

# SALMonADA: A Platform for Monitoring and Explaining Violations of WS–Agreement–Compliant Documents<sup>\*</sup>

C. Müller<sup>1</sup>, M. Oriol<sup>2</sup>, M. Rodríguez<sup>2</sup>, X. Franch<sup>2</sup>, J. Marco<sup>2</sup>, M. Resinas<sup>1</sup>, and A. Ruiz–Cortés<sup>1</sup>

<sup>1</sup> University of Seville, LSI

ISA research group, <http://www.isa.us.es/>, Seville (Spain)  
{cmuller, resinas, aruiz}@us.es

<sup>2</sup> Technical University of Catalunya

GESSI research group, <http://www.essi.upc.edu/~gessi/>, Barcelona (Spain)  
{moriol, jmarco}@lsi.upc.edu, {mrodrigues, franch}@essi.upc.edu

**Abstract.** Several research frameworks in both academy and industry aim at monitoring conditions stated in Service Level Agreements (SLAs). However, up to our knowledge, none of them present reasoning capabilities over the SLA with a clear explanation of the concrete statements that violate the agreement. In this paper we present SALMonADA, a platform to monitor SLAs specified with WS–Agreement, that provides agreement violations explanations by pointing both: violated terms of the WS–Agreement document, and violating measures of a monitoring management document.

## 1 Problem Motivation

There is a real need to have infrastructures and Service Based Systems (SBS) regulated by Service Level Agreements (SLAs). WS–Agreement is arguably the most widespread recommendation for defining SLAs. However, most developing environments do not include enough matured facilities to develop SLA–driven applications in general or WS–Agreement–driven applications in particular. The urgency to overcome this situation may change overnight if we consider the need for SLAs to regulate the increasing number of things that can be delivered as services over the Internet (the well–known XaaS or everything as a service). As an example, we find that several research frameworks in both academy and industry aim at monitoring conditions stated in SLAs to detect violations. However, up to our knowledge, between them there are only a couple of works providing monitoring facilities for WS–Agreement documents. Moreover, none of the proposals present reasoning capabilities over the SLA with a clear explanation of the concrete statements that violate the agreement. This lack of proposals to explain

---

<sup>\*</sup> This work has been partially supported by: S–Cube, the European Network of Excellence in Software Services and Systems; the European Commission (FEDER); the Spanish Government under the CICYT projects SETI (TIN2009–07366) and ProS–Req (TIN2010–19130–C02–01); and by the Andalusian Government under the projects THEOS (TIC–5906) and ISABEL (P07–TIC–2533).

violations in WS–Agreement documents pushed us to propose SALMonADA, a SBS comprising the monitoring capabilities of SALMon-aaS<sup>3</sup> and the analysis capabilities of ADA-aaS<sup>4</sup>, with the following architecture, novelties and functions.

## 2 Novelties

The novelty of our proposal is given by the following SALMonADA features:

*Full WS–Agreement compliant.* That is, it supports WS–Agreement documents with expressive terms including arithmetic-logic expressions relating several metrics inside the service level objectives (SLOs), and some elements not supported by other monitoring proposals yet, as far as we know, such as: (1) term compositors defining agreement variants inside an agreement; (2) several terms scopes denoting the affected service operations; and (3) qualifying conditions inside terms to enable or disable the SLO.

*SLOs violations explanation-aware* considering the afore mentioned expressive SLOs. That is, it provides both for SLOs violations: violated terms of the WS–Agreement document, and violating measures got at monitoring time and stored in a monitoring management document. For instance, an average response time may be calculated for each service operation allowing to include a SLO for each operation as follows: “AverageResponseTime <= x sec”. When a SLO violation is detected at monitoring, only such term which scopes to the violated service operation and which includes the violated SLO would be returned as violation explanation<sup>5</sup>.

*Asynchronous/Synchronous-compliant.* SALMonADA platform is designed and developed to support asynchronous and synchronous interaction styles with their clients. Thus, a client, based on its own benefit, may choose its preferred approach. Independently of the selected approach a client must start and stop the SALMonADA monitoring to be subscribed/unsubscribed as client.

*Decoupled Monitoring Management.* That is, it extracts a monitoring management document (MMD) as a monitoring view from a WS–Agreement document. Thus, SALMonADA comprised services use WS–Agreement documents and MMDs as operation inputs/outputs.

*Extensible.* to monitor new metrics by decoupling monitoring logic for each quality metric, making it easily upgradeable. Currently SALMonADA is extensible at deployment time but it will be so at run time in a nearby future.

