# Web Technology: Http, Web Server & Web Services

Minakshi & Happy Sharma
Department of Information Technology

*Dronacharya College of Engineering*, Khentawas, Farukh Nagar, *Gurgaon*, Haryana, India

minakshidhillon@yahoo.in & happysngh474@gmail.com

## Abstract

*Web services refer to a family of technologies that can universally standardize the communication of applications in order to connect systems, business partners, and customers cost-effectively through the World Wide Web. Major software vendors such as IBM, Microsoft, SAP, SUN, and Oracle are all embracing Web services standards and are releasing new products or tools that are Web services enabled. Web services will ease the constraints of time, cost, and space for discovering, negotiating, and conducting e-business transactions. As a result, Web services will change the way businesses design their applications as services, integrate with other business entities, manage business process workflows, and conduct e-business transactions. The early adopters of Web services are showing promising results such as greater development productivity gains and easier and faster integration with trading partners. However, there are many issues worth studying regarding Web services in the context of e-commerce. This special issue of the JECR aims to encourage awareness and discussion of important issues and applications of Web Services that are related to electronic commerce from the organizational, economics, and technical perspectives. Research opportunities of Web services and e-commerce area are fruitful and important for both academics and practitioners. We wish that this introductory article can shed some light for researchers and practitioners to better understand important issues and future trends of Web services and e-business.*

## 1. Introduction

The relationships among HTTP, Web Server and Web Services are a complicated set of functionalities and exchanges of information. Each component plays an important role in the thousands of functions users can access
and utilize on the Internet. HTTP allows users to interact with Web Servers and access information via the Internet. Web servers serve data and files to users who request them. Web Services allow cross-system, cross-language communication among various kinds of machines and enable inter-business transaction. Although each

technology works on its own and performs many useful functions, it is the combination of these technologies that has created the dynamic functionalities of the Web that are available today. This research paper will explore the inter-relationships between HTTP, Web Servers and Web Services technologies that have facilitated the functionalities and convenience of the Web.

## 2. HTTP

HTTP, or Hypertext Transfer Protocol, is the standard protocol currently used to access the Internet. According to the World Wide Web Consortium, HTTP "i s an application-level protocol for distributed, collaborative, hypermedia information systems." It is a very simple protocol that allows raw data to be transferred across the Internet. From this simple data transfer protocol, users of the Internet can easily perform functions and give commands to the Web Servers through a graphic user interface (GUI) the as a Web page viewed through a browser and not worry about the specific details of how the command is going to be transferred or interpreted by the computers involved. HTTP allows such exchange of information between the user's computer and the Web Server to take place rapidly and efficiently.

Just as HTTP allowed the World Wide Web and the Internet to become such a global phenomenon, the World Wide Web and the Internet also helped HTTP to become the global standard of data transfer protocol on the web. The World Wide Web global information initiative adopted HTTP as the basic data transfer protocol in 1990, allowing HTTP to develop into a global standard of data transfer protocol as the Internet expanded worldwide. However, just because HTTP is the current global standard does not mean that it is flawless. An example of HTTP's drawbacks would be the fact that HTTP is a stateless protocol, i.e. HTTP treats each command independently and does not string commands together. In another word, HTTP does not have any memories. HTTP does not remember past commands and will forget the current command as soon as it is executed. As such, it is very hard to make HTTP to be more interactive and dynamic without additional technologies such as JavaScript, cookies, PHP and other programming scripts. To put it simply, with pure HTTP users will not be able to customize their online shopping cart or CNN news home page.

Nevertheless, efforts have been made to improve HTTP. From the first HTTP 0.9, through HTTP 1.0, to the latest HTTP 1.1, numerous improvements have been made. Some of the improvements are:

"Faster response, by allowing multiple transactions to take place over a single persistent connection. Faster response and great bandwidth savings, by adding cache support.
Faster response for dynamically-generated pages, by supporting chunked encoding, which allows a response to be sent before its total length is known.
Efficient use of IP addresses, by allowing multiple domains to be served from a single IP address."

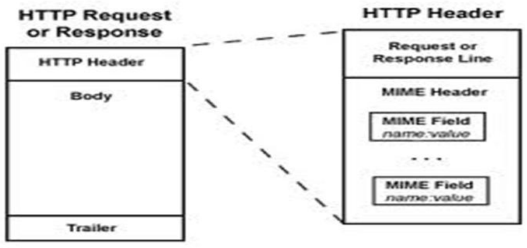

**Fig.2 HTTP Request or Response**

## 3. Web Server

A server is "a computer or device on a n etwork that manages network resources." There are m any different kinds of servers: the dial-up server that serves as a gateway for the user to access the rest of the Internet; the printer server that manages one or more printers connected to the network, allowing users to access various printers remotely; and Web Servers that stores web pages and other data and information that are provided to users upon request. A Web Server is the central nervous system of Web Site. It is the Web Server that hosts both the components of a Web page such as the actual Web page HTML files, CSS files and templates and all other essential technologies that make a Web site function the way it does. Although all Web servers function similarly, the set up and the way a server could be set can vary drastically.

