Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Multi-Level SLA Management Architecture for Cloud Computing

Ritu Juneja*, Deepti Sharma

Computer Science Department, ACTM,
Palwal, India

Abstract— The SLA for a service must be based on realistic, achievable targets (e.g. for performance and availability), and the achievement of these targets depends on the performance of the internal and external services that underpin the delivery of the main service. Putting it another way, SLAs must reflect the levels of service actually being delivered or that can be delivered. They are about what can be done rather than what we would like to be done. If a customer requires a different level of service, this would normally be dealt with by raising a Service Level Requirement. In this chapter, we present multi-level SLA management architecture and we discuss fundamental concepts of the framework and detail its main architectural components and interactions.

Keywords— cloud computing, SLA, SLM, security.

I. Introduction

Cloud computing provides various services on virtual machines allocated on top of a large physical machine pool which resides in the cloud. Cloud computing comes into focus only when we think about what IT has always wanted – away to increase capacity or add different capabilities to the current setting on the fly without investing in new infrastructure, training new personnel or licensing new software. The basis of cloud computing is to create a set of virtual servers on the available vast resource pool and give it to the clients. Any web enabled device can be used to access the resources through the virtual servers. Based on the computing needs of the client, the infrastructure allotted to the client can be scaled up or down. From a business point of view, cloud computing is a method to address the scalability and availability concerns for large scale applications which involves lesser overhead. Since the resource allocated to the client can be varied based on the needs of the client and can be done without any fuss, the overhead is very low.

Cloud computing provides three service models that provide different levels of control and security. These levels are, in decreasing order of control and increasing order of security:

1. Infrastructure as a Service (IaaS);
2. Platform as a Service (PaaS); and
3. Software as a Service (SaaS)

Each service model can be seen as a layer with IaaS at the base allowing full control of resources and storage, PaaS in the middle allowing development on an existing platform and finally, SaaS providing limited development opportunities but having appeal to end users. Each layer provides different development and/or deployment opportunities that can be matched to the resource requirements of individuals and businesses.

Security is one of the greatest concerns currently preventing large scale adoption of the cloud. This issue is emphasized in numerous recent literature articles, either stating that cloud computing security is still immature or just unreliable. Examples can be found in Everett (2009), Grossman (2009), Hutchinson et al. (2009), Kaufman (2009), Grobauer B et al. (2010) and Sloan (2009), which all raise the question of security as a concern in the cloud computing environment. Since cloud computing is such a new and talked about topic, numerous blog and web articles are also talking about security related concerns in a cloud.

A. *Service Level Agreement-*

A service-level agreement is an agreement between two or more parties, where one is the customer and the others are service providers. This can be a legally binding formal or an informal "contract" (for example, internal department relationships). Contracts between the service provider and other third parties are often (incorrectly) called SLAs – because the level of service has been set by the (principal) customer, there can be no "agreement" between third parties; these agreements are simply "contracts." Operational-level agreements or OLAs, however, may be used by internal groups to support SLAs.

SLAs commonly include segments to address: a definition of services, performance measurement, problem management, customer duties, warranties, disaster recovery, and termination of agreement. In order to ensure that SLAs are consistently met, these agreements are often designed with specific lines of demarcation and the parties involved are required to meet regularly to create an open forum for communication. Contract enforcement (rewards and penalties) should be rigidly enforced, but most SLAs also leave room for annual visitation so that it is possible to make changes based on new information.

SLAs have been used since late 1980s by fixed line telecom operators as part of their contracts with their corporate customers. This practice has spread such that now it is common for a customer to engage a service provider by including a service level agreement in a wide range of service contracts in practically all industries and markets. Internal departments (such as IT, HR, and real estate) in larger organizations have adopted the idea of using service-level agreements with their "internal" customers — users in other departments within the same organization. One benefit of this can be to enable the quality of service to be benchmarked with that agreed to across multiple locations or between different business units. This internal benchmarking can also be used to market test and provide a value comparison between an in-house department and an external service provider.

Service level agreements are, by their nature, "output" based – the result of the service as received by the customer is the subject of the "agreement." The (expert) service provider can demonstrate their value by organizing themselves with ingenuity, capability, and knowledge to deliver the service required, perhaps in an innovative way. Organizations can also specify the way the service is to be delivered, through a specification (a service level specification) and using subordinate "objectives" other than those related to the level of service. This type of agreement is known as an "input" SLA. This latter type of requirement is becoming obsolete as organizations become more demanding and shift the delivery methodology risk on to the service provider.

