International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6339
***Article Received**: 28 December 2022 **Revised**: 20 January 2023 **Accepted**: 10 February 2023*

Towards Automated and Optimized Security Orchestration in Cloud SLA

Shikha Singh[1], Dr. Ajay Kumar Bharti [2], Dr. Himanshu Pandey[3], Dr. Rakesh Kumar Yadav[4], Dr. Durgansh Sharma[5], Dr. N.R.Shanker[6]

[1]Research Scholar, Computer Science and Engineering,
Maharishi University of Information and Technology,
Lucknow, U.P., India
shikha.bims@gmail.com

[2]Professor, School of Computer Application,
Babu Banarasi Das University, Lucknow,
Lucknow, U.P., India
ajay_bharti@hotmail.com

[3]Assistant Professor, Computer Science and Engineering,
Lucknow University,
Lucknow, U.P., India
hpandey010@gmail.com

[4]Associate Professor, Computer Science and Engineering,
Maharishi University of Information and Technology,
Lucknow, U.P., India
rkmuit@gmail.com

[5]Associate Professor,School of Business Management,
Christ University Ghaziabad,
U.P., India
durgansh.sharma@christuniversity.in

[6]Professor, Computer Science and Engineering,
Aalim Muhamed Salegh College of Engineering,
Chennai, Tamil Naidu, India
nr_phd@yahoo.co.in

Abstract: In cloud computing, providers pool their resources and make them available to customers. Next-generation computer scientists are flocking to the cutting-edge field of cloud computing for their research and exploration of uncharted territory. There are still several barriers that cloud service providers must overcome in order to provide cloud services in accordance with service level agreements. Each cloud service provider aspires to achieve maximum performance as per Service Level Agreements (SLAs), and this is especially true when it comes to the delivery of services. A cloud service level agreement (SLA) guarantees that cloud service providers will satisfy the needs of large businesses and offer their clients with a specified list of services. The authors offer a web service level agreement–inspired approach for cloud service agreements. We adopt patterns and antipatterns to symbolize the best and worst practices of OCCI (Open Cloud Computing Interface Standard), REST (Representational State Transfer), and TOSCA (Topology and Orchestration Specification for Cloud Applications) with DevOps solutions, all of which API developers should bear in mind when designing APIs. When using this method, everything pertaining to the cloud service, from creation to deployment to measurement to evaluation to management to termination, may be handled mechanically. When distributing resources to cloud apps, our system takes into account the likelihood of SLA breaches and responds by providing more resources if necessary. We say that for optimal performance, our suggested solution should be used in a private cloud computing setting. As more and more people rely on cloud computing for their day-to-day workloads, there has been a corresponding rise in the need for efficient orchestration and management strategies that foster interoperability.

Keywords: Security, Orchestration, Cloud Computing, Service Level Agreement.

I. INTRODUCTION

The term "security orchestration" refers to a technique for linking together and coordinating various security-related instruments and infrastructures. The interconnected layer is what enables automation in the security domain. Researchers have only just begun to explore cloud computing, a cutting-edge technology where many concerns remain unanswered. Business organizations use cloud computing by placing orders for the necessary services. Enterprises enter into contracts with cloud service providers to guarantee the availability of critical

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6339
***Article Received**: 28 December 2022 **Revised**: 20 January 2023 **Accepted**: 10 February 2023*

services in accordance with service level agreements (Service Level agreement). Service Level Agreements (SLAs) are the contractual guarantees made by and between cloud service providers and their customers. In the past, Service Level Agreements (SLAs) in cloud computing were always individually negotiated between a client and a service consumer. A cloud service level agreement (SLA) guarantees that cloud service providers will satisfy the needs of large businesses and offer their clients with a specified list of services. Also included are the monetary penalties that will be applied if the service provider does not meet the promised conditions.

