



SOAP

Simple Object Access Protocol

Bob Brown
transentia
pty. ltd.

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 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

- 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 version 1.1)
 - 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

- *“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*"

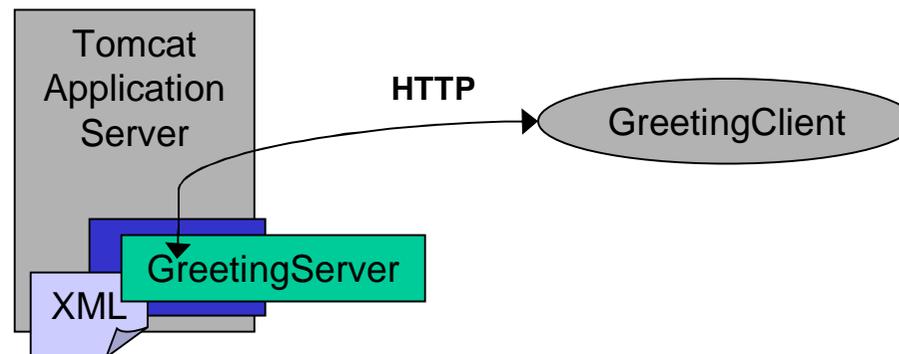
- Overcomes “the tyranny of the firewall”
 - Rides atop standard Internet transports such as HTTP/80; SMTP/25
 - Most firewalls are transparent to these...
 - *“Although HTTP will probably be the primary transport protocol for SOAP messages, the latest revision of the specification allows for sending 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 falling 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”*
 - SOAP will underlie much of .NET
 - ASP.NET; Visual Basic.NET, etc.
 - BizTalk server
 - Visual Studio.NET will make 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

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

XML-SOAP Admin

Deploy a Service

Service Deployment Descriptor Template

Property	Details								
ID	<input type="text"/>								
Scope	<input type="text" value="Request"/>								
Methods	<input type="text"/> <small>(Whitespace separated list of method names)</small>								
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></tbody></table>	Key	Value	<input type="text"/>					
Key	Value								
<input type="text"/>	<input type="text"/>								
<input type="text"/>	<input type="text"/>								
<input type="text"/>	<input type="text"/>								
Java Provider	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
```

```
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:
```

Request-Reply Interaction

```
boolean PlaceOrder([in] Title string,  
                   [in] Author string,  
                   [out] DaysToDelivery integer);
```

```
POST /BookServer HTTP/1.1  
Host: www.qwickbooks.com  
Content-Type: text/xml-SOAP  
Content-Length: nnnn  
SOAPAction: "Some-Namespace-URI#PlaceOrder"
```

```
<SOAP:Envelope xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1">  
  <SOAP:Body>  
    <m:PlaceOrder xmlns:m="Some-Namespace-URI">  
      <Title>Happy All The Time</Title>  
      <Author>Laurie Colwin</Author>  
    </m:PlaceOrder>  
  </SOAP:Body>  
</SOAP:Envelope>
```



```
HTTP/1.1 200 OK  
Connection: close  
Content-Type: text/xml  
Content-Length: nnnn
```



```
<SOAP:Envelope xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1">  
  <SOAP:Body>  
    <m:PlaceOrderResponse xmlns:m="Some-Namespace-URI">  
      <return>1</return>  
      <DaysToDelivery>7</DaysToDelivery>  
    </m:PlaceOrderResponse>  
  </SOAP:Body>  
</SOAP:Envelope>
```

- 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(?)

The screenshot shows a window titled "TCP Tunnel/Monitor: Tunneling localhost:8070 to localhost:8080". The window is split into two panes. The left pane, labeled "From localhost:8070", shows an HTTP POST request to "/soap/servlet/rpcrouter" with a SOAP envelope. The right pane, labeled "From localhost:8080", shows the corresponding HTTP 200 OK response with a SOAP envelope. A "Clear" button is located at the bottom center of the window. At the bottom left, the text "Listening for connections on port 8070 ..." is visible.