Service level agreements are also defined at different levels:

1) *Customer-based SLA:* An agreement with an individual customer group, covering all the services they use. For example, an SLA between a supplier (IT service provider) and the finance department of a large organization for the services such as finance system, payroll system, billing system, procurement/purchase system, etc.
2) *Service-based SLA*: An agreement for all customers using the services being delivered by the service provider. For example:
   A car service station offers a routine service to all the customers and offers certain maintenance as a part of offer with the universal charging.
   A mobile service provider offers a routine service to all the customers and offers certain maintenance as a part of offer with the universal charging
   An email system for the entire organization. There are chances of difficulties arising in this type of SLA as level of the services being offered may vary for different customers (for example, head office staff may use high-speed LAN connections while local offices may have to use a lower speed leased line).
3) *Multilevel SLA*: The SLA is split into the different levels, each addressing different set of customers for the same services, in the same SLA.
   Corporate-level SLA: Covering all the generic service level management (often abbreviated as SLM) issues appropriate to every customer throughout the organization. These issues are likely to be less volatile and so updates (SLA reviews) are less frequently required.
   Customer-level SLA: covering all SLM issues relevant to the particular customer group, regardless of the services being used.
   Service-level SLA: covering all SLM issue relevant to the specific services, in relation to this specific customer group.

The underlying benefit of cloud computing is shared resources, which is supported by the underlying nature of a shared infrastructure environment. Thus, service level agreements span across the cloud and are offered by service providers as a service based agreement rather than a customer based agreement. Measuring, monitoring and reporting on cloud performance is based upon an end user experience or the end users ability to consume resources. The downside of cloud computing, relative to SLAs, is the difficultly in determining root cause for service interruptions due to the complex nature of the environment.

As applications are moved from dedicated hardware into the cloud these applications need to achieve the same or even more demanding levels of service as classical installations. SLAs for cloud services focus on characteristics of the data center and more recently include characteristics of the network to support end-to-end SLAs.

Any SLA management strategy considers two well-differentiated phases: the negotiation of the contract and the monitoring of its fulfilment in real-time. Thus, SLA Management encompasses the SLA contract definition: basic schema with the QoS (quality of service) parameters; SLA negotiation; SLA monitoring; and SLA enforcement—according to defined policies.

B. ***SLA benefits for customer satisfaction***

Service level agreements SLAs enable you to ensure maximum customer satisfaction:

 By providing a guarantee that contracted services will be delivered as per agreement.
 By including penalties if contracted services are not delivered.
 By allowing you to negotiate the highest level a provider can guarantee.

C. **Functional Goals**

The primary functional goal of our SLA management framework is to provide a generic solution for SLA management that

1. supports SLA management across multiple layers with SLA composition and decomposition across functional and organizational domains;

2. supports arbitrary service types (business, software, and infrastructure) and SLA terms;
3. Covers the complete SLA and service lifecycle with consistent interlinking of design-time, planning and runtime management aspects; and
4. Can be applied to a large variety of industrial domains and use cases.

In order to achieve these goals, the reference architecture is based on three main design principles.

First, we put a strong emphasis on a clear separation of concerns, by clearly separating service management from SLA management and by supporting a well layered and hierarchical management structure.

Second, a solid foundation in common meta-models for SLAs as well as their relation to services and the construction of actual service instances is an important aspect to support clear semantics across different components of the framework.

Third, design for extensibility/adaptability is a key aspect in order to address multiple domains. Therefore, we clearly distinguish between generic solution elements and places where domain specific logic/models need to be provided. Furthermore, we seek for an architecture where even generic parts can be replaced by domain specific versions, which might be dictated by already existing (legacy) management functionality.

D. **Technical goals**

A set of technical requirements and goals has been collected from various industrial use cases and external stakeholders. They fall into the four categories of Framework Configuration & Setup, Framework Model Configuration, Framework Operation, and Framework Access.

Model-related requirements are mainly about model extensibility and are addressed by the design of the SLA model and the service construction model.

Other requirements relate to the usage of certain technology standards and are taken into account by the actual framework development

E. **Services, Resources and SLAs**

A first and fundamental concept for the architecture is the clear distinction between resources, services and SLAs. Following the ITIL definitions [10] and in accordance with the SLA@SOI glossary we can define them as follows:

1) ***Services***: A means of delivering values to customer by facilitating Outcomes Customers want to achieve without the ownership of specific Costs and Risks.
2) ***Resource***: A generic term that includes IT Infrastructure, people, money or anything else that might help to deliver an IT Service. Resources are considered to be Assets of an Organization.
3) ***SLA***: An agreement between service provider and a customer. The SLA describes the service, documents service level targets, and specifies the responsibilities of the service provider and the customer. A single SLA may cover multiple services or multiple customers.

