A framework to orchestrate security SLA lifecycle in cloud computing

8
Author(s)

Marco A. T. Rojas ; Nelson M. Gonzalez ; Fernando V. Sbampato ; Fernando F. Redígolo ; Tereza Carvalho ; Kazi W. Ullah ; M... View All Authors

Abstract:
Security issues of cloud computing environments are considered a major challenge for its full adoption. A Service Level Agreement (SLA) corroborates the shared management vision provided by the cloud computing paradigm, which can assist with related security issues. The necessity to address security requirements in cloud computing SLAs is considered important for both providers and consumers, along with the tools and mechanisms necessary to deal with these requirements. These issues are current research challenges; therefore, this paper aims at proposing a framework to orchestrate the management of cloud services based on security requirements defined by the SLA in an automated manner during its entire lifecycle. In addition, mechanisms to support the phases of the SLA lifecycle are proposed as a part of the framework. also the preliminary validation of the proposed framework. Finally, it is presented the integration of the framework with a cloud solution and in what manner the SLA lifecycle is supported by the framework.

Published in: Information Systems and Technologies (CISTI), 2016 11th Iberian Conference on

Date of Conference: 15-18 June 2016

INSPEC Accession Number: 16178584

Date Added to IEEE *Xplore*: 28 July 2016

DOI: 10.1109/CISTI.2016.7521372

ISBN Information:

Publisher: IEEE

SECTION I.
Introduction

Cloud computing providers deliver on-demand services based on shared and distributed infrastructure. This delivery approach is defined by the deployment and service models offered. The shared responsibilities and the segregation of the roles between cloud providers and consumers are also consequence of the adopted models.

Despite several benefits provided to consumers [1] and the increasing adherence to the services [2], the security issues of the cloud paradigm stand as a major challenge for its adoption in large scales [3]. According to Bouchenak [4], consumers aim at having at least equivalency between the security provided by the cloud and the one experienced by local environments. Besides, Huang [5] advocates that is necessary a joint effort between academia and industry to solve the pressing security issues in cloud.

SLA corroborates the shared management vision provided by deployment and service models of cloud. The cloud provider ensures application, data, and infrastructure based on the consumer requirements defined in the SLA [6]. However, this shared management is considered a security issue related to cloud management (governance, compliance and legal issues) also [7].

Unfortunately, the specifications and definitions of the security requirements for cloud computing SLAs are still in embryonic stages [8] because it has only recently been considered a necessity [4]. Luna [9] advocates for the need of cloud management architectures based on security SLA requirements.

To face the open challenge of cloud management architectures based on security SLA requirements, this paper proposes a framework to orchestrate the lifecycle of security SLA for cloud computing. The framework orchestrates the provisioning of cloud services and security mechanisms based on the security requirements defined by an SLA in an automated manner by following the four phases that compose the SLA lifecycle proposed by Rojas [10]. Furthermore, we present in what manner the framework can be transparently integrated to cloud solutions such as OpenStack, and support the phases of the SLA lifecycle.

This paper is organized as follows. Section II presents the concepts related to the security issues of SLA contracts for cloud computing domains, as well as the phases of an unified SLA lifecycle for cloud computing. Also, presents the related works to security SLA for the cloud computing context and gives a comparative analysis between the SLA lifecycle proposed and the related work. Section III presents the proposed framework in order to orchestrate the lifecycle of a security SLA by contemplating the components description and their relations, internally and externally. In addition, the integration with the cloud solution OpenStack and the SLA lifecycle are presented. Also, a preliminary validation based on functional requirements is presented. Finally, Section IV presents the conclusion and future work.

SECTION II.
Service Level Agreement

This section presents the necessity to address the security issues of SLA contexts for cloud computing domains. Also, the phases of the SLA lifecycle for managing the need to define the level of services, which is required by the consumer, are presented also. Furthermore, presents the related works to security SLA for the cloud computing context and comparative analysis between the SLA lifecycle and the related works.

A. Security Issues

Despite the security issues of cloud computing are considered similar to the ones presented in ICT domains [11] [12], the need to address security in SLA for cloud context was proposed only recently [4].

The current scenery of practices, obligations, recommendations and benefits related to addressing the security requirements in the SLA by the cloud providers and consumers were surveyed by Rojas [10]. Through the survey, the following challenges in cloud security SLA were identified: the architecture for managing security SLAs [8] [4], defining quantitative security metrics and not just qualitative metrics [13], security SLA representation and security service disclosure [14] [15]. Moreover, it was verified that the security aspects have been neglected in SLA contracts regarding the requirements specifications and its associated metrics. Furthermore, cloud providers do not have defined processes for managing the security requirements defined in the SLA and staff are not qualified for assisting and supporting customers during its definition.

