# Exposing Web Services to Business Partners: Security and Quality of Service Issue

Yann Le Blevec[1], Chirine Ghedira[1], Djamal Benslimane[1], Xavier Delatte[2] and Zahi Jarir[3]

1 - Liris Laboratory, Lyon I University, Villeurbanne, France {firstname.lastname@liris.cnrs.fr}; 2 - AVT, Annecy, France {xavier.delatte@adixen.fr}; 3 - Cadi Ayyad University, Marrakech, Morocco {zahijarir@ucam.ac.ma}

***Abstract* — Delivering QoS and ensuring security of Web Services (WS) based architecture is critical and constitutes a significant challenge because of its dynamic and unpredictable nature. This paper provides solutions to manage security and QoS problems when exposing Web Services to business partners. Based on a real case study, we first discuss the security issue in the context of an open industrial information system. Then an approach is presented to deal with the QoS issue in such systems.**

***Index Terms* — Information systems, Web services, Security, QoS, B2B, SLA.**

## I. Introduction

The web services (WS) paradigm is now established as the technology that aims at ensuring interoperability between web applications in heterogeneous and geographically distributed environment. Therefore, WS could support business processes across business partners. Of course, not whichever WS may be used. The business context of communication between partners (B2B) generates many constraints and requirements especially *security, reliability* and *trust*. As a consequence, elderly web services that are only built over SOAP, WSDL and UDDI aren't satisfying anymore.

Next generation services (advanced services or enterprise services) must be implemented with enhanced security and quality of service (QoS) features. This way, a service can be defined not only by its functionality but also by how the service is delivered. However, providing advanced services is a difficult task for an enterprise: many standards and products are available. Therefore, QoS and Security issues create a strong need for tools, and methodologies to support the design and the management of WS based architectures.

While creating and exposing services to partners, the enterprise should not lose focus on the main benefits of services in order to perennialize its investments: *flexibility*, *scalability* and *openness*. Flexibility means the possibility for the business processes and the information system to evolve easily in the future. Scalability aims at not increasing the complexity of the adopted solution while adding new partner organizations. Openness requires the enterprise to privilege technically loosely coupled systems and open standards.

Moreover, real enterprises have real constraints. An enterprise must leverage its existing infrastructure: legacy systems may cause strong technical constraints for future projects. The exposition must also follow global procedures, rules and policies of governance. Finally, the selected solution must be adapted to the enterprise's size and to its business objectives without being too expensive.

The remainder of this paper is organized as follow. Section 2 gives a short background of security and QoS concepts. Section 3 provides potential security solutions coupled with a technical discussion. Section 4 is focused on the QoS management architecture. Section 5 discusses about related work. Finally, we conclude and present future work.

## II. Background

### A. Web Service Security

An obvious fact is that security issue is considered as an important factor in any real business applications. It then becomes a critical factor when the application is accessible over a public network. The use of web services for B2B communication allows partners through a public network to access automatically functionalities that may interact with the backend systems of an enterprise. As a consequence the exposition of WS constitutes a huge threat for the information system of an enterprise. We focus in this paper on the danger incurred by the exposition of web services over an unsecured network.

WS must face three sources of threats. The first one is about exposing a software component. Every software component or system may have security faults. As a consequence WS are subjected to the security faults of the applications and devices used for their implementation. The second issue is related to the use of XML. Attackers may use XML messages (SQL injection, malicious schema, heavy SOAP attachments) to damage exposed systems. Finally the last threat deals with the

communication of information between business partners over an unsecured network, such as Internet. The content of exchanged messages may be sensitive and must be protected from attackers (e.g. man-in-the-middle attack).

As presented in Fig. 1, we will briefly introduce different layers of security measures for WS:

