RESEARCH ARTICLE OPEN ACCESS

Middleware Interoperability using SOA for Enterprise Business Application

T Sathis Kumar
Assistant Professor
Department of Computer Science and Engineering
Saranathan College of Engineering, Trichy
Tamil Nadu - India

ABSTRACT
In general a business level service can be implemented by multiple objects/components, or by a non-OO approach. SOA can be achieved with many types of middleware. The middleware works in terms of services. Quality of Service based on Security , Transactions, Routing, Logging, Failover, Protocols and of course Message formats. This may help to keep the business and implementation levels separate for the designers and implementers. To get an enterprise-wide SOA, we need to have middleware interoperability. The interoperability takes care of the ability of software and hardware on different machines from different vendors to share data. But middleware doesn't interoperate; even the "service" interfaces are different. A service has an interface, and this is the key to hiding details such as the operating system, programming language, network addresses, etc.But this interface doesn't hide/abstract the middleware itself. In this paper, I examine how the security applies to the Service Oriented Architecture (SOA) through the interoperability.
Keywords:- Web Services, SOA, Interoperability, SOAP, WSDL, WSARCH.

I. INTRODUCTION

A service is a Software component that is well-defined, well-structured, self-contained, and does not depend on the context or state of other services. Nowadays, service-oriented architecture (SOA) is used as an efficient solution to integrate the distributed applications in an enterprise model. In a SOA-based environment, Security is one of the major concerns when developing enterprise business applications.

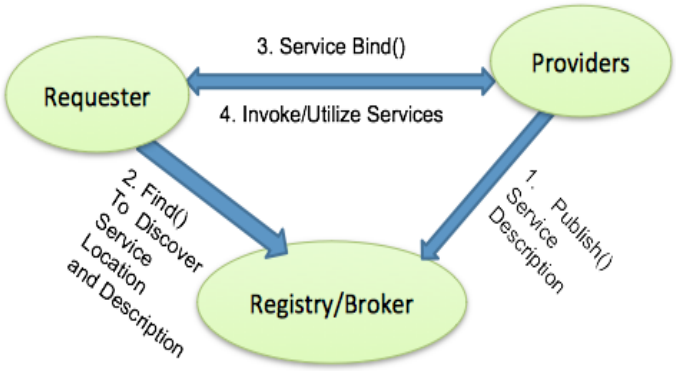

Fig. 1 SOA Components and Operations

SOA is used to provide an integrated, flexible, and cost efficient (Web) Service-based enterprise. It promises interoperability, reusability, loose coupling, and protocol independency of services as core principles of SOA ([1][2]). Normally, this standard-based approach uses Web Services as building block to support particular business tasks. Web Services are published with Web Services Description Language (WSDL) interface and they use Simple Object Access Protocol (SOAP) as a communication protocol. In the context of SOA, developers must focus the interoperability of services. Thus, SOA experts must provide the capability to secure the architecture instead of securing a service itself [1]. In order to overcome these matters, the various functional and non-functional security requirements are needed to be considered. Some of these requirements such as authentication, end-to-end security, interoperability, access control, auditing, secure configuration, assurance, and compliance have been presented by [1], [2], and [4]. In addition, some technologies and standards such as XML Signature [7], XML Encryption [8], WS-Security [9], XKMS [10], SAML [11], and XACML [12] have been developed to support the above requirements.

II. WEB SERVICE TECHNOLOGY STACK

Web Services are considered as an instance of a SOA. The Web services network is an application level network involving a number of participants like service providers, service consumers, and service registry operators. The stack starts at the service transport with the basic technologies that allow data transfer from one machine to another. Each layer is built on the lower layers and adds higher-level abstractions.

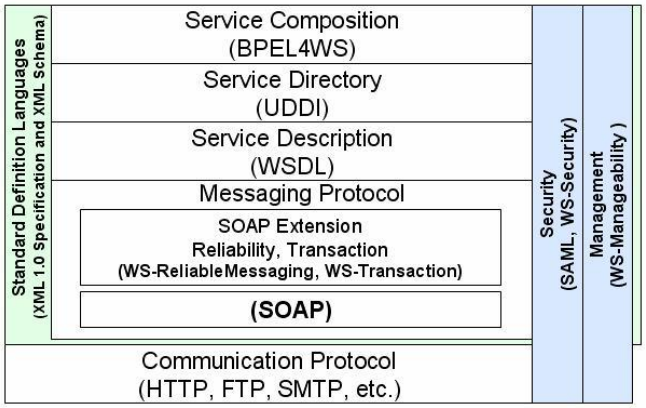

Fig. 2 Web Services Technology Stack

The upper layers of the stack do not necessarily depend on the lower layers and in some ways are orthogonal concerns. They are shown in this format to demonstrate the higher level of abstraction. Thus, the security may be guaranteed by the safety mechanism of present network level[3]. SOAP and HTTP are enough for interoperability with XML based messages transmission and WSDL describes the complete service.