B. SLA Lifecycle

A consolidated SLA lifecycle management for cloud context was proposed and discussed [10]. The management of the SLA lifecycle is divided by phases in order to achieve the defined level of service. Figure 1 illustrates this lifecycle,

Figure 1.
SLA lifecycle phases for cloud computing

The SLA lifecycle is composed of four phases and subtopics of interest and it has been explained in details by Rojas [10]. A short explanation of what comprises each phase is presented:

- Phase 1 - Definition and Specification: This phase is characterized by the specification and definition of the security requirements and parameters to be included in the SLA.

- Phase 2 - Negotiation and Deploy: In this phase, the financial conditions and acceptable level of services are traded and defined between cloud provider and consumer. The service is deployed to consumer based on requirements specified in the SLA. Moreover, sanctions are traded and defined for both parties in case of noncompliance with any of the agreed clauses.

- Phase 3 - Execution and Management: In this phase the required services are executed and managed in the cloud environment in compliance with the defined security SLA. The management performs tasks such as: real-time monitoring of the running instance, management of requirements to be compliant, emission of control reports, policy enforcement, corrective actions, reactive actions to be adopted and violations control of the required service.

- Phase 4 - Termination/Evaluation: At this phase the contract termination process is performed due to either contract expiration, violation of the contractual agreement or consumer request. The infrastructure deployed is disabled in addition to the release of the allocated resources and revocation of user access and grants. Then, billing is issued.

The phases of the SLA lifecycle assist the management concerns of cloud computing. Moreover, the comprehension of phases and required controls in order to improve the management of the shared responsibilities by consumers and cloud providers. Furthermore, the application of the SLA lifecycle for managing security concerns is fundamental for the adoption and evolution of the cloud paradigm.

C. Related Work

The necessity for addressing security requirements in cloud computing SLAs has become a research challenge for the academic community, standard institutes and enterprises. In this context eight related works were identified and analyzed in respect to SLA lifecycle.

- Garcia [8] proposed a framework to develop security metrics for evaluations by cloud providers, taking into account cloud models of service and deployment. This work contributes to Phase 1 of the SLA lifecycle because it addresses the issues related to the definitions and specifications of the security parameters.

- Luna [13] presents a set of metrics used to quantitatively compare security SLAs in cloud contexts. The CSA information base called STAR (Security, Trust & Assurance Registry) was a case study to compare the security practices applied by different cloud providers. Therefore, contributing to Phase 1 with security metrics that can be applied in SLAs. Also, it can be used to discover service providers and service offerings that are related to security features applied to Phase 1. In addition, it may be employed to assist negotiations (Phase 2), since enables customers to evaluate and compare the security features being offered by cloud providers.

- Luna [9] presents a quantitative and qualitative method to compare security requirements defined in the security SLA between cloud providers and consumers. This method is based on the QPT (Quantitative Policy Trees), an approach that enables the evaluation of security requirements by categories. This approach facilitates the comparison between security offers and service providers. The work contributes to Phase 1 due to the possibility of comparing security features by applying quantitative and qualitative metrics. In addition, it also contributes to Phase 2, since comparative results can be used in the negotiation process.

- Ullah [16] presents an automated tool to evaluate compliance levels related to security requirements adopted by cloud providers. This work supports Phase 1 because it identifies the security services offered by cloud providers, the security mechanisms applied and the related parameters.

- Silva [17] proposes a method to evaluate the provisioning of cloud resources based on hierarchical security metrics. The hierarchical metrics are developed applying the GQM (Goal-Question-Metric) approach. Thus contributing to Phase 2 due to the capacity of service provision based on security requirements and to Phase 1 due to generated security metrics that can be specified in SLAs.

- Rank [18] proposed an approach to deploy cloud resources based on the user roles and security policies defined in the SLA. The work contributes to Phase 2 due to the deployment of resources based on security requirements defined in the SLAs.

- Jegou [19] presented the VEP (Virtual Execution Platform) a component responsible for provisioning IaaS resources based on the requirements defined in the SLA. Furthermore, VEP is responsible for monitoring the instance deployed during the entire lifecycle. The work contributes to Phase 2 due to deployment resources based on security SLA and to Phase 3 due to the monitoring of activity during the execution of the deployed instance.