Cloud computing is a kind of distributed system that uses a network of several powerful computers to provide users with on-demand access to shared pools of computational resources in a virtualized environment. In cloud computing, resource and load requirements will fluctuate over time. In other words, the need for resources will be greatest after normal business hours. Since the provider's physical resources are dispersed across many locations, their workload may be divided up more evenly. Provisioning of virtualized resources to consumers as services, cloud computing is a sort of parallel or distributed system consisting of combinations of networked computer resources and virtualized computing resources. Cloud computing is characterized by temporal variability in resource requirements and loads. When it comes to Cloud, for instance, the demand for resources is often highest in the late evening, when people are typically not at work. Furthermore, a Cloud provider's physical resources (computing clusters) are geometrically spread, and the provider's resource burden may be allocated to such resources.

II. LITERATURE REVIEW

Islam, Chadni (2020) Companies use a wide range of security measures to thwart cybercriminals. Security solutions are available from a variety of providers, and they were built utilizing a wide range of different technologies and conceptual frameworks. Therefore, making security solutions operate together seamlessly is a difficult, if not impossible, task. The goal of security orchestration is to help SOC personnel by seamlessly integrating a wide variety of security products from different vendors that can work together swiftly and effectively (SOC). There is a growing body of literature on various elements of security orchestration solutions, reflecting the field's expanding relevance and significance. Unfortunately, no work has been done to rigorously examine the stated answers. Here, we provide the results of a Multivocal Literature Review, in which we carefully identified and examined academic and grey (blogs, web pages, white papers) literature on various aspects of security orchestration, spanning from January 2007 to July 2017. With the information gathered from this study, we are now able to define security orchestration and divide its primary functions into three categories: unification, orchestration, and automation. We've also broken down the drivers of security orchestration into technical and socio-technical groups, and uncovered the fundamental elements of a security orchestration platform. Additionally, we provide a security orchestration taxonomy that considers factors such as deployment type, mode of task, resource type, and execution environment. Thanks to this analysis, we've uncovered a number of promising avenues for future work in the field of security orchestration. CCS Ideas Generally speaking and for future use We will conduct a survey and broad overviews; Respect for one's personal space and safety; Informatization and software development Words & Phrases Used Regularly Synonyms include intelligent security assistant, multi-voice literature review, security automation, and security orchestration.

S., Shilpashree & Patil, Renuka & C, Parvathi (2018) the term "cloud computing" is used to describe a new kind of computing that relies on remote servers in the cloud. To put it simply, Cloud Computing is the practice of storing and distributing digital resources, such as programs and services, via a network rather than on individual devices. Cloud computing provides the majority of computational resources to customers as a service, according to the needs and preferences of each individual user. Larger storage space, high-performance servers, a selection of operating systems designed for use on a wide range of hardware, and a robust network are among the most important resources for Security is seen as a major issue in cloud computing, and yet consumer demand for these resources only continues to rise. This article has covered a lot of ground, including topics like cloud computing's architecture, different deployment patterns and kinds, applications, benefits, and drawbacks. The primary objective of this work is to improve readers' understanding of cloud computing and the challenges it presents for academics in a variety of fields.

Paladi, Nicolae & Michalas, Antonis & Dang, Hai-Van (2018) the cloud infrastructure is often deployed and managed using a cloud orchestration framework. Horizontally (across infrastructure, platforms, applications, and microservices) and vertically (across deployment on these elements) they play a crucial role (deployments from many distinct cloud resource providers). Despite orchestration's critical importance, however, common orchestration frameworks don't have ways to assure cloud operators' data is secure. Here, we take a look at the state of cloud orchestration frameworks for multi-cloud architecture from a safety perspective. Multi-cloud application deployments are made possible thanks to the threat scenarios

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6339
***Article Received**: 28 December 2022 **Revised**: 20 January 2023 **Accepted**: 10 February 2023*

we identify, the security enforcement enablers we design, and the cloud orchestration framework architecture we suggest.

