

# SLA-driven governance for RESTful systems

Antonio Gamez-Diaz.

Advised by Dr. Pablo Fernandez and Dr. Antonio Ruiz-Cortes

Universidad de Sevilla\*

agamez2@us.es

**Abstract.** Software distribution models are moving to SaaS paradigms where customers no longer need to buy a perpetual license. In this context, SaaS providers leverage the Service Level Agreement (SLA) concept to delimit the functionality and guarantees to which they commit to their customers. However, although formal specifications for the definition of SLAs have been proposed, providers usually have an ad-hoc approach with a low degree of automation. This approach confirms the fact that the SaaS industry has not incorporated the idea of an SLA model that can be implemented within the infrastructure as a decision mechanism. This instrumentation would be of special interest in RESTful microservice architectures in providing an automated governance framework for the service catalog and regulating the behavior of each component in the context of the agreements reached with each client.

This thesis project is divided in four stages: i) Establishing a sufficiently expressive specification for the description of RESTful microservices regulated by advanced SLAs; ii) Develop a catalog of SLA analysis and management operations to support the governance of micro-service architectures; iii) Implement a SLAs management ecosystem to support the government of RESTful microservices; iv) Consolidation of the Governify platform to validate the proposal in industrial environments.

## 1 State of the art

The *Software as a Service* (SaaS) paradigm has become entrenched in the industry as a deployment model, bringing flexibility to the customer in pay-for-use models and making unnecessary to maintain its own infrastructure. In particular, these types of systems present a business model that depends on the expected benefits for certain infrastructure costs.

The main architectural paradigm of SaaS systems are the service-oriented architectures (SOA) [10], [5]. They have a number of characteristics such as low coupling, abstraction, reusability, autonomy, and interoperability. These architectures have traditionally relied on a set of specifications (SOAP, WSDL) that

---

\* This thesis project is being partially supported by the FPU scholarship program, granted by the Spanish Ministry of Education, Culture and Sports (FPU15/02980) and the European Commission (FEDER), the Spanish and the Andalusian R&D&I programs (grants TIN2015-70560-R (BELI) and P12-TIC-1867 (COPAS))

allow varying degrees of granularity and provide complex interfaces that often require heavy deployment infrastructures.

In recent years, there has been a trend towards a new architectural style that has been called microservice. This style requires that each component (a microservice) can evolve, scale and deploy independently to the rest, increasing the flexibility of the system as a whole. This architectural style is employed by high-performance commercial systems architecture such as eBay, Amazon and Netflix [8].

A common element of these architectures in particular, and, in general, when defining and implementing microservices is that they follow the RESTful paradigm. This paradigm provides a unified approach to identify the granularity and operational interface of microservices that has a high degree of extensibility. In particular, RESTful provides a lighter approach for building, deploying and cloning microservices more effectively.

RESTful systems provide numerous advantages in terms of elasticity, fault tolerance and flexible architectures design [3]. Both industry [7] and academia [1] agree to identify the management and evolution of services as a key element to achieve a more agile integration of the systems and to have a technological infrastructure that responds quickly to the business needs. These challenges are magnified in the context of microservice architectures since the independent life cycle of each microservice must be coordinated as part of a service catalog that has a higher growth rate than that of classical architectures [9].

As a first step in solving this problem, with a similar approach that in other industries, SaaS providers use the Service Level Agreement (SLA) concept to delimit the functionality and guarantees to which they commit to their customers. However, although formal specifications have been proposed for the definition of SLAs that arise in the context of the classic SOAs (WS-Agreement [2]), providers usually have an ad-hoc approach with a low degree of automation. This approach confirms the fact that the SaaS industry has not incorporated the idea of an SLA model that can be implemented within the infrastructure as a decision mechanism when it comes to supporting the SaaS business model. This instrumentation would be of special interest in RESTful microservice architectures in providing an automated governance framework for the service catalog and regulating the behavior of each component in the context of the agreements reached with each client.

