis.Constants.XSD_STRING);

            String ret = (String) call.invoke( new Object[] { "Hello!" } );

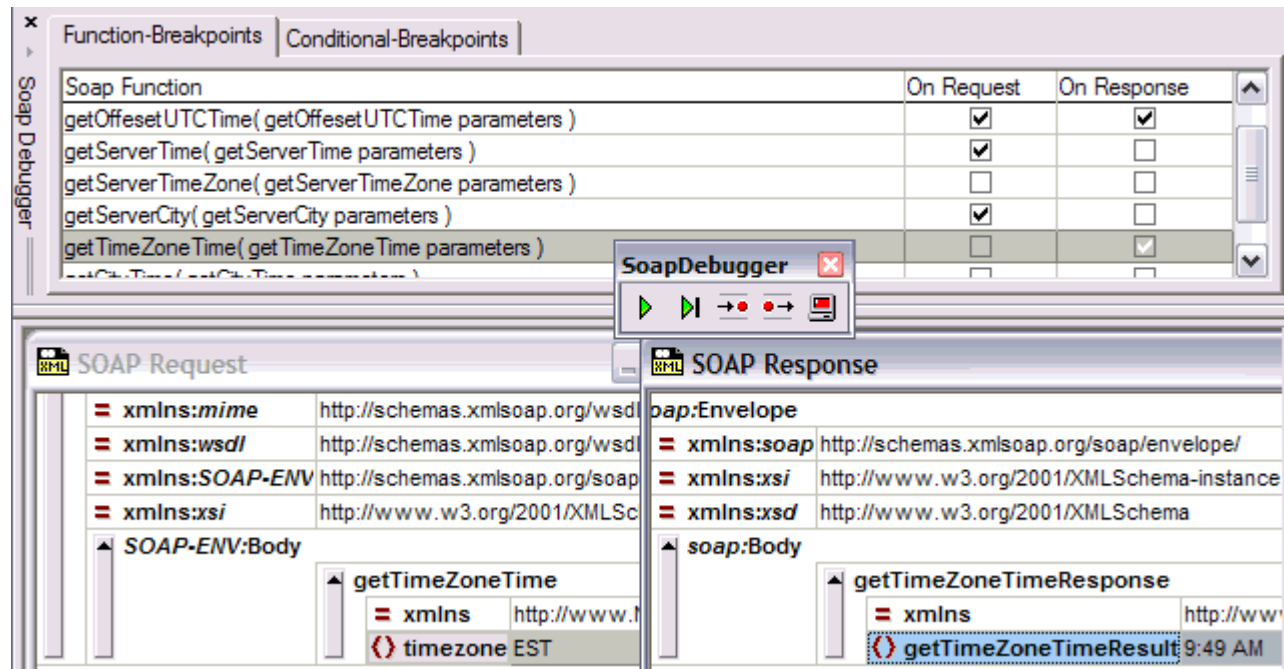
            System.out.println("Sent 'Hello!', got '" + ret + "'");
        } catch (Exception e) {
            System.err.println(e.toString());
        }
    }
}
```



- **Sun's Java Web Services Developer Pack**
 - Includes:
 - Java API for XML Messaging (JAXM)
 - Soap with Attachments API for Java (SAAJ)
 - Java API for XML Processing (JAXP)
 - Java API for XML Registries (JAXR)
 - Java API for XML-based RPC (JAX-RPC)
 - JavaServer Pages Standard Tag Library (JSTL)
 - Tomcat (Java servlet and JavaServer Pages container)
 - Administration Tool
 - Web Application Manager
 - Ant build tool
 - deploytool Web application deployment utility
 - Registry Server
 - Can be integrated with a J2EE server
 - Or at least, the reference implementation

- IBM's "showcase of emerging technology" in this area
- Contains:
 - Pre-installed environment (Xerces, Xalan, SOAP, etc.). Also instructions and configuration utilities for using WSTK with the WebSphere SDK for Web Services, WebSphere Application Server, and Jakarta Tomcat.
 - Client-side APIs for interfacing with a WSDL document; a UDDI registry; and APIs for publishing and binding Web services.
 - Demonstrations that can be used to test Web Services Tool Kit: Publish a service and then use a client that communicates with the UDDI registry to find the Web service and invoke it.
 - A set of Utility web services: User Profile, Metering, Accounting, Contract, Common Data, and Notification. Provided is a composite demonstration, which makes use of these utility services and makes use of Web Service Level Agreement (WSLA), Web Services Management Middleware, and Service Desk technologies.
 - Technology previews of security functions such as WS-Security, systems management, and SOAP technologies.
 - Documentation about the Web Services Architecture, WS-Inspection, and WSDL specifications.
 - An overview of Web Services Architecture, a white paper on developing Web services, and tutorials.
- Live demo on the alphaworks website

- XML Spy has a SOAP debugger tool
 - fancy version of the TCP Tunneller



- Rosy
 - Heavyweight support guarantees a future
 - Microsoft, IBM *and* the hacker community!
 - Minimal attempts at perverting/adapting
 - Being rapidly adopted
 - A foundation for new systems
 - UDDI, Web Services, etc.
 - Actually has a reason to exist
 - Not just a B.W.C. technology
 - C.f. WAP ☺!
- Issues remain
 - Security; performance; interoperability
 - Will be sorted eventually

- *“Thanks to SOAP, we finally have the middleware that will enable us to achieve client-server between applications and via Internet...For at least fifteen years, we have been waiting for a simple, lightweight RPC that lets you outside of the local network. Previous attempts failed because they weren't simple, and they weren't standard. It's not because CORBA, DCOM or even RMI are heavy, slow and painstaking to use that they were dropped by developers writing Web applications. It's above all because none could naturally cross the Firewall barrier...*

SOAP does not suffer this handicap, because it's based on HTTP. And this gives it one more advantage in terms of compliance with real standards!”

— ALAIN LEFEBVRE, Vice President, Groupe SQLI

Useful SOAP References

- The Web services (r)evolution (4 parts)
 - <ftp://www6.software.ibm.com/software/developer/library/ws-peer{1,2,3,4}.pdf>
- SOAP Software Directory
 - [http://www.soapware.org/directory/4/implementations](http://www.software.ibm.com/developer/library/ws-peer{1,2,3,4}.pdf)
- Why SOAP doesn't lack security while it does
 - <http://www.newtelligence.com/news/text01.asp>
- "Requirements for and Evaluation of RMI Protocols for Scientific Computing"
 - <http://www.extreme.indiana.edu/soap/sc00/paper/paper.html>
- SOAP: Simple Object Access Protocol
 - <http://www.techmetrix.com/trendmarkers/tmk1200/tmk1200-3.php3>
(<http://www.techmetrix.com/> is a good site!)
- Simple Object Access Protocol
 - <http://www.arsdigita.com/asj/soap/>
- Simple Object Access Protocol: A Step-By-Step Approach
 - http://www.vbip.com/xml/soap_syd.asp
- SOAP for Platform-Neutral Interoperability
 - <http://www.xmlmag.com/upload/free/features/xml/2000/04fal00/kb0004/kb0004.asp>
- XML and SOAP—Essential to .NET
 - http://www.dotnettoday.com/articles/art003_essentialdotnet.asp
- SOAP Weblog
 - <http://soap.weblogs.com/>