Proposition de recherche doctorale

Sécurité par aspects pour architectures basées sur des services

Mots clés :

● Directeur de thèse : yves ROUDIER
● Co-encadrant(s) :
● Unité de recherche : Laboratoire de recherche d'EURECOM
● Ecole doctorale : École Doctorale Informatique, Télécommunications, Électronique de Paris
● Domaine scientifique principal: Divers

Résumé du projet de recherche (Langue 1)
Résumé du projet de recherche (Langue 2)

1. Introduction Service-oriented architectures (SOAs) constitute a major architectural style for large-scale infrastructures and applications that are built from loosely coupled and well-separated services and that are subject to dynamic configuration, manipulation and evolution. SOAs today are the major structuring principle of a multitude of commercial infrastructures and applications that rely on service composition, in particular service orchestration, and that are frequently subject to evolution. They span a number of different organizations, and may involve powerful servers as well as resource-constrained devices (e.g., mobile devices). Two types of cross-boundary functionalities are essential to such compositions: • Functionalities that span administrative domains, such as security domains that are governed by different security policies, e.g., between warehouses and their clients. • Functionalities that span different technological domains, e.g., service infrastructures for fully featured service clusters as well as much more limited infrastructures supporting the mobile devices used by clients to access services. Similarly to other compositional structuring mechanisms, SOAs are subject to the problem of crosscutting functionalities, that is, functionalities that are scattered and tangled over large parts of the architecture and of the underlying implementation. Security functionalities, such as access control and monitoring for intrusion detection, are prime examples of functionalities with this problem: they cannot be properly modularized, that is, defined as well-separated modules, especially if they should scale across administrative or technological domains. Aspect-Oriented Software Development [Aks04] is an application-structuring method that addresses the problem of the lack of modularization facilities for crosscutting functionalities in a systemic way. The thesis will address this problem and contribute to the French-funded ANR project CESSA (“Compositional Evolutions of Secure Services with Aspects”). This project will enable the synthesis of correct by construction SOA-based applications and will allow the formal analysis of security properties of SOAs. It will also produce a specification of security aspects as part of a secure web service environment that will complement the specification of security aspects as part of the Business Process Execution Language for Web Services (BPEL) [WS05, part VII] is a de facto standard for expressing Web Service compositions. The difference between the SOA approach and traditional approaches using conventional middleware lies in the looseness of the different parts of the distributed system specified. Another key difference is the use of standard and uniform formats and protocols. Many efforts have been devoted to the formalization and the reasoning upon processes, as exemplified by the proceedings of the international workshop "Web Services and Formal Methods" [WSFM05?07]. More specifically, different formalisms have been applied to BPEL: Petri nets [BPEL?PN05], Spin, a model checker [Spin04, WSAT04], process algebras, like FSP ("Finite State Process") [FSP03], or CCS ("Calculus of Communicating Systems") [CCS04], for example. In a SOA, there exist complex interactions among functional, management, and infrastructure interfaces. Aspect Orientation approaches have been proposed to solve this issue in the context of existing orchestration services for SOAs (e.g., Padus [BVJ06] and AO4BPEL [CM07]). Similarly, the OoSL4BPEL approach [BRL08] eases the OoSL management in service compositions by the specification of aspects that concern the security aspects (i.e., aspects that result in the modification of BPEL service compositions) and vertical composition for Web Service and OSGi based SOAs.

2. State of the Art This thesis aims at addressing the problem of evolving large-scale SOAs based on aspects, in particular regarding their security models. Such SOAs involve traditional service compositions and refinements. The following sections briefly discuss the scientific and industrial state of the art of these different fields.

2.1. SOA Service-oriented architectures (SOAs) are considered as advanced component-based architectures for the construction of distributed systems. A service is a software application that can be located over a network, and whose interfaces and bindings can be defined, described and discovered by using standardized access means and formats. Services support direct interactions with other software agents using message exchanges over the network via well-defined protocols. Service compositions are computed to implement processes, whose specification is done using dedicated workflow languages. Web Services are a concrete realization of a SOA, which uses XML artifacts and Internet-based protocols [WS04, WS05]. The Business Process Execution Language for Web Services (BPEL) [WS05, part VII] is a de facto standard for expressing Web Service compositions. The difference between the SOA approach and traditional approaches using conventional middleware lies in the looseness of the different parts of the distributed system specified. Another key difference is the use of standard and uniform formats and protocols. Many efforts have been devoted to the formalization and the reasoning upon processes, as exemplified by the proceedings of the international workshop "Web Services and Formal Methods" [WSFM05?07]. More specifically, different formalisms have been applied to BPEL: Petri nets [BPEL?PN05], Spin, a model checker [Spin04, WSAT04], process algebras, like FSP ("Finite State Process") [FSP03], or CCS ("Calculus of Communicating Systems") [CCS04], for example. In a SOA, there exist complex interactions among functional, management, and infrastructure interfaces. Aspect Orientation approaches have been proposed to solve this issue in the context of existing orchestration services for SOAs (e.g., Padus [BVJ06] and AO4BPEL [CM07]). Similarly, the OoSL4BPEL approach [BRL08] eases the OoSL management in service compositions by the specification of aspects that concern the security aspects (i.e., aspects that result in the modification of BPEL service compositions) and vertical composition for Web Service and OSGi based SOAs.

