# Towards Security Awareness in Designing Service-oriented Architectures

Pascal Bou Nassar[1], Youakim Badr[2], Frédérique Biennier[2] and Kablan Barbar[3]
[1]*Agence Universitaire de la Francophonie, Mathaf, Beirut, Lebanon*
[2]*Université de Lyon, INSA-Lyon, LIRIS-CNRS, Villeurbanne, France*
[3]*Faculty of Sciences, Lebanese University, Fanar, Beirut, Lebanon*

Keywords: Security Management, Risk Management, Service-Oriented Architecture, Reference Models and Design Method.

Abstract: Many information security approaches deal with service-oriented architectures by focusing on security policies, requirements and technical implementation during service design, specification and implementation phases. Nevertheless, service-oriented architectures are increasingly deployed in open, distributed and dynamic environments, which particularly require an end-to-end security at each phase of the service's lifecycle. Moreover, the security should not only focus on services without considering the risks and threats that might be caused by elements from business activities or underlying hardware and software infrastructure. In this paper, we develop a model highlighting the dependency between elements at business, service and infrastructure levels, defining the design context. In addition, we develop a holistic approach to define a security conceptual model, including services, security risks and security policies and guides all phases in a typical design method for service-oriented architectures.

## 1 INTRODUCTION

Changes in economic environments and business opportunities impose new organizational strategies and require interoperable information systems to facilitate collaboration between enterprises. In addition, the exponential growth of services accessible via the Web enable the emergence of the service-oriented architecture (SOA) as de facto architectural style to build agile information systems and support the interconnection of collaborative business processes by virtue of composing processes from distributed services. Although SOA ensures business and information systems alignment, it requires the establishment of security constraints that entirely cover the information system and is not only being limited to service design and composition levels. That is to say that information security should not be limited to technological solutions and has to simultaneously take into account business, organizational and technological dimensions. Given the fact that services (i.e., Web services) are not isolated from their environments, but are instead a part of an evolving ecosystem and depend on business and organizational elements such as partners, actors and organizational structure to mention a few. Moreover, services depend on their hosting infrastructure elements, including Web containers, application servers, operating systems, and networking devices. Dependencies between different elements should be explicitly identified and security objectives must be defined and attached to these elements to cover the whole SOA lifecycle and optimize security investments as well as the sustainability of security measures.

The SOA lifecycle comprises a series of phases, including domain analysis, service design, development, testing, deployment and administration (Erl, 2005). These phases are aligned with Oracle's SOA lifecycle model that clearly separates the design-time by identifying business processes, service design, build and development, and the run-time by managing service publish and provision, integration and deployment (Wall, 2006). Papazoglou proposes a complementary service model that starts with an initial phase of planning, followed by a series of phases iteratively repeated such as analysis and design, construction and testing, provisioning, deployment, execution and monitoring (Papazoglou and Van Den Heuvel, 2006). The

Bou Nassar P., Badr Y., Biennier F. and Barbar K..
Towards Security Awareness in Designing Service-oriented Architectures.
DOI: 10.5220/0004541034703550
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 347-355
ISBN: 978-989-8565-61-7

planning phase as a preparatory phase aims to streamline and organize the remaining phases. During the planning phase, the project feasibility, goals, rules and procedures are established and requirements are gathered.

The integration of security concerns in the planning phase will ensure security awareness in the services' design throughout the service lifecycle. Establishing an end-to-end security in SOA design methods requires a common model or a reference model to identify all relevant security concepts (i.e., security objectives, risks, security measures, etc.) in the SOA lifecycle and ensure a coherent guideline to design SOA as well as tie all elements at business, service and infrastructure levels. Based on our survey in section 2, current service reference models do not fully cover an end-to- end security strategy through the service lifecycle and consider information security in a context that goes beyond services to cover business, service and infrastructure levels and potential dependencies between their elements.

In this paper, we overcome these limitations by introducing a security reference model to be used in the planning phase in order to reduce gaps between actors involved in SOA design. We also propose a dependency model to establish causal relationships between business, service and infrastructure elements. These models also extend our previous work (Bou Nassar et al., 2012), by which we have proposed a secure SOA design method that tackles security from a risk management perspective by integrating risk management and SOA design steps and incorporating both technological and organizational levels (e.g., infrastructure and business level).

