Service-Oriented Architecture (SOA) and Web Services

Claudiu PIRNAU[1], Mihai Alexandru BOTEZATU[2]
[1]Politehnica University of Bucharest, Romania
[2]Romanian - American University, Romania
claude.pyr@gmail.com, mihaibotezatu@gmail.com

From a functional perspective, a service is a standalone unit, such as taking a request from an e-commerce site. By this definition, a service is an operation that can be invoked discreetly. Services can be combined to provide the functionality of a high scope software. SOA makes it easier for different software components on computers connected via a network to cooperate in order to accomplish a certain functionality. SOA comprises several dimensions that must work together effectively to be successful. The adoption of service-based technology itself will not allow organizations to achieve benefits associated with SOA.

Keywords: *SOA, Web Services, XML, WSDL.*

1 Introduction

The main advantages of the web services are:

Interoperability between applications; reusability of existing services; easy distribution of information between consumers; fast development.

The developed application is the key element but the technology used for the application server has a major impact on the results of the implemented application. A web container is a part of a server that manages servlets, Java Server Pages (JSP) as well as other web components. The technology of the web container has a very important role in determining the performance and adaptation capability of the web service components. The web services can be implemented by using different platforms. An example for this is Oracle GlassFish Server (Sun GlassFish Enterprise Server and, previously, Sun Java System Application Server) that represents a platform for delivering server side Java applications and web services. This is an application server Java EE6 certified and it represents an essential component of the Java Enterprise System, that supports the integrated development tools.

Oracle GlassFish Server provides an environment to develop and implement Java TM–XML applications as well as web services. GlassFish Server applications include standard characteristics for Java Platform, Java EE platform as well as specific characteristics for GlassFish Server. GlassFish Server includes the following modules:

- Web Module – represents a collection of servlets, EJB's, HTML pages and classes that can be implemented on a number of Java EE application servers.
- EJB Module –it's a software unit that is composed from one or more enterprise beans as well as an implementation descriptor EJB.
- Connector Module –it's a software unit with the purpose to offer portability to the EJB components for accessing the data and information system.
- Application Client Module is the client application module and it is that software unit composed of one or more classes and descriptors that have a role in implementing the client application.

- Lifecycle Module represents the module that has the functions to define the system lifecycle and it has the purpose to execute Java based tasks in the GlassFish server environment.

The principles of service-oriented architecture are independent from any provider, product or technology. According to OASIS (Organization for the Advancement of Structured Information Standards), Service Oriented Architecture (SOA) is an architectural style that supports service orientation based on a way of thinking in terms that are specific to services and development focused on services and the results of their implementation. As the name suggests, SOA also involves architecture [1-5]. The results of adoption and promotion of SOA capabilities allow:

- Increase speed of reaction of the users: service enables implementation of flexible systems, and architecture cantered on business and technology enables impact of changes to be isolated and business processes to be modified more easily and quickly in order to meet performance requirements;
- Simplify the delivery of enhanced services: SOA and business models based on "services" allow effective management of cooperation by simplifying access to services and value streamlined chains beyond organizational boundaries;
- Streamlining business administration: TO facilitate investment leading to reduced leverage level of public and private sector (legacy of the economic crisis) through a model centered on the reuse of existing capacities, eliminate undesired redundancies and existence of visionary project architects in the IT field;
- Share information: SOA offers a new and effective approach to implementing reusable data exchange, logic interoperability from multiple activities of faster data modeling evolving towards physical interoperability;
- Transparency, security and resiliency in operation: SOA has an infrastructure built on performant standards enabling consolidation, simplification and optimization of IT infrastructure, which in turn will allow a greater transparency and auditing capacity, as well as improving business continuity [6-10].

2. SOA architecture and design

SOA architectures depend on data and services, which are described by metadata that should meet the following criteria:

- The metadata may be used by other applications, such as a catalog service to perform automatic discovery of services without changing the functionality of a service contract;
- Metadata should be provided in a form in which system designers can understand and manage services at a reasonable cost and effort.

SOA services require independent connection to operation systems and other technologies that underlie applications.