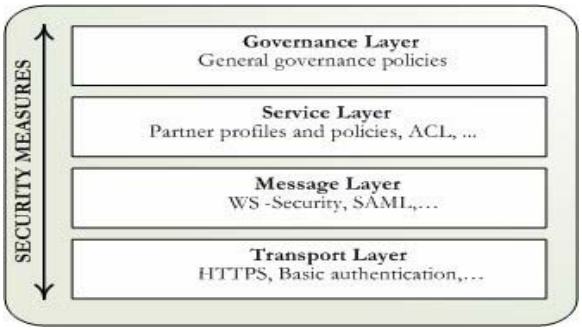


Fig. 1. Layers of security measures for WS

- *Transport Layer*: Depending on the chosen transport protocol different security measures may be applied on the transport layer. For example, with HTTP, the invocation of a service may be controlled by a certificate or a basic authentication (login/password). HTTPS may also protect data between two points (point-to-point tunneling).
- *Message Layer*: The W3C consortium has proposed three core web services security standards for XML [4]: XML Encryption, XML Digital Signature and XML Key Management System. OASIS relied on these works to create several standards. Since 2002, WS Security [1] has provided a way to insert XML security standards in SOAP messages. With SAML, OASIS also defines a XML schema that allows trust assertions (authentication, authorization or attribute) representation in XML and request/response protocols to perform authorization, XML authentication, and attribute assertion requests.
- *Service Layer*: This layer specifies which security measures are to be used for a specific service. WSDL doesn't have such functionality. WS-Policy [11] provides a generic syntax to express such requirements. Moreover, service level security also matters who has the authorization to use a service: the enterprise must maintain business partner profiles and Access Control List (ACL). XACML, also proposed by OASIS, defines an XML vocabulary for specifying the rules from which access control decisions can be enforced.
- *Governance layer*: Finally, the governance layer specifies general guidance and global policies. As an example, certified products and procedures for WS lifecycle management could be listed here. These measures are detailed in written documents. No automation is required for the governance layer because of its strategic and long term nature.

To decide which security measures should be used at which level, a threat analysis must be conducted. Afterwards, depending on the existing applications *(the constraints*), the level of security required (*the needs*) and the IT budget (*the money*), a coherent solution may be elaborated. However, one should not forget that B2B main issue is the interoperability: semantic, technical, and process. The final solution must be based on widely accepted and mature standards to ensure technical interoperability. Exotic specifications and home-made developments should be prohibited. This statement is not true when using WS for internal communication (SOA).

To ensure an optimal use of WS for B2B, we also have to deal with QoS issues as described in the next section.

### *B. Quality of Service*

The very first step is to define quality for WS. In [6], quality is expressed referring to observable parameters, relating to non-functional property (e.g. response time). Early, the literature [8] has identified the importance for services not only to declare their functionalities but to formalize their non-functional properties.

**1. QoS criteria for web services.** The more common criteria are related to the execution of the services and come from the domain of networks: *Availability, Reliability, Cost, and Execution tim*e. Thanks to [9], [14], a more complete list can be built: *Availability, Scalability, Accessibility, Integrity, Robustness, Accuracy, Performance, Reliability,* and *Regulatory*. By definition this list of criteria is not exhaustive and cannot apply to every service. What is functional for a service or a domain may be considered as non-functional for another service or domain. As a consequence the list of selected criteria may be rearranged for every service [15].

Finally, other works have tried to create new criteria. In [12], the widely-used concept of reputation was modified so that users' evaluations are pondered by the conformity of the service and the results of previous executions ("objective" reputation).

**2. QoS modelization.** An explicit definition is particularly important in an environment transcending organizational boundaries. Therefore, three levels of modelization have been studied through the literature to improve QoS qualifications: ontologies and taxonomies for the concept of quality of service, languages for consumers and providers to express level of QoS, and models to evaluate the quality of a service.

Works like [5] have presented ontologies to define more precisely the concept of quality of service. Another approach presented in [3] was to define taxonomy for quality of service in the context of web applications. Even, UML has specified a profile to model the QoS.

Moreover, WS-Policy and its extensions allow consumers to express the expected level of QoS and providers to declare their level of QoS. WSLA [7]

proposed by IBM and its "successor" WS-Agreement also define languages to provide such functionalities. Meanwhile, others works have decided to use their own language regardless of existing standards. XML increases the level of interoperability of these languages.

Finally, [5] and [15] have proposed an extensible model of quality of service that can evaluate the quality of a service regardless of the selected criteria.

**3. Metric acquisition.** In order to evaluate the QoS criteria, many measurements must be performed first. However, the modality of the measurement is difficult to determine, and to describe without ambiguity [6]. Three questions need to be answered: Who is in charge of the measurements (provider, requester or third party)? How is the measurement performed (e.g. time between two measures)? Can a measurement be trusted?