```
From localhost:8070
POST /soap/servlet/rpcrouter HTTP/1.0
Host: localhost:8070
Content-Type: text/xml; charset=utf-8
Content-Length: 450
SOAPAction: ""

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<ns1:getGreeting xmlns:ns1="urn:GreetingService" SOAP-ENV:encoding="base64">
<who xsi:type="xsd:string">Bob</who>
</ns1:getGreeting>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

From localhost:8080
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 515
Set-Cookie2: JSESSIONID=kzwzkrzvg2;Version=1;Discard;Path=/
Set-Cookie: JSESSIONID=kzwzkrzvg2;Path=/soap
Servlet-Engine: Tomcat Web Server/3.2.1 (JSP 1.1; Servlet 2.2;

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<ns1:getGreetingResponse xmlns:ns1="urn:GreetingService" SOAP-ENV:encoding="base64">
<return xsi:type="xsd:string">[Mon Apr 23 10:54:33 GMT+10:00]
</ns1:getGreetingResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Clear

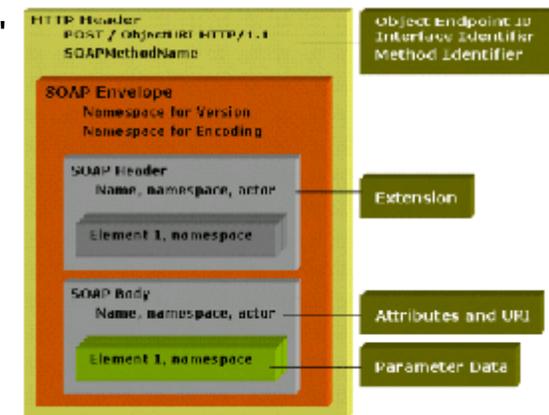
Listening for connections on port 8070 ...
```

- SOAP message has 3 parts
 - Envelope: defines namespaces, etc.
 - Header: optional for carrying auxilliary information (authentication, transaction, payment, etc.)
 - Body: payload data
 - Uses XML Schema data types for tagging

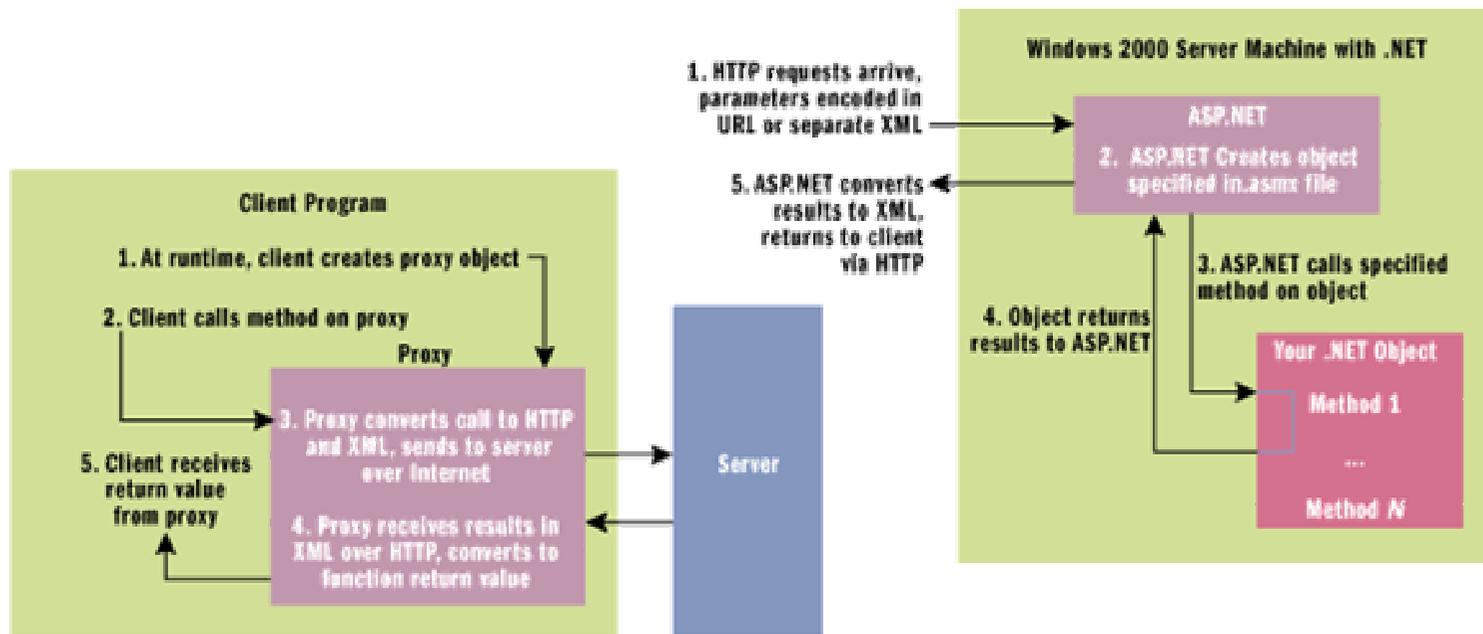
```
POST /EventManager HTTP/1.1
Host: www.techmetrix.com
Content-Type: text/xml;
charset="utf-8"
```

