Design and Implementation of an e-Transcript System using Web services

Ramani Garikipati, Billy B. L. Lim
School of Information Technology
Illinois State University
Normal, IL 61790-5150, USA

Abstract- Web services technology is a burgeoning technology that has received tremendous amount of attention in the software industry in recent years under the broader umbrella of service-oriented architecture (SOA). It is fast becoming an important emerging technology that utilizes the power of the Web to produce an integrated, interoperable solutions based on standard Web protocols that are easy to consume in a heterogeneous network. The paper showcases various Web service features and how they can be used for a secure online transcript transmission system. It discusses how the latest state-of-the-art technology can be used to solve the antiquated problem of transcript processing faced by many universities and students around the world presently. The experience gained and the possible future enhancements of the work are also described.

I. INTRODUCTION

For many years, software reuse and systems interoperability have been primary goals of many IT organizations as means to curb software cost. These organizations have software applications that use the Internet/Web to transfer data, interoperate, and conduct business transactions. Object-oriented (OO) technology has been utilized to accomplish these goals with relative success over the years. Nevertheless, OO technology itself could not overcome many of the hurdles.

One of them is due to lack of standards. A software component developed in one vendor’s technology cannot easily communicate with another vendor’s. This is evidenced from the failure of major distributed OO technologies such as CORBA, DCOM, and RMI from becoming mainstream software development technologies. Another difficulty is due to the fact that the majority of software applications reside behind firewalls – security barriers that restrict communication between networks. Here, even if two systems use the same protocol to communicate, the security of firewall prevents the communications from taking place.

Web service [1, 2, 3, 4] is the latest buzzword in the industry to address the problems identified above. Web service model is one that utilizes loosely-coupled platform and language neutral framework for designing the next generation distributed systems. It is based on technologies by the W3C, the international standard body that oversees various Web related technologies. It also has strong support from major industry players such as Microsoft, IBM, Sun, HP, and Oracle, some of the major companies that are members of WS-I, an open industry organization chartered to promote Web services interoperability across platforms, operating systems and programming languages [5]. As such, and given that Web services operate on HTTP [6], a firewall friendly protocol, it is projected that many IT organizations will investigate and adopt Web services if proven viable.

In fact, recent studies on Web services have also shown its growth and acceptance. According to ZapThink, a market research firm, the market for Web services platforms, application development suites, and management tools is projected to expand from a $380 million (US) market in 2001 to over $15.5 billion (US) in 2005. Also, according to a comprehensive Web services usage survey conducted on a number of organizations including Global 1000 corporations during Feb 2004, 70 percent of them are currently using Web services either internally or externally [7]. Another important observation is that the use of Web services is now widespread in many industries including retail, financial services, national security, transportation, etc.

This paper focuses on the B2B design and implementation of Web services technology to integrate and automate the e-Transcript transmission process by colleges/universities. This addresses the present scenario where colleges/universities and students face the antiquated problem of paper-based transcript transmission during admission process, which traditionally requires tremendous time and efforts. By incorporating the latest WS-* security specifications such as WS-Security [8, 9, 10], WS-Policy [11], WS-Trust [12] and WS-SecureConversation [13], this paper showcases how the aforementioned problems together with security issues can be addressed using some of the burgeoning Web services technologies.

II. WEB SERVICES ARCHITECTURE

Web services architecture follows the ubiquitous Service Oriented Architecture (SOA) [14], an architectural style whose goal is to achieve loose coupling among the services, the providers, and the consumers. It consists of three main roles: the Service Provider, the Service Requester, and the Service Registry (Fig. 1). The Service Provider provides the Web service that is capable of doing a particular task, which may benefit a certain group of consumers. The Service Provider publishes its service in a Service
Registry, which stores a directory of services. The consumer or the Service Requestor searches the Service Registry for a service of interest. A client interface is then created at the Service Requestor end to bind it to the Web service so that communications can take place.

Fig. 1. Web Services Architecture

A. Web Service Programming Stack

There are various standardized protocols and application programming interfaces (APIs) used while engaging in Web service development so that the applications can easily locate and utilize them. There are four layers that constitute the Web Service Programming Stack (Fig. 2) and they are described below.

