



## Service-Oriented Architecture and Web Services

**Zahid Hussain Khuhro\***

*Institute of Business & Technology, Biztek, Pakistan*

**Noor Zaman\***

*College of Computer Science & IT, King Faisal University Saudi Arabia*

**Zia Ahmed Shaikh\***

*Institute of Business & Technology, Biztek, Pakistan*

### ABSTRACT

Web services are the service based Platform; web services provide the services to combine different environments of different technologies, where as there are other requirements such as privacy of data, confidentiality it and accuracy of data. When user send the query to the web servers the possibility of two security types are there one is point - to - point and second is end -to - end security. After deployment of web services requirements monitor is used to analyze the confirmation of a web service to fulfill such requirements. The integrity of the confirmation is based upon the integrity of monitor specially web services reduce the risk factor for service oriented architecture. In this paper, we proposed a hardware-based dynamic attestation mechanism to validate the integrity of the requirements monitor

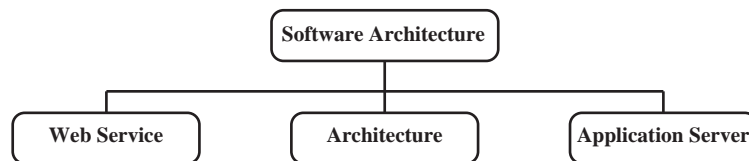
**INSPEC Classification :** C7210, D2080, D5020

**Keywords :** Web services, Web security, SOA

### 1. INTRODUCTION

Service-oriented architecture is an information technology approach or services wok as a bridge that provide some services to collect the information from different environment on the world wide web (WWW). Service-oriented architecture is a concept for web services is conceptual architecture is Called Service-oriented architecture (SOA).

**Figure. 1**  
Service-Oriented Architecture.



\* The material presented by the authors does not necessarily portray the viewpoint of the editors and the management of the Institute of Business and Technology (Biztek) and King Faisal University Saudi Arabia.

\* Noor Zaman : nzaman@kfu.edu.sa

\* Zahid Hussain Khuhro : zahid\_khuhro@yahoo.com

\* Zia Ahmed Shaikh : ziaahmedshaikh@gmail.com

© JICT is published by the Institute of Business and Technology (Biztek).  
Ibrahim Hydri Road, Korangi Creek, Karachi-75190, Pakistan.

Implementation of service-oriented architecture can involve developing applications that use services, making applications available as services so that other applications can use those services, or both. A service provides specific function, typically a business function, such as analyzing an individual's credit, history or processing a purchase order.

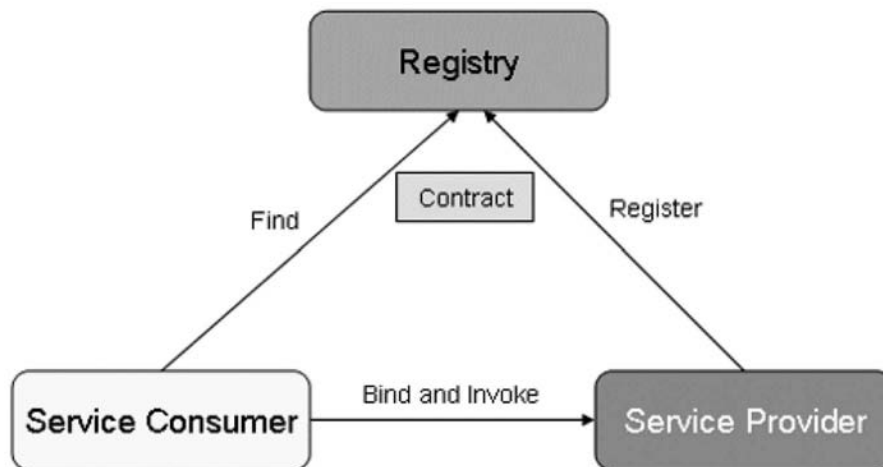
A service can provide a single discrete function, such as converting one type of currency into another, or it can perform a set of related business functions, such as handling the various operations in an airline reservations system. Services that perform a related set of business functions, as opposed to a single function, are said to be "coarse grained." Multiple services can be used together in a coordinated way. The aggregated, or composite, service can be used to satisfy a more complex business requirement. In fact, one way of looking at an SOA is as an approach to connecting applications (exposed as services) so that they can communicate with (and take advantage of) each other. In other words, service oriented architecture is a way of sharing functions (typically business functions) in a widespread and flexible way (Seros, 2008).