II. **MULTI-LEVEL SLA ARCHITECTURE**

The overall SLA Framework is conceived as a possibly distributed, hierarchical management system providing consistent SLA management across the service delivery stack.

At the highest level, we assume the operation of the SLA serves ultimately to satisfy the goals of some business entity. Consequently, all management activities supported by the framework should eventually relate to the needs of the business entity.

Technically, the architecture is built along the following design goals:

1. Having division between different components e.g.:-infrastructure, software, platform, and business manager.
2. Roles are defined at each layer and having interaction between each other and require different setups of the framework
3. Simplicity, which is important to make the whole architecture understandable to a large audience and to make the actual framework adoptable for industrial use cases. Design goals 1 and 2 are indispensable in order to support the different scenarios that are introduced by the different industrial use cases. Goal 3 is a more pragmatic goal but may in some cases also conflict with goal 2 as flexibility typically increases complexity. The main approach taken to resolve such conflicts is by providing default implementations of respective components or interaction channels.

The Business Manager is at business layer and have the responsibilities for controlling all business regarding information and communication with customers and cloud providers. For example, it realizes the customer relation management necessary to efficiently sell the offered services. Furthermore, the Business Manager concerns and controls the Business SLA manger, Software SLA manager, and Infrastructure SLA manager. For this purpose, SLA Managers have to adhere to business rules defined by the Business Manager ('control') and have to inform the Business Manager about their current status and activities ('track').

The (Business-, Software- & Infrastructure-) SLA Managers are responsible for the management of all SLA related concerns. The Business SLA Manager, Software SLA Manager, and Infrastructure SLA Manager are specializations of an abstract generic SLA Manager. SLA Managers are responsible for the negotiation of SLAs, and for the SLA Management of services (subject to SLAs). All SLA Managers can act as "service customers"; negotiating SLAs with other SLA Managers inside the same framework, or with external (3rd party) service providers (including other framework instances). As "service providers" all SLA Managers can negotiate SLAs with other SLA Managers in the same framework. Only the Business SLA Manager can negotiate with customers who are external to the framework. Finally, all SLA Managers can consult Service Evaluation to a priori evaluate the potential quality of a service (<<evaluate>>). This evaluation can be based on prediction, historical data, or predefined quality definitions, and supports the SLA Manager in finding service realizations with an appropriate quality.