- Ferreira [15] proposes an architecture for monitoring security based on SLA for the IaaS service model. The monitoring approach avoids the necessity of installing a monitoring agent on the host. The work can support the Phase 3 due to the process for monitoring the execution of the deployed resources. Also, it contributes to Phase 2 due to deployment of resources based on security SLAs, and to Phase 1 due to an approach for representing security policies in SLAs.

The Table I summarize the relation between the related works found in the literature and the SLA lifecycle phases.

Table I. Relation between related works and sla phases

Related Work	Phase 1	Phase 2	Phase 3	Phase 4
Garcia [8]	1			
Luna [13]	1	2		
Luna [9]	1	2		
Ullah [16]	1			
Silva [17]	2	1		
Rank [18]		1		
Jegou [19]		1	2	
Ferreira [15]	3	2	1	
Total	6	6	2	0

It can be verified that Phase 1 is the most addressed, being the focus of four works and related in three others. Meaning that to define and to specify security metrics and security parameters is important and fundamental for the security management of cloud solutions. Methods to define this metrics and parameters are equally important. Phase 2 is the second most addressed with three related works focusing on it and other three related indirectly. The provisioning and deployment of cloud resources based on security metrics were the challenges faced by these works. Phase 3 has one related work to it directly and another related indirectly. The work is focused on the necessity for monitoring security. It can be verified that more development of this phase is needed in terms of mechanisms and methods for managing security in cloud environments, along with monitoring and auditing tools. Phase 4 has no contribution. It can be considered a consequence related to the necessity of evolving the other phases. However, it does not justify the lack of related works despite the issues appointed. In this case it can be considered a fertile area of future works. From the results of our analysis it can be verified the necessity of more research in the security SLA area and solutions related to Phases 3 and 4 due to their importance in the security context. Furthermore, the results corroborate the necessity of improving security in cloud environments as appointed by consumers and providers.

Based on related works it can be verified that there are no proposals that encompass the four phases of the lifecycle, just proposals to a specific phase. To meet this need a framework to orchestrate the security SLA lifecycle phases in an automated and integrated manner is proposed, as advocated by Bouchenak [4] and Garcia [8]. The proposed framework aims to fill this gap related to cloud management based on security requirements advocated by the academia and appointed as a necessity by cloud consumers and providers.

SECTION III.
Security SLA Framework

This section presents the details of proposed framework and how it supports the phases of the SLA lifecycle, also their integration with the cloud solution adopted. By analyzing the related works, it was verified the necessity for solutions to orchestrate the lifecycle of security SLAs. In this context, a framework is proposed to orchestrate the security requirements addressed in the SLA lifecycle in

a framework is proposed to orchestrate the security requirements addressed in the SLA lifecycle in an automated and integrated manner, also solutions for individually supporting each phase of the SLA lifecycle are presented.

The framework is composed of two sides, user and cloud provider. The user side has available interfaces for communicating with the cloud provider side. The cloud provider side comprises the proposed framework and their integration with the cloud solution infrastructure. Figure 2 illustrates the framework.

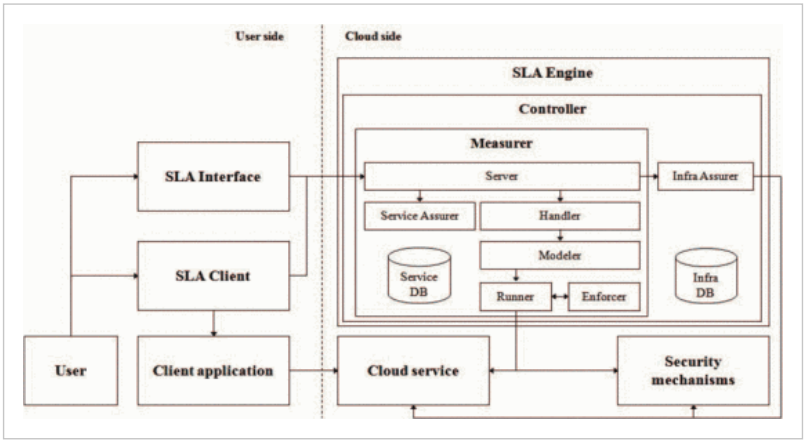


Figure 2.
Security SLA framework proposed for cloud computing

The user side provides the interface to allow users to access the cloud side and define the security SLA required. It is compounded of three communication interfaces to support the interactions between both sides:

- *SLA Interface*: It is the GUI used to specify the security parameters and mechanisms to be applied by the cloud provider for the deployment the service requested. This interface could be integrated with the Client Application.