There are two common ways to setting up Web Servers: P2P and Client-Server. P2P, or Peer-to-Peer, indicates a direct connection of individual computers to one another where each computer can specify what data it is willing to share. This kind of network is very easy and cheap to set up. Furthermore, the speed of file transfer in a P2P network is not constrained by the capability of any single server/computer. Since each computer in the network is capable of becoming a server on its own, a file could be shared and transferred from multiple servers at the same time, thus increasing the file transfer rate. However, since each computer in the network is a server, each computer on the network needs to be set up individually. The responsibility of managing the system lies in the hands of every single owner of every single computer that is connected to the network. Consequently, the management of a P2P network is extremely difficult. Due to the decentralized management of the network, servers with a P2P connection are susceptible to virus and worm attacks.

The Client-Server network, on the other hand, is a highly centralized network system with one central computer as the server. This set up is easy to manage and secure. Yet, maintaining a centralized network requires tremendous amount of resources ranging from manpower to hardware. As a result, the cost of a client-server network is very high. Another drawback of a client-server set up is that the speed of file transfer between the client and the server slows down when the number of clients accessing the server at a time is too high. Nevertheless, because of its ease to manage and good security, client-server network is still the dominant set up of Web Servers. Apache, a free server technology, is currently one of the most popular server technologies in use today. The first version of Apache, based on the NCSA http Web Server, was developed in 1995 by a "loosely-knit group of [about 20] programmers. " Apache provides full source code and an unrestrictive license. Apache users can easily change, adapt, or modify the software in accordance with the needs of their particular organization. Additional modules, either written by the user or downloaded free of charge from the vast Apache module library online, could easily be added to accommodate any specific needs of the user. Apache is also capable of performing many functions such as DBM database authentication, multiple Directory Index directives, unlimited flexible URL rewriting and aliasing, content negotiation and virtual hosts.

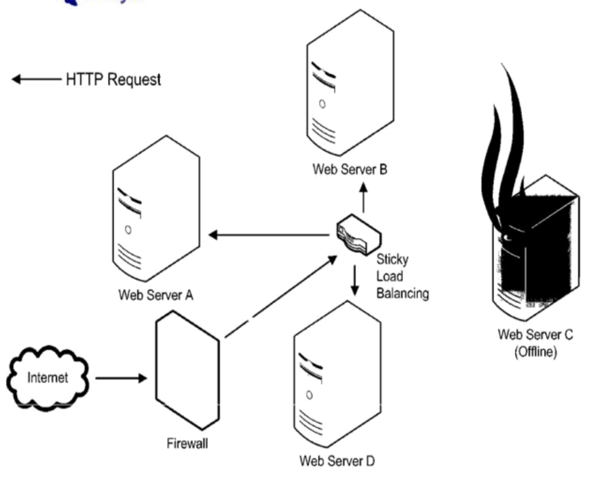

**Fig.3.1 Web Server**

## 4. Web Services

Web Service is a very powerful tool that has greatly enhanced the efficiency and communication among businesses. According to the World Wide Web Consortium, "a Web Service is a software system designed to support interoperable machine-to-machine interaction over a network." Alternatively, Zeldman defines Web Servi ces as a "reusable software components based on XML and rela ted protocols that enable near zero-cost interaction throughout the business ecosystem." In other words, Web Services is a software system that allows machines (including servers) to communicate with each other regardless of each individual machine's operating systems and programming languages. 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".

Extensible Markup Language (XML) is the universal markup language that all machines are capable of understanding. In the process of inter-machine communication via Web services, XML is used to tag the data involved. Web Services Description Language (WSDL), on the other hand, is being used for describing the services available. Then Universal Description, Discovery and Integration (UDDI) lists the services available from that particular machine. Lastly, Simple Object Access Protocol (SOAP) is used to transfer data for each exchange of information between machines and servers, which typically involve "HTTP with an XML serialization in conjunction with other Web-related standards." Cons equently, Web Services "are not tied to any one ope rating system or programming language." As a result, via W eb Services, Java based programs will be able to talk to servers running C++ based programs and a Windows machine will be able to communicate with a LINUX machine.

While serving a similar function as the Web, Web Services do have some significant differences. The most prominent difference between Web services and the Web is that instead of a user interface, Web Services functions via application interfaces. In other words, the machines communicate with each other application to application. Such exchanges limit possible user errors and thus increase the efficiency of the exchange.