The concept of an SOA is not new. Service-oriented architectures have been used for years. What distinguishes an SOA from other architectures is loose coupling. Loose coupling means that the client of a Service is essentially independent of the service. The way a client (which can be another service) communicates with the service doesn't depend on the implementation of the service. Significantly, this means that the client doesn't have to know very much about the service to use it. For instance, The client doesn't need to know what language the service is coded in or what platform the service runs on. The client communicates with the service according to a specified, well-defined interface, and then leaves it up to the service implementation to perform the necessary processing (Motta, 1999). However what is relatively new is the emergence of web services-based SOAs. A web service is a service that communicates with clients through a set of standard protocols and technologies. These web services standards are implemented in platforms and products from all the major software vendors, making it possible for clients and services to communicate in a consistent way across a wide spectrum of platforms and operating environments. This universality has made web services the most prevalent approach to implementing an SOA (Deri, 2004).

## **2. ROLE OF SOA**

SOA describes a distributed application construction that has been in principle in use for many years. CORBA (Common Object Request Broker Architecture), in use since the late 80ies, promoted by the Object Management Group and mainly deployed in telecommunication applications is one example. The DCOM (Distributed Component Object Model) of Microsoft is another. Both concepts provide interface definition languages (IDL, MIDL) and tightly coupled distribution mechanisms that allow local or remote invocations. In principle, SOA allows designing applications that provide services to other applications through published and discoverable interfaces. In this sense SOA might be defined as an application architecture in which all functions are implemented as independent services with well-defined inviolable interfaces, which can be called in sequences to form business processes. However in contrast to CORBA and DCOM, SOA based on Web services constitutes a loosely coupled model, as depicted in Figure 1, provides platform independency (regarding the two camps, Java and Microsoft), as well as language, transport and message format independency (Wollrath, 1996).

**Figure. 2**  
SOA's Find-Bind-Execute Paradigm (Martin, 2003).



### 2.1 Reusability

What drives the move to SOA is reuse of business services. Developers within an enterprise and across enterprises (particularly, in business partnerships) can take the code developed for existing business applications, expose it as web services, and then reuse it to meet new business requirements.

Typically, solutions developed in different enterprises, even different departments within the same enterprise, have unique characteristics. They run in different operating environments, they're coded in different languages, they use different programming interfaces and protocols. You need to understand how and where these applications and systems run to communicate with them. The work involved in doing this analysis and the development effort in tying these pieces together can be very time consuming (Fensel, 2002).

### 2.2 Interoperability

The SOA vision of interaction between clients and loosely-coupled services means wide spread interoperability. In fact, web services provide exactly that. Web services comprise a maturing set of protocols and technologies that are widely accepted and used, and that are platform, system, and language independent (Keynot, 2008).

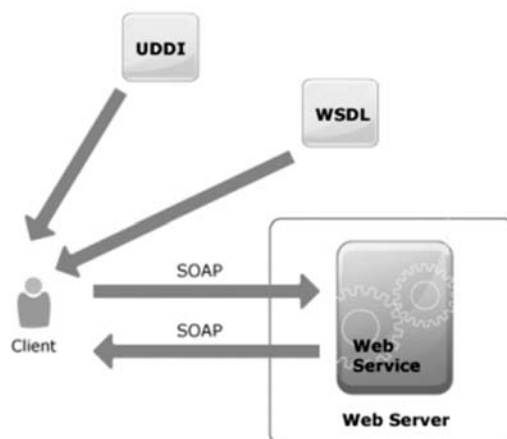
### 2.3 Scalability

Because services in an SOA are loosely coupled, applications that use these services tend to scale easily -- certainly more easily than applications in a more tightly-coupled environment. An example of a document-oriented service might be a travel agency service that accepts as input a document that contains travel information for a specific trip request. An asynchronous service performs its processing without forcing the client to wait for the processing to finish. A synchronous service forces the client to wait. The relatively limited interaction required for a client to communicate with a coarse-grained, asynchronous service, especially a service that handles a document such as a purchase order, allows applications that use these services to scale without putting a heavy communication load on the network. (<http://e-articles.info>, Aug. 2008).

