SECURE WEB SERVICES FOR JXTA FRAMEWORK

İlhami Görgün
Supervisor: Asuman Dogaç

Middle East Technical University,
Department of Computer Engineering,
December, 2003

ABSTRACT

Web Services introduce a new paradigm for distributed computing, and the technology that it introduces constructs a new breed of Web application. This technology covers services that are self-contained, self-describing and modular applications which can be published, located and invoked through the Web. *Web Services* perform functions, which can be anything from simple requests to complicated business applications, and they can be described as any piece of software that makes it available over the Internet and uses a standardized XML messaging system. Once a *Web Service* is deployed, other applications (and other *Web Services*) can discover and invoke the deployed service.

In this work, a peer-to-peer approach is used to exploit *Web Service* technologies by providing *Web Services* security service for JXTA peer-to-peer networks. Peer-to-Peer (P2P) refers to a class of systems and applications that employ distributed resources (computing power, data, network bandwidth) to perform a critical function (distributed computing, data/content sharing, communication, collaboration) in decentralized manner.

JXTA is a network programming environment for the P2P platform. The vision of the JXTA project is to provide an open, innovative collaboration platform that supports a wide range of distributed computing applications and enables them to run on any device with a digital heartbeat.

The technology that the *Web Services* security introduces is that it defines a SOAP Header element to carry security-related data. If XML Signature is used, this header can contain the information defined by XML Signature that conveys how the message was signed, the key that was used, and the resulting signature value. Likewise, if an element within the message is encrypted, the encryption information such as that conveyed by XML Encryption can be contained within the *WS-Security* header. *WS-Security* does not specify the format of the signature or encryption. Instead, it specifies how one would embed the security information laid out by other specifications within a SOAP message. *WS-Security* is primarily a specification for an XML-based security metadata container.

In order to perform and adapt the *Web Service* security service for the JXTA framework, the JXTA components “security”, to make use of the cryptography toolkit, and “soap binding” , to make use of the Web Service technology exploitation in JXTA, are to be benefited from. In addition, as the peer-to-peer network implementation, the JXTA framework is to be used for the justification of the work.

TABLE OF CONTENTS

Title	**Page**
Abstract	
Introduction	3
Background	3
Peer-to-Peer Computing	3
Platforms	4
JXTA	5
Web Services Security	5
General Approach	7
Environment	7
Architecture	7
Advantages of the System	8
Conclusion	8

1. INTRODUCTION

The web is evolving into a provider of services such as information providing services like flight information providers, a variety of e-commerce and business-to-business applications. Interoperation of web services is needed to make such services computer interpretable, to create a semantic web of services whose properties, capabilities and interfaces are in machine readable form.

In this work, a framework is proposed for enabling secure web services technology exploitation in a peer-to-peer approach such that different services can be accessed in a secure manner in accordance with the web services security specification.

2. BACKGROUND

In this section, the background information about the core technologies related with the approach is described; namely, an overview about the related work in peer-to-peer environment for web service exploitation.

2.1 Peer-to-Peer Computing

The term peer-to-peer (P2P) refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized manner. P2P is increasingly receiving attention in research, product development and investment.

P2P enables:

- valuable externalities, the whole is made greater than the sum of its parts; by aggregating resources through low-cost interoperability

- lower cost of ownership and cost sharing; by using existing infrastructure and by eliminating and distributing the maintenance costs

- anonymity/privacy; by allowing peers a greater degree of autonomous control over their data and resources

P2P gained visibility with Napster's support for music sharing on the Web. However, it is increasingly becoming an important technique in various areas, such as distributed and collaborative computing both on the Web and in ad-hoc networks.

Assuming that "peer" is defined as "like each other," a P2P system then is one in which autonomous peers depend on other autonomous peers. Peers are autonomous when they are not wholly controlled by each other or by the same authority, e.g., the same user. Peers depend on each other for getting information, computing resources, forwarding requests, etc. which are essential for the functioning of the system as a whole and for the benefit of all peers.

Conceptually, P2P computing is an alternative to the centralized and client-server models of computing, where there is typically a single or small cluster of servers and many clients. In its purest form, the P2P model has no concept of server; rather all participants are peers.

