



A Proposed Strategy for Vendors of Web Service Providers

N Suresh Rao

Computer Centre

Computer Science and IT Block

University of Jammu

Jammu. INDIA.

Abstract- *This paper investigates web services tools and their related technologies available. Then the paper focuses on the choices available in web services and which one, will be the best choice out of the available choices. This paper starts with a bird's eye view on the technologies that make Web Services work. Then we discuss the Sun Java Systems we then discuss the sun java system application server v 9.1 based on GlassFish v2, to be the best. As vendors are available who provide other features apart from web services support that are more robust and better than those provided by GlassFish and may become market leaders in future. In this paper, finally strategies were proposed for vendors that could be incorporated so as to be more productive and efficient with related to web service computing tools.*

Keyword- SOAP , WSDL, XML, WCF, WSIT

I. INTRODUCTION

Web can thought as new name for distributed computing. Web services allow businesses class to interoperate among themselves. Today various different vendors are available in the market competing for perfection. But business has two choices for implementing their enterprise solutions. The choices are .NET or JavaEE. This paper focuses on the initiatives such as interoperability among their applications. As an example the effort taken by Sun Microsystems towards WSIT, is a part or the project named Tango. Developers working in this project did indeed spend And developers working in this project did indeed spend much time working with their Microsoft counterparts at Red-Mont where Web Services are developed under WCF (Microsoft Communications Foundation), also called by the name Project Indigo. As of today MS provide support for Web Services [1], [3], [5] created by Sun and vice-versa. Of course these supports are yet to be extended in the actual business arena. There are many vendors in the market having largest shares with respect to implementation of the Java EE. One reviews the new entry to the market such as GlassFish t sees its seat in future. WSIT [6] which also called as the Metro Web Services, stack implements many of the WS-* specifications that are required for interoperability with MS applications. To review this stack a comparison was made with some application servers such as Axis, JBoss etc. Finally, certain techniques and methods are proposed to find a solution for universal interoperability.

II. WEB SERVICES

Unlike traditional point -to-point architectures, SOAs comprise loosely coupled, highly interoperable services. These services interoperate based on a formal definition (or contract) which is independent from the underlying platform and programming language (e.g., WSDL). The interface definition encapsulates (hides) the vendor and language-specific implementation. A SOA is independent of development technology (such as Java and .NET). The software components become very reusable because the interface is standards-compliant and is independent from the underlying implementation of the service logic. So, for example, a C# (C Sharp) service could be used by a Java application and vice versa. Web services [5] can be used to implement SOA, but service orientation does not necessitate the use of Web services protocols, nor does the use of Web services protocols ensure that the overall system is SOA. Services provide something of value to those who know how to request and consume them, without having to know how to produce that value. SOA is an approach to building software applications as collections of autonomous services that interact without regard to each other's platform, data structures, or internal algorithms. Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing capabilities that may be under the control of different ownership domains. To appreciate why SOA is currently more popular for many enterprises, due to the following reasons.

- Improved software reuse with minimal redundancy
- Improved portability of software on different architectures.
- Availability of better tool set and hence less time for on integration.
- SOA based systems can be better governed.
- Testing is more transparent.

- Error determination is easy
- Good opportunities for real time performance.

Many more reasons can be attribute for SOA.

III. ARCHITECTURE OF SOA

The SOA structure includes Web services such as:

- Message exchange
- Web service description
- Web service publication and discovery

The basic structure defines a machine to machine interaction for the exchange of messages between service requesters and providers. The Requester finds a note of services available and then invokes the required service. A requestor can be software agent or a web service i.e anything that can send a SOAP message. Therefore required execution of service takes place. Provider of web service must ensure a description of the service they provide. The various are the following components such as:

- **XML** - All data to be exchanged is formatted with XML tags.
- **HTTP** - Main transport protocol for web services, but they can go across any number data such as FTP, SMTP etc.
- **WSDL** - Stands for Web Services Description Language and is a way to describe a web services, it includes data type and structure information, identifies the MEP (message exchange pattern), and contains the address of the service provider. Examples of MEPs include one-way, request/response, publish/ subscribe, and broadcast.
- **UDDI** - The web service information is published using a UDDI registry. It should enable applications to look up web services information.
- **SOAP** - The protocol for formatting XML documents to be sent over the wire (normally using HTTP). SOAP is a protocol that supports one or more intermediaries in a message path between requester and provider. Intermediaries may perform certain functions associated with the message such as routing, security, management, or other operations.

