

Web Services Interoperate as Distributed (Systems) Based on Trust

Mohammad Rostami¹⁺, Esmail Bagheri²

¹Member of Young Researchers Club, Islamic Azad University, Dehaghan Branch

²Department of Computer, Islamic Azad University, Dehaghan Branch

Abstract. Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. Web applications are independent of plat form. They can available in the form of a set of function in web by address. Web Service is separated from the other available technologies because of using XML and some other standards technology such as SOAP, WSDL and UDDI. Once a web service is deployed, other applications are able to discover and invoke the deployed service .Web Services are reference point in the architecture of distributed applications on network. Undoubted, one of the most important evolutions in context of distributed applications is, discussing web services that will be followed by many impacts communicated with soft ware statues especially on network.

A distributed application is one that its processing potentials might be provided by some physical computers and its data could be set in some physical places. A web service that is running on servers can receive a request from a distant computer and send computer necessity information for it after necessary processing on network. Web services are main components to build distributed processing. Web services interoperate with one another as distributed system. It should be used of Ws-Trust protocol for cooperating these web services. In this paper, it is explained that, the web services can be interoperated as distributed based on trust.

Keywords: Web service, distributed system, discover, invoke, Ws-Trust.

1. Introduction

Web services were built by XML technologies to aid IT developers to help heterogeneous applications. The architecture of prevalent web services have three main roles: service provider, service consumer and integrated registry and discover current technology Web services discovery based on UDDI are designed for such Web services that can provide the necessity of user dutiful and their necessity of user undutiful such service qualities are greatly ignored in discovery.

With increasing the number of web services on web, it may be caused problems as services on web, it may be caused problems as services are returned that are useless Regardless details, web services are applications that can be published, located and invoked by network, for example: [9].

- Taking information of portfolio cost
- Obtaining weather report
- Reserving airport ticket

Web services may use of other web services to perform their works [5] [9].

In simple word, it can be said that web service is a traditional web service (like nominating service, weather report service producing electronic and etc) that is available by network. Web service follows of a set of standards that is caused service-consumers who follow of them to obtain them by network. These standards make the architecture core of web component [2].

⁺. Corresponding author. Tel.: +989132893902.
E-mail address: mohamad.rostami10@yahoo.com.

A web service is a kind of web component. This component makes application which use of it able to apply this web service methods [7].

Nowadays the security problems are the biggest preventative ways on deploying the web services. However many efforts have made in the context of the web services. Still, some of the problems haven't been solved totally. Such as, to compatible of access control in the web services.[13][15]

The architecture with weak and distributed connection for the web services creates challenges in surveying on patents and applying the access control on different references. Such architecture hasn't been yet created to control the access of web services, as they are composed for a special operation. Different departments have roles with different accesses, so, the registration of specifications or clients roles is difficult between different systems.[13][17]

2. CORBA

CORBA is a method based on open standards for distributed calculations. The main advantage of CORBA, it is that, service consumers and service producers can be encoded by every programming language. It is possible, because the objects have been defined by Interface Definition Language (IDL) with high level section. After defining an IDL file, the compiler will provide the registration of IDL file to a special programming language. The communication between objects, service producers (servers) and service consumers (clients) performs by ORB s. If it is necessary to a high efficiency in distributed applications, CORBA will still be among the optional tool. Anyway, the encoding of distributed applications with CORBA is difficult. In order to translate IDL to different languages should limit itself to the contexts found in all supported languages.[14][8]

3. DCOM

Microsoft's Distributed Components Object Model allows away objects to be invoked by using of a layer placed on top of DCERPC mechanism and interacted with DCOM run time services. DCOM service producer (server) publishes its methods for service consumers (clients) by supporting of multiple interfaces. Interfaces have been encoded in IDL that is similar to c++.