Kalliola, et al (2018) to keep virtualized network services safe in the cloud, we outline the design and implementation of the security wrapper idea. In the network forwarding graph, the security wrapper is the transparent protective envelope around a group of virtualized resources. Both the size and capabilities of this envelope are adaptable. We demonstrate a working prototype of the security wrapper and provide an analysis of its behavior across a range of use cases. Data on wrapper orchestration delays, resource overhead, and effect on data plane traffic suggests the proposed technique may be used in virtualized networks with little disruption to service and minimal impact on the privacy of traffic flowing through the security wrapper.

Khan, Imran &Alam, Mahfooz (2017) the term "cloud computing" refers to a kind of computing that is delivered through the internet and makes use of a network to make available shared computing resources and data to various devices. Today, cloud computing is a very new area of study. Cloud computing offers several advantages to organizations, including cost savings, a competitive edge, ease of management, security, stability, scalability, and automated upgrades. However, cloud computing has several challenges, including cost, service availability, power consumption, energy management, resource provisioning, virtual machine provisioning, load balancing, security, and more. This presentation addresses the problems and difficulties associated with cloud computing. The author also discusses the current research gaps in a cloud setting.

III. RESEARCH METHODOLOGY

SLA-based Resource Management for Adaptability

If the resource allocation system can dynamically accommodate the changes according to the changing expectations of the user, then the resource management system becomes more adaptive and effective, as monitoring and analysis of SLA parameters paints a clear picture of the timely requirements of the service consumer. In order to decrease the amount of SLA breaches in an opennebula cloud environment, we seek to better manage resource allocation by supplying more pictures to running workloads. Our method for incorporating dynamic management of cloud-based resources is shown here.

Algorithm 1:Identifying an SLA Violation

Input: Monitored SLA parameters.
Output: Flag status of SLA violation.

1. Start
2. Let *response time* and *job execution time* be two SLO values. Let *flag value* be a boolean value for controlling the trigger signal in adaptive resource allocation.
3.
   a. Initialize SLA violation limit and SLA threat limit.
   b. Reset the *flag value* to zero.
4.
   a. Monitor SLA parameters using monitoring tool.
   b. Measure SLO values like *response time*, *job execution time* and *number of resources*.
5. Compare the measured SLO values with SLA threat limit.
6. If the SLO value $\geq$ threat limit then
   a. Detect it as a possibility of SLA violation.
   b. Set the value of *flag* to 1.
   c. Give trigger signal to dynamic resource allocation.
7. Stop

Algorithm 2: Allocating Resources in an Adaptive Manner

Input: Computing resources like CPU, RAM and Memory.
Output: Computing resources after allocation.

1. Start
2. Read the availability of resources like CPU, RAM, Memory, VM image, Network etc.
3. Allocate normal resources to the job.
4. Check whether any trigger signal at adaptive resource allocation.
5. If yes
   a. submit resource request to resource provisioning unit.
   b. Release and allocate extra resources by providing VM images.
   c. Update estimation of job execution time, response time and resource availability.
   d. Convey back to SLA negotiator and resource provider about the resource allocation/re-allocation.
6. Stop

A flag value indicating the status of an SLA violation is the output of Algorithm 1, where the input is the measured SLA values obtained through the monitoring tools. This flag value operates the trigger signal used by Algorithm 2 to regulate adaptive resource allocation. If the flag is set, it indicates that a potential SLA violation has been identified, but if the flag's value remains unchanged, no such violation is anticipated. When possible SLO violations are detected, the measured SLO values, such as 'response time' and 'task execution time,' are compared with threat threshold limits of violation. The second algorithm takes as input a list of accessible resources such as central processing unit, random access memory, memory, virtual machine images, etc., and returns the same list of resources (after being used). The resource provisioning unit, which handles resource allocation and re-allocation, receives requests for assigning more resources to tasks. Following

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6339
***Article Received**: 28 December 2022 **Revised**: 20 January 2023 **Accepted**: 10 February 2023*

allocation, a revised estimate of available resources is calculated and sent to the SLA negotiator and the units providing the resources. Both techniques have time complexity O(n), where n is the number of tasks.