SOA separates functions into distinct units or services, which developers make available through a network in order to allow users to combine and to re-use them in making applications. These services and their consumers communicate with each other by passing data into a common well-defined format, or by coordinating an

activity between two or more services, and these include the following [11-14]:

- Services of reusability - logic is divided into services with the intention of re-using them.
- Autonomy of services - services have control over encapsulated logic from design time and runtime.
- Services status - minimizing resource consumption produced by management of status information when needed.
- Contract of standardized services - services adhere to an agreement of communication and are defined collectively by one or more descriptors.
- Loose coupling service - service maintain a relationship that minimizes dependencies and only requires that they maintain a reference to the other.
- Abstraction of services - beyond their description in the service contract, services hide logic from the outside world.
- Discovery services - services are complemented with metadata through which they can be traced effectively and interpreted by different service registries.
- Services composability - services are participants to the composition of other services, regardless of size and complexity of composition.
- Service granularity - as design consideration it ensures the purpose and optimal level of granularity necessary to provide functionality in a service operation.
- In some cases, services are deformalized for different purposes (optimize performance, access, aggregation, etc).
- Services optimization - quite obvious that for effective implementation are preferred high-quality services to those of inferior quality.
- Services relevance - functionality should be present at a level of granularity recognized by the user as a significant service for him[13-14].

In the case of a reference model for SOA,OASIS has a standard Reference Model for SOA, shown in Figure 1, which is not directly tied to any standards, technologies, or other concrete implementation details.

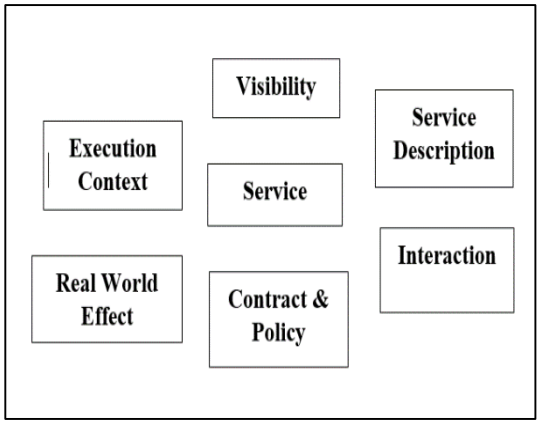

Fig. 1. The core OASIS Reference Model for Service Oriented Architecture

3. Approach of web-services

These services may be either new applications or encapsulate existing old systems to make them active on the network. Each SOA block can play one or both of the following two roles: service provider and /or service consumer according to Figure 2.

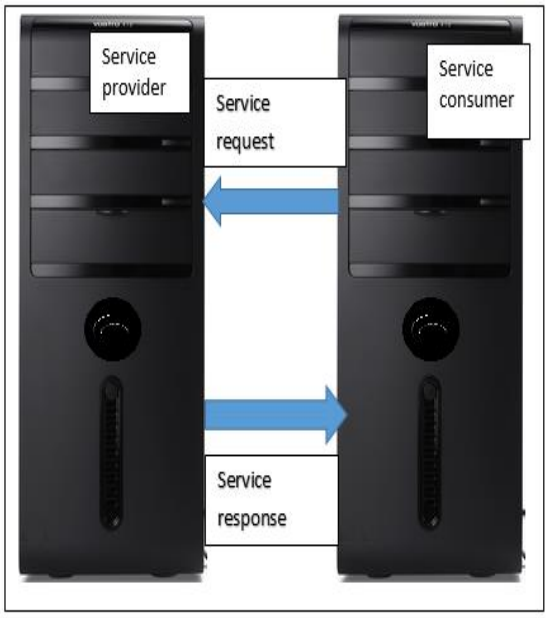

Fig. 2. Web Services

Service provider- the service provider creates a web service and, possibly, publishes interface and access information to the service register [10]. Each provider should decide what services exposes, how to compromise between security and easy availability, how to easily charge services, or (if no fee applies) how/whether to exploit them for other value.

Also, the supplier has the obligation to decide in which category services should be listed for a broker type service and what kind of partner agreements is the service required to use. Supplier records what services are available in the register and list all potential recipients of services. The implementing party then decides on broker incidence [12-15].

Public brokers are available on Internet, while private brokers are available only to a restricted public, such as the company intranet users. In addition, the amount of information provided should be decided from the beginning.

Service consumer - regardless of what service the service consumers want, they need to take it from brokers' register making a bind with the respective service and then using it. They can access multiple services if the service broker offers multiple services.

In the future you can deploy A Virtual Cloud Computing Provider for Mobile Devices [16]. Web Services Description Language (WSDL) is a format for describing Web services specifications. It is a way to describe services and how they should be related to specific network elements. WSDL have three elements: definitions, operations and services connections (bindings). Definitions are generally expressed in XML and include both data type definitions and definitions of messages, which use definitions of data types.

XML represents a condensed form for Standard Generalized Markup Language (SGML) that allows developers to create customised labels and provides flexibility in presenting information. XML is composed of 2 metalanguages, both described in the same document.