DCOM binary level specifications allow using of different languages for encoding service producer (server) objects.[14][12]

4. Surveying on Web Service Standards

In old world of service producer-service consumer, the service producer presented several functions that could be used or invoked (called) by service consumer. Some kind of explorer services operated as interface between service producer and service consumer. As far as the web services presented only another kind of distributed applications, they consist of three same components: A service interface that operates as an explorer service between service provider and service applicator. Service provider that publish its services for service interface. Service applicator that ask from service interface where is found a suitable service provider and then connect itself to provider. Anyway web services use of XML on HTTP [2].

XML¹ is widely an acceptable format for exchanging information and rival meanings that is approximately main and fundamental section for each other layer which is used for web services. Generally, these layers make web services stack that consists of below parts:[2] [5]

-XML is a markup language that provides a format to describe data. Like html, it consists of tags, etc. XML was provided as structured self-describing method to represent data that is totally independent of application, protocol, terms, operating system or even programming language. Many nominate XML as language mixed with other business languages, because it is widely used by all companies to transport loadable business data [8].

Using of XML shows a vast set of security challenges. Both XML and HTML are the formats based on text which obtain from the same root. XML was first developed for dominating on HTML limits, that, it is

¹. Extensible Markup Language

good to describe how should be shown the objects. But it is weak to describe what data are shown. XML based on text is too useful for the web services, because it is readable for human. There is no tool for analyzing and translating the data. The simple text tools and editors are sufficient to manipulate it.[2]

-SOAP² is communicating protocol for web service such html for web and etc.

Regarding to web services are executed in heterogeneous environment, used protocols to transport data between functions should be independent of each run time environment. SOAP is a protocol that has these characteristics. SOAP itself doesn't define any application context such as programming method or special deploying concepts that mean distributed garbage collection..

It more defines a simple mechanism to explain application concepts by providing modular package model and coding mechanisms to code data in modules [2].

-WSDL³ describes net services by using of language order of XML. Also, it provides documentation for distributed systems and enables programs to communicate with each other by automatic method. While SOAP describes communication between applicator and provider, WSDL describes provider services (an ending point) and it may be used as producing order of suitable SOAP message to access to the service [4].

-UDDI⁴ is a designed standard to provide list of search business web services. So, it presents a service interface that enables service applicators to find its suitable service provider. UDDI has been designed on different methods such a phone booklet. [6] [10].

5. Web Services Interoperate as Distributed (Systems)

Web services exchange data as distributed or interoperate with each other to perform a work. To interoperate with these web services, it should be made a protocol between them to able cooperate together in addition of preserving security. For this purpose, it can be used of WS-Trust protocol that describes a model to make direct and indirect trust communications with interface.[16]

When credentials are published in a schema and then are presented in another, the main problem of WS-Trust occurs. For example, your manager, ABC, is a provider for XYZ transport company and it has decided to present itself vital business system for its providers. An strategy for XYZ transport company, it is that, to publish credentials for every individual provider, directly, and to manage them. It means that even if you don't work for XYZ, you will receive a credential from the company for presenting to web service, as using of its Bid to order to web. But how will XYZ transport company know if you are recalled or upturned?

It is very difficult but it isn't impossible, for XYZ company to manage credentials given to a number of particular ABC providers employees who have directly permission to order in XYZ . Another method that XYZ can choose, is sightless trust to ABC provider and all its employees. Of technical view, this strategy may mean that X.509 certificates published by ABC provider(perhaps in defined characteristics) Can be presented to XYZ to access to its web. It may immediately seem to be your trust. Perhaps, these are very sensitive facilities that XYZ company allows its access, as even its employees should be personally presented to receive the credentials. Assume that XYZ transport company has periodic difficult methods for rebuilding the credentials and for canceling them, when they are removed or employees change the responsibility. If ABC provider company hasn't similar difficult methods, XYZ company will seriously reduce the security by weaker ABC standards. This level of trust become the field of contracts and low because it is the way that companies explain their trust or lack of it and exceeding of the trust become practible. Before having the safe communications, each unit needs to define whether the credentials of other units are reliable.WS-Trust defines junctures for WS-Security that provides the following cases:

- The methods for distributing and exchanging security tokens
- The methods to configuration and access to the trust communications

Similar to SAML, the WS-Trust defines a request/response mechanism to obtain security token.[1][7]

6. Ws-Trust Specifications

² . Simple Object Access Protocol

³ . Web Services Define Language

⁴ . Universal Discover Description Integration

The most important WS-Trust specifications are as below

- The communication between web service applicator, web service provider and security token service (STS) which is shown in the figure 6-1.

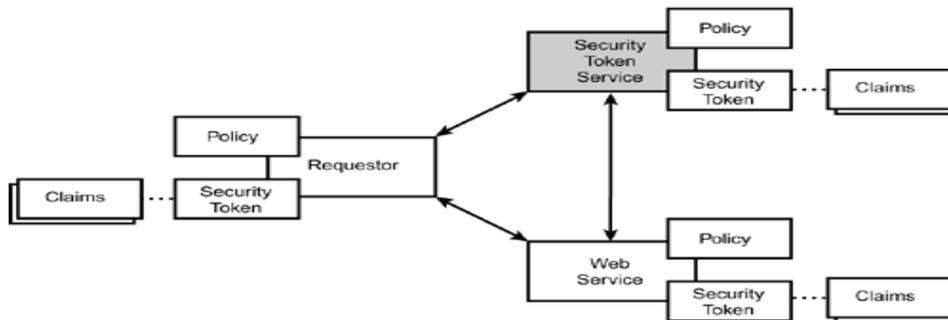


Figure 6-1: WS-Trust model to obtain a trust interface of security token service in supporting of WS-Security

- Both Request Security Token and Request Security Token Response models [17][11]

security token services form the fundamental of trust by publishing security tokens which are able to use for interface trust communications among different certificated domains.. As shown in figure 6-1, a security token service is a prepared reliable interface (mediator) that tends to provide a web service with a method to realize whether the arrived request from a different reliable schema (probably unknown) can be trusted. A politic statement in the web service provider informs the web service applicator that a security token is needed to use the service. The request agent provides such toke, both sides obtain it before or a , < Request Security Token > is sent to the security token service by request agent now. While requesting service from web service producer, such a compatible security token is provided. Also WS-Trust establishes a challenge-response protocol to validate the impact and to make the request agent token appropriate.[1]

The web service security model defened in WS-Trust based on a procedure which is necessary for a web service to confirm a set of request by an entry message.(for example name, key, credit, ability and etc). If a message entered without having needed document for requests, the service will ignore or reject whole message. A service can point to needed requests and related data on its policy that has been described by WS-Policy and WS-Policy Attachment. A request agent can send those messages which show its ability to confirm in the set of its needed requests by depending security token on messages and placing messages signatures which represent possessing document of tokens.

WS-Trust shows in figure 6-1 that a request agent (an applicant) may be a service and security token service is a web service with explaining policy and needed security token. This public messaging - requests, policies and security tokens- support some particular models, such as security based on identity, lists of accessibility control and security based on abilities. It allows using of available technologies such as public key X.509 certificates, tokens supported on XML, Kerberos communal private labels and even passwords. Public model in composition with WS-Security and WS-Policy to build exchanging high level key, admission of reliability (trust), accessibility decisions supported on policy, search and complicated trust communications.[1][3]

7. Web Services Based on Trust

It should be used of Ws-Trust protocol for cooperating web services.

A short definition of WS-Trust model is that a web service has an applied policy on it, it receives a message from an applicator that may consist of security tokens and may have a number of applied security or safety on it by using of WS-Security mechanisms. The following steps are done by a WS-Trust motor:

- Controlling the requests in security token to be sufficient for compatibility with provider policy and the message to compatible with this policy.