Selecting a P2P approach is often driven by one or more of the following goals:

- **improved scalability/reliability:** with the lack of strong central authority for autonomous peers, improving system scalability and reliability is an important goal

- **resource aggregation and interoperability:** by aggregating compute resources at thousands of nodes, they are able to perform computationally intensive functions

- **increased autonomy:** in many cases, users of a distributed system are unwilling to rely on any centralized service provider. Instead, they prefer that all data and work on their behalf be performed locally

- **anonymity/privacy:** much of the cost sharing is realized by the utilization and aggregation of otherwise unused resources (e.g. SETI@home), which results both in net marginal cost reductions and a lower cost for the most costly system component. Because peers tend to be autonomous, it is important for costs to be shared reasonably equitably

- **improved scalability/reliability:** with the lack of strong central authority for autonomous peers, improving privacy. A user may not want anyone or any service provider to know about his or her involvement in the system. With a central server, it is difficult to ensure anonymity because the server will typically be able to identify the client, at least by Internet address. By employing a P2P structure in which activities are performed locally, users can avoid having to provide any information about themselves to anyone else. FreeNet is a prime example of how anonymity can be built into a P2P application

- **dynamism:** P2P systems assume that the computing environment is highly dynamic. That is, resources, such as compute nodes, will be entering and leaving the system continuously. When an application is intended to support a highly dynamic environment, the P2P approach is a natural fit

2.2 Platforms

There are a number of candidates competing for future P2P platform. .NET is the most ambitious one, going beyond P2P to encompass all service support on the client and server side. JXTA is another attempt, taking a bottom up and strong interoperability approach. Most other systems also have some level of platform support, such as Groove covering enterprise domain and Magi, covering handheld devices domain. However the JXTA platform, formed by Sun Microsytems will be used in the work.

2.3 JXTA

In this section a brief introduction about the P2P platform, JXTA is given.

The vision of the JXTA project is to provide an open, innovative collaboration platform that supports a wide range of distributed computing applications and enables them to run on any device with a digital heartbeat. JXTA provides core functionality in multiple layers, including basic mechanisms and concepts, higher level services that expand the capabilities of the core, and a wide range of applications that demonstrate the broad applicability of the platform.

Goals. The goal of JXTA is to provide a "general-purpose" network programming and computing infrastructure. Its goals are:

• Interoperability: by enabling inter-connected peers to easily locate each other, participate in community based activities and offer services to each other seamlessly across different P2P systems and different communities

• Platform independence: JXTA is designed to be independent from programming languages (such as C or Java), system platforms (such as Microsoft Windows and UNIX operating systems), and networking platforms (such as TCP/IP or Bluetooth)

• Ubiquity: JXTA is designed to be implementable one very device with a digital heartbeat, including appliances, desktop computers, and storage systems

Peer groups are the core of JXTA's infrastructure. A peer group is essentially a partitioning of the world of peers for communication, security, performance, "logical locality" and other reasons. A single participant can be in multiple groups at one time.

JXTA uses asynchronous uni-directional communication channels, called pipes, for sending and receiving messages. All data interchange in JXTA is in the form of XML formatted documents.

2.4 Web Services Security

WS-Security specifies the methods to embed security within the SOAP message itself, exploring the concerns WS-Security addresses: authentication, signatures, and encryption.

Over HTTP, one can authenticate the caller, sign the message, and encrypt the contents of the message. This makes the message secure in several dimensions: the caller is known, the receiver of the message can verify that the message did not change in transit, and entities watching the wire traffic cannot figure out what data is being exchanged. For those looking at SOAP messaging to solve bigger problems, however, HTTP-based security simply isn't enough. Many of the bigger problems involve sending the message along a path more complicated than request/response or over a transport that does not involve HTTP. The identity, integrity, and security of the message and the caller need to

be preserved over multiple hops. More than one encryption key may be used along the route. Trust domains will be crossed. HTTP and its security mechanisms only address point-to-point security. More complex solutions need end-to-end security baked in. WS-Security addresses how to maintain a secure context over a multi-point message path.

WS-Security addresses security by leveraging existing standards and specifications. This avoids the necessity to define a complete security solution within WS-Security. The industry has solved many of these problems. Kerberos and X.509 address authentication. X.509 also uses existing PKI for key management. XML Encryption and XML Signature describe ways of encrypting and signing the contents of XML messages. XML Canonicalization describes ways of making the XML ready to be signed and encrypted. What WS-Security adds to existing specifications is a framework to embed these mechanisms into a SOAP message. This is done in a transport-neutral fashion.

WS-Security defines a SOAP Header element to carry security-related data. If XML Signature is used, this header can contain the information defined by XML Signature that conveys how the message was signed, the key that was used, and the resulting signature value. Likewise, if an element within the message is encrypted, the encryption information such as that conveyed by XML Encryption can be contained within the WS-Security header. WS-Security does not specify the format of the signature or encryption. Instead, it specifies how one would embed the security information laid out by other specifications within a SOAP message. WS-Security is primarily a specification for an XML-based security metadata container.

