Interoperability and Functionality of WS-* Implementations

Andreas Schönberger, Distributed Systems Group, University of Bamberg, Germany

Johannes Schwalb, Senacor Technologies AG, Germany

Guido Wirtz, Distributed Systems Group, University of Bamberg, Germany

ABSTRACT

Recently, the Web Services Interoperability Organization (WS-I) has announced to have completed its interoperability standards work. The latest deliverables include the so-called “Basic Security Profile” and the “Reliable Secure Profile.” This gives rise to the question whether or not Web Services adopters can rely on interoperability and functionality of Web Services stacks, in particular in terms of security and reliability features. To answer this question, the authors thoroughly analyze two important Web Services stacks for interoperability of WS-Security and WS-ReliableMessaging features. Their analysis shows that security and reliability features are far from being implemented in an interoperable manner. Additionally, they reveal that some of those interoperability problems are not even covered by WS-I profiles and therefore conclude that WS-I’s work has not yet resulted in Web Services interoperability. Finally, the authors investigate support for the so-called “Secure WS-ReliableMessaging Scenario” in order to find out whether WS-* adopters can at least rely on the availability of real-world functionality in homogeneous environments.

Keywords: Interoperability Testing, Secure WS-ReliableMessaging Scenario, Web Services Interoperability, WS-ReliableMessaging, WS-Security

INTRODUCTION

Quality-of-Service (QoS) features such as security and reliability are brought to the Web Services world by the so-called WS-* standards. WS-Security 1.1 (OASIS, 2006) and WS-ReliableMessaging 1.2 (OASIS, 2009a) are prominent representatives of WS-* standards that define data formats and processing instructions for extending the SOAP (Gudgin, Hadley, Mendelsohn, Moreau, Nielsen, Karmarkar, & Lafon, 2007) messages that implement Web Services exchanges. For example, an XML Signature tag together with SignedInfo, SignatureValue and KeyInfo tags would have to be inserted into the SOAP Header tag to provide integrity protection. For convenience, a Web Services developer is not supposed to ‘manually’ insert all that information into SOAP messages. Instead, Web Services Security Policy 1.3 (WS-Sec-Pol; OASIS, 2009b) and Web Services Reliable Messaging Policy Assertion 1.2 (WS-RM-Pol; OASIS, 2009c) can be used to extend the WSDL definition of a Web service with assertions that instruct the Web Services stack implementations (WS stack in the following) in
use to apply WS-Sec and WS-RM features to the SOAP message exchanges. This policy-based realization of QoS features for Web Services has been postulated in several publications (Martino & Bertino, 2009; Gruschka et al., 2006, 2007; Curbera et al., 2008; Khalaf et al., 2006; Nezhad et al., 2006; Menzel, Warschofsky, & Meinel, 2010) and promises better interoperability than letting all Web Services developers implement QoS features their own way.

On November 10th 2010, the Web Services Interoperability Organization (WS-I) released a press announcement (Cassidy, 2011). This announcement followed the release of new versions of WS-I’s core deliverables, most notably the Basic Security Profile (BSP) (WS-I, 2010a) and Reliable Secure Profile (RSP) (WS-I, 2010b) in 2010. WS-I profiles are the core deliverables of WS-I and document “[...clarifications, refinements, interpretations and amplifications of those specifications which promote interoperability]” (WS-I, 2010b). The main target of BSP and RSP are WS-Sec and WS-RM. As WS-I promotes interoperability and has declared its standardization work to be completed, this means that Web Services adopters should be able to rely on the interoperable implementation of Web Services security and reliability features which are pivotal for Web Services according to Nezhad et al. (2006) and Moser et al. (2007). However, WS-Sec and WS-RM as well as WS-Sec-Pol and WS-RM-Pol are highly complex specifications so that interoperability across WS stacks does not come easy. Consistently, Martino and Bertino (2009) stress that “although one of the main purposes of the standard [i.e., a WS security standard] is to guarantee the interoperability between different platforms, it might be necessary to test it on the field.”

This paper investigates in how far WS-I’s work has led to interoperability across WS stacks regarding the implementation of WS-Sec(-Pol) and WS-RM(-Pol) and whether real-world functionality is available. In order to operationalize the question whether or not a Web Services adopter can rely on interoperability between different WS stacks, we use two ‘optimistic’ hypotheses:

**H1:** The overwhelming majority of WS-Sec-Pol/WS-RM-Pol features are implemented by Web Services stacks.

**H2:** Out of those WS-Sec-Pol/WS-RM-Pol features implemented by two platforms, the overwhelming majority is implemented in an interoperable manner.

For analyzing availability of real-world functionality, support of the so-called Secure WS-ReliableMessaging Scenario (SecRM scenario) (Gavrylyuk et al., 2005; Backes et al., 2006) is investigated. The SecRM scenario describes the WS-Sec and WS-RM based implementation of important security and reliability properties that have been identified as crucial for integration scenarios (Schönberger et al., 2010) and therefore provides a good benchmark for real-world functionality. In anticipation of the interoperability results of section “Interoperability Assessment,” the SecRM scenario is tested in homogeneous environments only.

This paper is an extended version of Schönberger et al. (2011) and the main new content is the analysis of the SecRM scenario presented in section “Functionality Assessment.” The contribution described in this paper is fourfold where the first three contributions originally have been published in Schönberger et al. (2011). First, we assess the coverage of WS-Sec-Pol/WS-RM-Pol specifications by two major Java-based WS stacks. Second, we assess the interoperability of implemented features. Based on these results, both hypotheses (H1, H2) will have to be rejected. Third, we analyze in how far the detected interoperability issues could have been avoided by strict compliance to the BSP and RSP. Fourth, we present a WS-Policy operationalization of the SecRM scenario and we show that this advanced real-world WS-* functionality is available at least for one platform.
Early Capacity Testing of an Enterprise Service Bus
www.igi-global.com/article/early-capacity-testing-enterprise-service/34106?camid=4v1a

Knowledge Discovery and Big Data Analytics: Issues, Challenges, and Opportunities
www.igi-global.com/chapter/knowledge-discovery-and-big-data-analytics/217829?camid=4v1a