Both security and QoS are very active domains. The next two parts show how an enterprise may deal with these works to expose advanced services to business partners.

## III. WS SECURITY ISSUE

### *A. Network constraints*

By offering Web services to business partners, we try to increase the openness of an information system. The information system of an enterprise is always located in a private network. This network is often considered as a trusted and secured environment. As a consequence, every access point from a public network (such as Internet) to the company private network must be strictly controlled. Therefore, Demilitarized Zones (DMZ) are used to safeguard this private network. Logically, administrators try to lower the number of DMZ and to limit the number of applications that are hosted within a DMZ. A DMZ is a complex zone composed by firewalls, proxies and tight governance policies. An enterprise may choose to externalize the administration of a DMZ to a third party.

The aim of any enterprise when exposing WS is to prevent its information system from all threats by the use of a secure architecture. Therefore, the information system of an enterprise must evolve with the use of new B2B components and new governance policies.

### *B. A case study*

In our case of study, the enterprise belongs to a group[1]. Therefore, it doesn't control the boundaries of its private network. Every connection with the outside world must validated by the group that manages all DMZ.

**1. Architecture.** Concerning the DMZ, we had three possibilities (Table 1). The first one (1) was to let the enterprise implement its own DMZ and take all responsibilities. The second possibility (2) was to deport the enterprise B2B component in the group DMZ. This leads to shared responsibilities: the group is in charge of the infrastructure whereas the enterprise provides the business logic. Finally, solution (3) was to reuse an existing middleware previously deployed by the group. The group would be responsible for the exposition of WS.

TABLE 1
BENEFITS AND DRAWBACKS FOR EACH POSSIBLE SOLUTION

| # | Benefits | Drawbacks |
|---|---|---|
| **1** | + Highly fexible solution for the enterprise | - Missing skills within the enterprise<br>- Expensive solution<br>- Group policies' violation |
| **2** | + Lot of flexibility for the enterprise<br>+ Short path followed by XML messages<br>+ Product customizable by the enterprise | - Enterprise is responsible of all security issues<br>- The product must be validated by the group<br>- New products for the group in the DMZ |
| **3** | + Leverage existing infrastructure and people expertise<br>+ No new entry point for the private network | - No flexibility for the enterprise<br>- Long path followed by the messages<br>- Solution already deployed |

The chosen solution was to reuse the existing infrastructure (3) as described in Fig. 2. The reverse proxy and the integration are already operational and under the responsibility of the group. The main goal of the reverse proxy is to publish web services to distant business partners over the Internet. It should filter incoming requests and forward only the correct ones to the central integration server. Then the integration server is responsible for the correct routing of incoming messages to the appropriate backend web services.

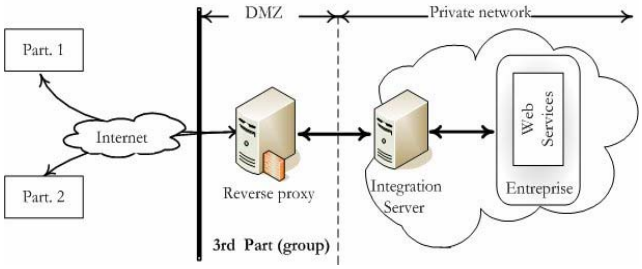


Fig. 2. Sample of architecture exposing WS over internet

The decision to choose this solution was taken to quickly answer the need of the enterprise and to leverage the existing infrastructure. The major problem of solution

[1] Alcatel Vacuum Technology is the enterprise where several web services were exposed. This SMB is a subsidiary of the Alcatel Group.

2 was its lack of reusability. What if another subsidiary of the group wants to expose WS? With the solution 3, the group will have the same answer to this subsidiary. Whereas with the second solution the group may have to add another product to its DMZ leading to obvious scalability issue. As a conclusion, among all possible solutions, the choice made seems to be a fair one.

**2. Adopted security measures.** At *transport level*, the protocol used between the enterprise information system and its partners is HTTPS with client/server certificate authentication. Each business partner is required to have a valid certificate and must provide it to the group at design time. Thus the group must maintain a PKI. HTTPS guarantees the integrity of the message from its emission to its reception. Thanks to certificate, both parties are clearly identified. At *message level,* the requesting partner must insert its DUNS number in the SOAP message. This measure will help the integration server to check the partner identity. At *service level*, business profiles and access control lists must be provided by the enterprise to the group. The group is then able to reconfigure its platforms. At run time, when an incoming message arrives in the integration server a check is automatically performed to validate if the identified business partner (certificate + DUNS) is granted to access the requested service (ACL). Finally, at *governance level*, the group doesn't modify any policies.