## 2.4 Flexibility

Loosely-coupled services are typically more flexible than more tightly-coupled applications. In a tightly-coupled architecture, the different components of an application are tightly bound to each other, sharing semantics, libraries, and often sharing state. This makes it difficult to evolve the application to keep up with changing business requirements. The loosely-coupled, document-based, asynchronous nature of services in an SOA allows applications to be flexible, and easy to evolve with changing requirements. (<http://www.w3schools.com>, Sep 2008).

**Figure. 3**  
Web services Model (<http://java.sun.com>, Sep 2008).



## 3. ROLE OF WEB SERVICE

Web services is service based architecture, collect the information from different server on world wide web. Web service combine the different environment in information technology. Web service provide an implementation-independent way for applications to communicate with each other. Currently, many companies use electronic-data-interchange (EDI) systems to communicate with their business partners. Web services, which are based on SOAP messages that wrap XML documents, provide a flexible infrastructure that leverages the ubiquitous HTTP (or HTTPS) over TCP/IP. In addition, thanks to XML's structure and flexibility, each partner can extract only the information it needs from a message, which gives participants a great deal of freedom. For example, you can create a Web service operation that uses a Web service operation from another provider to give its consumers (also known as service requestors) information tailored to their needs. Web service operations are akin to the methods of a Java class; a provider is an entity that publishes a Web service, while the entities that use the Web service are called consumers (Wollrath, 1996).

Listing Code Method of RPC SOAP message

```
<soapenv:Envelope
xmlns:soapenv="soap_ns"
xmlns:xsd="xml_schema_ns"
xmlns:xsi="type_ns">
<soapenv:Body>
<ns1:getStockPrice
xmlns:ns1="app_ns"
```

```
soapenv:encodingStyle="encoding_ns">
<stockSymbol xsi:type="xsd:string">AAPL</stockSymbol>
</ns1:getStockPrice>
</soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope
xmlns:soapenv="soap_ns"
xmlns:xsd="xml_schema_ns"
xmlns:xsi="type_ns">
<soapenv:Body>
<ns1:customerOrder
soapenv:encodingStyle="encoding_ns"
xmlns:ns1="app_ns">
<partNumber>88</partNumber>
<description>Blue pen</description>
<quantity>250</quantity>
</item>
<item>
<partNumber>563</partNumber>
<description>Red stappler</description>
<quantity>30</quantity>
</item>
</orderInfo>
</order>
</ns1:customerOrder>
</soapenv:Body>
</soapenv:Envelope>
```

Current Web service technology allows an organization to easily integrate its systems, creating an enterprise-wide solution that leverages the work that is performed best by smaller groups within the enterprise. For example, the Payroll system is the one that should deal with an employee's compensation, while the human resources system is more appropriate for the management of vacation and sick-leave time. However, an Employee Information system should gather the information that both the Payroll and Human Resources systems contain, but should not duplicate it. Payroll and vacation information would be available through Web service operations provided by separate applications tailored to their particular objectives (Wollrath, 1996).

#### **4. CONCLUSION**

Service Oriented Architecture is one of the most popular architectural paradigms today, but without any standardized reference model. It is an architecture that provides seamless Enterprise Information Integration between loosely coupled distributed applications or services over the network. SOA does not necessarily mean Web Services, J2EE, or CORBA. In fact, these are the implementations of SOA.

#### **5. REFERENCES**

- Seros (2008) Enterprise Service Bus and an open standards-based reference architecture in a successful SOA initiative.
- Motta E (1999). "Reusable Components for Knowledge Modelling." IOS Press, Press, Amsterdam, The Netherlands.
- Deri (2004). "Web Service Modelling Ontology Project." DERI Working Drafts.
- Wollrath, A. (1996). "A Distributed Object Model for the JAVA System." Proceedings of the USENIX 1996 Conf. on Object-Oriented Technologies (COOTS), pp. 219-231.

- Martin, D. (2003). Semantic Markup for web services. " OWL-white paper 2003-12 27.
- Fensel, D., Bussler, C. (2002) "The Web Services Modelling Framework WSMF" Electronic Commerce: Research and Applications. Vol. 1. (2002). 113-137.
- Keynot. (2008) "Keynote" is an industry leader in tackling the thorny issues of capturing real end-user quality of experience stats and then providing triage for insights into where the problems might lie.  
<http://e-articles.info/> Aug. 2008  
<http://www.w3schools.com/webservices/default.asp> Sep 2008.  
<http://java.sun.com/developer/technicalArticles/WebServices/soa/> Sep 2008