- *SLA Client*: It was designed to be a command line tool for testing the SLA engine and their integration with other functions during the framework development. At the end, it can also be used as an auditing tool by both user and service provider, or even an independent auditor.

- *Client Application*: It is the primary interface (GUI and CLI) provided by the cloud solution to the user (e.g. Horizon the web-based management interface for OpenStack).

The cloud provider side contains the engine for orchestrating the management of the lifecycle of cloud services based on a SLA specified according with user, cloud infrastructure and security mechanisms related. The SLA Engine is composed of nine modules:

- *Controller*: responsible for managing the execution of modules. Ensures that the correct order of operations is performed and provides access to the database (*Infra DB*).

- *Measurer*: is the interface for writing and reading information related to the operations performed and accountability. It also provides access to the database (*Service DB*).

- Server: is the server side *SLA Interface* and SLA *Client*. Listens for incoming requests and updates information related to operations performed by the *SLA Engine*.

- *Handler*: Manages the incoming requests related to services, parameters and security mechanisms to be applied in the deployment of the service.

- *Modeler*: Translates the input parameters, related to services and security mechanisms, into adequate data structure for managing the deployment of service. It can be customized to generate the proper data structure in order to support the cloud solution adopted (e.g. OpenStack, OpenNebula, Eucalyptus, CloudStack, etc.). Then, more than one cloud solution could be adopted.

- *Runner*: Dispatch run orders, executing the adequate programs, sequences and security mechanisms. The executing task could also support distinct cloud solutions.

- *Enforcer*: Enforces the right commands and execution order (workflow). The order of execution and parameters are hashed and stored to support compliance and audit tasks.

- *Service Assurer*: Assures that services are being provided according to the specified security SLA and following the defined workflow. In addition, control information is stored in the database (Service DB). The Service DB stores the information about the service usage (requests received, operations performed, information assurance, user data). Moreover, the information stored is hashed to guarantee its integrity. Nonetheless, it is a data source that can support audit tasks.

- *Infra Assurer*: Collect information related to services and security mechanisms provided by the cloud available through the SLA Interface. It provides assurance on assessments of the hardware and software pieces regarding trust and correct operation. If the cloud infrastructure has TPM (Trusted Platform Module) and vTPM (Virtual Trusted Platform Module) resources, their primitives can be used to improve the assurance process [20]. In addition, control information is stored in the database (Infra DB). The Infra DB stores information related to the engine operations and cloud infrastructure resources (services, security mechanisms, trusted hardware).

The cloud provider side supplies cloud solutions, services and security mechanisms (*SecMecs*). The *SecMecs* can be native mechanisms related to the services provided or external designs provided by third parties integrating with the cloud solution. The current integration process between the cloud resources and framework are not automated. The automatic discovery of resources and their integrating to framework is a challenge that needs research.

The framework proposed here is independent from the cloud solution adopted for providing services and security mechanisms to users. The modules *Modeler, Runner* and *Infra Assurer* can support this feature. Besides, the framework can orchestrate cloud services and security mechanisms provided by more than one cloud solution.

A. Lifecycle Integration

The framework was designed to support the SLA lifecycle phases. The relation between the lifecycle phases and framework modules, along with the lifecycle workflow is illustrated in the Figure 3.