Model for Deploying Cloud-Based Service Level Agreements

The primary ways in which a web service level agreement (SLA) may be described are by using one of two standards. Two such agreements are 1) Open Grid Forum's (OGF) Web Service Agreement (WS-Agreement) and 2) IBM's Web Service Level Agreement language and architecture (WSLA). As far as we are aware, SLAng is the other major active research effort working on SLA standard. Rule-based Service Level Agreement (RBSLA) is the focus of another relevant research. RBSLA takes a know-how-based approach by defining the SLA using Rule ML. For the sake of this introduction to network SLA material, we will use WSLA as an example. The WSLA is an XML language and a collection of ideas. There are three main sections to this document: first-party recipients Two) SLA Norms Third, we aim to provide a certain degree of service (SLO). Information regarding the service provider, the user, and any third parties is described by the parties. In this case, third-party agents were entrusted with the responsibility of measuring the service entity; they mediated communications between the user and the service provider and kept sensitive data private.

Orchestration

One definition of Cloud Resource Orchestration (CRO) is "the act of automatically organizing and coordinating various Cloud Resources" (server, database, network, application, container, etc.). Choosing cloud resources, putting them into action, keeping tabs on how they're doing, and making adjustments as needed throughout execution are the four pillars of orchestration. The following describes these steps in detail.

i. The action of "resource selection" (both during development and operation) gives program owners the ability to choose between hardware and software options.

ii. This includes the development and execution of the many software resources (services or components) that make up the application, on a chosen cloud platform (referred to as "deployment").

iii. To guarantee that quality of service characteristics are tracked in real time, it is necessary to do the following operation.

iv. Managing resources (during operation) by responding to events and launching remedial actions according to predefined rules.

Cloud Standards

OCCI:

The Open Cloud Computing Interface (OCCI) is an open standard for Cloud computing that seeks to solve issues of heterogeneity, interoperability, integration, and portability. A cloud provider's internal management system may benefit from OCCI's RESTful Protocol and API, which provides a service front-end for cloud providers. The OCCI's place in this setup is shown in Figure 1. In this context, a service consumer may be an actual human user or another component of the system.

- **REST:** To transfer one's representational state from one entity to another, or REST, as defined in Roy Thomas Fielding's doctoral dissertation. Web application designers and developers may benefit from REST since it is an architectural style that defines a set of guidelines for the design of distributed systems.

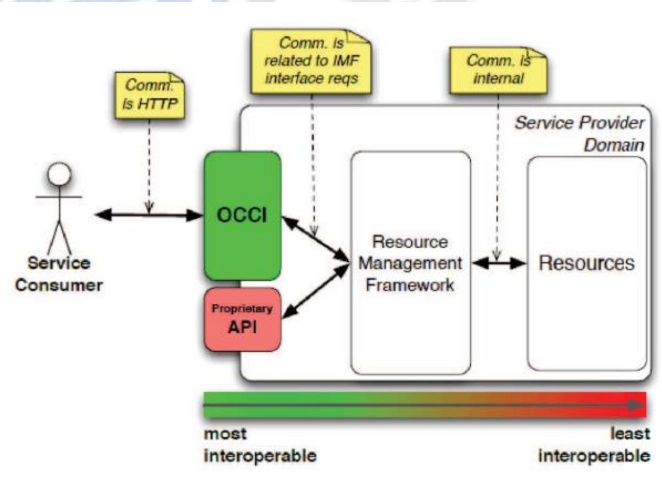

Figure 1: Position of OCCI in a provider's architectural design

TOSCA:

TOSCA stands for Topology and Orchestration Specification for Cloud Applications, which is the official name of an OASIS standard. One of its ultimate goals is to simplify the process of deploying and managing cloud-based applications by providing a standardized, portable description of such applications. To define composite applications, TOSCA offers a structured XML-based/YAML language and a metamodel that is not tied to any one technology.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6339
***Article Received**: 28 December 2022 **Revised**: 20 January 2023 **Accepted**: 10 February 2023*