The first language is a set of rules for developing XML documents, and the second represents a set of rules to develop definitions for the type of the XML document, or DTD (Document Type Definition) with the purpose to allow validation of the XML document according to defined constrains.

Basically, XML has a dual nature, as a metalanguage that allows the description of new document and vocabulary structures as well as a language used to express the structure and vocabulary for a document. An XML document contains text, that is

usually composed of a number of marks and character data.

Vocabulary within an organization could be designed especially for that organization. It may or may not be based on an industry-wide vocabulary. If the definitions of the types of data and messages should be shared between organizations, then most likely an industry-wide vocabulary shall be used [11].

However XML language is not absolutely necessary for definitions. For example, OMG Interface Definition Language (IDL) could be used instead of XML. If a different definition language were used, senders and recipients should agree on the language and vocabulary. However, over time, vocabulary and XML-based messages have dominated. XML namespaces are used to ensure uniqueness between XML element names in definitions, operations, and service connections [17-21].

4. Metadata management

Metadata management includes information description about a Web service needed to build the body of the message (including data types and structures) and message headers, so that a service consumer may invoke a service. Public service provider posts metadata to enable consumer discovery and use them so as to build messages, which can be successfully processed by the supplier [12].

So when consumer invokes a service, it is important for him to understand not only the types and structures of data to send, but also understand additional qualities of the services provided, such as security, reliability, or transaction capacity. If one or more of these features are missing from the message, this can hinder the successful processing of the message.

Specifications for metadata include:

- XML-Schema - for the expression of data and more important for structuring and expressing a policy.
- WSDL – for the combination of messages and message exchange patterns with names and addresses of network services.
- WS-Addressing – for addressing an endpoint and reference properties associated to that endpoint. Many of the other extended specifications require WS-Addressing to define endpoints and reference properties in models of communication [7][19].
- WS-Policy - for associating quality of services with a WSDL definition. WS-Policy is a framework that includes policy type statements for various aspects of security, transaction capacity and reliability.
- WS-Metadata Exchange - to interrogate and discover metadata associated with a Web service, including the ability to bring a WSDL file and associated WS-Policy [22-25].

5. Ensure SOA security

Ensuring SOA security in an ecosystem of covered regulated services has implications over distributed policy and nature of the mechanisms used to secure SOA [8].

Security expressed through security policy messages should follow the same architectural implications for policies and contracts involved. Security policies should have support, storage, and distribution mechanisms for describing them. The descriptors of services should include a sufficiently rich meta-structure to indicate clearly which security policies are

necessary and where policy options are possible.

The mechanisms that make up the context of execution in SOA-based systems should: protect privacy and integrity of message exchanges; messages should be distributed so as to ensure the availability of policy-based identification, authentication and authorization; ensure the availability of services for consumers; be able to scale to support safety services to a growing ecosystem of services; be able to support security between different means or channels of communication; contain a framework for resolving conflicts between security policies.

Security issues apply to every level of SOA stacks and require a variety of mechanisms of protection against the many challenges and threats coming (some of them) from distributed computing architectures.

6. Conclusions

Most times it is necessary for protection mechanisms to be used in combination with specific solutions to protect against a specific threat or combination of threats.

In the world of Web services and SOAs it is particularly important to assess the need for protection at network and transfer of messages level, as well as data contained in the message [26].

Basic security mechanisms are built around encryption, authentication mechanisms, and authorization mechanism and usually they include extensive logging and audit methods to detect potential problems. Industry has reached consensus around a single framework of specifications, WS-Security, although efforts continue to complete the profiles and related additional specifications.

Providing specialized experience often requires intensive interaction of data and introduces new challenges in managing data between client and server levels. Data synchronization is a key concept and requires users to work within a distributed system. Such services allow developers to focus on applications on business logic instead of bothering with infrastructure issues.

Modern service orientation and server-based infrastructures enable automatic management of temporary disconnection, ensuring secure delivery of data to and from client application.

Some services offer data-push functions, allowing data to be pushed automatically to the client application, without a prior query.

This can be achieved through intuitive or deductive methods to ensure provision of data according to customer requirements. This very scalable capability allows to push simultaneously data to thousands of users, for example applications of stock traders, human resource monitoring or automating various logistical activities.

7. Reference

[1] A. Singhal, T. Wingrad, K. Scarfone, NIST Guide to Secure Web Services, National Institute of Standards and Technology, 2007

[2] E.Thomas, Service-Oriented Architecture. Concepts, Technology and Design, Pearson Education, Inc. ISBN 0-13-185858-0, USA, 2005.

