IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, March 2021
Impact Factor: 5.731

Comparing of Soap and Distributed Object Technologies: A Case Study

Dr. Manish L Jivtode
Assistant Professor, Department of Computer Science
Janata Mahavidyalaya, Chandrapur, Maharashtra, India

Abstract: *Web service technology has emerged as a popular way for building distributed applications involving distributed databases. It is the next generation technology in the long journey from functions to objects to components to services. Today's comparing SOAP as a wire protocol to the commonly used distributed object technologies and their wire protocols in use. SOAP makes use of openly available technologies that, when combined, specify a wire protocol. This protocol can be used to facilitate highly and ultra-distributed architecture. SOAP commonly uses the HTTP protocol to transport XML-encoded serialized method argument data from system to system. This serialized argument data is used on the remote end to execute the client's method call on that system, rather than the client's local system. This case study provides a more details comparison of the SOAP and Distributed objects.*

Keywords: SOAP, XML, HTTP, RPC, XMOP

I. Introduction

SOAP (Simple Object Access Protocol) is a message protocol that allows distributed elements of an application to communicate. SOAP can be carried over a variety of lower-level protocols, including the web-related Hypertext Transfer Protocol (HTTP). SOAP defines a header structure that identifies the actions that various SOAP nodes are expected to take on the message. The textual information is encoded in an XML format. It has specific rules for encoding and processing. The actual transmission of the XML data is managed by the transport protocol that is HTTP. The combination of the open XML encoding style and the pervasive HTTP protocol makes SOAP possibly the most interoperable wire protocol.

II. SOAP And Distributed Objects Technologies

SOAP is compatible with most of the aspects of distributed computing, but the implementation of some of these aspects is outside the scope of a wire protocol.

There are several considerations that go into selecting a distributed architecture. Some of the important ones are scalability, performance, state management, garbage collection and security. Table shows how SOAP compares with the common distributed architectures based on these criteria.

	CORBA	DCOM	JAVA-RMI	SOAP
Protocol Name	General Inter-ORB Protocol (GIOP)	Object Remote Procedure Call (ORPC)	JRMP	Any transport protocol.
Scalability	Corba uses stateful programming model which is not as scalable.	Least scalable. Clients ping the server at regular intervals to ascertain that it is still available. This pinging process limits scalability when large # of connections are involved.	Relatively scalable. Uses RMI Registry which could limit scalability if it is located on one server.	Most scalable of the four.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, March 2021
Impact Factor: 5.731

Performance	Once an object reference is obtained, CORBA permits direct client-server communication. Hence subsequent communication is very fast.	Requires several round-trips to activate and use the remote object. Once object's reference is obtained, direct object access without DCOM can take place from client.	Good performance. Works for Java language only and hence is fine-tuned for it.	Currently low. Overhead of extracting SOAP envelope, parsing XML, creating appropriate objects and converting parameters.
State Mgmt.	Connection - oriented and stateful.	Provides location transparency. Is stateful.	Very flexible. Provides both stateful and stateless sub-protocols.	Not addressed by SOAP. If HTTP is the protocol used, it is stateless.
Garbage collection	CORBA does not address distributed memory manage-ment Vendor-specific implementations exist.	Provides automatic garbage collection using the pinging mechanism	Excellent garbage collection	SOAP does not address garbage collection.
Security	No intrinsic support for authentication, authorization or identity.	Very security-oriented. Provides support for authentication, authorization or identity. User can set appropriate level of security.	Since Java RMI works with java programming language it inherits the security built into Java. Use of RMI Security Manager can enable dynamic class loading thus providing additional security.	Since SOAP is a wire protocol, it does not address security. Security is determined by the transport protocol that it uses. For example, HTTPS using secured socket layer (SSL) when HTTP is the transport protocol.