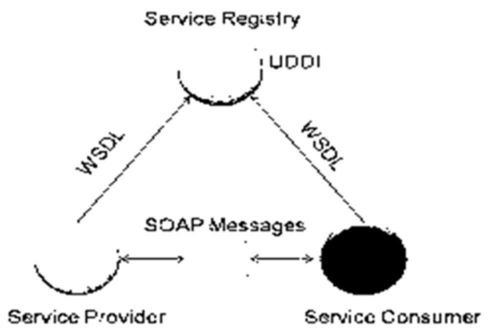

**Fig.4.1 Web Services**

## 5. Web Services Standard and Infrastructure

Web services rely on a set of standards to support interoperability among applications developed in different languages and running on different platforms or operating systems. One way to understand Web services is to understand Web services standards. Core Web services standards including SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery, and Integration), and other emerging ones will be discussed in this section.

### 5.1 Web Services Definitions

The basic idea of Web services is the use of SOAP messaging protocol to invoke software method in remote systems. This is often described by some technologists as Remote Procedure Calls (RPC) over the Internet protocols (e.g., HTTP). A SOAP message consists of an "Envelo pe", an optional "Header", and a mandatory "Body". The SOAP "Body" carries application-specific contents i ncluding the method name and the serialized values of the methods' input or output parameters (Scribner and Stiver, 2002). Parameters of a Web services method can be a simple value or a compound value (structure or array). Serializing a Web services message in (pure text) XML format allows the SOAP XML to pass through Internet firewall.

The Web services can be considered as a set of callable interfaces to software programs or components, regardless of their implementations. They can be invoked remotely via SOAP messaging. Therefore, these programs can provide services to other applications using Internet protocols. W3C's Web Services Architecture Working Group refers to the services provided by those programs as Web services. A detailed definition of Web services by the W3C Web Service Architecture Group stated that (Austin et al., 2002):

"A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols."

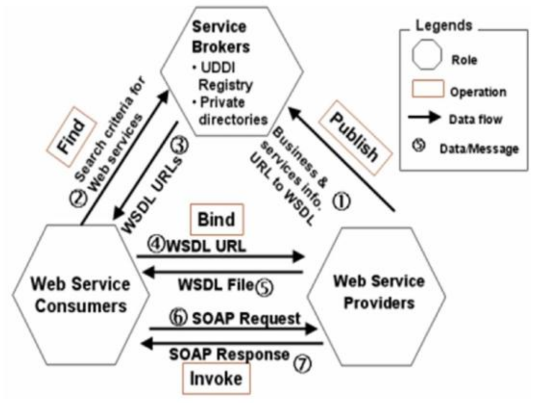

**Fig.5.1 Web Services Based Architecture**

## 6. How HTTP, Web Server and Web Services Work Together

The interaction among HTTP, Web Servers and Web Services is simple: HTTP is a simple protocol browsers use to communicate with Web Servers. Web Servers, on the other hand, fulfill users' requests and store the information users provide. Meanwhile, Web Services allow different Web Servers to communicate and interact with one another in order to process the request and/or commands of the user.

A good example of how the interconnectivity among the three technologies works would be a user trying to buy a plane ticket online. The user would access a travel agency's Web page to query for the availability of seats, date and time of the flight and prices of the plane ticket. In this querying process, HTTP acts as the language that users end up using to communicate with the Web server that actually can access the information of flight date, time, seat availability and prices from the airlines database. According to the values users input into the Web page (i.e. GUI) and transmitted to the Web server via HTTP, the Web server performs the command of search by sending out commands of this query to each individual airline's flight schedule databases using an application to application interface, i.e. Web services. Web services translate whichever markup language the Web server uses into the universally understood XML that gets relayed to the databases of all the airlines. When the XML is received by the airline databases, Web services then translates the XML into whatever programming language that each database is using so that the database would be able to understand the command the Web server sent out. After the query has been completed, the result would be transmitted back to the Web server through Web services again. Then the Web server would relay these search results to the user via HTTP which would present the information to the user through an HTML file that could be interpreted by a browser.

In sum, the simple function of querying for flight schedules and seats requires all three technologies, HTTP, Web server and Web services, to work together. Without any of these technologies, the query would fail or the scope of the search would be drastically limited.

## 7. Conclusions

The functionalities that HTTP, Web Servers and Web Services provide dramatically changed the way companies, as well as individuals, conduct business online. While each technology was created for one specific purpose, it is the combination of these technologies that has greatly enhanced the transfer of information online. The example of users purchasing plane tickets online shows how critically important a role each technology plays in one of the most common tasks users can accomplish on the Internet today. Without any one of these technologies, e-commerce would not have boomed and the convenience users enjoy would not have existed today.

## 8. Acknowledgments

The preparation of this paper has been greatly aided by the discussion that we have had with our friends and faculty members of our college.

## 9. References

[1]Marshall, J. (1997). HTTP made really easy: a practical guide to writing clients and servers. Retrieved October 13, 2013

[2]Webmonkey (1999). Introduction to Apache. Retrieved October 31, 2003

[3]Arkin, A., Business Process Modeling Language, BPMI.ORG, November 13, 2002