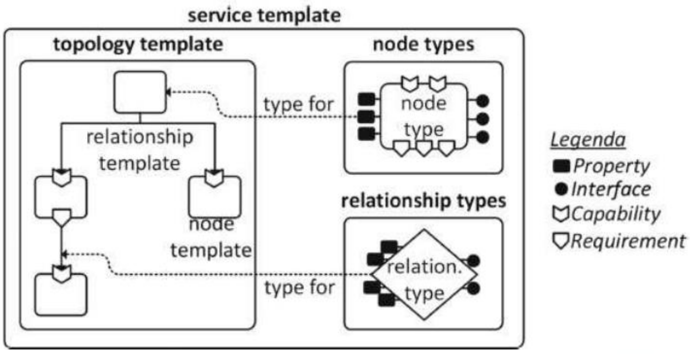

Figure 2: The meta-model for TOSCA

IV. DATA ANALYSIS

The findings are based on the use of opennebula and other monitoring technologies, such as Ganglia, to create a private cloud environment. Each server in the private cloud may create several virtual machine images, and the whole system is managed by a front-end controller. In this case, the response time SLA metric is used to identify service level agreement breaches. The studies are run on an opennebula 4.6-based private cloud. Sunstone is a graphical user interface (GUI) software used for submitting, monitoring, and controlling computing services housed in virtual machines on numerous host computers. The GMOND module of the Ganglia monitoring tool has been used to keep an eye on and assess performance in real time. The front-end controller, often known as the "cloud provider," is where the ganglia tool will be deployed. The gathered data is then sent to an event stream processing system, where an Esper engine is used for the event processing. With Java Messaging Service (JMS), data may be shared across applications. An XML parser is used to retrieve the SLA parameters and SLO values from the document containing the negotiated SLA. MySQL database is created for database administration. To ensure that SLA violations are recognized promptly, we set the threshold limit for the SLA parameter (response time) at 2 seconds; if the measured parameter values exceed this violation threshold limit, it is considered a violation. In the same way, we set the time limit for a danger to reach a certain threshold at 1.7 seconds for forecasting.

Figure 3 depicts the preliminary findings achieved for the identification and prediction of SLA violation. Over the course of many days, the detection module tallied the number of violations and predictions, and the graph shows the relationship between each of the three variables. The data collected and its interpretation are as follows: If there is no recognized SLA violation, then all hosts have access to sufficient computational resources to run their applications without degrading the service quality. However, any breach or potential breach is a clear sign that service quality has been compromised. The average monthly SLA violation count for our proposed module is 20–25, while the average monthly prediction count is 45–50. Clearly, our detection module has been successful in anticipating the vast majority of SLA breaches, as shown by these findings. As can be seen in Figure 4, our adaptive resource management system yielded the desired outcomes. Time taken for four distinct tasks to complete on separate virtual machines (VMs) after using and without using dynamic resource allocation is shown. Our adaptive allocation of resources is efficient, reducing work time in three situations while adding time in the fourth. In other words, if we standardize the criteria, our method outperforms the competition in around 75% of the instances. Overall, the data supports the viability of our SLA-accountable adaptive resource management system in a private cloud setting, where it leads to increased resource allocation and processing efficiency.