III. XML The Block Of SOAP

SOAP is the technology that is taking XML 'out-of- process'. SOAP is not the only distributed technology that uses XML as its foundation. Several other protocols for executing Remote Procedure Calls (RPC) using XML already exist. SOAP outshines the other protocols because it provides a standard way of executing RPC using XML. Some of the other XML approaches are as follows –

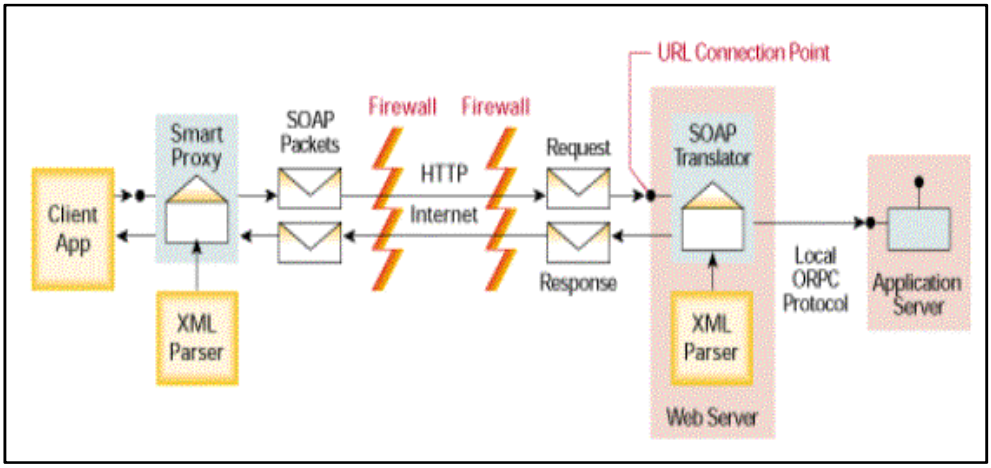
1. **XML-RPC:** This protocol serializes RPC requests and responses into XML documents, which are transmitted across a HTTP connection between the client and the server. SOAP originally started out as XML-RPC before additional features were added to it. SOAP is far richer than XML-RPC as it offers support for namespaces, longer element names, XML Schemas, enumerations, arrays, and custom types.
2. **XMOP:** XML Metadata Object Persistence (XMOP) is a protocol that allows interoperation between various object technologies such as Java on different platforms, Microsoft COM and CORBA. To use XMOP developers write XML based serialization for their classes. This is achieved by implementing the s**erializable** interface in Java and IPersist interface in Microsoft COM/DCOM. Using these XML files objects can be marshalled between different object systems such as COM/DCOM, CORBA and Java on various platforms.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, March 2021
Impact Factor: 5.731

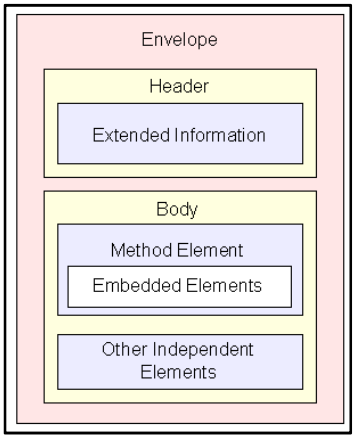
3. **EbXML:** Electronic Business XML (ebXML) is a more ambitious standard for sharing information between businesses. It uses messaging containing XML documents to transfer data between business systems. It is much larger in scope than SOAP because it encompasses the following features - Envelope and headers for messages/documents, Reliable messaging and error handling, Message routing Security, Audit trails, Quality of service, Platform independence, Restart and recovery.

IV. SOAP ARCHITECTURE

Figure shows the architecture of a generic system built using SOAP. This system uses HTTP protocol to pass the SOAP message between the client and the server. The client application calls a client-side proxy object using its native RPC protocol (such as COM for Microsoft platform and CORBA for UNIX). The proxy object uses an XML parser to convert the call into a SOAP packet. This SOAP packet is then transmitted over the Internet/Intranet to the web server using the HTTP protocol. The Web server handles the URL connection point of the remote service, and launches a SOAP translator which may be an ASP page, an ISAPI extension, a CGI program, a Perl script, etc. This translator uses a local XML parser to parse out the object name, method name and parameter values from the SOAP package. It uses these values to call the particular method of the server object by the local ORPC protocol, and packages the results into a response SOAP packet. This response is unpackaged by the proxy and presented to the client.



SOAP requires calls to use proper XML syntax. SOAP recommends that namespaces be used because they provide a mechanism to scope elements and attributes to various contexts. The namespace 'urn:schemas-xmlsoap-org:soap.v1' is the proposed namespace value for SOAP. Within a SOAP payload it uses the id/href attribute pairs to distinguish between unique entities. This provides the ability to multi-reference elements in the request and response payloads, such that an element that is serialized may be referenced as many times as necessary.