```
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="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="www.techmetrix.com/Event">
      <Enterprise>SQLI</Enterprise>
      <Address>Paris</Address>
    </m:NewCustomer>
  </SOAP:Body>
</SOAP: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
```

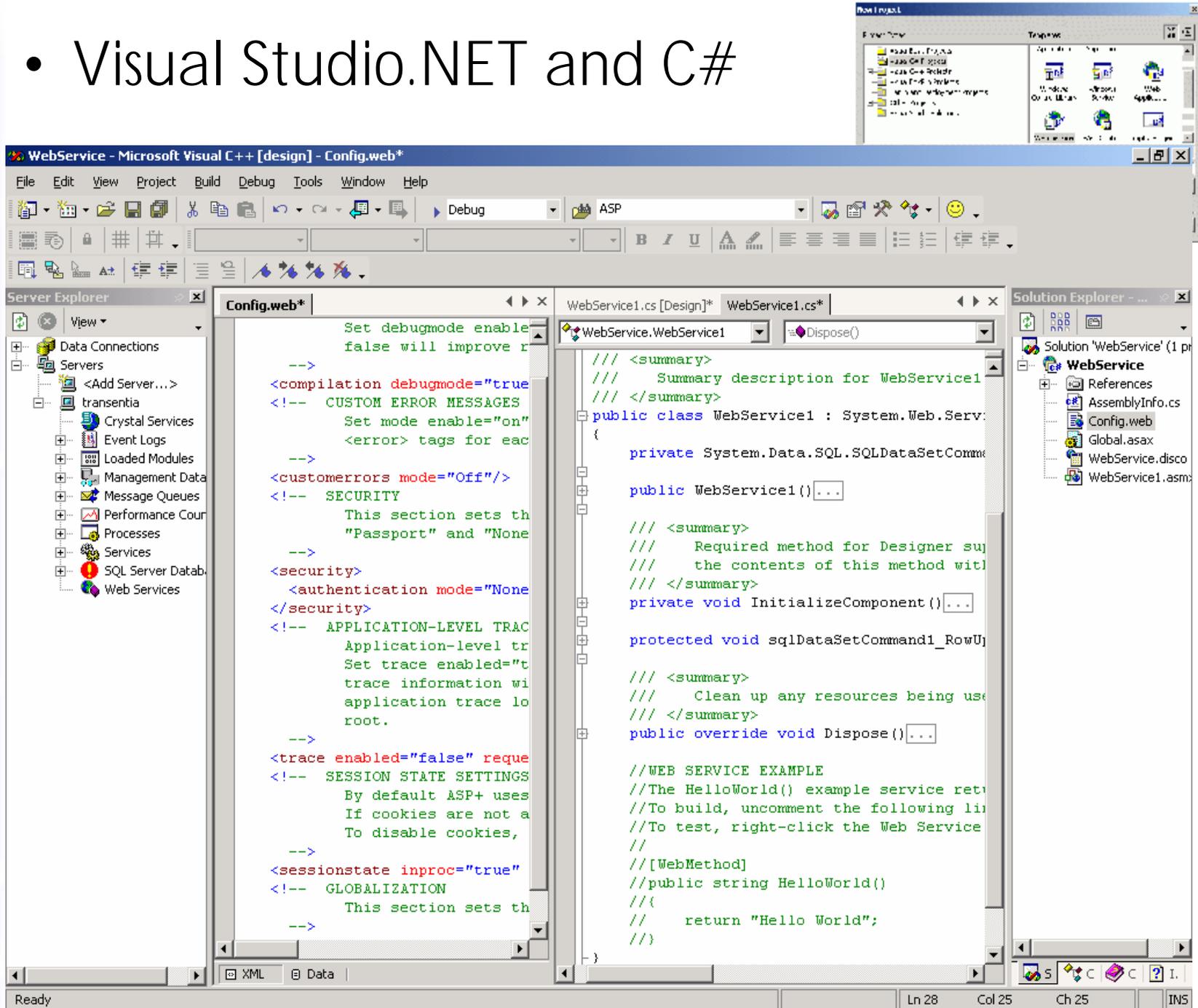
```
Serializer.Init Connector.InputStream  
Serializer.startEnvelope  
Serializer.startBody  
Serializer.startElement "pingHost", "urn:xmethodsSoapPing", , "namesp01"  
Serializer.startElement "hostname"  
Serializer.writeString CStr("www.yahoo.com")  
Serializer.endElement  
Serializer.endElement  
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
 - ASP example
 - Shows how to hand-hack interactions
 - Not necessarily the best way, but sometimes...

```
<%  
Set objReq = Server.CreateObject("Microsoft.XMLDOM")  
  
'Load the request into XML DOM  
objReq.Load Request  
  
'Query the DOM for the input parameter  
strQuery = "SOAP:Envelope/SOAP:Body/m:GetSalesTax/SalesTotal"  
varSalesTotal = objReq.SelectSingleNode(strQuery).Text  
  
'Calculate the sales tax  
varSalesTax = varSalesTotal * 0.04  
  
'Prepare the return envelope  
strTmp = _  
"<SOAP:Envelope xmlns:SOAP=""urn:schemas-xmlsoap-org:soap.v1"">" & _  
"<SOAP:Header></SOAP:Header>" & _  
"<SOAP:Body>" & _  
"<m:GetSalesTaxResponse xmlns:m=""urn:myserver/soap:TaxCalc"">" & _  
"<SalesTax>" & varSalesTax & "</SalesTax>" & _  
"</m:GetSalesTaxResponse>" & _  
"</SOAP:Body>" & _  
"</SOAP:Envelope>"  
  
'Write the return envelope  
Response.Write strTmp  
%>
```

