

WEB SERVICES: A PARADIGM SHIFT IN WEB DEVELOPMENT PROCESS

Er. Ashish Kharvar¹, Darshan R. Chauhan²

Assistant Prof., Information Technology Department, SCET, Surat, Gujarat, India

Assistant Prof., Computer Department, SNPITRC, Bardoli, Gujarat, India

ABSTRACT: Web services can be seen as a revolutionary change in the Web Application Development process. Web services attempt to solve problems, such as service coordination and service composition that distributed information systems have addressed in various ways (but not always successfully) in the past. This paper will look at the web services from the technical perspective. In addition, the paper will discuss the differences between the traditional Web-based applications and the one which uses Web services. Web Services will make computing resources, both hardware and software, accessible to you through the Internet just like electricity is made available to you. Web Services will do for computing what the Internet did for data. They would encourage a pay-per-usage model and make dynamic collaborations possible. Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across the network.

1. INTRODUCTION

Most of us are familiar with accessing the Web through software called Web browser, which provides a human-oriented interface to information and user-oriented services such as on-line shopping and online library system. When a Web page is requested by a user, the request is processed by a remote Web server, which returns the information in hypertext markup language (HTML)—a form that allows the browser to present it using a selection of fonts, colors and pictures, all factors that make it more useful and appealing to a human. “Web services can be defined as a loosely coupled distributed software components that provide information to applications rather than to humans, through an application-oriented interface”. The information is structured using eXtensible Markup Language (XML), so that it can be parsed and processed easily rather than being formatted for display. In a Web-based retail operation, for example, Web services that may be running on widely separated servers might provide account management, inventory control, shopping cart and Credit card authorization services, all of which may be invoked

multiple times in the course of a single purchase. Some of the early products in Web Services started appearing in 1997 when Sun announced its Jini platform and Hewlett-Packard its e-speak. After that, many big players such as IBM and Microsoft joined this race. The Web Services arena picked up steam after the big players roped in and several small players also joined hands for what was perceived as the next Internet wave. Server- standard body consortiums were formed, which developed numerous standards on different aspects of Web Services. Some of the key standard bodies consortiums are: W3C, Oasis, JCP, OMG, and several individual efforts by a group of companies.

2. WEB SERVICE INFRASTRUCTURE

The Symon's Extensible Markup Language (XML) Page provides a very nice formula that neatly defines the major components of Web Services: “Web services = XML + SOAP + WSDL + UDDI”

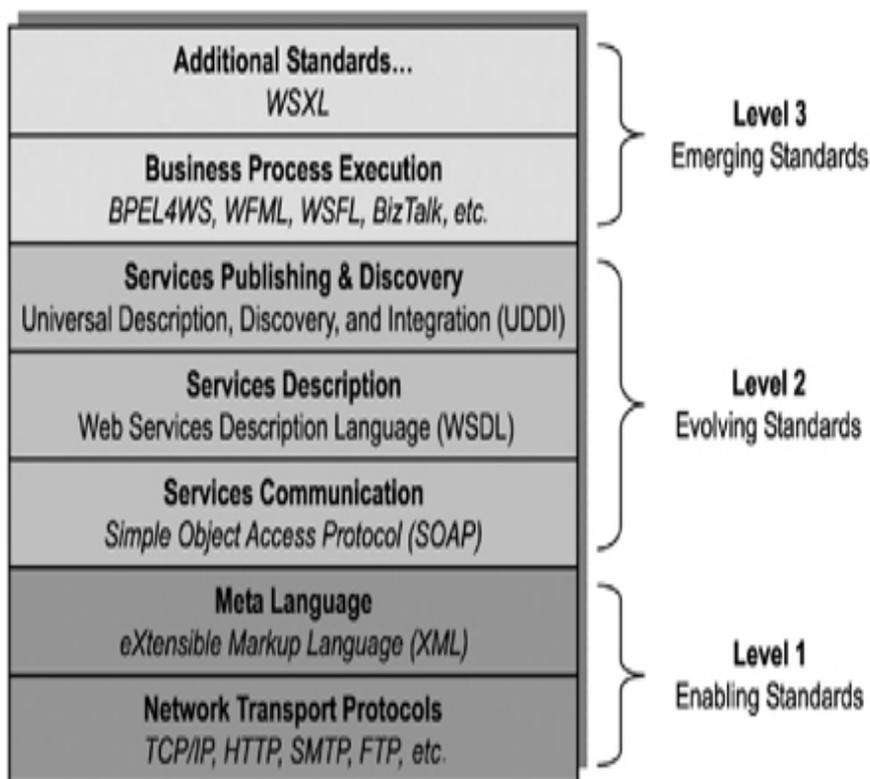


Figure 1 : Web Service Stack(WSS)

3. WORKING OF WEB SERVICES

As can be seen from Figure 1, the key components used by Web services are:

XML (eXtensible Markup Language)—a markup language for formatting and exchanging structured data. **SOAP** (originally Simple Object Access Protocol, but technically no longer an acronym)—an XML-based protocol for specifying envelope information, contents and processing information for a message.

WSDL (Web Services Description Language)—an XML-based language used to describe the attributes, interfaces and other properties of a Web service. A WSDL document can be read by a potential client to learn about the service.

UDDI Universal Discovery Description and Integration is the yellow pages of Web services. UDDI is a standard sponsored by OASIS (Organization for the Advancement of Structured Information Standards). Often described as the yellow pages of Web services, UDDI is a specification for creating an XML-based registry that lists information about businesses and the WebServices they offer. UDDI provides businesses a uniform way of listing their services and discovering Services offered by other organizations.

The Web Services model follows the publish (Advertise), find(Discover),and bind(interact) paradigm.

