SeaClouds Project

D2.3.3 - Periodic Standardization report

Project Acronym	SeaClouds
Project Title	Seamless adaptive multi-cloud management of service-based applications
Call identifier	FP7-ICT-2012-10
Grant agreement no.	610531
Start Date	1st October 2013
Ending Date	31st March 2016

Work Package	WP2 Requirements Analysis, overall Architecture and Standardization
Deliverable code	D2.3.3
Deliverable Title	Periodic Standardization report
Nature	Report
Dissemination Level	Public
Due Date:	M30
Submission Date:	
Version:	1.0
Status	Final
Author(s):	Román Sosa (ATOS), Jacopo Soldani (UPI), Andrea Turli (Cloudsoft)
Reviewer(s)	Jacopo Soldani (UPI)

Dissemination Level

Project co-funded by the European Commission within the Seventh Framework Programme		
	Public	X
	Restricted to other programme participants (including the Commission)	
	Restricted to a group specified by the consortium (including the Commission)	
	Confidential, only for members of the consortium (including the Commission)	

Table of Contents

Executive Summary 4
1. The Importance of Standards in SeaClouds 5
2. Relevant Standards and Open Source Systems, and Our Work in SeaClouds 6
2.1 Application Topology 6
2.1.1 CAMP 6
2.1.2 TOSCA 6
2.2 Service Level Agreement Languages 7
2.3 WS-Agreement 10
3. Conclusions 12
4. References 13

Executive Summary

This deliverable, D2.3.3, summarises the main standards relevant to our work in SeaClouds, in particular outlining the areas where we are consuming the standards, contributing to the standards, or fostering adoption of the standards.

The structure of this document is the following:

- Section 2: This section outlines why we view standards as important to our research and to the wider community
- Section 3: This section outlines the key standards in four areas:
    - Application Topology
    - Service Level Agreements

1. The Importance of Standards in SeaClouds

Cloud computing is the widely adopted solution for modern distributed architecture. It offers many benefits to organizations by making information technology (IT) services available as a commodity and accessible from the web.

SeaClouds builds on top of cloud computing, and provides a platform that simplifies adopting and migrating to heterogeneous cloud environments. This is obtained by adopting a standard-based solution, which increases the interoperability, the portability across clouds and avoids the well-known problem of vendor lock-in.

In summary, some tangible examples of how standards can benefit cloud computing:

- Standards-based descriptions are attractive to users -- especially in enterprises -- as they do not develop on top of proprietary formats.
- Systems can be easily tested with existing descriptions– end-users are free to select solutions based on their functionalities, instead of being forced to use a specific solution because of vendor lock-in issues.
- Standards allow end-users to invest and work in features, instead of re-inventing description syntaxes.

2. Relevant Standards and Open Source Systems, and Our Work in SeaClouds

2.1 Application Topology

The leading standards in representing cloud application topologies are OASIS CAMP [1] and OASIS TOSCA [2]. SeaClouds is member of the OASIS standardisation group since the very beginning of the project, and the members of the SeaClouds consortium have attended and actively participated to various e-meeting of CAMP's and TOSCA's technical working groups.

During the last period of the SeaClouds project, we envisaged an increasing interest for the Simple Profile of TOSCA in YAML [3], and we decided to focus on such profile for modelling cloud-based application topologies[1].

2.1.1 CAMP

OASIS CAMP — Cloud Application Management for Platforms — describes a REST API for runtime management of applications. It supports investigating topology, sensors, and operations, and it supports deployment of applications. Deployments are presented as "plans" based on a declarative YAML syntax consisting of artifacts and services together with their requirements and characteristics.

2.1.2 TOSCA

OASIS TOSCA — Topology and Orchestration Specification for Cloud Applications — permits specifying portable cloud applications, and to automate their deployment and management. TOSCA permits describing the structure of a cloud application as a typed topology graph, whose nodes represents the application components, and whose arcs represent the dependencies among such components. Each node of a topology can also be associated with the corresponding component's requirements, its observable properties, the operations to manage it, and the capabilities it features (to satisfy other nodes' requirements). Inter-node dependencies associate the requirements of a node with the capabilities featured by other nodes.

[1] Further information about relevant research and interoperability projects for IaaS and PaaS cloud platform can be found in the previous deliverables D2.3.1 and D2.3.2.