The network layer is the foundation layer, built on the HTTP protocol, through which the services are made available. Above it is the XML-based [15] messaging layer that helps the Web service and its clients to communicate with each other. It commonly uses SOAP [16] for all the required operations to bind the service consumers (clients) and service providers (servers) over HTTP. The next is the description layer, which uses WSDL (Web Service Description language) [17] to describe the Web service to the client. WSDL is an XML notation that contains details such as the operations included, message format, required policies and so forth about service provider. These three layers are required in order to have interoperable Web services.

The fourth is the service flow layer. This is an optional layer and is used to publish a Web service in order to give its access and location to its prospective consumers. It contains the WSDL and the URL pointing to the Web service. UDDI (Universal description, discovery and integration) [18] is a standardized directory for registering and querying the metadata of Web services. Some of the public UDDI services are the xMethods, IBM UDDI, and Microsoft UDDI.

Fig. 2. Web Service Programming Stack

III. E-TRANSCRIPT PROJECT

A. Present Scenario

With the advent of the Web, many universities/colleges have started online process for submitting admission applications. However, due to technological limitations, they are not successful in electronically gathering all the information needed from an applicant via an online system. Among the documents required for an application, official documents such as transcripts from the applicant's old colleges/universities are the most difficult to submit online for obvious reason—transcripts need to be authentic and it is not easy for the recipient to determine if an electronic copy is authentic, especially if it comes directly from the applicant. Thus the colleges/universities and students are forced to face the problem of paper-based transcript transmission during the admission process.

This current problem/scenario is described in Fig. 3. The student, who wishes to obtain a transcript from a particular college/university, has to initially send a request to the registrar office responsible for issuing the transcript, either by mail, fax, or by person. Then the office issues a paper-based transcript after some processing time. Then this transcript is mailed to the particular college/university the student requested. This in turn involves some transfer time. If the requested college/university is international, then the transfer time is even longer. Thus the student has to make sure that sufficient time is given for the whole process to unfold so that the transcript reaches the destination on time.

Fig. 3. Present Scenario for Transcript Transmission
The college/university issuing the transcript too has to utilize considerable time and efforts during the issuance of the paper-base transcript. For each request, it has to repeat the whole process of retrieving the student information, issuing transcripts, and transferring them to each and every college/university the student requested. From the admission office point of view, since the application materials are not delivered at the same time in one package (e.g., transcript is delivered weeks after the initial application date), it is error-prone and time consuming to collect documents and information from many different places and assemble them into a single application package for a particular applicant.

Lastly, another shortcoming of the present scenario is the non-standard transcript formats used by different colleges/universities. Virtually every institution has its own format for the transcript and do not follow any specific standards. This makes efficient processing of transcripts impossible and thus contributes to the overall delay of the admission process.

B. Web Services Scenario

With the aim to address the above drawbacks of a typical college/university admission system and to conduct a proof-of-concept of various emerging technologies, the proposed Web services solution sets out to build a Web-based e-Transcript transmission system to facilitate the student application process. This is done by means of allowing secure transmission of electronic transcripts over the Web between two colleges/universities. It uses secure Web services technology to automate the whole process of transcript transmission, there by reducing the time and efforts required by both the students and colleges/universities. It extends a previous project [19] by incorporating more security features and by using the XML-based Postsecondary Transcript Schema, released by PESC (Postsecondary Electronic Standards Council) and AACRAO (American Association of Collegiate Registrars and Admissions Offices) [20]. This permits the standardization of the transcript format for use by all colleges/universities that adhere to the PESC/AACRAO’s recommendation.

Fig. 4 provides an overview of transcript transmission using the Web services architecture. In this scenario, the student can interact with a client interface (provided by the college requesting the transcript), which in turn interacts with the Web service (provided by the college that the transcript is to be obtained from). The aim of the project is to enable e-Transcript processing through an integrated and interoperable system that is independent of the platforms involved.

Also, considering the sensitivity and importance of a transcript, another focus of this project was to use and implement the latest security specifications in the realm of Web services. The WS-* security specifications are implemented in order to provide message integrity and confidentiality.