III. WEB SERVICE TECHNOLOGIES

The web service paradigm has grown so quickly, hence several competing technologies are attempting to provide the more capability techniques [4]. However, the web service vision of seamless worldwide business integration will not be feasible unless the core technologies are supported by every major service based company in the world. 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.Infact, over the recent years; three primary technologies have emerged as worldwide standards that make up the core of today's web services technology such as SOAP, WSDL and UDDI.

1) ***Simple Access Object Protocol (SOAP):*** 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 "Envelope", an optional "Header", and a mandatory "Body". The SOAP "Body" carries application-specific contents including the method name and the serialized values of the methods' input or output parameters. 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

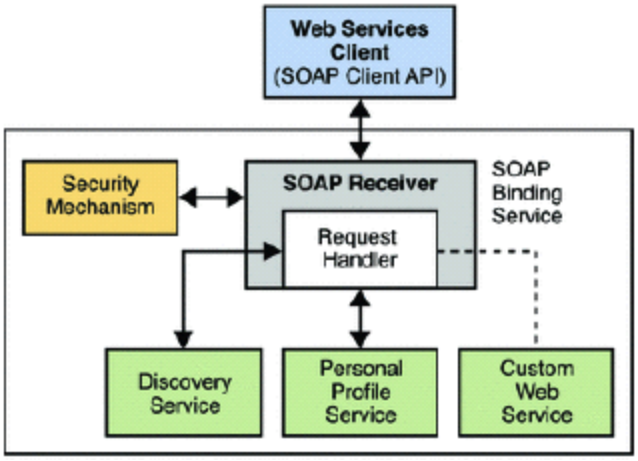

Fig. 3 Web Services Stack Design

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.

2) ***Web Service Description Language(WSDL):*** WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. WSDL describes the interface of a web service in a standardized way. WSDL standardizes how a web service represents the input and output parameters of an invocation externally, the structure of function, the nature of the invocation and the service protocol

binding. WSDL allows unperceptive clients to automatically understand how to interact with a web service[5]. A Web services developer usually uses a software tool to generate the WSDL file from the source code of a Web services program in which some public methods are marked as Web services operations to be accessed by Web services consumers[6]. A Web services consumer needs to generate a Web services proxy (client) to handle the encoding and messaging of an actual Web services call. A Web services generation software tool that uses a WSDL file as the input can generate the Web services proxy. Once the proxy is generated and referenced, the client application can use the Web services program via a set of the published Web operations as if they were local procedures or objects. The Web services client proxy will handle the serialization and de-serialization between internal data types and SOAP data representation

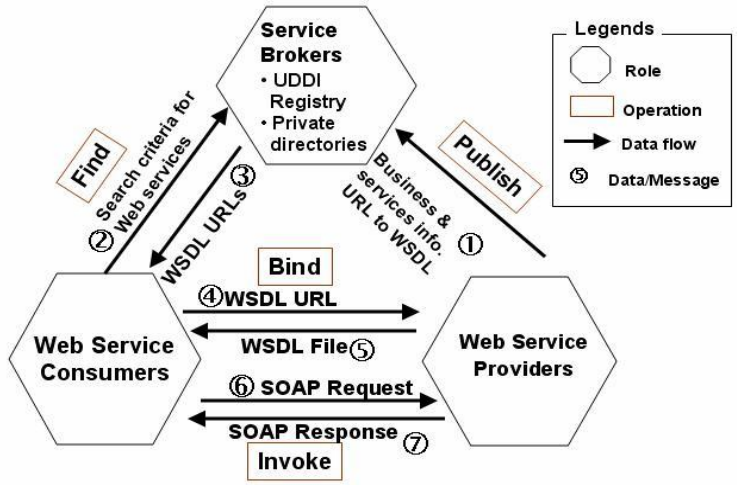

Fig. 4 Web Services Based Architecture

A WSDL file contains service definitions for distributed systems to support the automatic creation of client-side stubs or proxies, and the binding to the Web services[13]. It describes the interfaces to a Web services implementation in terms of format of the messages, binding of the abstract messages to a concrete protocol, and address of the endpoint. It is a "take -it-or-leave-it" technical contract offered by a Web services provider to Web services consumers.

*3) **Universal Decsription Discovery Language(UDDI):***

The UDDI specification defines a platform-independent framework for businesses to describe their publicly available services, discover other services, and share information about points of interaction in a global registry[14]. UDDI is a building block that enables organizations to quickly find and conduct business transactions with other organizations using open standards.

The UDDI specification is an attempt for organizations to accomplish the following:

i) Discover the right partner to conduct business with one out of the millions that exist on the Internet today.

ii) Create an industry-accepted standardized approach to reach partners and customers with information on their services.

iii) convey the preferred method for integration between disparate systems.

iv) Characterize how business transactions are used in commerce, once a preferred partner is finalized.