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, March 2021
Impact Factor: 5.731

Figure - shows the SOAP payload structure. The *Envelope* is the first element in a SOAP message. It encapsulates the various parts of the message and identifies it as a SOAP message. The envelope is followed by the optional Header element. It contains extended information about the message such as authorization or transaction information.

Next comes the Body element which contains the application-specific data. In the listing 1 below the SOAP Body represents a remote procedure call to CheckAccount. Encoding rules describe the syntax used to encode data in the body.

The SOAP architecture matches very well with the HTTP request-response model. SOAP requests are transported in a HTTP POST or HTTP M-POST and the SOAP response is sent back in HTTP response.

The Content-Type of the SOAP message is set to "text/xml". An additional HTTP header SOAPAction helps identify incoming SOAP requests to firewalls. The child element of the Body represents a method call, and the child elements of the call are parameters. The convention is to use the method name for the request payload and method name + "Response" for the response (ex. Add &AddResponse). For the response, the specification states that the return value must appear as the first child element of the payload, and that it's name is not important (you know it by its position, not its name).

Listing 1 and 2 represent an example of the request and response messages respectively generated by a SOAP call.

```
POST /PartServer.pl HTTP/1.1
Host: www.mcp.com
Accept: text/*
Content-type: text/xml
Content-length: nnnn
SOAPAction: the-method-uri#FindPart

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<m:FindPartxmlns:m="the-method-uri">
<PartNo>12345</PartNo>
<GroupID>7</GroupID>
</m:FindPart>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 1 Request message generated by a SOAP call

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 31 Jan 2001 07:21:19 GMT
MessageType: CallResponse
Content-Length: nnn
Content-Type: text/xml
Expires: Wed, 31 Jan 2001 07:21:20 GMT
Cache-control: private

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

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, March 2021
Impact Factor: 5.731

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
< m:FindPartResponse xmlns:m="the-method-uri">
<PartName>Fan Belt #18</PartName>
</vb:GetSecretIdentityResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 2 Response message generated by a SOAP call

V. CONCLUSION

In this paper, SOAP is to specify a wire protocol that can be used to facilitate highly and ultra-distributed architectures. SOAP specifies a very lightweight protocol form an administrative and use perspective. And finally, when comparing SOAP as a wire protocol, SOAP fares well in most areas. It might take more time to process the SOAP packet than other protocols, because the XML information must be parsed. SOAP protocol is a very scalable and relatively efficient. It is flexible for growth because of XML properties.

REFERENCES

[1]. Wagh, K. "A Comparative study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host", Journal of Information Engineering and Applications, ISSN 2225- 0506 (Online), 2(5), 2012.

[2]. Mironela, P., "The Importance of Web Services using the RPC and REST Architecture", IEEE, International Conference on Computer Technology and Development, 2009.

[3]. Halili, F., &Dika, A. (2012). "Choreography of Web Services and Estimation of Execution Plan", IEEE International Conference of Information Technology and e-Services (ICITeS), pp.168-174, Sousse, Tunis, March 2012. ISBN: 978-1-4673-1166-3, IEEE CN: CFP1245SART, 2012

[4]. Adamopoulos, V. T., "The Effectiveness of Promotional Events in Social Media.", Workshop on Information Systems and Economic, 2014

[5]. Mumbaikar, S., Padiya, P., "Web Services Based On SOAP and REST Principles", International Journal of Scientific and Research Publications, 3(5), ISSN 2250-3153, 2013.NCES

[6]. "From SOAP/1.1 to SOAP Version 1.2 in 9 points", https://www.w3.org/2003/06/soap11-soap12.html

[7]. Halili, F., &Kasa, M., "Analysis and Comparison of Web Services Architectural Styles, and Business Benefits of their Use", Proc. Book of International Conference of Information Technologies and their importance in the economic development, 701-712, Tirana. ISBN: 978- 99956-59-13-4, 2011.

[8]. Kumari, V., "Web Services Protocol: SOAP vs REST", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2015.

[9]. Wagh, K., &Thool, R. (2015). A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host. Marathwada MitraMandals Institute of Technology Pune, India - Shri Guru GobindSinghji Institute of Engineering & Technology, Nanded, India.