### *C. Discussions*

Several remarks must be formulated to highlight important issues addressed by this case study.

**1. Relying to much on transport level measures.** In this case study, data protection and authentication rely obviously on the transport layer with HTTPS.

However, with XML messaging and WS, one can expect security to be related to the content of the message, adapted to architectures with n-nodes (communication platforms, backend business systems,…) and ready for message-style communication. What if we want to implement different levels of security messages depending on the criticality of the message? What if we want to encrypt only a specific part of the message? HTTPS brings only a global solution. The thin granularity available with XML structures is not used at all. Moreover, for each HTTPS connection, an autonomous authentication is required that will check the requester's credentials. Authentications can't be mutualized to create a Single-Sign-On architecture (SSO). Finally, because of the importance of HTTPS, this solution is tidily coupled to HTTP. It would then be difficult to use another protocol.

**2. Missing security measures at message level.** Thanks to several new security standards and recommendations (Table 2.), many of the problems pointed out in the previous paragraph can be solved. With WS-Security [1], each message sent by the partner can be digitally signed by the sender to guarantee that its content wasn't modified. XML encryption can encrypt selected XML nodes and protect data from being read by unauthorized people.

SAML authentication statement can also be inserted into XML messages. If a SAML *Authentication Statement* is present in the XML message, applications can identify the client and how and when the client was authenticated.

TABLE 2
MESSAGE LEVEL SECURITY MEASURES (FROM SAP©)

| | **XML encryption** | **XML signature** | **User token** | **X. 509 token** | **Time stamp** |
|---|---|---|---|---|---|
| Tampering | M | H | | | |
| Eaves dropping | H | | | | |
| Spoofing | | | M | H | |
| Repudiation | | M/H | | | |
| Mess. Replay | | | | | M/H |

Legend: L/M/H= low/medium/high protection

**3. No automation at service level.** In this case study, the enterprise has the business knowledge and the group has the infrastructure. Right now, the enterprise shares manually its knowledge with the group and does not have any mean to modify automatically the exposition of its services. In a near future, internal services could be provided by the group in order to manage the lifecycle of exposed services: open/close the exposition of a service, modify its configuration or its ACL. The enterprise would also benefit from other services like periodical reports about the activity of the services observed by the group.

**4. Evolution of the architecture**. To improve the exposition of WS, the architecture deployed by the group should evolve. However, one should not forget two basic constraints. From an economical point of view, an enterprise cannot invest each year on the new trendy technology. From a technical point of view, high risk zones such as DMZ require stability. For these reasons, and because such issues deals with governance policies, migration projects in DMZ are always long-term projects.

Nevertheless, the group believes that WS will be more and more used in tomorrow B2B communications. Among all possibility, the integration server could be replaced by a standalone reverse proxy.

As stated before business transactions need not only security, but also reliability, and trust. Next section evaluate the QoS of a WS.

## IV. QUALITY OF SERVICE ISSUES

Many factors degrade performance of delivering QoS of web service to the clients. This degradation may come

from Web environment and/or Provider environment. Web environment implication concerns the changes in traffic patterns, the denial-of-service attacks, the effects of infrastructure failures, the low performance of Web protocols, the security issues over the Web and the bandwidth degradation. Whereas Provider environment concerns for example the number of connection, the offered service, the server hosting platform, server computational resources…

In a competitive business environment, QoS becomes a main key for differentiating between providers offering similar WS. A formal contract, called Service Level Agreement SLA, may be defined to establish more precisely all bounds on various QoS metrics of the offered service. It defines mutual understandings and expectations of the provider and consumers for a particular service. The service guarantees are about what transactions need to be executed and how they should be executed. Parties may also define rewards or penalties for each objective.

Therefore, providers are interested in gaining a good understanding of the relationship between what they can promise in an SLA and what their IT infrastructure is capable of delivering.

To select the best WS according to the user QoS requirements, there is a clear need to classify firstly these WS according to the offered QoS, and secondly to validate their proposed QoS parameter using measurement.