As shown in the Figure 2, in the first step, a service provider publishes a Web Service in a Web Service registry. Secondly, a client (Service Requestor) who is looking for a service to meet their requirement searches in a registry. After successfully finding multiple matches, it chooses a service. The client then chooses a service based on its preferences. The client then downloads the service description and binds with that to invoke and use the service.

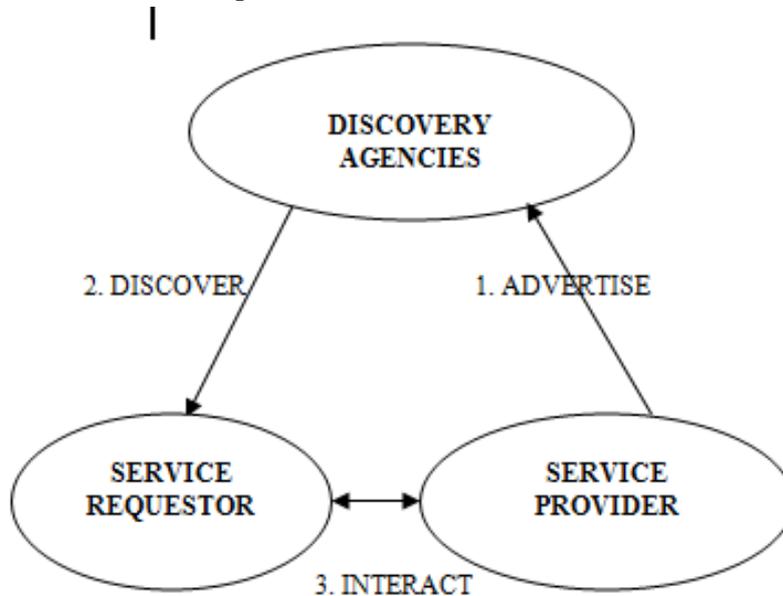


Figure 2 :Service Interaction Scenario in SOA

Here's how a Web service works. A client intending to invoke a function formats a request with SOAP XML encoding, and sends it across the server over any mutually agreeable communication protocol (typically HTTP, but SMTP is possible). The server runs some sort of a listener that accepts the incoming SOAP calls, reads the information from the XML SOAP packets, and maps them to business logic processing "application" software on the server. This application layer on the server processes the request and returns output to the listener, which formats the output into a response packet in the SOAP XML encoding and

returns it to the client. A separate XML file contains a description of the services provided by the server in an encoding format called SDL (Service Descriptor Language), analogous to a type library in COM or an IDL file in CORBA.

Three of the key problems solved by Web Services over earlier distributed systems such as CORBA, DCOM, RPC, and so forth were:

Interoperability: Earlier distributed systems suffered from interoperability issues because each vendor implemented its own on-wire format for distributed object messaging. By using XML as an on-wire standard, the two camps of Java/J2EE and .NET/C# now could speak to each other.

Firewall traversal: Collaboration across corporations was an issue because distributed systems such as CORBA and DCOM used non-standard ports. As a result, collaboration meant punching a hole in your

firewall, which was often unacceptable to IT. Hence, this did not allow any dynamic collaboration, as it required going through a manual process for collaborating with partners. Web Services use HTTP as

a transport protocol and most of the firewalls allow access through port 80 (for HTTP), leading to easier and dynamic collaboration. The dynamic nature of Web Services interaction offers several exciting services for the users.

Reusability: Web services provide not a component based model of application development, but the closest thing possible to zero-coding deployment of such services. This makes it easy to reuse web service

components as appropriate in other services. It also makes it easy to deploy legacy code as a web service.

4. ISSUES AND CHALLENGES AHEAD

Although the simplicity of Web services is an advantage in some respects, it can also be a hindrance. Web services use plain text protocols that use a fairly verbose method to identify data. This means that Web service

requests are larger than requests encoded with a binary protocol. The extra size is really only an issue over low-speed connections, or over extremely busy connections.

Web services are exposed to the public using http-based protocols. As Web services are publicly available, it must be implemented using authentication and authorization mechanisms and using SSL-enabling encryption of the messages for securing the usage.

Adopting open security standards like SAML, XML Encryption, XML Signature, or XACML may be a solution.

CONCLUSION

From our perspective, Web services are the most important element in the process of new generation web application development. Moreover, Web services are self-describing business functions that are available

from any Web-connected device making them globally accessible in more efficient manner than the traditional web applications which are not using the web service concept. Hence, Web services have enabled a whole new world of online interactivity by allowing anyone to create new and innovative applications on top of existing websites.

REFERENCES

Web Resources:

- [01] [http://h71028.www7.hp.com/ERC/downloads/SOA- Grid-HP-WhitePaper.pdf](http://h71028.www7.hp.com/ERC/downloads/SOA-Grid-HP-WhitePaper.pdf)
- [02] <http://home.earthlink.net/~symonchang/xml.htm>
- [03] <http://www.webopedia.com/TERM/W/WebServices.html>
- [04] <http://www.w3.org/DesignIssues/WebServices.html>
- [05] [http://social.msdn.microsoft.com/Forums/en-](http://social.msdn.microsoft.com/Forums/en-US/asmxandxml/thread/435f43a9-ee17-4700-8c9d-d9c3ba57b5ef)
- [06] [US/asmxandxml/thread/435f43a9-ee17-4700-8c9d- d9c3ba57b5ef](http://social.msdn.microsoft.com/Forums/en-US/asmxandxml/thread/435f43a9-ee17-4700-8c9d-d9c3ba57b5ef)
- [07] <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>

Books

- [01] XML Web Services in the Organization- Chris Boar - Microsoft Press