The remaining sections of the paper are organized as follows: In section 2, we discuss work related to reference models in service-oriented architectures. We discuss how current reference models underestimate security requirements in open environments (i.e., end-to-end security) and could not identify risks and vulnerable assets and their impact on security policies. To overcome these limitations, we propose in section 3 a dependency model and a secure service's conceptual model. In section 4 we introduce the conceptual model, comprising three distinct models; the service's conceptual model (section 4.1), the risk's conceptual model (section 4.2) and the security policy's conceptual model (section 4.3). In section 5, we conclude our work and provide future trends.

## 2 RELATED WORK

Given a domain of interest, a conceptual model is defined as an abstract representation of basic concepts, their main characteristics and relationships among these concepts. In service-oriented architectures, conceptual models can be used to present the structure and relationships between different service elements.

In SeCSE's project (Colombo et al., 2005), a conceptual model has been proposed to provide a clear definition of the service's concepts such as publication, discovery, composition, execution and supervision. This model was designed to be a common reference for the involved partners describing the actors and relevant activities as well as the relationships between them. The model presented in (Emig et al., 2008) aims to associate the concepts of business process and services and highlights the dependencies between the design and deployment phases' of the service lifecycle. This allows, in a model driven approach, to transform the business processes into deployed services.

Standards organizations like OASIS, the Open Group and OMG have developed complementary reference models and architectures, showing a broad consensus on the basic concepts:

- The reference model proposed by OASIS (Matthew et al., 2006) focuses on the fundamental concepts of SOA: service, description, visibility, interaction and execution context. This abstract model does not consider the services' deployment.
- The reference document "SOA Ontology" (The Open Group, 2010) developed by the Open Group emphasizes on the services' description and creates a common language for describing SOA concepts.
- The reference architecture proposed by OASIS (Estefan et al., 2008) describes how to implement a service oriented architecture, provides guidance and helps manage SOA with several partners. In addition, this architecture defines a trust model for secure collaboration. This reference architecture can be used along with The SoaML Modeling Language (OMG, 2009) for services' modeling.
- The Open Group Service Integration Maturity Model (The Open Group, 2009) can be used to assess the services' maturity to start a SOA project.
- The "SOA Governance Model" (The Open Group, 2009) can be used to define the service oriented architecture governance.

The OASIS reference model highlights the models' relations (Matthew et al., 2006). The

concepts defined by reference models are intended to be the basis for describing references architectures and patterns. Concrete architectures arise from a combination of reference architectures, architectural patterns and additional requirements. Another work, carried out in (Kreger and Jeff, 2009), compares the above listed models and shows how they complement each other.

In the literature, it is possible to point out that the reference models have gained a high maturity's level in the design, deployment and governance of service oriented architectures. However, we note that:

- None of the models mentioned above have been developed with the aim of designing secure service oriented architectures.
- In models that tackles security aspects and trust networks, such as OASIS's reference architecture, business and organizational security aspects are not taken into account.
- None of these models could be used to develop security patterns.

For these reasons, we consider that security awareness should be improved by developing a model highlighting the dependencies between the elements defining the service's design context and a secure service's conceptual model that could be used to develop security patterns.

Security management standards and methods highlight the concepts of dependency modeling in order to leverage an end-to-end security. As an example the ISO 27001 specifications (ISO/IEC 27001, 2005) which employ a PLAN-DO-CHECK-ACT (PDCA) model are based on finding dependencies between assets following a top down approach. The risk management method EBIOS (ANSSI, 2010) base its analysis on assets' dependencies. OCTAVE (Alberts, 2003) and CORAS (Lund, 2010), two other risk assessment methods are based on threat modelling with focuses on dependency modelling as well. However, these standards and methods target assessing risk in static information systems and should be adapted to meet dynamic services' environments. To fill this gap, a dependency model was proposed in (Hafner, 2009). The model details the dependency between the service and the hosting infrastructure. However, it pays less attention to the business, organizational and legal elements.

## 3 DEPENDENCY MODEL

To meet users' requirements, services' composition provides a new perspective in creating applications by using existing distributed services. In order to highlight the dependency between the elements defining the design context, we have introduced the concept of "essential elements" which highlight the fact that the service is not isolated and that its security depends on the essential elements' security. The dependency model (Figure 1) shows the "top-down" approach in identifying those elements.

