SECURE AND CORRECT TRANSACTION MANAGEMENT IN SERVICE-ORIENTED ARCHITECTURES

Csilla Farkas and Michael N. Huhns

Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29208 USA
{farkas,huhns}@engr.sc.edu

ABSTRACT: *In this position paper we present an overview of technologies needed to provide security and reliability for Web-based applications and discuss limitations of current support. We propose promising, new approaches to enable correct and secure transaction management in the larger context of a Service-Oriented Architecture. In particular, we study vulnerabilities that are primarily due to the distributed and loosely coupled nature of the service-execution environment, the dynamic run-time selection and composition of services, and the autonomy of the individual services. We argue that safeguarding these new technologies requires new ways of thinking of and developing security capabilities.*

INTRODUCTION

One of the newly evolving technologies that will shape future computing is Service-Oriented Architecture (SOA). SOA aims to provide flexible, reusable, and cost-effective ways to support complex business applications. To achieve these goals, SOA applications must support a dynamic and secure combination of services that incorporates security-levels of individual services, and establishes a security-level of the composite service. Current development [4,5,6,7,8,9], targeting Web Services (WS) technologies, support for business processes, and basic security, addresses some of the issues but in an insufficient manner. *Current research lacks rigorous formulation on the desired security and transactional properties of WS compositions.*

In this position paper we emphasize two crucial aspects of SOA applications: correctness and security. We argue that both of these aspects must be addressed during the development of **technologies** that support SOA applications, as well as during the development of the SOA **applications** themselves. Our approach is two-faceted. While the development of individual services and their combination must be secure and correct, the technology, that supports the applications using these services, must have the capability of expressing security and correctness requirements. In particular, we argue that a framework and corresponding methodologies must be developed to support business processes that may require the composition of services that range over different security domains, have long duration, and lack central coordination.

We focus on two main aspects of WS transactions. First, we study the available security technology to secure WS transactions and propose new methods to enhance security. Our main focus is SOA security from the perspective of service-level security. We argue that each member service must be securely designed, developed, deployed, and maintained. In addition, we discuss the need for research to address security assurance of service compositions and develop methods to dynamically support secure and correct service compositions. Second, we elaborate on the need to develop concepts to define secure and correct execution of WS transactions. While the concepts, such as atomic transactions, locking protocols, etc., have been used by the database community, their definition must be adjusted to be applicable for

SOA, and new concepts, that properly capture the needs of SOA applications, must be developed. Furthermore, well known database problems, e.g., deadlock detection and failures, become even more complex in the SOA context.

LAYERS OF SECURITY

We propose an approach to provide security for SOA applications. Our approach addresses security needs of individual services, their combinations, and support of SOA applications over insecure communication channels.

Individual service-level security incorporates traditional software security practices, such as incorporating security needs in the Software Development Life Cycle (SDLC) [1], and methods to assign security attributes to individual services. **Service composition** (or business)**-level security** aims to provide flexibility and ease of use to combine diverse services into combined services to fulfill business needs while guaranteeing security properties of the combination. Security needs may range from the need to protect corresponding metadata to limit the information available for malicious users, e.g., partial disclosure of Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) specification, to the enhancement of business process composition languages to express security requirements. **Network-level security** targets security research that arise from the distributed and open nature of WS and WS-based SOA applications. Research directions include federated architectures, identity management, authentication, trust management, and secure communication.

We argue that high assurance security can only be achieved if all three of the above aspects are considered. Any security model that addresses only some of these aspects will be vulnerable to malicious attacks.

CORRECT AND RELIABLE EXECUTION

Correct execution is a crucial component of WS transactions. The question is whether we can rely on traditional database transaction management concepts to provide security and consistency for SOA application. Emerging research [3,8,9], such as compensation-based transaction management, addresses some of the related problems but does not provide sufficient support to establish the achieved level of correctness. Moreover, traditional database concepts, such as serializability, are not suitable to guarantee correct WS executions and may also limit system efficiency. On the other hand, special characteristics of SOA applications, such as the availability of several compensating services, seem promising to increase the robustness and efficiency of the system.

We propose that rigorous research, addressing the transactional properties of SOA applications, must be developed. Additional security threats resulting from the special characteristics of SOA applications, e.g., deadlocks created by service-level (transactional) dependencies, must be evaluated and appropriate safeguards developed.

We argue that along with new types of computing paradigms and associated vulnerabilities, we need to harness new approaches to protect SOA applications. For example, we can employ collaborative defense systems that pool knowledge of individual nodes across organizational boundaries to make a variety of simultaneous diverse approaches to defense [2]. For example, a multiagent-based approach can increase system robustness by supporting multiple, independent versions of the services. This produces a flexible and adaptable platform to handle multiple versions.

CONCLUSIONS

We argue that **Web-based applications**, built on top of existing technologies to support Web Services, Semantic Web, dynamic business processes, etc., will become the main **shaping forces of future computing**. These applications must satisfy two crucial requirements: **correct** and **secure** execution. However, current technologies fail to guarantee the satisfaction of these requirements. Our belief is that additional research and development must be performed to develop data and application security and reliability methods and technologies to support Web-based applications.

REFERENCES

[1] J. Epstein, S. Matsumoto, and G. McGraw, Software Security and SOA: Danger, Will Robinson!, Building Security In, 2008.

[2] Morton Swimmer, "Using the danger model of immune systems for distributed defense in modern data networks," *Computer Networks*, vol. 51, no. 5, 11 April 2007, pp. 1315-1333.

[3] M. Verma, "Web services transactions," IBM Publications, 2005.

[4] W3C Schools, WSDL and UDDI, http://www.w3schools.com/WSDL/wsdl_uddi.asp , 2008.

[5] OASIS Web Services Business Process Execution Language (WSBPEL), 2007, http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=wsbpel , 2008.

[6] W3C, Simple Object Access Protocol (SOAP), 2007, http://www.w3.org/TR/soap/ , 2008.

[7] OASIS Web Services Coordination (WS-Coordination), 2007, http://docs.oasis-open.org/wstx/wscoor/2006/06 , 2008.

[8] OASIS Web Services Atomic Transaction (WS-AtomicTransaction),2007, http://docs.oasis-open.org/wstx/wsat/2006/06 , 2008.

[9] OASIS Web Services Business Activity (WS-BusinessActivity), 2007, http://docs.oasis-open.org/ws-tx/wsba/2006/06 , 2008.