Securing Collaborative Business Processes: A Methodology for Security Management in Service-based Infrastructure

Pascal Bou Nassar[1, 2], Youakim Badr[1, 2], Frédérique Biennier[1, 2], Kablan Barbar[3]

[1] Université de Lyon. CNRS
[2] INSA-Lyon. LIRIS. UMR5205. F-69621. France
{pascal.bou-nassar, youakim.badr, frederique.biennier}@insa-lyon.fr
[3] Faculty of Sciences, Lebanese University, Fanar, Lebanon
kbarbar@ul.edu.lb

Abstract. In order to secure collaborative business processes, we present a methodological approach that early integrates security and risk management throughout the design process of service-oriented architectures. We develop our methodology based on two complementary axes: the first being the business needs while the second, is ensuring a consistent security between partners at the runtime. The information security is globally applied to business needs, service specifications and infrastructure deployment. Finally, we annotate services with security parameters that could be used to improve the selection of secure services in run-time.

Keywords: Business Processes, Service Oriented-Architecture, Methodology, Risk Management, Business Process Modeling.

1 Introduction

The frequent changes in the organization nowadays, need an alignment of business processes on business strategies, which, at the same time, require a set of methods, tools, management practices and the adaptation of information and communication technologies. Besides, the development of technology led enterprises to go beyond their own boundaries and establish collaborative business processes. This trend increases the need for interoperability not only at the organizational level (i.e. the partners should share common business objectives and management rules) but also at the technological level (i.e. the informational flows should be understood by the different partners). [1]

Recently, the Service-Oriented Architectural style (SOA) proves to be an efficient design method of business processes from independently developed and deployed services. A service consists of a logical representation of a repeatable business activity that has a specified outcome, such as “check customer credit”, “provide weather

data", or "consolidate drilling reports". A service is also self-contained, may be composed of other services, and appears to be a "black box" to its consumers [2]. By such SOA guarantees flexibility and adaptability, which are fundamental characteristics of modern business requirements by enabling services to be dynamically selected and composed [3].

As an attractive solution in implementing business processes and ensuring technical interoperability, SOA seems to be a convenient architecture for inter-enterprises collaboration since it is based on semantic information modeling, which describes the information of the business domain and permits the design of business documents to be used in service interactions between partners. Despite these advantages, SOA raises new challenges related to the information security level between partners. Information security is determining what needs to be protected and why, what it needs to be protected from, and how to protect it. Establishing security involves securing data communications, data in rest, managing access rights, etc. In SOA, a coherent security view must be shared, a trust network should be established, and global security policies' enforcement should be accomplished. For these reasons, managing security is crucial aspect of online business collaboration and should support an end-to-end security strategy at the organizational and technical levels and consider the entire services' life cycle. Services Life cycle is characterized by two main time intervals: Design-time and Run-time, during which, services are transformed from concepts into deployed entities. Services, which are established at design time, should be analyzed, designed, constructed and tested. At the run-time, they are managed, monitored and controlled to ensure their execution as expected.

In our work, we shed the light on both the design and the run-time of secure service-oriented architecture for business collaboration and base our approach on two complementary axes: the first is the business needs for evolving the architecture and optimizing the return on investment; while the second, is ensuring an end-to-end consistent security.

The rest of the paper is organized as follows: Section 2 examines previous methodologies in securing service oriented architectures and discusses the need for improvements. Section 3 presents our proposed methodology covering the design of a secure SOA. Section 4 introduces our proposed architecture for dynamic selection of secure services. Section 5 presents future work and our conclusion.

2 Related work

Due to the success of service-oriented architectures, many design methodologies have been developed and lead to the maturity in service design [4] [5] [6] [7] [8]. Some of these methodologies focused on business aspects of SOA, while others were more technical. Particularly, the 'Service-oriented design and development methodology' proposed by M. Papazoglou covers both the design and the run-time phases of

the service's life cycle. However, we have found that none of the above listed methodologies meets our needs in developing a secure service-oriented architecture.