Figure 1 depicts what will become a fairly common message flow:

Figure 1. Typical message flow.

3. GENERAL APPROACH

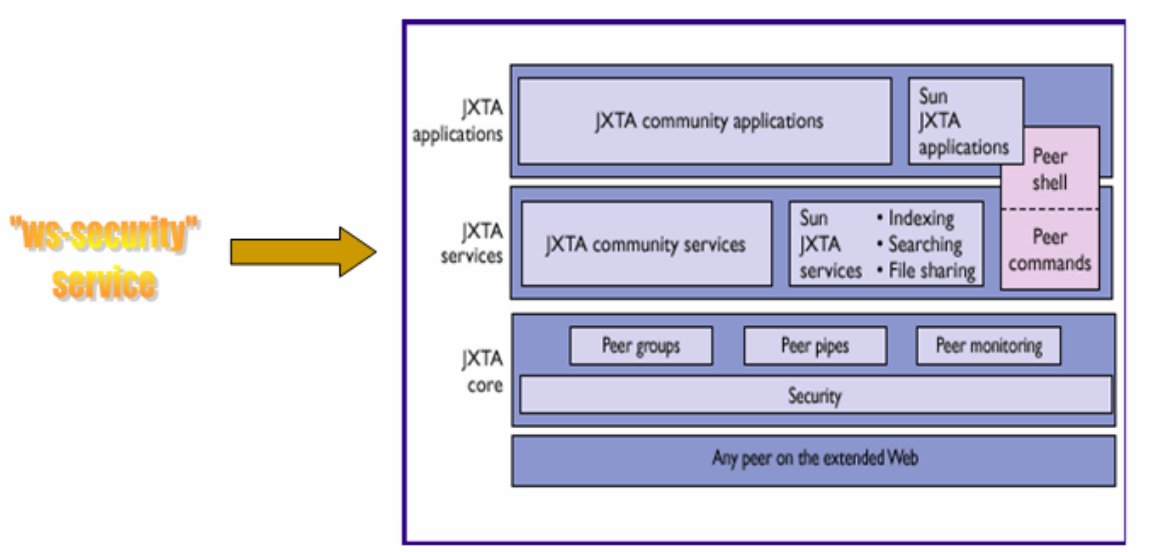
3.1 Environment

The proposed system is to make use of the JXTA framework to establish a peer to peer network. JXTA provides a peer to peer environment to publish, search and invoke web services through its SOAP Binding component. It also allows secure message exchange through its Security component which provides whole message security.

The main aim of this study is to provide secure SOAP message exchange in accordance with the Web Services Security specification.

3.2 Architecture

The following figure indicates the architectural appearance of the JXTA framework with the newly introduced service component with this study:



The newly introduced service component relies on the JXTA framework key management mechanism and the cryptography toolkit provided by the Security component. It is to cover security token, XML Signature, and XML Encryption implementations. In addition, it is to provide an API for JXTA web service applications via the SOAP Binding component.

The WS-Security service API is designed in two alternatives to make use of it:

1. by parsing the DTD file of the specified SOAP message; and with a user interface, encrypting and/or digitally signing the depicted XML element nodes by the SOAP message sender

2. encrypting and/or digitally signing the XML element nodes of the SOAP message depicted in the service advertisement

3.3 Advantages of The System

Considering the nature of the XML documents that they can travel through many originators within the scope a scenario, and the popularity of the JXTA framework becoming a common programming environment for peer-to-peer networks; the gap for web services invocation in JXTA peer-to-peer networks is to be overcome with this study relying on the newly conceived security specification for web services.

4. Conclusion

With this study, a new approach is taken to exploit web services to its full power conforming to the web services security specification. Considering the potential of the peer-to-peer networks and the promises of the JXTA framework in order to provide a common peer-to-peer programming environment, the study is to be based on the architecture of the JXTA framework; and its capabilities to exploit web services technology.

5. References

1. Peer-to-Peer Computing Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja1, Jim Pruyne, Bruno Richard,

2. O'Reilly p2p conference (http://www.oreillynet.com/) .

3. Exploiting Web Service Semantics: Taxanomies vs. Ontologies , Asuman Dogac, Gokce Laleci, Yildiray Kabak, Ibrahim Cingil

4. Jxta official site (http://www.jxta.org).

5. World Wide Web Consortium, XML Encryption Standard, http://www.w3.org/TR/xmlenc-core/

6. World Wide Web Consortium, XML Signature Standard, http://www.w3.org/TR/xmldsig-core/