A. Leading Web Services

- **W3C** - Standards organization working on the standards such as WSDL [10], SOAP [8], [9].
- **OASIS** - A standard organization working on many of the WS-* standards WS Interoperability Organization (WS-I) - provides clarifications and constraints on those de facto standards that will enable vendors to implement so they interoperate.
- **IETF** - Working on some of the lower level security and transport standards.
- **Vendors** - Microsoft are moving ahead with .Net, J2EE community with many web services packages, and just about every other software vendor has some sort of Web services support.
- **Users** -Ultimately leading SOA Web services by buying software's with more features.

IV. CHOICES AVAILABLE

Today enterprise applications are built either using .NET or JavaEE. And the IDE with respect to Microsoft is Visual Studio .NET. But there are numerous vendors implementing the JavaEE specifications. One can choose from IBM's Web Sphere, BEA's Web logic, Oracle 10G Application Server, Sybase Enterprises Application Server and many more. Last but not the least the GlassFish Application Server V2. This paper focuses on GlassFish as a competitor among the major players in the market. IBM and Oracle vie primarily for the lion's share of spending on application server platforms by large enterprises. Microsoft dominates spending by medium enterprises but also has an expanding position in large enterprises. Many customers select the largest vendors primarily because of a long-standing relationship or perceived safety as a supplier, as opposed to product features. And as is evident in the Forrester Wave™ data, IBM, Microsoft, and Oracle each have huge numbers of customers and have built the largest partner networks in this market. Fig. 1 depicts this fact from Forrester survey. And clearly Sun is a very strong competitor.

V. GLASSFISH TECHNOLOGY

Well, this is the Application Server that is an open-source and that implements most of the recent WS-* specifications which even the other commercial major players are at their early stages. What more, GlassFish [4] has WSIT implemented which provides interoperability with its Microsoft counterpart through the WCF (Windows Communications Foundation). The best part is the SPECj –Applications Server results are much better. The other thing is as of now; the SPECj results have not included Web Service performance into consideration. Even without this, the results say a 10 % performance boost over Web logic and a 30% boost over IBM's Web sphere v 6.1. And according to Gartner Report, September, 2007, Sun and the GlassFish open-source community have delivered the first production-scale open-source Java EE 5 application server. This challenges the dominance of market leaders like Red Hat and IBM. SJSAS v 9.1 is based on GlassFish in the sense, it is the

binary distribution from sun after accepting the licensing agreement, whereas GlassFish is the open-source implementation based on SJSAS v 9.1 that is delivered as a jar file download and is said to be used at “owner’s risk”.

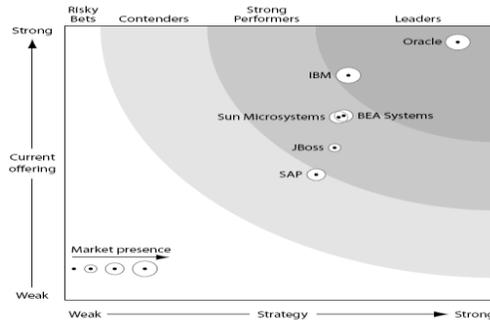


Fig. 1 Forrester Wave™: Application Server Platforms Based On Java/J2EE, Java EE, Q3 '07
VI. CONSTRAINTS AND SUGGESTIONS