2.2. Security SOA-related security is a rather recent field of research. Security in SOA has previously been addressed mainly from an application point only of view, in terms of enforcement of properties like authenticity or confidentiality of messages, in particular in early SOA middleware (COSS and Jini), then with more acuity and by handling additional complexity in the case of Web Services. In particular, the loose coupling assumption of SOA made it necessary for security to be implemented as a service, thereby avoiding tightly binding security concerns with “functional” services themselves. Such an architecture avoids compositional impossibilities. However, even though this approach successfully introduced security in the SOA arena so far, it also has inherent limitations with respect to the security concerns that can be addressed. Whereas nothing limits the expressiveness of the Security Assertion Markup Language (SAML) [CKP+09], an XML-based standard used to control access, the enforcement of an access control policy through a unique service makes it essentially unsuitable for cross organizational concerns. It is becoming apparent that the enforcement of security properties is a fundamental problem that has to be addressed and that is an inherently crosscutting concern. It was only recognized in recent years that SOA clients and services are themselves susceptible to various attacks through the messages exchanged, in addition to classical network?level attacks or web?based attacks. XML rewriting attacks have, for instance, been demonstrated [RRS06, GLS07]. Such concerns can be partly addressed by a more careful design of communication protocols. However, implementation bugs are hard to track and deserve to be explored to effectively ensure security enforcement: vulnerability testing, test case generation, and fuzzing [Bei84, Bei90, GJM94] may be used to that effect, as well as approaches [CGP06, GKS05, MKK07] aiming to explore the alternative execution paths of an application to increase the analysis and test coverage of dynamic techniques. Application gateways also offer an interesting, even though under-explored structuring mechanism for both testing advanced vulnerabilities and countering attacks. Gateway services have for instance been used to inject attacker patterns and to replay these with some degree of automation. Regarding countermeasures, gateways have long been used for mitigation as demonstrated by Scott and Sharp’s application?level firewall [ScSh92]. XML and SOAP firewalls have developed in recent years [Wiki09], in particular in form of network appliances for sanitizing the XML encoding used for REST or SOAP messages before they actually reach any client or server in a domain, as well as for enforcing access control policies. Still not much has been done to date with respect to the deployment of such appliances and to exploit their capabilities for the development of dynamic applications. Furthermore, vulnerability detection and appropriate countermeasures in the context of cross?boundary security properties still constitute an open research issue.

Résumé du projet de recherche (Langue 2)
3. Thesis topic

The thesis will provide a comprehensive treatment of security functionalities for SOAs in the presence of horizontal and vertical composition as well as evolution using aspects. Given a specification of security requirements for a service-oriented architecture, three problems will be investigated: • How to synthesize secure services to satisfy the security requirements. • How to certify that the services synthesized effectively satisfy the security requirements. • How to ensure that properties satisfied by a service-oriented architecture are preserved after evolutions using aspects. Aspect Oriented Programming will ease the synthesis of secure composed services by separating security concerns from other concerns in SOAs, such as its business logic, transaction handling and so forth. Likewise, the certification of the services synthesized will benefit from AOP, which will be helpful to automate the control process: AOP can often easily ensure a property by introducing access control or data flow checks when some condition is detected. The thesis will also focus on the definition of a formal architecture description language (ADL) for SOAs with aspects. This ADL will support techniques for the description of functionalities that go across administrative and technological domains through horizontal and vertical composition of services, as well as their evolution using aspect-oriented programming (AOP). It will be formally defined in order to enable reasoning about correctness properties, specifically security properties. 3.1. Language support for service composition with aspects A large part of the thesis will focus on refining the architectural model by language?support for specific kinds of protocol. Concretely, the language should provide the following features: • Expression of horizontal and vertical compositions using composition operators over service protocols. • Expression of service evolutions using history?based aspects whose application is governed by aspect?aware service interface: aspects may only be applied if aspect applications are enabled for a service. 3.2. Language support to express security specifications A prerequisite to the synthesis of secure services is the definition of a language to express security specifications. The language will allow the description of expected security properties and security countermeasures to be adopted in the presence of an abnormal behavior of the system compared with the initially defined requirements. Means for the definition of security properties in the presence of aspects will be developed, including property definitions over vertical and horizontal compositions involving aspect?aware interfaces. Properties of compositions between large?scale SOAs and embedded devices will be handled. 3.3. Property preservation in the presence of aspects The evolution of SOAs with aspects may break some underlying properties satisfied by the original SOA. Indeed, many evolutions and hence aspects have to be invasive (to some extent). This task will explore the impact of the use of aspects to secure SOAs. Aspect?aware service interfaces will be exploited to provide strict guarantees on the effects aspect may exert on the different architectural layers, thus reconciling aspects with the strong encapsulation properties of SOAs. All the above mechanisms will be applied within the context of Enterprise Service?Oriented Architectures, in particular through experiments performed with existing SOA protocols and with use cases supplied by SAP. In particular, the thesis will produce a methodology for applying the above mechanisms, and will compare the use of the language supporting the definition of security aspects with traditional security best practices.