## 3 SALMonADA Architecture

As shown in Fig. 1, we have developed SALMonADA as a SBS with the following elements:

<sup>3</sup> [gessi.lsi.upc.edu/salmon/web/](http://gessi.lsi.upc.edu/salmon/web/)

<sup>4</sup> [www.isa.us.es/ada/](http://www.isa.us.es/ada/)

<sup>5</sup> This scenario is included in a violating SLA test case of the web application developed as SALMonADA client

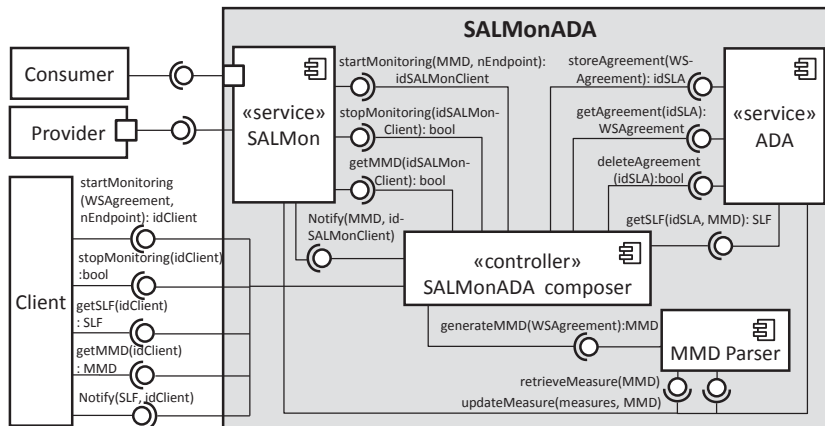


Fig. 1. Architectonic Model of SALMonADA

*Client* provides the SLA to monitor expressed in WS-Agreement. It is able to retrieve either the MMD or the analysis result, and if desired, it can also receive asynchronously notifications when the SLA has been violated.

[carlos:] *Client a la izquierda*

*SALMonADA composer* is the service that composes the internal services of the platform. It provides the interface to the client and manages the execution process of the system. It also adds an independence layer on the interaction required between the analysis of the SLAs (performed by ADA-aaS) and the monitoring of the services QoS (performed by SALMon-aaS). Such a decoupled structure allows to add or modify the internal components in a very flexible manner. (i.e. allows to replace the monitor or the analyzer without affecting the other elements of the platform).

*SALMon* is the service responsible for monitoring the services QoS.

*ADA* is the service responsible of managing and analyzing the WS-Agreement documents. It supports the analysis of WS-Agreements with expressive assertions inside guarantee terms.

*MMD Parser* is the service that extracts the MMD from the SLA, and it also implements the functionality to interact with the MMDs (retrieve or update values). Thus, the MMD structure, whose information is used by all platform components, is decoupled from both ADA and SALMon. Therefore, different MMD structures can be developed, if needed.

## 4 Functions

We have developed a web application for SALMonADA<sup>6</sup> for demonstration purposes that supports afore mentioned novelties. For instance, Fig. 2 depicts the web appli-

<sup>6</sup> it can be tried at [www.isa.us.es/ada.source/SLAnalyzer/](http://www.isa.us.es/ada.source/SLAnalyzer/) and a screencast is available at [gessi.lsi.upc.edu/salmon/ada/](http://gessi.lsi.upc.edu/salmon/ada/)

cation highlighting as violation explanation that the `AverageResponseTime` of `explainNonCompliance` operation is the violating metric because it was measured as 3.421 seconds, while the ADA-aaS SLA guarantee term obligates the provider to respond in less than 2 seconds.

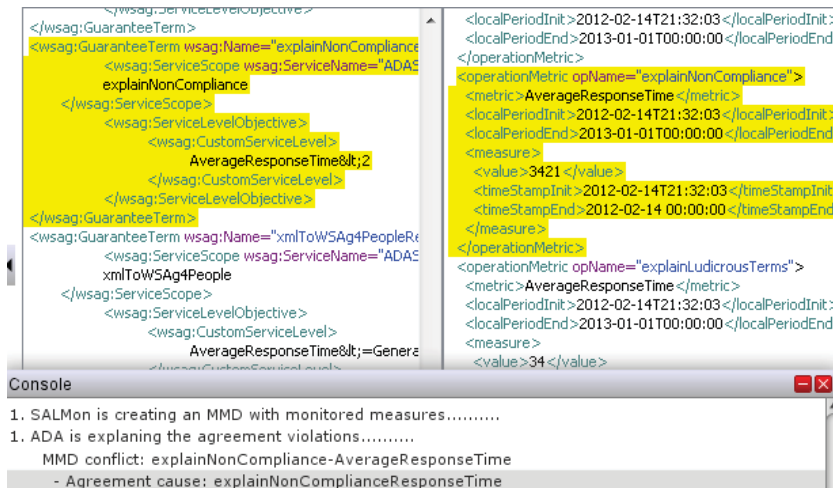


Fig. 2. Reporting a violation with SALMonADA

The web application provides also some additional functions such as: SLA violation checking that informs about the agreement violating state without providing any kind of explanation; violating and non-violating WS-Agreement documents of SALMon-aaS and ADA-aaS, to test the platform; and an edition panel to: (1) change/edit the SLA to be monitored and analysed; (2) show the MMD without and with monitored measures; and (3) show in a parallel view the SLOs violation explanation including both: the SLA highlighting violated terms, and the MMD highlighting the monitored violating measures. Finally, the web application also includes a log console to show the overall process.