In its original form, TOSCA provides an XML-based syntax to describe nodes and relationships, as well as to define their types and to associate them implementation details. OASIS has recently released a draft of a simpler YAML-based "profile", which is currently under public review.

Why Relevant

The description of an application's topology is essential as input to SeaClouds and useful as intermediate representations and presentation back to a user. By following open standards, we increase the potential for SeaClouds to interoperate with other tools on the inbound and outbound sides.

Our Contributions

The members of the SeaClouds consortium (from ATOS, Cloudsoft, POLIMI, UMA, UPI) are involved with both technical committees, suggesting improvements to the specifications based on our activity. A concrete example is the participation of SeaClouds members (from UPI) in the "Instance model ad-hoc group" of the TOSCA TC. The purpose of this ad-hoc group is to standardise the modelling of concrete instances of TOSCA applications, and SeaClouds members are providing feedback and suggesting improvements based upon the experience earned in the SeaClouds project. Fig. 1 shows a snapshot of a presentation of SeaClouds held during one of the meetings of the OASIS TOSCA TC.

Fig. 1 A snapshot of the main slide of a presentation of SeaClouds to the Instance Model Ad-Hoc group of the TOSCA TC.

Furthermore, many of the SeaClouds members have been contributing to the open source project "Apache Brooklyn", which provides an implementation of both CAMP (earlier version as of this writing) and TOSCA.

Finally, to ease the design of TOSCA-based applications (thus helping the widespread adoption of TOSCA) the SeaClouds Discoverer has been equipped with a web-based GUI (Fig. 2), which permits retrieving the TOSCA-based representation of available cloud offerings.

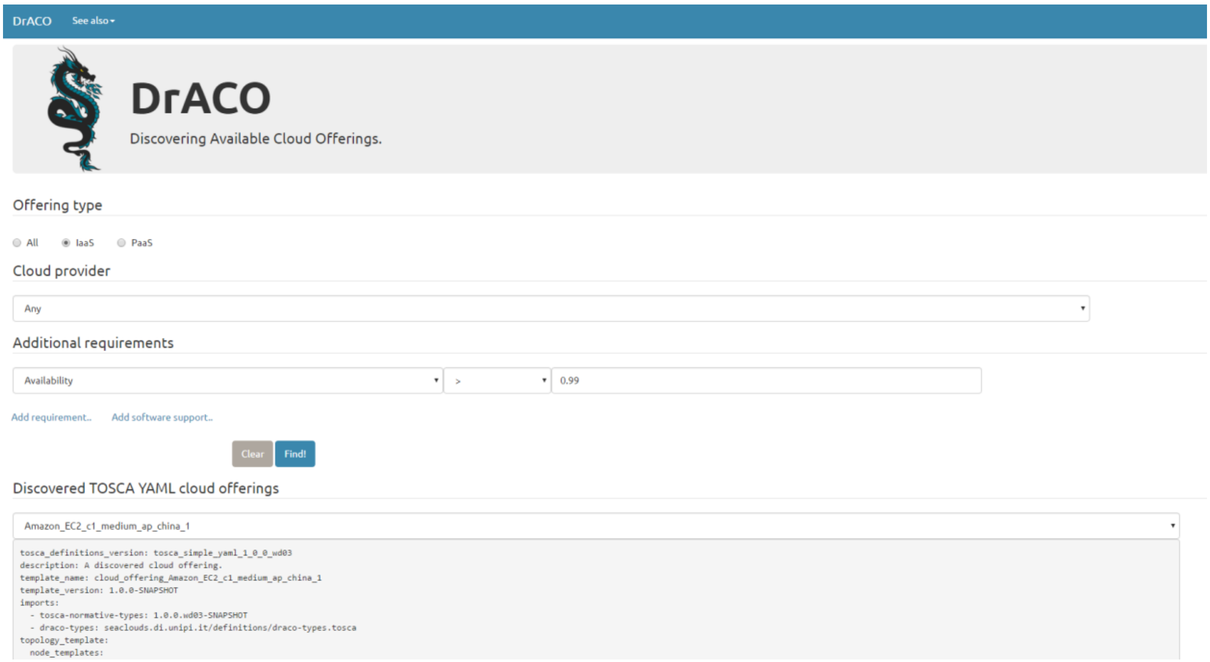

Fig. 2 A snapshot of DrACO, a web-based GUI for the Discoverer.