### *A. QoS properties for Web Services*

WSDL only describes the syntactic signature for a Web service and does not specify any semantics aspect such as QoS. In this direction, several approaches are proposed by integrating WS QoS constraints and properties: DAML-S, WSLA [7], UDDIe [2], QoS proxy, QoS Broker, etc. These approaches consist in selecting the required WS according to the user's QoS requirements. However less attention has been paid to:

- Control and validate the QoS declared in SLA of selected WS continually during the session,
- Save all violations of SLA in a database and notify both the client and the provider,
- Manage the provided QoS, the WS Environment and the Web Environment (end-to-end QoS).

To fulfil those requirements, we propose in the next section a third party component to resolve the QoS issues.

### *B. General description of our approach*

Our approach as depicted in Fig. 3 is based on the main component QoS Manager, which takes into account the following features:

- Retrieve WS according to the user's QoS requirements described in the QoS contract. Contracts are registered in a database while WS providers publish their WS in a UDDI registry,
- Compare the offered QoS with the required end-to-end QoS by calling a WS, named Service Level Management (SLM), which consists in performing continually metrics and notifying the QoS Manager if some constraints are violated. SLM is a collection of probes that measures QoS provided between a WS and the user location,
- Bind user application with the selected WS, Notify first the provider if the QoS performance is in a degradation state, in order to anticipate the readjustment of the violated properties. If QoS constraint is still degrading, QoS Manager should also notify the client.
- Maintain and adapt the consumption of WS, in case of QoS constraints violation, according to the user permission. The adaptation is made by finding optimal situations of the interaction between WS and user location according to available computational resource and the end-to-end QoS.

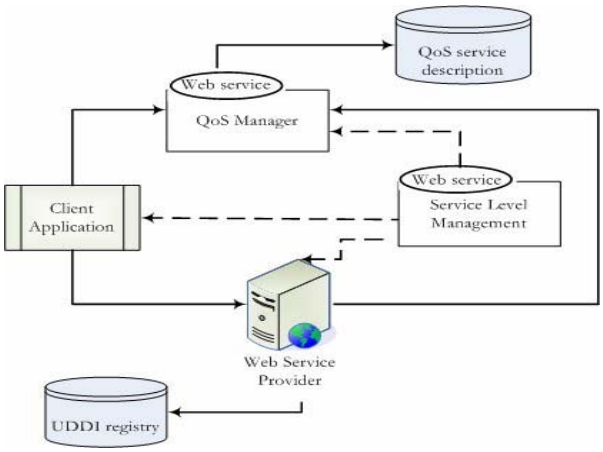


Fig. 3. QoS management

## V. RELATED WORK

In security, the use of policies is usually geared towards the specification of security mechanisms that must ensure authentication, message privacy, and authorization of Web Services. We report in this chapter such efforts.

Many languages for policy specification exist such as the Web Service Policy Framework (WS-Policy) [11]. A WS-Policy specification defines a syntax and semantics for service providers and service requestors to describe their requirements, preferences, and capabilities. The syntax provides a flexible and concise way of expressing the needs of each domain for policies.

Other service-specific policies have been proposed. Privacy policies discussed in [16] are an example. Yee and Korba propose a privacy policy negotiation approach to protect privacy of Web services users. Along the same direction, Indrakanti and al. use the XML Access Control

Language (XACL) to specify authorization policies for patient records in healthcare systems implemented as WS..

To support Web service QoS according to SOA, [17] classifies current approaches into three main concerns: extending UDDI registry, extending the format of SOAP messages or QoS-Enabled Web Service Certification [13]. Extending UDDI means that UDDI registry may be advanced to publish QoS-enabled web services by extending UDDI functionality to describe specific QoS information of a web service. [2] proposes extension within UDDI, called UDDIe, by integrating QoS descriptions and search-operations capabilities. Other approaches consist in extending UDDI functionality outside UDDI registry by adding a third party QoS broker to SOA. [10] is an example of this category. This work proposes an improved UDDI model for dynamic QoS based interactive service choice-making. This model is based on configurable fuzzy synthetic evaluation system, which evaluates web service QoS dynamically according to the service context. Other researches propose a new infrastructure for specifying QoS issues associated with Web services. An example is WSOL project (Web service offering language) [14]. The targets of this project are the creation of service offerings, definition of QoS constraints, management statements, reusability, and a mechanism for switching between services called Service Offering Dynamic Relationship (SODR).