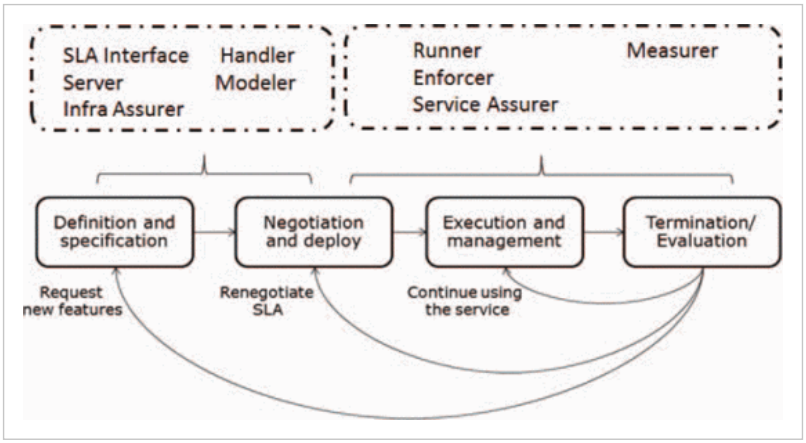


Figure 3.
Relation between SLA lifecycle and proposed framework

It can be verified that the *Definition and Specification* phase support the customer activities, to compose a contract for the desired SLA, which has to be established and accepted by the both parties, service provider and consumer. This activity is supported by the framework modules the *SLA Interface, Server* and *Infra Assurer*. After defining the SLA, it has to be *Negotiated* and then *Deployed* in the cloud solution. For the *Negotiation* activity the modules applied by previous phase are referred to. For the *Deployment* activity the framework provides the modules *Runner, Enforcer* and *Service Assurer*. After the deployment of the services and security mechanisms related, the requested service is applied. The adequate deployed environment has to be *Executed* and *Managed* to ensure that the security levels defined are attended to. To support the phase activities the framework provides the modules *Runner, Enforcer* and *Service Assurer*. For the service release instances, there is a *Termination* phase followed by *Evaluation* related to accounting and billing. To support *Termination* the framework provides the modules *Runner, Enforcer* and *Service Assurer*. To support *Evaluation* the framework uses the module *Service Assurer*. Considering the workflow of a SLA lifecycle, it is possible to migrate from the last phase back to the first phase (e.g., if the customer requests new features or if the contract has to be changed). It is also possible migrate back to the second phase, if the SLAs have to be renegotiated. Finally, it is possible migrate back to the third phase if the customer simply wants to continue using the service with the previously set SLA definitions.

B. Openstack Integration

To support the proposed framework, a cloud solution is necessary in order to provide services and the security mechanisms related. The integration process between the cloud solutions require open access to information related to internal architecture and source code of services provided (e.g., virtual machines, storage, images, etc). Each service possesses its internal communication and security mechanisms, and was identified through source code analysis. The OpenStack Icehouse release series was nominated as cloud solution to support this integration due to their importance in the cloud computing scenery, for providing the necessary resources and for its open source code. Furthermore, OpenStack is considered the largest most active community when compared to solutions such as OpenNebula, Eucalyptus and CloudStack [21].

The Nova service source code, which provides compute resources (VM) to users, was analyzed in relation to their existing security mechanisms. Two functionalities and four security mechanisms are going to be presented. They are examples of native, non-native, and proposal of new mechanisms. The security mechanisms presented had their impact analyzed in terms of the CIA triad [22] based on the NIST security standards [23] [24]. Figure 4 illustrates the mechanisms and their security analysis.

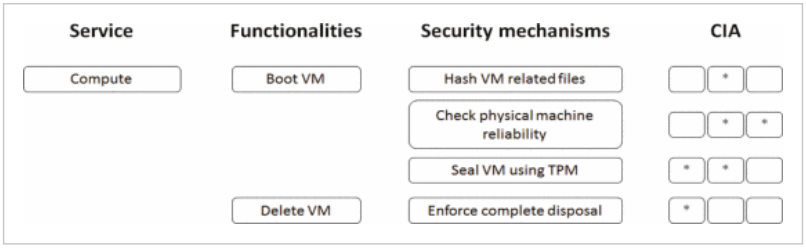


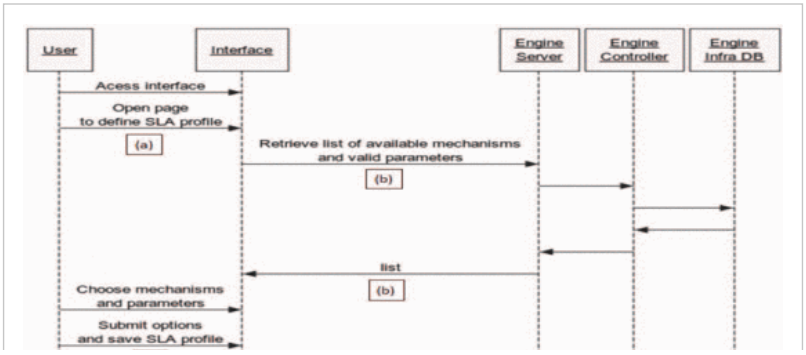
Figure 4.
Security mechanism for nova service and their analysis

Three functions of the *Boot VM* functionality can be improved. The first is hash VM related files, this action increases the service integrity. This is an example of native mechanism. The second checks the reliability of the physical machine (TPM-based infra), this action increases both integrity and availability. The third seals the VM by using TPM or vTPM (VM assurance), this action increases service confidentiality and integrity. The cloud infrastructure may have physical computing resources TPM-based for deployment to customers and vTPM to support the VM deployment. The TPM primitives (e.g. attestation, binding, sealing, etc.) can support the nonnative second and third functions. The *Delete VM* functionality can be developed to enforce the complete disposal process, this action increases service confidentiality. This is an example of security mechanism that can be developed by the cloud provider or third parties.