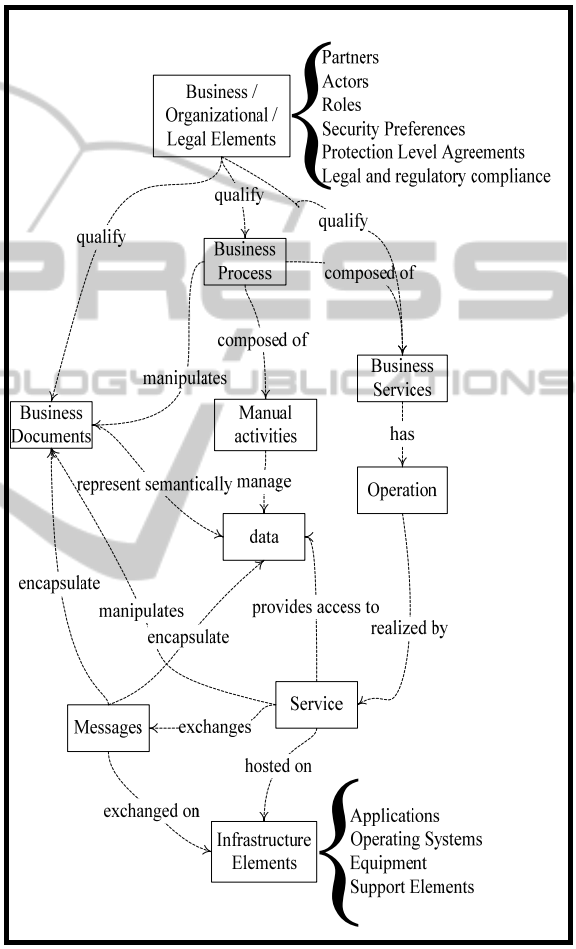


Figure 1: Dependency model.

The essential elements are:

- The business elements, describing the business, organizational and legal context include the business processes, business documents, partners, actors, roles, protection level agreements, security preferences and legal and regulatory compliance
- The atomic or composite services that implement the operations of business services
- The data that can be stored or exchanged by services
- The message exchanged between services

containing business documents and data necessary to perform a particular activity
- The infrastructure's elements, describing the services' hosting context, consist of the applications, systems and infrastructure's nodes

We start by identifying the business elements, which conceive the business processes, business services and business documents followed by the elements associated with the service (offered operations, exchanged messages, encapsulated data) up to the infrastructure's elements hosting the services. The dependency model is used to develop the service's conceptual model in the next section.

# 4 SECURE SERVICE'S MODEL

The conceptual secure service's model that we propose is built from three distinct models:
- a service's conceptual model that it built from the dependency model and that highlights the essential elements associated with the service.
- a risk's conceptual model putting in evidence different types of security risk.
- a security policy's conceptual model associating security objectives to security measures.

In the following sections, we will describe these models and create the association between them in order to leverage the conceptual secure service's model.

## 4.1 Service's Conceptual Model

The service's conceptual model (Figure 2) associates the essential elements identified in the dependency model: the business process is composed of business services (automatic and semi-automatic activities) and manual activities. Each business service provides operations that are realized by one or more services.

A business service exchanges messages encapsulating business documents or data and is implemented by services that are hosted by the infrastructure's elements.

To manage relationships between the provider and the client (dotted line), we added the association class 'contract'. The contract specifies the interface which defines the operations, security assertions and quality of service protection. The contract includes the functional and non-functional description of the service. We present in the following section the security policy's conceptual model associating security objectives to essential elements and security measures.

## 4.2 Security Policy's Conceptual Model

A service is a member of an ecosystem; it has capacities, a clearly defined role, responsibilities and rights. The frequent changes in a service environment, requires new strategies to secure resources. Similarly, security solutions must be adaptable to change.

A security policy is an efficient security measure that could be used to define the service's requirements and the usage constraints. For these reasons, we develop a security policy's model by associating security objectives to the business, organizational and technological essential elements.

In fact, security should not be limited to securing the transmission or stock of data and therefore to a technological view. For this reason, we integrate classification of information assets and partners, management of access rights. The security policy must address both the business security objectives (information classification and resources' usage) and technological security objectives (e.g. security assertions implementing security requirements).

Figure 3 presents our security policy's conceptual model in which we combine business, technology and support security objectives.

In this model:
- Security objectives are achieved by applying the constraints imposed by the contract and the security policy.
- Constraints apply to essential elements and accomplish security objectives.
- Security policy depends on the context defined by the identified essential elements.