## 2 Research challenges

The thesis project focuses on the study of the ways to improve and automate the management of microservice architectures based on RESTful systems regulated by SLAs through the definition of methodologies, techniques and tools. Specifically, we have identified four main challenges:

- **D1: Establishing a sufficiently expressive specification for the description of RESTful microservices regulated by advanced SLAs**

This challenge would be focused on defining a specification that has the necessary elements to support two main requirements: i) Establishing the service properties necessary to regulate government infrastructures that allow aligning the behavior of the system with the business model; in this way, these properties would guide the automated decision making in the different components of the architecture (e.g. their scalability strategies); ii) Defining a price model that has sufficient degree of expressiveness to contemplate the actual variability of the purchasing processes taking into account aspects such as discounts or compensation [6].

From the research context point of view, this challenge would represent the extension of the industrial specification for RESTful systems most used (OAI) to meet the requirements presented.

- **D2: Develop a catalog of SLA analysis and management operations to support the governance of micro-service architectures.**

This challenge would address the characterization of the requirements to support the government of real SaaS scenarios regulated by SLAs with micro-service architectures. In particular, these scenarios can be classified into two types: i) End-user information systems containing a backend-frontend (applications) schema; ii) Integrated information systems that have only one backend component (APIs).

These two typologies contain specific requirements to be characterized as part of the challenge in order to define a complete set of SLA analysis and management operations that allow the extraction of adequate information to support automated decision making in the provision of services and definition of business strategies.

- **D3: Implement a SLAs management ecosystem to support the government of RESTful microservices.**

Based on the catalog identified in D2, this challenge aims to address the evolution of current SLAs management libraries to an ecosystem scheme articulated in microservices as new components within the Governify platform. This ecosystem would allow integrating SLAs defined based on the specification proposed in D1 to achieve an automated SaaS instrumentation with independent domain tools.

- **D4: Consolidation of the Governify platform to validate the proposal in industrial environments.**

Since Governify has been designed to avoid ad-hoc approaches in SaaS governance, it has been designed as an open set of components that define a unified and homogenous interface to foster the development of new research and industrial ICT projects.

### 3 Preliminary results

In particular, during the first thesis year we have addressed the contribution C1, giving as a result the following activities:

- **C1.1: Technical workshops** dealing with the technological ecosystem that is closely related to the thesis topic: i) *REST APIs Security* invited talk for master students; ii) *Docker* and *Kubernetes* workshop for master students. iii) Technical meetings of *servitization* given to nodes of the RCIS network in order to share the research results with another nodes.
- **C1.2: Conferences**, both national and international: **C1.2.1** *Towards SLA modeling in RESTful APIs*, presented at the JCIS national conference; **C1.2.1** *An analysis of RESTful APIs offerings in the industry*, accepted at 15th IC-SOC edition.
- **C1.3: Journals**: we are preparing a journal extension for the paper accepted on IC-SOC as well as we are incepting another one on the specification of SLAs for RESTful APIs.

## 4 Timeline

A four-year work plan divided in in four stages is proposed for addressing the identified challenges. Each stage is related to a certain contribution (C) and a expected paper (P). They are as follows:

- **C1**: facing challenge D1. This contribution will be carried out in the context of a joint project between the research group and a local IT company.
  - **P1**: *Specification and support of RESTful APIs governed by advanced agreements.*
- **C2**: facing challenge D2. We will be able to define various governance scenarios in order to support the complete lifecycle of an API-based SaaS system: ranging from design, construction and testing to the evolution of the system and the business model.
  - **P2**: *Governify: an ecosystem for SaaS management driven by SLAs.*
- **C3**: facing challenge D3. This task would be carried out through a decoupled microservice architecture, so that an SaaS governance ecosystem (AME) needs to be developed.
  - **P3**: *Operations Catalog for RESTful APIs driven by SLAs.*
  - **P4**: *Evolution of a monolithic library to a microservice architecture.*
- **C4**: facing challenge D4. This contribution aims to provide new knowledge in the field as well as technology that can be used as within ICT projects for the research group.
  - **P5**: *Agreement Management Ecosystem for RESTful Services.*

Figure 1 shows the thesis project plan taking into account the most unfavorable average response times in the published journals, according to the experience of the group. We assume that an article will take three quarters in preparation state (P), one waiting state (E) and at least one in the state of applying revisions (R).