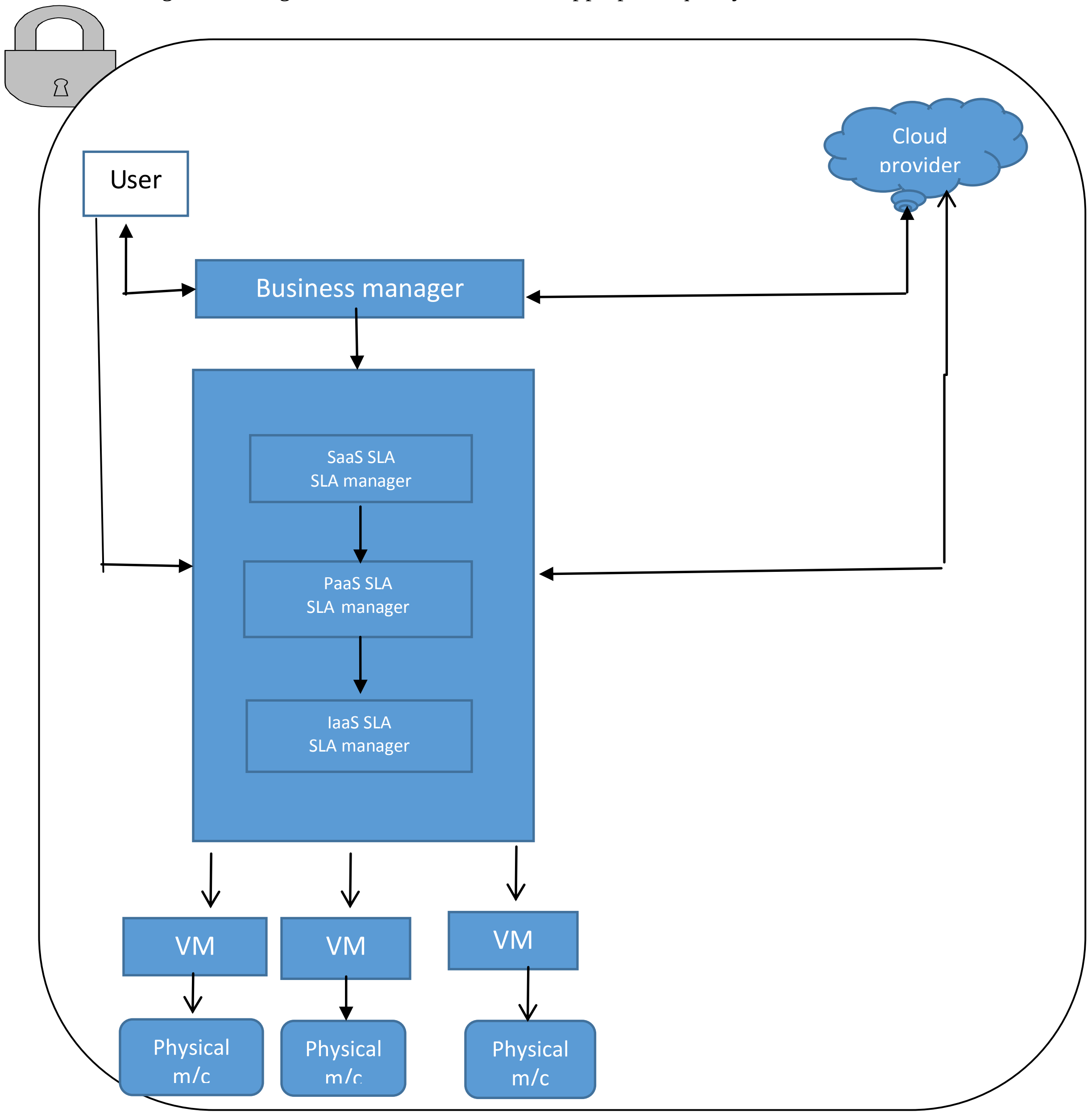

Fig1: ARCHITECTURE OF MULTI LEVEL SLA

Infrastructure- and Software Service Managers encapsulate all service-specific details. Both are specializations of the abstract Service Manager concept.

A. Business manager-

SLA Management at the business layer is encapsulated by the Business Manager & Business SLA Manager components. From the business perspective an instance of the SLA@SOI Framework constitutes a single 'service provider' - which offers & delivers products (the services exposed by the framework to the outside world). The business layer is responsible for selling products to business-customers (i.e. customers who are external to the framework), and for all related business-customer relations. The business layer does not, however, have complete responsibility for dealing with external service providers. Instead, for provider-facing relations, each SLA Manager within the framework can act (semi)autonomously as a 'service customer' to external service providers (as well as to each other) - but only within limits specified at the business-layer.

B. Software SLA Manager-

The management of SLA enabled software services follows the separation of concerns defined in the top- level SLA@SOI architecture. The Software SLA Manager is responsible for all software SLA related management activities while the Software Service Manager takes care of all software service management activities.

C. Infrastructure SLA Manager-

The Infrastructure SLA Manager manages all aspects of SLAs concerning IT infrastructural resources. It is a specialization of the Generic SLA Manager, including infrastructure specific implementations of the Planning and Optimization Component (POC) and Provisioning and Adjustment Component (PAC). The Infrastructure POC is responsible for the planning and optimization of Infrastructure SLAs. It receives requests for infrastructure, queries the Infrastructure Service Manager for potential provisioning solutions, selects and reserves the optimal one and requests the Infrastructure PAC to provision the selected plan as appropriate. If local resources cannot satisfy the request (e.g. due to lack of availability or specification discrepancies), the Infrastructure POC can attempt to outsource to third party providers to satisfy the request.

The Infrastructure PAC is responsible for the provisioning and adjustment of Infrastructure SLAs. It directs the Infrastructure Service Manager to provision as per the plan supplied by the POC. It also decides on any adjustments required, e.g. to avoid potential SLA violations.