The same analysis can be made to other OpenStack services related to security mechanisms and their security impacts. The security mechanisms can be applied by the framework throughout the entire SLA lifecycle, supporting the need for management and enforcement of contracted security requirements.

To present the integration of the framework proposed (*SLA Interface* and *SLA Engine)* within cloud service (Nova) and security mechanism (TPM-based infra) the sequence diagrams were elaborated. The cloud operations (define SLA and deploy VM) and security mechanisms (TPM-based infrastructure and VM assurance) were selected to produce the sequence diagrams that illustrate the modules operations. The framework side of the *SLA Interface* defines a SLA and selects the *SLA Engine*.

1) Sequence Diagram - Define SLA
The process of SLA definition is illustrated in Figure 5.



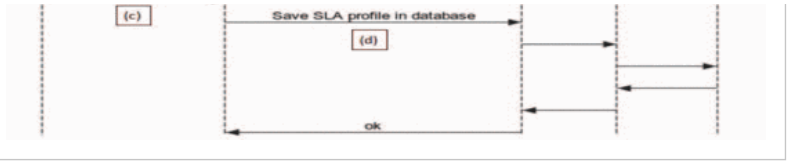


Figure 5.
Sequence diagram of define SLA process

The process is illustrated by the following steps: (a) user accesses the *SLA Interface* (GUI) and opens the page for defining the SLA. (b) The interface retrieves the information to the user from the *SLA Engine* by presenting the available security mechanisms related to the Nova service. The user then sets the desired SLA definitions. (c) The set configuration of the security mechanisms and parameters related are submitted by the user. (d) The SLA is configured as a profile which is stored in the *Engines Service DB* and the provider confirms the operation to the user.

2) Sequence Diagram - Deploy a VM
The VM deployment process is presented in two phases. The first is a definition of the features related to VM is illustrated in Figure 6.

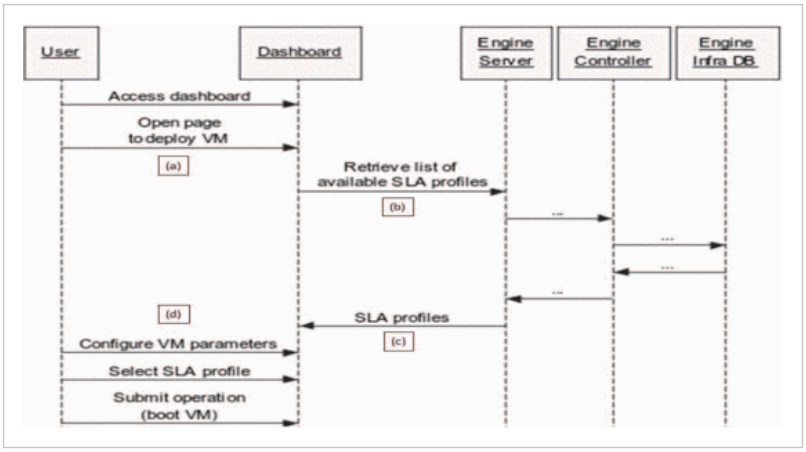


Figure 6.
Sequence diagram of deploy VM process — first phase

The process is illustrated by the following steps: (a) the user accesses the OpenStack Dashboard and open the create VM option. (b) The interface retrieves information from the SLA profile. (c) The *SLA Engine* queries the *Service DB* and sends the defined and available profiles. (d) The user sets up the VM parameters (flavor, memory, image, etc.) and the desired SLA profile. Finally, the user submits the request to OpenStack. In the second phase the VM parameters and SLA profile submitted are handled by the SLA Engine. This phase is illustrated in Figure 7.