Moreover, in this model we have defined three generic security objectives to cover the business aspects of security in a service's ecosystem:
- The 'management' of essential business elements is the creation and administration of contracts, rights and obligations, e.g. the management of access rights to business documents, the management of the agreements on the quality of protection offered during the service's delivery, etc.
- The 'classification' of business processes and business documents by assigning them a sensitivity level while adopting the scale proposed by (Badr et al., 2010): Black for private data, gray for data requiring a standard level of protection and white for public data.
- 'Trust' associated with the classification of actors and the establishment of a sharing network based on specific security policies. Trust networks allow

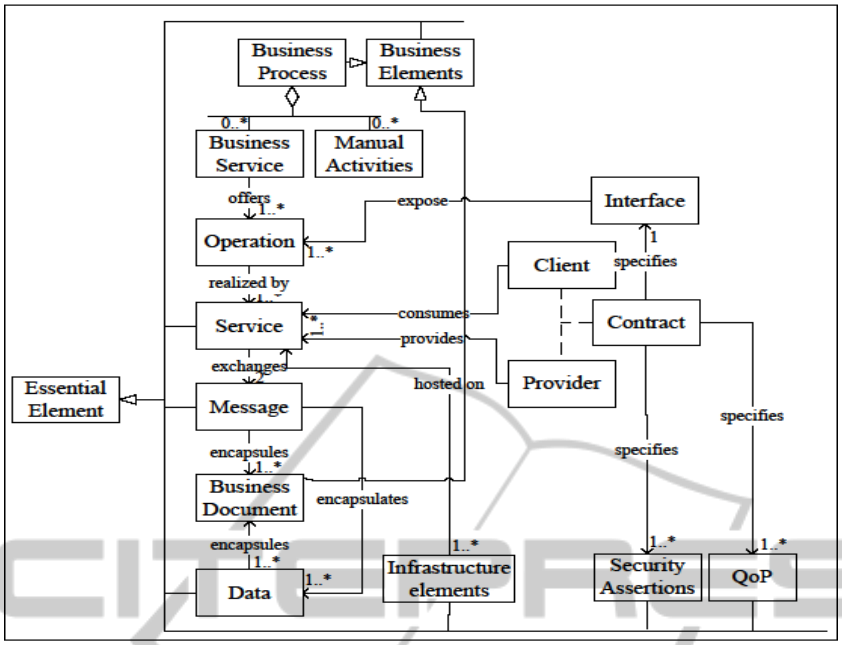


Figure 2: Service's conceptual model.



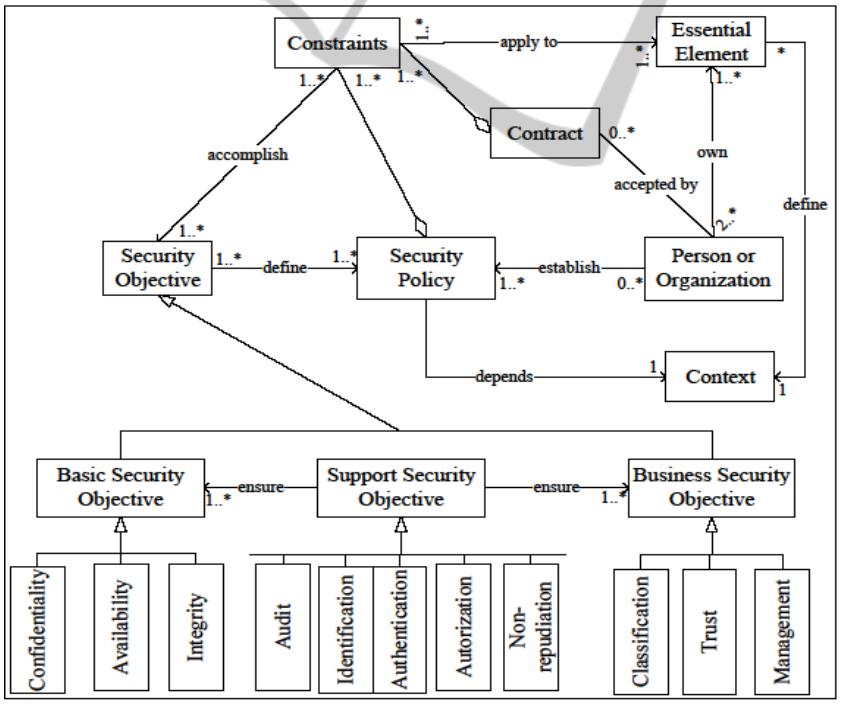


Figure 3: Security policy's conceptual model.

identity propagation across different domains and management of collaboration between actors.