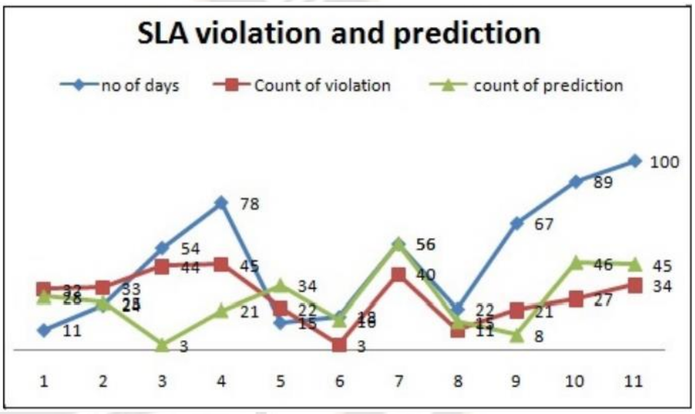

Figure3: Consequences of SLA Violations and Prediction

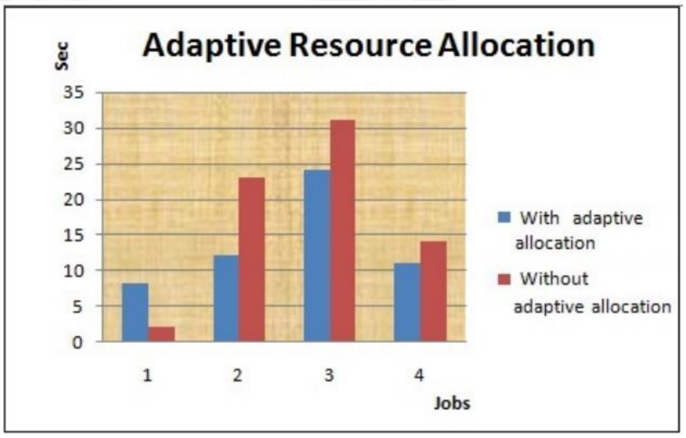

Figure4: Adaptive allocation of resources

On the foundation of the WSLA paradigm, we propose the SLA model known as CSLA (Cloud Service Level Agreement). There are really two models in this framework. The first is a model of coordination, while the second is a model of management. There are two components of service level agreement negotiation for a service composition.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6339
Article Received: 28 December 2022 Revised: 20 January 2023 Accepted: 10 February 2023

Ensuring end-to-end quality of service requires a number of moving parts, one of which is the coordination of negotiation for numerous services. The other part involves discussion and compromise between a service buyer and one or more service sellers. Here, we outline the three main components of cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Customers are free to mix and match the services they purchase. Since each service has its own unique Quality of Service (QoS) needs, we create a Coordinator Agent (CA) to handle QoS negotiation for the whole suite of services. SaaS SLA (Service Level Agreement), PaaS SLA (Service Level Agreement), and IaaS SLA (Infrastructure as a Service) will all exist if a customer employs all three varieties of cloud service (ISLA). California State Aggregation will combine them into one. The client's SLA will then be sent to the library's SLA templates.

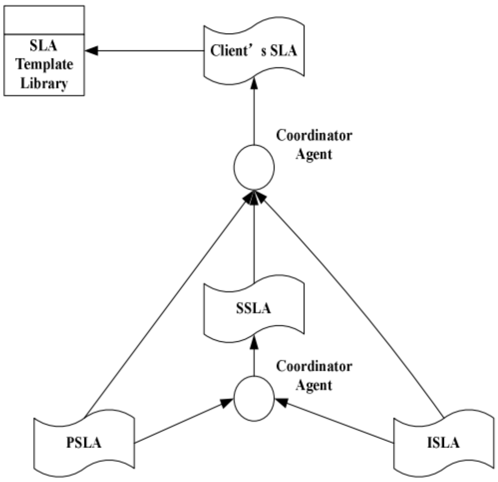

Figure5: Model of CSLA cooperation

We only use SLA management for a single cloud service in our management strategy. To begin, users can never be sure of the exact position of their data in the cloud because of the fluid nature of both the cloud and the resources it uses. Therefore, we augment the traditional SLA paradigm with real-time tracking. The measuring service relied on it to track Service Level Agreement (SLA) observables. Second, the data will be transported to systems that the consumers don't understand, thus the security issues must be included in to the SLA model as well. In this case, we still set up a credible third party to guarantee the security of the service and supplement it with a reliable method of negotiation. Last but not least, the cloud's inherent flexibility necessitates more regular service evaluations. Our SLA assessment services will provide dynamic scheduling of evaluations based on a variety of metrics and situations.