- Controlling the applicator characteristics are proved by signatures. In the interface trust models, maybe, the signature doesn't control the applicator identity(ID). It may control interface identity(ID), every one may simply contend applicator identity(ID). Requests are stabilized and not based on policy.

- Controlling the security token publishers (consisted of all related security token) are reliable for publishing the requests. Trust motor may need to send tokens to a security token service for exchanging them with others that can directly be used in evaluation. If these states have compatibility and applicator has permission to implement operation, service can process the service request in defined trust model.[17][1]

8. Conclusions

Web services are new technologies that provide interaction between machine to machine on a network by using of a set of standardized technologies consists of WSDL, SOAP, and UDDI. Under these technologies, UDDI is designed to find the services that provides an access to WSDL documents and it includes four core types of information: business Entity, business Service, binding Template, t Model.

Web services enable desktop and web applications to access functions in them with communication on HTTP and to receive its conclusion in the standard format platform of XML. Web services don't depend on programming language, platform and special protocol. It means that you can use of Web services with each programming language on each platform. A Web service sends the result of its processing to applicator program in XML format platform by HTTP. XML format is a standard way for communicating between two adverse systems. Web services messages are sent by using of HTTP because of easily receiving to applicator on network and without blocking by fire walls.

Web service is a software which provides accessibility to data and also data processing systems independent of different hardware and software platforms by using of standard protocols of exchanging net data.

(These standards include the reuse of HTTP to standardise XML which lead onto SOAP).

Web services interoperate with each other as distributed WS-Trust protocol must be used to cooperate these web services. This protocol make a model to provide direct and indirect trust communications with interface. That credentials is published in a schema.WS-Trust defines junctures for WS-Security.

9. References

- [1] Hoops.*Web Services as Distributed Systems*.Part 1, December 7, 2010.
- [2] Richard Monson; Haefel.*Web Services: XML SOAP WSDL UDDI WS-I JAX-RPC JAXR SAAJ JAXP*.
- [3] M.Tamer Ozsu, P. Valduriez. *Principles of Distributed Database System*. Prentice Hall, USA, 1999.
- [4] Christensen E., et al., *Web Services Descriptionlanguage(WSDL)1. 1*, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>,March 2001.
- [5] Ethan Cerami. *Web Services Essentials*, February 2002.
- [6] Ehnebuske D., et al., *UDDI Version 2.04 API*, http://uddi.org/pubs/programmersAPI_V2.04-published-20020719.htm,july 2002.
- [7] Blum A., Carter, F. *Representing Web Services Management*.
- [8] chase,Nicholas, *understanding Web Services Specifications*. Aug 2006.
- [9] BellwoodT., et al., *UDDI Version 3.0*, <http://uddi.org/pubs/uddi-v3.htm>, July 2002.
- [10] Ehnebuske D., et al., *UDDI Version 2.03 Data structure Reference*, http://uddi.org/lubs/Datastructure_V2.03 Published-20020719.htm, July 2002.
- [11] Bertino, Elisa, Martino, Lorenzo, Paci, Federica, Squicciarini, Anna. *Security for Web Services and Service-Oriented Architectures*. Springer, 2009.
- [12] O'Neill, Mark. *Web services security*. McGraw-Hill Professional, 2003.
- [13] Galbraith , Ben. *Professional web services security*. Wrox, 2002.
- [14]Periorellis, Panos. *Securing Web services: practical usage of standards and specifications*. IGI Global snippet,

2008.

- [15] Papazoglou, M. *Web services: principles and technology*. Pearson Prentice Hall, 2008.
- [16] Zhang, Liang-Jie, Jeckle, Mario. *Web services: European conference*, ECOWS 2004, Erfurt, Germany, September 27-30, 2004.
- [17] Birman, Kenneth Paul. *Reliable Distributed Systems: Technologies, Web Services, and Applications*. Springer, 2005.