III. ADVANTAGES OF MULTI-LEVEL SLA

SLAs are important so they will rarely be agreed without negotiation between the IT service provider and the customer, beginning with a statement of intent that sets out the terms conditions and targets to be agreed. It has to be in a language that both sides will understand and this means in the language of the customer and not the technical language of the provider. The SLA defines in language that has meaning to the customer preciously what is to be delivered and when and where it is to be delivered. It also defines the standard of quality to be delivered, responsibilities of both the service provider and the customer. This is important. It makes little sense for a service provider to commit to deliver a service without making it clear what is expected from the customer. Where one party delivers services to another, it is a good idea to have some kind of agreement setting out the basis on which the service is provided. Such agreements would normally contain among other things, a description of what is to be provided, the key performance indicators, the way the service is to be charged for (where relevant) and the responsibilities of each of the parties. In service level management, the agreement between the internal IT service provider and the business customers that it supports are known as SLAs. And it is through SLAs that SLM manages the relationship between itself and its customers. It would be hard to find IT services provided by IT companies like TCS, Infosys etc. where they do not have any accepted agreed upon SLAs with their customers. In order to be effective the SLA must be a written document signed off by all parties affected by it.
The SLA will include contact details, what should happen if something goes wrong, the way any dispute should be handled, any provision for redress, the mechanism for getting the SLA changed if necessary and the period over which the agreement will stand unless otherwise changed by agreement. If the service is to be charged for, then the way charges are to be determined and arrangements for invoicing should be included. Charges may also be included in a separate document, the tariffs referenced in the SLA. It is common in IT for individual services to be shared by a number of customers and its individual customer will use a range of services. This means that there is a choice in designing SLAs. They can be customer based where the SLA covers a range of services delivered to a particular customer, or they can be service based where a common SLA covers all customers of a given service.

IV. CONCLUSION

In this paper a reference architecture for multi-level SLA management that supports the comprehensive management of possibly complex service stacks is presented. Service Level Agreements or SLAs are used for managing the non-functional aspects of the complete service lifecycle. Also, SLA translations across different layers allow for consistent interlinking of complete service networks and hierarchies. The architecture which is presented here is based on the experiences gained from an SLA framework built around a specific reference application. The main achievement of this work is the generalization of the concepts so that the architecture can serve a large variety of domains. The example use case demonstrates the applicability and flexibility of the architecture. Additionally, future work is planned on assessing the business benefit of multi-level SLA management in the presented use case. Technically, the SLA management framework will be extended in various dimensions, such as support for managing specific non-functional properties (e.g. reliability), a library of SLA planning algorithms and finally advanced interfaces for harmonized cloud infrastructures.

REFERENCES

1. http://www.cloudcomputingforum.com/
2. W. Theilmann, R. Yahyapour, and J. Butler, *Multi-level SLA Management for Service-Oriented Infrastructures.* Towards a Service-Based Internet (2008)324–335
3. M. Comuzzi, C. Kotsokalis, C. Rathfelder, W. Theilmann, U. Winkler, and G. Zacco, *A Framework for Multi-level SLA Management.* Proc. of 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing (NFPSLAM-SOC'09), November 23, Stockholm, Sweden.
4. Rajkumar Buyya1,2, Saurabh Kumar Garg1, and Rodrigo N. Calheiros.SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions 2011 International Conference on Cloud and Service Computing
5. Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst., 25(6):599–616, 2009.

6. Andreas Klein, Christian Mannweiler, Joerg Schneider, and Hans D. Schotten. Access schemes for mobile cloud computing. In Eleventh International Conference on Mobile Data Management (MDM), 2010, pages 387–392, 2010.
7. Jon Oberheide, Kaushik Veeraraghavan, Evan Cooke, Jason Flinn, andFarnam Jahanian. Virtualized in-cloud security services for mobile devices. In MobiVirt '08: Proceedings of the First Workshop on Virtualization in Mobile Computing, pages 31–35, New York, NY, USA, 2008. ACM.
8. OpenMobster. http://code.google.com/p/openmobster/. 2010.
9. Rajiv Ranjan and Rajkumar Buyya. Decentralized overlay for federation of enterprise clouds. CoRR, abs/0811.2563, 2008.
10. ABI Research. http://www.abiresearch.com/. 2010.
11. B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, L. Llorente,R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galan. The RESERVOIR Model and Architecture for Open Federated Cloud Computing. IBM Journal of Research and Development, 53(4):Paper 4, 2009.
12. Xinwen Zhang, Joshua Schiffman, Simon Gibbs, Anugeetha Kunjithapatham, and Sangoh Jeong. Securing elastic applications on mobile devices for cloud computing. In CCSW '09: Proceedings of the 2009 ACM workshop on Cloud computing security, pages 127–134, New York, NY, USA, 2009. ACM.
13. Zehua Zhang and Xuejie Zhang. Realization of open cloud computing