In [9], the author has integrated a risk managing process in designing SOA applications; however, the work appears to target web services' technology, without emphasizing on the business aspects, which is very important since SOA is an architecture style and not a technology. In our work we will try to fill the above mentioned gaps and we emphasize on the fact that security does not only concern internal actors of the enterprise but also the partners. Therefore, will provide a mean for the partners to express the security capabilities of their services and guarantee the security requirements needed for the provision of a secure collaborative business process.

3 Methodology for Designing Secure SOA

By integrating security and risk management in the SOA design process, we have developed a new methodology (Fig. 1) and this to achieve two complementary objectives: on one hand, it will ensure business alignment, flexibility and reuse of services while on the other hand; it will leverage secure SOA applications by treating the identified security risks.

A successful application of the methodology is based on creating a common dialogue between business and technical managers; stages of the methodology should be well prepared to acquire all available information in order to take the best decisions. The methodology consists in integrating a risk management process into the process of services' design. As illustrated in Fig. 1, we have separated between the 'identification and specification', 'risk management' and 'annotation' phases in order to cover separation of concerns between services, risk management and services' annotations.

In the following subsections, we will emphasize the above mentioned phases.

3.1 Phase 1: Services Identification and Specification

We start in step 1 by identifying the elements of the business domain which are the business objectives, the list of actors and partners, the business strategies and finally, the list of global interactions in between the domain actors. Step 2 is dedicated for the business process and the business document modeling. These tasks are accomplished in two parallel steps because they share a common element: the business document. In step 2A, we focus on modeling the business processes in order to meet business objectives; this is an important step that leads to the identification of business services. In step 2B, we establish a data structured model, which allows us to determine the documents to be exchanged within processes and ensures business interoperability among partners. Subsequently, we identify in step 3, security objectives that cover the business level such as the identification of legal and organizational constraints, the definition of a Protection Level Agreement (PLA) for managing secure interactions between partners. Step 3 must be conducted by the business managers supported by

security analysts, brainstorming sessions are also required to determine the appropriate security objectives based on business needs.

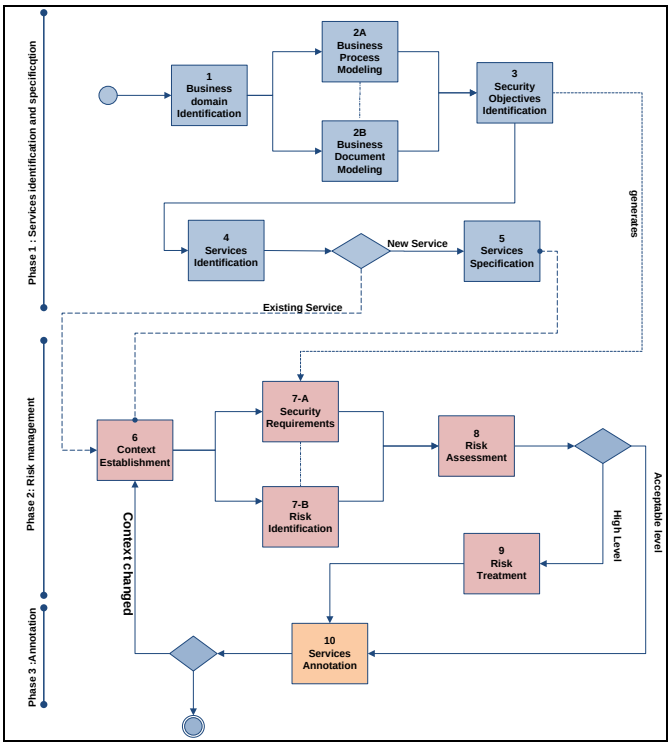

Fig. 1. Methodology for designing secure SOA

In a collaborative business process, activities are realized through local services in addition to the partners' services. Step 4 is dedicated to the identification of these services. We rely on the business process model, established in step 2A, to identify the existing services, the partners' services and the services to create. Step 5 is dedicated to the specification of new services in which we approach the business and technical specifications excluding security aspects, they will be determined following the risk management process. The specification of a service must provide information about the service capabilities and its requirements while hiding the implementation details.