We present in the following section the risk's conceptual model by defining risks at different levels of abstraction (business, organizational and technological risks). This model also considers the dynamic environments; given that the risks vary depending on the context.

## 4.3 Risk's Conceptual Model

To leverage the risk's conceptual model (Figure 4),

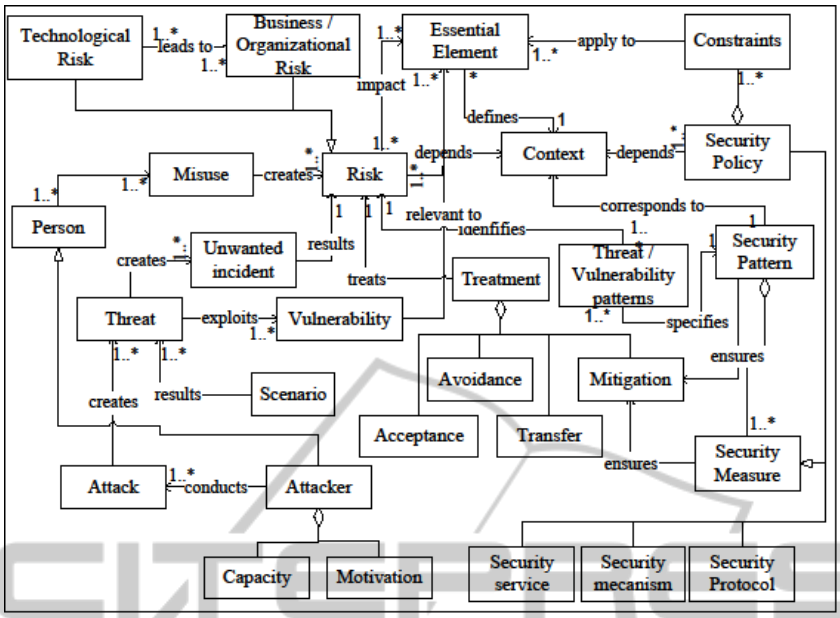


Figure 4: Risk's conceptual model.



we highlighted the following concepts:

- Identification of different risks' types of risk (business, organizational and technological) impacting the essential elements, eg :
  - Unavailability of the business essential element "business process" is a business risk.
  - Modification of the organizational essential element "access rights" is an organizational risk.
  - Unavailability of the technological essential element "router" is a technological risk.

The classification of risk types leads to an improved risk identification in brainstorming sessions carried by the business and technical leaders. Besides, the integration of risks' relationships can include causal chains in the model and thus simplify the identification process. For example, a business risk may result from technical problems (the unavailability of a router may cause the unavailability of a business process) or the opposite (an 'business' denial of service attack on a business process can lead to overloading the infrastructures' elements and therefore the router)

- Association of the risk to the context defined by the essential elements e.g, the risk level differs if the service is hosted within the company or is outsourced.
- Definition of a treatment category (acceptance, avoidance, transfer or reduction).
- Creation of a relationship between risk and security measures reducing risks. A security measure can be a security policy, a security protocol, a security mechanism or security service (In a service ecosystem, a security service is a service implementing a security objective, e.g. an authorization service.

In the following section, we establish associations between the service, security policy and risk conceptual models to leverage the secure service's model.

## 4.4 Secure Service's Conceptual Model

In the three models developed previously, we have identified the following common elements: essential element, risk, context, constraints and security measures. We rely on those elements to create the associations between the models and build the conceptual secure service's model (Figure 5), focusing on the following factors:

- The service is not isolated: its security depends on the essential elements (business, organizational and technological) defining the context.
- Risk identification is accomplished by identifying the unwanted incidents that may harm the essential elements while referring to threats and vulnerabilities patterns.
- Security patterns defined from threats' and vulnerabilities' generic patterns are used to mitigate the identified risks.