[3] D. Nickul, L.Reitman, J.Ward, J.Wilber, Service Oriented Architecture (SOA) and Specialized Messaging Patterns. Technical White Paper. Adobe Systems Incorporated, USA, 2007.

[4] N.Endrei, J.Ang, A.Arsanjani, S.Chua,

Patterns: Service-Oriented Architecture and Web Services, IBM - International Technical Support Organization, April 2004

[5] M. Pîrnău, C. Pîrnău, M. Apetrei, G. Badea, The SOAP protocol used for building and testing Web services, Proceedings of the World Congress on Engineering 2011, Vo l IWCE 2011, 2011.

[6] T. Soddemann, Web Services and Service Oriented Architectures, The Garching Supercomputing Center – RZG, Germany, Delaman Workshop 2004

[7] P. Bianco, G. Lewis, P. Merson, Service Level Agreements in Service-Oriented Architecture Environments. Technical Report CMU/SEI-2008-TN-021, CMU/SEI (2008)

[8] N.M. Villegas, H.A. Muller, G. Tamura, Optimizing Run-Time SOA Governance through Context-Driven SLAs and Dynamic Monitoring.In: 2011 IEEE International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA 2011), IEEE (2011)

[9] https://www.oasis-open.org/

[10] http://www-935.ibm.com/services/us/cio/pdf/ciw03035usen.pdf

[11] https://msdn.microsoft.com/en-us/library/bb833022.aspx

[12] https://msdn.microsoft.com/en-us/library/bb972954.aspx

[13] http://docs.oracle.com/cd/E17904_01/doc.1111/e15020/introduction.htm#OSBCA107

[14] http://www.oracle.com/technetwork/articles/cloudcomp/keyes-cio-guide-to-oracle-chap5-2262726.pdf

[15] http://docs.oracle.com/cd/E23943_01/dev.1111/e15866/ws_policy.htm#OSBDV1599

[16] Barca Cristian, Barca Claudiu, Cucu Cristian, Gavriloaia Mariuca-Roxana, Vizireanu Radu, Fratu Octavian , Halunga Simona - A Virtual Cloud Computing Provider for Mobile Devices. Proceedings of the International Conference ECAI-2016

[17] A. Oluwaseyi, Service Oriented Architecture & Web Services Guidelines for Migrating from Legacy Systems and Financial Consideration, Master Thesis in Computer Science, School of Engineering at Blekinge Institute of Technology, Sweden, 2008.

[18] N.M. Iacob, “Information security for web and SQL services”, Proceedings of the 9th International Conference on Virtual Learning, Proceedings of ICVL 2014 (ISSN 1844-8933)

[19] [20] R.T. Watson, Information Systems, University of Georgia, Global Text Project, 2007

[20] https://msdn.microsoft.com/en-us/library/bb833022.aspx

[21] http://www.w3.org/TR/wsdl

[22] https://en.wikipedia.org/wiki/Service-oriented_architecture

[23] W3C, Web Services Description Language (WSDL) 1.1 - W3C Note, W3C, March 2001, http://www.w3.org/TR/wsdl.

[24] https://sites.google.com/site/spkinsight/soainreality

[25] https://www.w3.org/Submission/ws-addressing/

[26] http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-95.pdf

Claudiu PÎRNĂU graduated from Politehnica University of Bucharest, Faculty of Engineering and Management of Technological Systems in 1989, and holds a master degree in Management Information Systems since 2007 and a PhD in the field of Engineering and Management since 2015. I am an engineer in the Faculty of Engineering and Management of Technological Systems, having eleven years of engineering experience and twenty years of teaching experience. I am member of different scientific and educational organisations and I was a member of four national projects too. My specific area of competences is in web technologies, databases, knowledge management, creativity and sustainable development.

Mihai Alexandru BOTEZATU graduated from Echological University of Bucharest, Law Faculty, in1999, as licensed legal counselor; he graduated as well from the Romanian-American University, European Integration Economic Studies Faculty in 2004, as licensed economist.

He has a PhD in the field of Economy - Cybernetics and Economic Statistics, from the Economic Studies Academy, Bucharest, since 2010. He graduated a special course for programming specialization as programming analyst.

He has extensive experience in management and coordination activities, both in public bodies and in private companies of different types. He has extensive experience in management and development of European funded projects, but he performed applicative research as well within research projects, as an expert and associated member of Romanian-American University.

He has a rich publishing activity, with more than 15 articles presented in Scientific Conferences that are ISI indexed and/or published in well-known scientific magazines that are indexed in international databases. He is the author of the manual for **IT Projects Management** and member in 3 research teams financed through research projects; one of them is an international project.