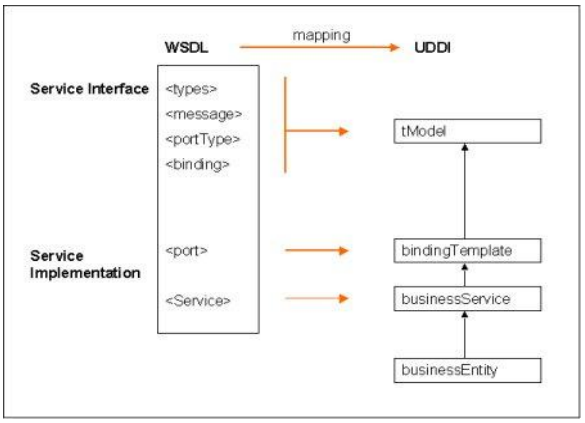

Fig. 5 Service Interface of UDDI

UDDI can be the solution to many business problems. By simplifying B2B interactions, a business can discover other business, independent of the choice of standards and protocols.

IV. WEB SERVICES ARCHITECTURE

The WSARCH (Web Services Architecture) is an architecture which allows accessing Web services using a combination of functional and non-functional aspects of Quality of Service (QoS). These QoS aspects aim at evaluating the performance of Web services in order to achieve QoS in a service-oriented architecture for enterprise application. These QoS attributes were mapped to the components participating in a service-oriented architecture

that incorporates quality of service [15]. The architecture provides the monitoring of service providers and the data obtained are used to locate the most appropriated service. A prototype for the WSARCH allows performance evaluation studies being conducted considering different components of the architecture, algorithms, protocols and standards. By now, we want include security attributes in this architecture involving all the components (UDDI, Broker, clients and providers).

The following steps are necessary for the execution of a request:

1. Client makes request to the Broker, which has updated information from the service provider (load, type of service, client class).
2. Based on the QoS information requested by the client, the Broker performs a search in a service repository in order to find the most appropriate service.
3. Broker gets the specification of appropriate service and information about the QoS of the service provider.
4. With the location of appropriate service in the service repository and information of the service providers (this QoS information is propagated periodically), the Broker chooses the best candidate provider (Services Selection).
5. After selecting the service, the Broker performs the request (invocation of Web Service) to the service provider.
6. After being made the request, the reply is returned directly to the Broker.
7. Finally, the reply is sent to the client that originally requested the Web service.

Other activities occur in parallel with the request of the service client. The QoS information of service providers are updated periodically in the UDDI service registry for each of the registered providers. The information is obtained through the use of the Ganglia Monitoring System running in the service providers and also in the UDDI. Service providers have slave monitors that collect and send information to a master monitor in the UDDI registry, so that the Broker can use it as QoS information (performance index) for selecting the best service provider.

V. CONCLUSIONS

In this paper, a brief description of conceptual security for enterprise service oriented architecture is provided. 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. The emerging Web services standards and technologies enable companies to provide software functions and business services over the Web to be integrated by internal business processes or with trading partners. Web services have been proclaimed as "bigger than the Internet" by many advocates. These supporters believe that Web services are "the" technology to bridge the gap between IT and business. It generates a renewed interest and excitement in B2B electronic commerce and mobile commerce, as well as enterprise application integration (EAI). 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.

REFERENCES

[1] Ramarao,k. & Prasad, C.(2008). SOA Security. USA: Manning Publication.

[2] Eric Pulier & Hugh Taylor. (2006). Understanding Enterprise SOA. USA: Manning Publication.

[3] Web Service Activity, http://www.w3.org/2002/ws/

[4] Kalantari, A., Khezrian, M., Esmaeili, A. and Taherdoost, H. (2011). Enabling Security Requiremets for enterprise Service Oriented Architecture. International Journal of Recent Trends in Engineering and Technology, The Association of Computer Electronics and Electrical Engineers (ACEEE), 6(1), 75-81 .

[5] New to SOA and Web Service, http://www.ibm.com/developerworks/ webservices/newto/service.html

[6] M. Schumacher, D. Witte. (2007). Secure Enterprise SOA: known and new security challenge, Datenschutz und Datensicherheit.

[7] XML- Signature. (2001). Retrieved 2009, from W3C: http://www.w3.org/Signature/

[8] XML-Encryption. (2002). Retrieved 2009, from W3C: http://www.w3.org/Encryption/

[9] S. Thompson. (2003, 04 01). Implementing WS-Security. Retrieved 2011, from IBM: http://www.ibm.com/ developerworks/webservices/library/ws-security.htm

[10] W. Ford. (2001, 03 30). XML Key Management Specification (XKMS). From W3C: http://www.w3.org /TR/xkms/

[11] Security Assertion Markup Language (SAML). From OASIS: http://docs.oasis-open.org/security/saml/v2.0/

[12] eXtensible Access Control Markup Language (XACML). From OASIS: http://www.oasis-open.org/committees/ tc_home.php?wg_abbrev=xacml

[13] Buecker et al. (2007). Understanding SOA Security: Design and implementation. USA: IBM Publication.

[14] Candolin. (2007). A Security Framework for Service Oriented Architectures. Military Communications Conference, 29-31 Oct. MILCOM 2007: IEEE.

[15] Arsanjani, A.(2007). S3: A Service-Oriented Reference Architecture. IEE Computer Society, 9(3), 10-17.