## VI. CONCLUSION

Beginning Business partnerships and enterprise processes can significantly profit from the interoperability offered by web services. However enterprises require advanced WS with built-in quality of Service and security features. We propose, in this paper, a solution to resolve the security issues involved by the communications between WS, based on widely accepted standards. Our approach consists in a middleware component named Integration Server, coupled with a frontal reverse proxy, that is responsible of exposing specific services from an internal network to the Internet. Furthermore, we designed the mainlines of an architecture that aims to manage the QoS of Web Services and to select services according to the consumers' needs.

We are looking forward to implement our QoS solution in order to validate the feasibility of the ideas presented in this paper. However, several key points must be solved first. How to design business contracts for services? Which standards, languages and protocols should be used to support our QoS architecture?

## REFERENCES


[1] A. Nadalin, C. Kaler, P. Hallam-Baker, R. Monzillo,"Web Services Security: SOAP Message security 1.0", OASIS Standard 200401, 2004.

[2] A. ShaikhAli, O. Rana, R. Al-Ali., D.W. Walker, "UDDIe: An Extended Registry for Web Services"; Proceedings of the Service Oriented Computing: Models, Architectures and Applications, SAINT-2003 IEEE Computer Society Press. Orlando, USA, 2003.

[3] B. Sabata, S. Chatterjee, M. Davis, J. Sydir, T. Lawrence, "Taxonomy for QoS Specifications," Proceedings of the IEEE CS 3rd Int. Workshop on Object-oriented Real-time Dependable Systems, Newport Beach, CA, 1997.

[4] C. Gutierez, E. Fernandez-Medina, M. Piattini, « Web service Security : is the problem solved ? », Proceedings of the second International Workshop on Security in Information Systems, WOSIS 2004, In Conjunction with ICEIS 2004, Porto, 2004.

[5] E.M. Maximilien, M.P. Singh, "Toward autonomic web services trust and selection", Proceedings of the 2nd Int. Conference on Service Oriented Computing ICSOC '04, ACM Press, New York, NY, 2004, pp. 212-221.

[6] H. Ludwig, "Web Services QoS: External SLAs and Internal Policies - Or: How do we deliver what we promise?", Keynote Speech at the WISE Workshop on Web Services Quality, Rome, 2003.

[7] IBM, "Web Service Level Agreement (WSLA) Language Specification Version 1.0", http://www.research.ibm.com/

[8] J. O'Sullivan, D. Edmond, A. Ter Hofstede, "What's in a service? Towards accurate description of non-functional service properties", Distributed and Parallel Databases. Vol. 12, 2003, pp. 117-133.

[9] M. Anbazhagan, A. Nagarajan," Understanding quality of service for Web services", IBM Developerworks website, http://www.ibm.com/developerworks

[10] Mou Yu-Jie, Cao Jian, Zhang Shen-Sheng, Zhang Jian-Hong, "Interactive Web service choice-making based on extended QoS model", Journal of Zhejiang University SCIENCE, Vol. 7, No. 4, 2006.

[11] P. Nolan. "Understand WS-Policy Processing". Technical Report, IBM Corporation, 2004

[12] S. Kalepu, S. Krishnaswamy, S-W. Loke, "Reputation = f(User Ranking, Compliance, Verity)", Proceedings of the IEEE Int. Conference on Web Services, 2004.

[13] Shuping Ran, "A Model for Web Services Discovery With QoS", ACM SIGecom Exchanges, 4(1), 2003

[14] V. Tosic, K. Patel, B. Pagurek, "WSOL – A Language for the Formal Specification of Classes of Service for Web Services", Proceedings of the 2003 Int. Conference on Web Services ICWS'03, CSREA Press, Las Vegas, 2003.

[15] Y. Liu, A. H. Ngu, L.Z. Zeng, "QoS computation and policing in dynamic web service selection", Proceedings of the 13th Int. WWW Conference on Alternate Track, New York, NY, 2004.

[16] Yee G., and L. Korba; "Privacy Policy Compliance for Web services", Proceedings of the IEEE Int. Conference on Web Servives ICWS'04, San Diago, CA, USA, 2004