IV. PROJECT IMPLEMENTATION

This project was implemented using Microsoft’s .NET Framework and its WSE (Web services Enhancement) environment. It consists of two Visual Studio .NET solutions—one for the Web service and the other for the Web client application. The user/student interacts with the Web client application to retrieve the transcript from the Web service of a system at, say, University of Illinois (service provider) and submit it directly to another system, say, Illinois State University (service consumer). Here the student has to provide student ID and PIN as the authentication information for the Web service to authenticate the request. Upon receiving the request from the client, the Web service sends the e-transcript as a response. The step-by-step details are given below.

A. Project Implementation Architecture

From Fig. 5, it can be seen that the first SOAP request is directed to the Security Token Service (STS) where the client token is exchanged for a Security Context Token (SCT), which is a derived key based on the security tokens of the client (Username Token) and the Web service (X509 certificate). The SOAP response delivers the SCT to the client, which uses it for encrypting and signing all the future SOAP messages that are communicated between the client and the Web service until the session expires.

The second SOAP request is signed and encrypted using the derived SCT with input parameters. After some database processing by the Web service, the appropriate XML based e-transcript
is sent as a SOAP response. Signing and encrypting the message follows the WS-Security specifications and the exchange of Security Context Token follows the WS-Trust and WS-SecureConversation specifications. An external XML file as per the WS-Policy and WS-SecurityPolicy specifications is used to specify the policies of the Web service.

Notwithstanding the initial difficulties (of getting supporting documents), the investment to use WSE2 paid off easily as it vastly reduces the amount of manual coding with respect to enforcing and verifying the security tokens. Much of the work was done via a policy file which can be generated automatically using the wizard. Further, if required, manual alterations can be made within the policy file in case of implementing custom policies.

As with any other tool, WSE2 too has its own limitations. It does not have any support for WS-PolicyAttachment, another WS-* specification. Thus, the senders and receivers need to use some other means (e.g., e-mails) to exchange information about their policies, i.e., the policy files. WSE2 also does not allow for custom SOAP fault messages concerned with policy enforcement or verification.

As mentioned earlier, this project was implemented using .NET, running on Windows XP. Needless to say, to get a better proof-of-concept, developing the project using different programming languages and platforms for the Web service and the client would be more realistic to showcase Web services interoperability. But due to time limitation and the non-availability of any other tool (at the time of project implementation) that supports the WS-* security specifications implemented in the project (i.e., WS-Policy, WS-Trust, and WS-SecureConversation), the resulted choice was to implement the project using the combination of .NET Framework and WSE2 entirely.

V. CONCLUSIONS AND FUTURE WORK

The paper briefly discusses Web services and their potential for re-shaping the software development industry. It showcases the collection of technologies behind Web services and how it can be used to implement a secure electronic transcript transmission system that can help both the students and the academic institutions to transfer e-transcripts over the Internet.

There are a number of future extensions that can be applied to the work presented here. Because of the limited support of tools that are available in the market today, the e-Transcript project used only a subset of the entire WS-* specifications that are available. Further specifications that may be incorporated include WS-* Reliability [24], which concentrates on message reliability, WS-* Metadata [25], which concentrates on the format of metadata information about Web service exchange such as WSDL, policies, and so forth, and WS-PolicyAttachment [26], which concentrates on how to attach policies to WSDL and UDDI.
Another potential future work is to re-architect the project using Microsoft’s Indigo [27], which is considered as the next step in the evolutionary path that “provides a rich set of .NET Framework based technologies for creating, consuming, and transmitting messages among connected systems”. Unlike WSE2, which is built on top the .NET framework, Indigo is being built into the Vista (previously called “Longhorn”) operating system to utilize the extensibility of open standards. The features of WSE2 will eventually be rolled into Indigo, which will be available to every Vista application and as separate download for Windows XP and Windows Server 2003.

In spite of the various advantages of Web services, an IT project needs to engage the proper system development life cycle to brainstorm if the scope and usability of the project can leverage the benefits provided by Web services and decide whether or not to use the technology. The adoption of Web services for the e-Transcript transmission system seems to facilitate the students and the colleges/universities in the overall processing of transcripts. But it would have real benefit only if the academic institutions adopt the technology by providing their own Web services for sending the transcripts of their students and conform to some standard XML transcript schema. With all these, the transfer of online transcript would be a secure, reliable, convenient, and painless mechanism that benefits all the parties involved.

REFERENCES