Creating instances of this model is done in three steps:

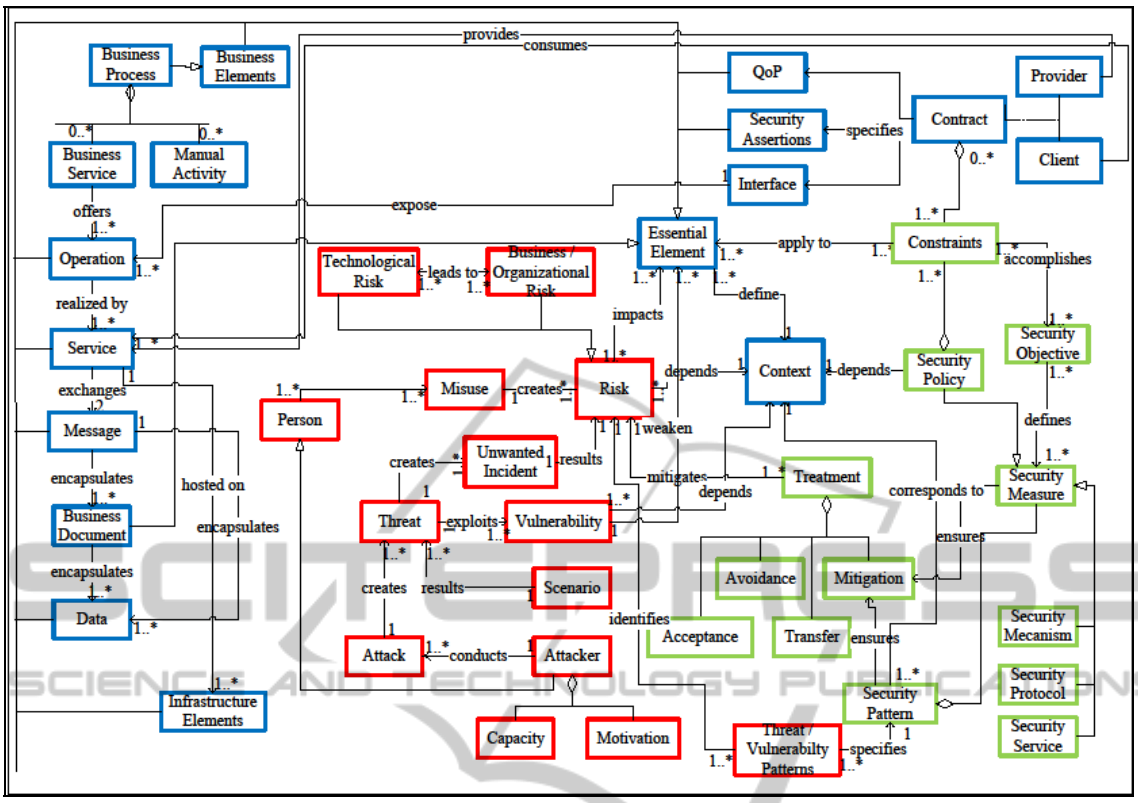


Figure 5: Secure service conceptual model.

- Context Establishment: Identification of the essential elements forming the context (concepts framed in blue).
- Risks Identification: Identification of risks that may harm the identified essential elements (concepts framed in red).
- Risks Treatment: Identification of security measures to treat risks and meet the security objectives (concepts framed in green)

As for the context establishment, we start by identifying the business, organizational and legal essential elements related to business processes. Second, we identify services that compose the identified business processes and specify their interfaces (offered operations, exchanged messages and data). In turn, the services are used to identify the infrastructure's elements that host them. This top down approach improves a secure services' design while aligning security to business needs.

To illustrate the use of our model, we propose the example of a travel agency offering online booking services. We start by establishing the context and identifying the partners (hotels and airlines), the business processes (travel reservation, online assistance, quality management), the actors (personnel, clients), their roles and access rights.

In our example and for purposes of brevity, we will focus on the 'travel reservation' business process, which is composed of the following services:

- The 'trip' service is used to book flight tickets through the 'bookTicket' operation. This operation returns the reservation result to the process and manipulates the "reservation" business document.
- The 'hotel' service is used to reserve rooms by calling the 'reserveRoom' operation which returns a confirmation message and the room number (data).
- The 'transportation' service is used to rent a car by calling the 'rentCar' operation which returns a confirmation message and the car type.