2.2 Service Level Agreement Languages

There are several techniques to describe service level agreements between Cloud Service Provider (CSP) and a Cloud Consumer (CC). QoS is usually defined in static manually managed SLAs.

However, systematic and dynamic languages to represent these contracts are becoming increasingly necessary because they make it possible to interpret these contracts so that, for instance, service providers can be chosen on the fly. SLAs define acceptable service levels to be provided by the CSP to its customers in measurable terms. The ability of a CSP to perform at acceptable levels is consistent among SLAs, but the definition, measurement and enforcement of this performance varies widely among CSPs. A cloud consumer should ensure that CSP performance is clearly specified in all SLAs, and that all such agreements are fully incorporated, either by full text or by reference, into the CSP contract.

In the context of SeaClouds the SLA management framework provides a generic end-to-end solution for SLA definition and operational management embracing multi-clouds services at IaaS and PaaS level. It provides an operational management with SLA composition and decomposition across functional and organizational Cloud domains; and covers the complete SLA and service lifecycle with consistent interlinking of planning and runtime management aspects; and can be applied to a large variety of industrial domains and use cases.

Standardization Needed:

- SLA description language
- SLA evaluation and Penalties

2.3 WS-Agreement

WS-Agreement (GFD.192) [4] defines a language and a protocol for advertising the capabilities of service providers and creating agreements based on creational offers, and for monitoring agreement compliance at runtime. This is supported by WS-AgreementNegotiation (OGF), which defines a protocol for automated negotiation of offers, counter offers, and terms of agreements defined under WS-Agreement-based service agreements.

SeaClouds explored additional SLA description models as well as negotiation strategies and protocols to enable an end-to-end SLA creation, the development of monitoring and feedback mechanisms to observe the commitments met by an SLA, and the development of adaption strategies to mitigate the effects of possible SLA infringements.

Why Relevant

SeaClouds requires a language for expressing quality of service, to be embedded in the abstract model supplied by the user and passed to the monitoring systems. Thus an understanding of the standards and available systems is very relevant to our work.

Our Contribution

SeaClouds selected WS-Agreement as the specification to comply in the implementation of the SLA Service[2].

SeaClouds is primarily a consumer of the WS-Agreement standard. We did not envision attempting to advance it, although there are several places in the specification that could have been extended or defined by domain specific languages. Therefore, we made use of

[2] Further information about other relevant initiatives for standardising SLAs in cloud computing can be found in the previous deliverables D2.3.1 and D2.3.2.

this property to achieve our goals. Concretely, SeaClouds reused Tower4Clouds for the evaluation of Service Level Objectives (SLOs), and defined a simple language to evaluate SLOs in the context of the SLA Service. On the other hand, the standard business values part corresponding to an SLO was not expressive enough for our requirements. For example, a simple business requirement like “I want to scale up in case two violations of this SLO occur in 10 minutes” is not possible in the standard penalty specification of WS-Agreement. As such, SeaClouds used the custom part of the business values to define a custom penalty format.

3. Conclusions

This deliverable describes the standardization strategy implemented by the SeaClouds project.

SeaClouds architecture has been aligned with two major standards for cloud interoperability: the OASIS CAMP (Cloud Application Management for Platforms) and TOSCA (Topology and Orchestration Specification for Cloud Applications), promoting them in research and industrial communities, although an increasing interest for OASIS TOSCA has been taken into account to be more appealing for enterprises that have recently showed a significant interest in that standardization effort.

The Application Topology work achieved all the three main objectives of the initiative: consume, contribute and foster adoption.

The Service Level Agreements work mainly by implementing the tools (standard consumption) and fostering the adoption of SLA languages in the cloud computing. This enables SeaClouds as an SLA management tool. SeaClouds complies with WS-Agreement and extends this specification where necessary to cover the SeaClouds needs.

References

[1] OASIS: Cloud Application Management for Platforms.
http://docs.oasis-open.org/camp/camp-spec/v1.1/camp-spec-v1.1.pdf

[2] OASIS: Topology and Orchestration Specification for Cloud Applications.
http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.pdf

[3] OASIS: TOSCA Simple Profile in YAML
http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/TOSCA-Simple-Profile-YAML-v1.0.pdf

[4] OGF: Web Services Agreement Specification (WS-Agreement)
https://www.ogf.org/documents/GFD.192.pdf