3.2 Phase 2: Risk Management

Implementing a service-oriented architecture never starts from scratch; we always rely on existing resources and services. The integration of a risk management cycle

(steps 6 to 9) allows us to study the design context by identifying the existing security measures on relevant elements, improving these measures or creating new ones:

We begin this cycle by establishing the context in step 6; this is done by gathering the business and technical elements forming the design context. Subsequently, we identify in steps 7A and 7B the security requirements and the risks. These steps are accomplished in parallel because they share a common entry: the context establishment. Security requirements are the specification of security objectives; we rely on the security objectives determined in step 3 to derive the security requirements at each of the elements forming the design context.

In our previous work [10], we showed that, in a service environment, managing security and handling threats requires advanced methodologies for risk management. In order to accomplish the risk identification in step 7B, we relied on a threat modeling approach, particularly the Coras approach [11] due to its flexibility to be used in service environments. We also hinge on the context's essential elements and the relevant security requirements to identify the risks, which consist of threat scenarios, threats, vulnerabilities and unwanted incidents. In step 8, we assess identified risks based on their impact and the probability of their occurrence. We recall that the risk is the probability of occurrence of an unwanted event and its impact on resources. We establish the impact's scale in terms of value reduction of the assets (e.g., insignificant, minor, moderate, major) and a probability scale for a fixed period of time. (e.g., rare, unlikely, possible, certain), finally we elaborate a risk evaluation matrix.

In step 9, we take the necessary measures to reduce, transfer, avoid or accept the risks. Security protocols, security mechanisms, security policies, security services are different types of measures that could be implemented or improved. In the next section; we will discuss the security annotation phase.

3.3 Phase 3: Service Annotation

In the service design process, we have decided to annotate the services for multiple reasons:

- Enrich the services' description: Security annotations describe security capabilities.
- Improve the dynamic selection of services: Security annotation could be used in run-time to improve services' selection based on security requirements in addition to functional requirements. This would allow the interconnection of collaborative business processes on the fly.

The main challenge with security annotations is that they must not disclose the service weaknesses' and this by providing private information about essential elements while guaranteeing the security on both service and infrastructure levels. To solve this issue:

1. We set the following security concepts to be included in the annotation:

- Availability: specifying service availability based on mechanisms of redundancy or automatic restoration, for example.
- Privacy: This element must ensure that private data, exchanged or stored by the service is protected against unauthorized access.
- Monitoring: specifying the services of auditing and logging. For example: Policies must be clear, enforced by the partners' systems, service operations must work as expected, etc…

2. We generate a calculated value for each of the above security concepts based on the implemented security measures. The calculated value results from an overall security assessment. **Table 1** illustrates a conceptual example in calculating the availability of service "check customer credit". For simplicity purposes, we have chosen three technical elements that constitute the hosting environment of the service.

	Essential element	**Security measure**	**Value**
check customer credit service	Application server	None	0
	Web server	Periodic security updates	1
	Internet Connection	Redundancy	1

Table 1. Example calculating the service availability

The availability value will be 2/3 = 66, 6 % and will represent a level of an overall availability of the technical elements relevant to the service's security. This value in addition to the values of other concepts will form the service's security annotation.

4 Architecture for Secure Services' Selection.

Service oriented architectures improve application and business alignment by adapting or creating business processes from distributed services. In a closed environment, invocation and interaction with services are static and service registry is accessible to the enterprise and to its partners. However, in a distributed environment, static interaction with services becomes obsolete and their invocation must be accomplished dynamically. A dynamic SOA provides means to dynamically adapt the architecture on runtime. It allows services to communicate even if they aren't recognized in advance [12].

Service oriented architectures are commonly built using web services standards that have gained broad industry acceptance. Web services architecture is composed of three phases where, in the first phase, service description is published to a registry. The second phase consists in inquiring a service based on its functional parameters. Finally, the binding phase consists in invoking a service from the list of candidate services populated in the registry. In this section, we propose extending the Web services architecture by adding an intermediate service 'Security Broker'. This service allows:

- Publishing secure services' description in the registry.
- Checking and validating of security annotations.
- Selecting web services based on their functional parameters and their security annotations.

The Security Broker is composed of the following modules:

- The publishing module manages publishing functional and security parameters into the registry. It adds the security annotation into the 'security annotation registry' as well.
- The selection module searches for the best service candidate in the registry.
- The 'Security Annotation registry' stores for each service its annotations for further validation.