These services are hosted on the following infrastructure components:

- Glassfish Application server.
- Apache2 web server.
- MySql Database.
- Linux / Debian Operating System.
- Servers with redundant disks (Raid5)

The combination of the above listed business and technological essential elements forms the design context.

Once the context is established, we identify the business and technological risks that may harm the essentials elements. For example, the 'unavailability of the travel reservation business processes' risk can result from various unwanted events caused by the unavailability of:

- Partners (partners prefer other travel agencies or stop providing specific services)
- Trip, hotel, or transportation services (due to denial of service attacks e.g. XML-DOS)
- Components of the infrastructure (due to denial of service attacks).

Finally, the identified risks are assessed based on their likelihood and consequences. In fact, the risk analysis process provides information on whether risk needs to be treated as well as the most appropriate cost-effective treatment. Given that, high availability of the 'travel reservation' business process is needed, the unavailability of this process should be treated by implementing security measures to reduce the impact and / or probability of occurrence of unwanted incidents, for example, we implement:

- Protection level agreements specifying the availability of services between partners (business level)
- Security pattern 'Message inspector gateway pattern' to intercept the traffic and filter the requests at the service's level.
- Filtering mechanisms (firewall, routers, etc.) at the infrastructure level.

# 5 CONCLUSIONS

Securing collaborative business processes from the early design phases is highly necessary. Security parameters must be taken into account as any other functional parameter. In this work, we have presented a service security conceptual Model for improving security awareness in service design methods. As a reference model to manage information security in service-based infrastructures, it also can be used to develop security design patterns.

In our future work, we are working to support the service security conceptual with the development of ontologies, defining the essential elements and their relationships and the development of a risk treatment reasoning system to simulate risks and infer security measures, with respect to global security objectives.

# REFERENCES


Alberts, C., 2003. Managing Information Security Risks : the OCTAVE Approach, Boston: Addison-Wesley.

ANSSI, 2010. EBIOS: Expression des Besoins et Identification des Objectifs de Sécurité. Available at: http://www.ssi.gouv.fr/

Badr, Y., Biennier, F., and Tata, S., 2010. The Integration of Corporate Security Strategies in Collaborative Business Processes. *IEEE Transactions on Services Computing*, 4(3), pp. 243–254.

Bou Nassar, P., Badr, Y., Biennier, F., Barbar, K., 2012. Securing Collaborative Business Processes: A Methodology for Security Management in Service-Based Infrastructure. *Advances in Production Management Systems (APMS)*, pp. 480-487

OASIS, 2006. OASIS Reference Model for Service Oriented Architecture 1.0. Available at: http://docs.oasis-open.org/soa-rm/v1.0/.

Colombo, M., Di Nitto, E., Di Penta, M., Distante, D., Zuccalà, M, 2005. Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems. *Service-Oriented Computing*, 2005, p.48–60.

Emig, C., Krutz, K., Link, S., Momm, C., and Abeck, S. 2008. Model-Driven Development of SOA Services. Cooperation and Management, Universität Karlsruhe (TH), *Internal Research Report*.

Erl, T., 2005. *Service-Oriented Architecture : Concepts, Technology, and Design*, Upper Saddle River NJ: Prentice Hall Professional Technical Reference.

OASIS, 2008. OASIS Reference Architecture for Service Oriented Architecture Version 1.0. Available at: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/.

Hafner, M., 2009. *Security engineering for service-oriented architectures*, Berlin: Springer.

ISO/IEC 27001, 2005. Information Technology, Security Techniques, Information Security Management Systems and Requirements.

Kreger, H., Jeff, E., 2009. Navigating the SOA Open Standards Landscape Around Architecture.

OMG, 2009. SOA Modeling Language (SoaML). Available at: http://www.omg.org/spec/SoaML/

Lund, M., 2010. Model-Driven Risk Analysis : the CORAS Approach, Berlin: Springer.

Papazoglou, M. P., Van Den Heuvel, W. J., 2006. Service-Oriented Design And Development Methodology. *International Journal of Web Engineering and Technology*, 2(4), p.412–442.

The Open Group, 2010. Ontologies for SOA. Available at: http://www.opengroup.org/projects/soa-ontology.

The Open Group, 2009. SOA Integration Maturity. Available at: http://www.opengroup.org/projects/osimm.

Wall, Q., 2006. SOA Service Lifecycle Design. Available at: http://www.oracle.com/technetwork/articles/entarch/soa-service-lifecycle-design-096035.html