- 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>"
%>

```

Issue: Interoperability

- Initially not so good
 - Getting better now
 - ‘Interopathons’ are now being discussed
 - TechMetrix’s matrix:

SOAP Client – TechMetrix Platform		SOAP Servers: Implementation (hosting location)		
		Apache SOAP (XMethods)	SOAP::Lite (XMethods)	Microsoft SOAP Server Toolkit 2.0 (ITFinity)
Apache SOAP	Medium API	Service: [Traffic Info California] => OK	Service: [BabelFish translation] => OK	#Datatype problem
SOAP::Lite	High API	Service: [Traffic Info California] => OK	Service: [BabelFish translation] => OK	Service: [Generation of a Globally Unique Identifier] => OK
	Medium API	Service: [Temperature of USA] => OK	Service: [BabelFish translation] => OK	Service: [Generation of a Globally Unique Identifier] => OK
MS SOAPToolkit2.0 Beta1	High API	#Datatype problem	Service: [BabelFish translation] => OK	Service: [Generation of a Globally Unique Identifier] => OK
	Low API	#Datatype problem	Service: [BabelFish translation] => OK	Service: [Generation of a Globally Unique Identifier] => OK

- 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
 - ShortDeserialiser, 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 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.”*

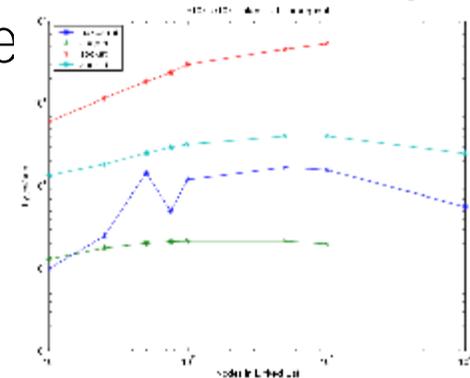
- *“New verbs have been introduced in an HTTP protocol extension proposal. The verbs, such as M-POST, are designed to simplify administration of firewalls and proxies.”*
- *Several commentators have expressed unease at SOAP: they worry about permitting arbitrary communication across a firewall*
 - *First we close all ports except 80, then we allow arbitrary objects to pass through port 80...*
 - *Note, however, that we are not passing the actual executable code. This may help us salvage our security...text/xml is easier to grok than binary executables*

```
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
```

- Experiments at Indiana University
 - 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
`<double> 3.141592653589793E+000 </double>`
 - » If Unicode is used, this doubles!
 - IBM estimates an *average* message size of 60K!
 - Speed

“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”

- Serializing Java objects into SOAP-encoded XML data takes approximately ten times more memory than the binary representation....Serialization speeds...are approximately 100 times slower and their throughputs are also a 100 times lower



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...
 - Hmmm...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 **J**

- 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 **J**!
- Issues remain
 - Security; performance; interoperability
 - Will be sorted eventually

- Working Draft 9 July 2001
 - reflect changes in XML Schema/datatypes specs.
 - support for XML Namespaces
 - better support for arrays
 - *“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.”*

- *“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

- 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>
- 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/2000/04fa100/kb0004/kb0004.asp>
- XML and SOAP—Essential to .NET
 - http://www.dotnettoday.com/articles/art003_essentialdotnet.asp
- SOAP Weblog
 - <http://soap.weblogs.com/>