Although the choices are many, there are still unsolved problems that need to be addressed with respect to implementation of Web Services in them. For one thing all these vendors are not compatible in implementing all the current WS-* specifications. So, there had been only few real life demonstrations on the much talked about subject. And since WSIT is the leader that implements the most needed specifications, let us first make a comparative study of some the Applications servers that comply with the Metro Web Services stack. Table I shows this. Most of the required WS-* specifications are implemented by GlassFish that are required for communication with WCF and vice versa. Following strategies could be implemented for the incompatibilities:

- Up gradation: The older consumer Application Server might be upgraded/migrated to the one that will be compatible with the provider Application Server. The limitation is it may not solve the problem completely.
- Use a Gateway proxy/ Data Power: A proxy that converts older Web service to the one required at the provider might do the job. This may work to an extent but may not scale well.
- Use an EJB proxy at the provider: This might be a more probable solution and may scale well, but will encounter problems when the component is basically based on SCA.
- Each AS to support WSIT plug-in. This might be the best solution as Tango is doing its job with implementing the required specifications. So, other application Servers should have an option to support the WSIT as a plug-in and the jar files loaded by the respective class loaders correctly at runtime. Tomcat already has this support.

Net beans 6.0 are the IDE helps in creating Web Services based business applications. This frees the developer for the deep learning. Also, the reliability, authenticity, security, integrity concepts required for a Web Service (through WSIT) [6] can be specified declaratively.

Table I :Stack Comparison based on WS-*

Feature	Axis 1 x	Axis2	Celtix	Glue	JBossWS	XFire	Metro@GlassFish	OracleAS 10g
WS-Addressing	X	X	X	X	X	X	X	with BPEL
WS-Atomic Transaction	X	X					X	
WS-Business Activity		X						
WS-Coordination	X	X					X	
WS-Eventing		X			X			
WS-Metadata Exchange		X [10]					X	
WS-Notification	X	X [12]		?		?		
WS-ReliableMessaging	X	X	X				X	
WS-Policy		X					X	X
WS-Secure Conversation		X					X	
WS-Security Policy		X					X	
WS-Security	X	X		X	X	X	X	X
WS-Trust		X					X	
WS-Transfer		X						
WSDL 1.1 Support	X	X	X	X	X	X	X	X
WSDL 2.0 Support		X						

VII. CONCLUSIONS AND DISCUSSIONS

Web Services and their implementations are starting to appear and certainly business will be much more productive than earlier with this growing revolution. Vendors have to make sure that they implement the WS-* specifications in a standard way so that interoperability is actually achieved. Alternatively vendors implementing the JavaEE specs may have a plug-in support for WSIT, as WSIT implements the WS-* a spec required and is a part of GlassFish.

REFERENCES

- [1] Bosworth Adam. "Developing Web Services", IEEE, 2001
- [2] Elfwing, R., Paulsson, U., and Lundberg, L. Performance of SOAP in Web Service Environment compared to CORBA, In Proceedings of the Ninth Asia- Pacific Software Engineering Conference, IEEE, 2002
- [3] Eric Armstrong et al. "The Java Web Services Tutorial", http://java.sun.com/web_services/downloads/web_services_tutorial.html
- [4] Paul Krill. "GlassFish app server to swim in the enterprise", http://www.javaworld.com/channel_content/jwdevelopment-index.html
- [5] CERAMI, E. Web Services Essentials. London: O'Reilly Media, 2002.
- [6] OASIS. Web Service Security. OASIS. [S.l.].2004.
- [7] J. Kangasharju, S Tarkoma, .; K. Raatikainen, Comparing SOAP performance for various encodings, protocols, and connections. Personal Wireless Communications. Venice: Springer-Verlag., p. 397-406. 2003.
- [8] Chiu, K. Investigating the Limits of SOAP Performance for Scientific Computing. Edinburgh: IEEE, 2006.
- [9] Garcia, J.; Blanco, R.; Blanco, A. A comparative performance evaluation of different implementations of the SOAP protocol. Coruna: University of Coruna, 109-118 p,2007
- [10] W3C. Extensible Markup Language. W3C. Available at:<http://www.w3.org/XML/>>. Accessed on: 23 Jun 2011.