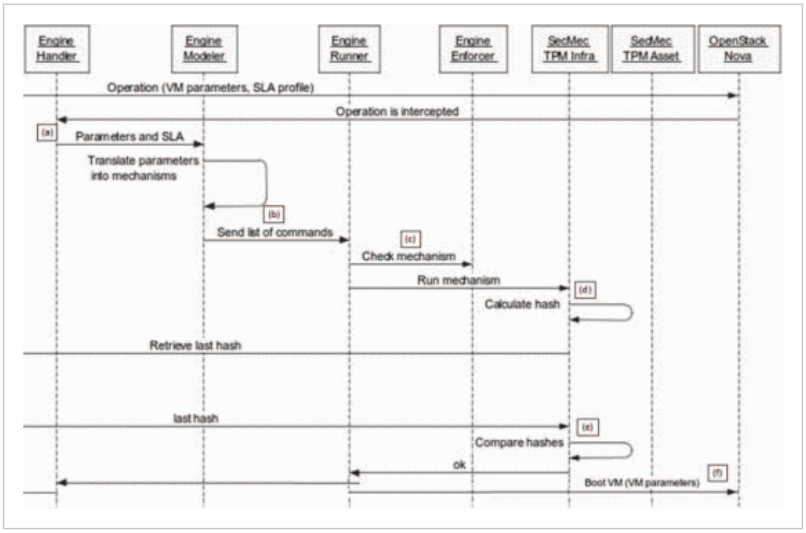


Figure 7.
Sequence diagram of deploy VM process - second phase

The process is illustrated by the following steps: (a) the request is intercepted by *Handler*, which reads the SLA profile information and sends it to *Modeler*. (b) *Modeler* receives the parameter of

the SLA profile and defines the adequate commands; then, in the correct order, sends it to be run by *Runner*. (c) *Runner* executes the commands while checking with the *Enforcer*, which guarantee that the right commands are being executed in the right order (in compliance with the SLA). (d) *Runner* executes the command which gathers the information of infrastructure (hardware and software) and hashes it. The information is processed by the TPM module and stored in *Infra DB*. This hashed information is also compared to the previously stored database in order to verify modification occurrences of the infrastructure. (e) After confirmation, *Runner* executes the command to Boot VM with the requested parameters.

Through the sequence diagrams are presented how the user defined the SLA and how this SLA is applied within the Nova cloud services in order to deploy a VM with security features provided by the mechanism. Then, the communication and interaction between the framework modules and cloud solution are presented.

C. Preliminary Validation

The proposed framework was designed based on functional (FR) and non-functional (NFR) requirements defined by the team members attending the necessity to orchestrate the deployment of security SLAs for the cloud environment. The validations of requirements (VFR/VNFR) are presented:

- FR1: The solution must provide a graphical interface to allow easy interaction between the user and the mechanisms, providing visualization tools to define the required SLA levels.
- VFR1: To satisfy the requirement the *SLA Interface* and SLA Client was designed, providing the required interface.
- NFR2: The solution must provide a command line interface to allow easy integration between itself and other programs or scripts.
- VNFR2: To satisfy the requirement the *SLA Client* was designed, providing the required interface.
- NFR3: The solution must provide a transparent communication interface to interact with any type of interface (e.g. via HTTP requests).
- VNFR3: This requirement was satisfied by the *Server* module that provides a transparent communication interface.
- FR4: The solution must handle the input parameters defined by the user, thus defining which security mechanisms and cloud services are necessary to provide the SLA levels specified by the user.
- VFR4: To satisfy the requirement the *Handler* was designed, providing the adequate services and mechanisms to be applied in the deployment of the service.
- FR5: The solution must allow integration to multiple cloud environments (e.g. OpenStack, OpenNebula, AWS), without being restricted to any specific environment.
- VFR5: This requirement was satisfied by the *Modeler* module that provides a translate mechanisms to the adequate cloud solution.
- FR6: The solution must define the execution order of commands while providing the SLA levels specified.
- VFR6: To satisfy the requirement the *Runner* was designed, providing the interface to execute the right commands to the cloud solution adopted.
- FR7: The solution must provide assurance regarding both the execution order and the SLA levels actually delivered, and the assurance must cover both how the service is delivered and how the infrastructure is deployed.
- VFR7: This requirement was satisfied by the *Service Assurer* and *Infra Assurer* modules that provide the adequate mechanism to guarantee that execution order and service comply with SLA agreed. Through these modules is possible to audit the environment, verifying that the SLA was complied by the cloud provider.
- FR8: The solution must provide accountability related to its own operations and also the services and actions performed in the cloud, and the information should be stored in

adequate databases.