	1 <sup>st</sup> year			2 <sup>nd</sup> year			3 <sup>rd</sup> year			4 <sup>th</sup> year		
<b>Contributions</b>												
C1	■	■	■	■	■							
C2				■	■	■	■					
C3								■	■	■	■	
C4											■	■
<b>Papers</b>												
P1	P	P	P	W	R							
P2				P	P	P	W	R				
P3							P	P	P	W	R	
P4									P	P	P	W
P5										P	P	P
											W	R

Fig. 1. Thesis roadmap.

## 5 Relevance

This thesis project extends the scope of *BELI*, a Spanish national government project, whose objectives include the development of a governance system for hybrid services systems that integrate information systems (computer services) and teams of people. In this context, the initial stages have been covered with the Governify<sup>1</sup> ecosystem, which provides a set of microservices that support the design, monitoring and implementation of SLAs.

The objectives of the project are closely aligned with the IMPACT (H2020) European project, which aims to develop a set of techniques and tools to support the evolution of information systems to cloud platforms in the context of public administrations. In this context, the present thesis project will try to provide the instrumentation mechanisms that can be incorporated in the integration of public cloud APIs to verify the fulfillment of the providers guarantees and the regulations required by public administrations.

The relevance in the industry of the thesis topic is evidenced in the numerous ICT projects. Specifically, we highlight *SLA@OAI* project, developed in conjunction with *Icinetic*, which aims to develop an operational extension to define SLAs within the framework of the Open API Initiative (OAI) specification<sup>2</sup>, which currently represents the reference standard, widely used in the industry, to define functional interfaces of RESTful services.

Furthermore, several European projects have been framed by this idea of service governance (SLALOM, SLA-Ready, CELAR, S-CUBE, RESERVOIR and SLA@SOI). These projects approach the topic of the Service-Oriented Applications (SOA) governance from different perspectives, but all of them place SLAs as a key element for the advanced management of infrastructure[4].

<sup>1</sup> <http://www.governify.io/>

<sup>2</sup> <https://www.openapis.org/>

## References

1. Stephan Aier, Philipp Offermann, Marten Schönherr, and Christian Schröpfer. Implementing Non-functional Service Descriptions in SOAs. In *Trends in Enterprise Application Architecture*, pages 40–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 11 2006.
2. Alain Andrieux, Karl Czajkowski, A. Dan, Kate Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web Services Agreement Specification (WS-Agreement). Technical report, Open Grid Forum, 2004.
3. Bruno Costa, Paulo F. Pires, Flávia C. Delicato, and Paulo Merson. Evaluating REST architectures-Approach, tooling and guidelines, 2 2014.
4. Elisabetta Di Nitto, Carlo Ghezzi, Andreas Metzger, Mike Papazoglou, and Klaus Pohl. A journey to highly dynamic, self-adaptive service-based applications. *Automated Software Engineering*, 15(3-4 SPEC. ISS.):313–341, 9 2008.
5. D. Georgakopoulos and Mike P. Papazoglou. *Service-oriented computing*. 7 2009.
6. Sahil Kansal, Gurjeet Singh, Harish Kumar, and Sakshi Kaushal. Pricing Models in Cloud Computing. In *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies - ICTCS '14*, pages 1–5, New York, New York, USA, 2014. ACM Press.
7. Mansour Kavianpour. SOA and Large Scale and Complex Enterprise Transformation. In *Proceedings of the 5th international conference on Service-Oriented Computing*, pages 530–545, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
8. T. Mauro. Microservices at Netflix: Lessons for Architectural Design.
9. Sam Newman. *Building Microservices*. O'Reilly Media, 2015.
10. Mike P. Papazoglou and Willem-Jan van den Heuvel. Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal*, 16(3):389–415, 7 2007.