Fig. 2 illustrates the interactions in services' publishing and selection.
Step 1: The provider publishes the service's description via the publishing module.
Step 2: The security broker publishes the service's description into the registry.
Step 3: The security broker saves a copy of the security annotation into the security annotation registry.
Step 4: The client sends a request to the security broker to select a service based on functional requirements.
Step 5: The security broker searches the registry for services that meet with the functional requirements
Step 6: The security broker validates the security annotations with the security annotation registry and searches for the best secure service.
Step 7: The client binds to the chosen service.

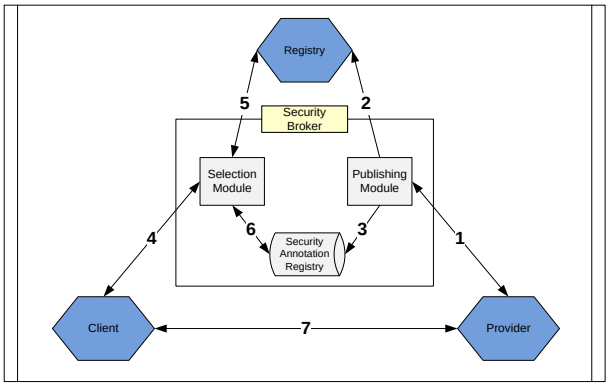

Fig. 2. Architecture for secure service selection

5 Conclusion

Securing collaborative business processes from the early design phases is a must. Security parameters must be taken into account as any other functional parameter. In

this work, we have presented a methodology for designing secure service-oriented architectures. Our methodology is based on integrating risk management cycle into the service design process. Besides, we use the security measures to annotate the services with security parameters. Annotations will be used to design new secure business processes or improve the security of business processes created on the fly.

We would like to note that the methodology presented in this work is an overview of the developed methodology which consists of models and best practices to guide the analysts in their work. The details were omitted due to the paper limit space.

Besides, we have presented an extended web services architecture which allows selecting services based on their functional and their security parameters as well.

In our future work, we will enrich the methodology by developing: (1) a secure service model which will be used to improve the security awareness in service's design. (2) a security annotation ontology representing the annotation elements (3) a process defining the security assessment for calculating the security annotation elements' values.

References

[1] Y. Badr, F. Biennier, et S. Tata, « The Integration of Corporate Security Strategies in Collaborative Business Processes », *IEEE Transactions on Services Computing*, 2010.

[2] The Open Group, « SOA white paper ». [Online]. Available: https://www2.opengroup.org.

[3] S. Chaari, Y. Badr, et F. Biennier, « Enhancing web service selection by QoS-based ontology and WS-policy », in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, p. 2426–2431.

[4] M. P. Papazoglou et W. J. Van Den Heuvel, « Service-oriented design and development methodology », *International Journal of Web Engineering and Technology*, vol. 2, n°. 4, p. 412–442, 2006.

[5] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Gariapathy, et K. Holley, « SOMA: a method for developing service-oriented solutions », *IBM Syst. J.*, vol. 47, n°. 3, p. 377–396, 2008.

[6] C. Emig, K. Krutz, S. Link, C. Momm, et S. Abeck, « Model-driven development of SOA services », *Cooperation & Management, Universit\ät Karlsruhe (TH), Internal Research Report*, 2008.

[7] C. Bate, A. Mulholland, et U. K. Capgemini, « A methodology for service architectures », 2005.

[8] O. Zimmermann, N. Schlimm, G. Waller, et M. Pestel, « Analysis and design techniques for Service-Oriented Development and Integration », *IBM Deutschland*, 2005.

[9] N. Kokash, « Risk Management for Service-Oriented Systems », *Web Engineering*, p. 563–568.

[10] P. B. Nassar, Y. Badr, K. Barbar, et F. Biennier, « Risk management and security in service-based architectures », in *Advances in Computational Tools for Engineering Applications, 2009. ACTEA'09. International Conference on*, 2009, p. 214–218.

[11] M. Lund, *Model-driven risk analysis : the CORAS approach*. Berlin ;;London: Springer, 2010.

[12] D. Parigot, B. Boussemart, et others, « Architecture Orienté Service Dynamique: D-SOA », 2008.