- VFR8: To satisfy the requirement the *Service Assurer* and *Infra Assurer* modules was designed, each module store the operations performed by the cloud in the respective database, providing the accountability of the environment.

It can be verified that the proposed framework can support the definition of security requirements through SLA and orchestrate the deployment of cloud service and security mechanism based no defined SLA. Beyond that the framework can provide the auditability, which is an interesting and requested feature for cloud providers and consumers.

SECTION IV.
Conclusions and Future Work

The need for addressing security requirements in cloud computing SLA, as a fundamental aspect of development, consolidation, advancement and management of cloud technology was presented. The process defined by the SLA lifecycle can assist in the management of cloud services due to the shared responsibilities between providers and consumers required by the cloud environment. This management is accomplished by well-defined phases and it can also improve the control and mitigation of security issues desired for this environment, thereby reducing the risks from both sides. The proposed framework was presented and validated as a solution to manage the lifecycle of security SLAs for the cloud context in an automated and flexible manner. Thus, improving the security and compliance required by consumers. The outcome of the analysis advocates that security research requirements in cloud computing are currently at embryonic stages. Therefore, efforts from the security communities to adequately define those security requirements for cloud environments, as well as the metrics to properly measure cloud services, are needed in order to improve the desired management of security issues required by consumers and cloud providers. For future work, it is necessary to fully implement the framework, test and evaluate the results, along with their integration with OpenStack. Furthermore, automated discovery and integration of security mechanisms are interesting topics for future research.

Keywords

IEEE Keywords
Security, Cloud computing, Measurement, Context, Monitoring, Contracts, Computational modeling

INSPEC: Controlled Indexing
security of data, cloud computing, contracts

INSPEC: Non-Controlled Indexing
cloud solution, security SLA lifecycle, security issues, cloud computing environments, service level agreement, security requirements, cloud services management

Author Keywords
lifecyle, security, sla, cloud computing, framework

Authors

Marco A. T. Rojas
Computing and Digital Systems Department, School of Engineering - University of São Paulo - Brazil

Nelson M. Gonzalez
Computing and Digital Systems Department, School of Engineering - University of São Paulo - Brazil

Fernando V. Sbampato
Computing and Digital Systems Department, School of Engineering - University of São Paulo - Brazil

Fernando F. Redígolo

Computing and Digital Systems Department, School of Engineering -
University of São Paulo - Brazil

Tereza Carvalho
Computing and Digital Systems Department, School of Engineering -
University of São Paulo - Brazil

Kazi W. Ullah
Ericsson Research Ericsson, Stockholm, Sweden & Jorvas, Finland

Mats Näslund
Ericsson Research Ericsson, Stockholm, Sweden & Jorvas, Finland

Abu Shohel Ahmed
Ericsson Research Ericsson, Stockholm, Sweden & Jorvas, Finland

Related Articles

SMICloud: A Framework for Comparing and Ranking Cloud Services
Saurabh Kumar Garg; Steve Versteeg; Rajkumar Buyya

Collaboration-Based Cloud Computing Security Management Framework
Mohemed Almorsy; John Grundy; Amani S. Ibrahim

Security SLAs for Federated Cloud Services
Karin Bernsmed; Martin Gilje Jaatun; Per Hakon Meland; Astrid Undheim

Security as a Service Using an SLA-Based Approach via SPECS
Massimiliano Rak; Neeraj Suri; Jesus Luna; Dana Petcu; Valentina Casola; Umberto Villano

Cloud monitoring for optimizing the QoS of hosted applications
Khalid Alhamazani; Rajiv Ranjan; Fethi Rabhi; Lizhe Wang; Karan Mitra

SecAgreement: Advancing Security Risk Calculations in Cloud Services
Matthew L. Hale; Rose Gamble

Security Services Lifecycle Management in On-Demand Infrastructure Services Provisioning
Yuri Demchenko; Cees de Laat; Diego R. Lopez; Joan A. Garcia-Espin

REST-Based SLA Management for Cloud Applications
Alessandra de Benedictis; Massimiliano Rak; Mauro Turtur; Umberto Villano

A Certification-Based Trust Model for Autonomic Cloud Computing Systems
Marco Anisetti; Claudio A. Ardagna; Ernesto Damiani

A service level agreement framework of cloud computing based on the Cloud Bank model
Fengchuan Zhu; Hao Li; Joan Lu