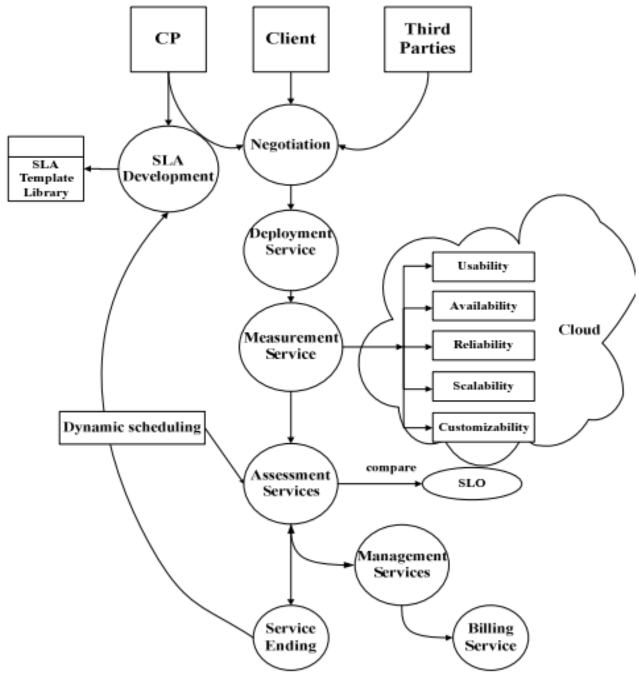

Figure6: Strategy for managing in a CSLA environment

A cloud provider (CP) will design their own SLA templates and then share some of their instantiations with their clients. They may also be sent to the service agency's resource library, where end users can browse for and choose the best SLA model for their needs. Once the fundamentals of an SLA have been settled upon, the parties involved may mutually agree upon a third party to oversee the agreement. Chiefly, they do the shady business of running things that CP and users can't manage. After negotiations are finalized, the Service Level Agreement (SLA) is signed by both the CP and the end customers. The Deployment Service is accountable for monitoring the efficiency of SLAs and assigned tasks for the sake of the supporting organization. The supporting entity just has to get the relevant executable data of function. Measurement, evaluation, and management of the service are ongoing processes including CP, users, and other stakeholders. In order to keep track of the state of the system's configuration and operational data, the management service must monitor business performance in real time in accordance with the SLA criteria established in the document. In this paper, we propose standard SSLA settings for use in cloud environments. 1) Ease of use, as in intuitive interfaces; 2) Availability, or the length of time users can access the program; 3) Reliability, or the program's ability to continue functioning in the vast majority of situations; 4) Scalability, or the program's ability to accommodate a large or small user base; and 5) Customizability, or the program's adaptability to meet the needs of a wide range of users. The duty of determining how well runtime SLA parameters align with the signed SLO falls on the evaluation services. In order to get the assessment

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6339
***Article Received**: 28 December 2022 **Revised**: 20 January 2023 **Accepted**: 10 February 2023*

findings, it compares the SLA threshold with the measured SLA parameters.

Patterns of OCCI (anti) detection

The detection outcomes of 24 OCCI patterns and anti-patterns on the five Cloud RESTful APIs are summarized in Table 1. Table 1 shows that when it comes to management-related (anti) patterns, the most common ones are "Compliant Delete" and "Compliant Update." The OCCI recommendations for removing and updating a cloud resource are followed by Cloud API designers either directly or implicitly. The most common anti-patterns in this group are Non-Compliant Trigger Action and Non-Compliant Create. Most Cloud RESTful APIs, in fact, don't need you to describe the resource category when you create it. The REST operation to initiate an action on a resource appears to be disregarded by the designers of cloud APIs, along with the query disclosing the term of the action and its associated HTTP category specifying its functionality.

Table 1: The OCCI (Anti)pattern Detection Outcomes

Cloud REST API	OOi (28)	COAPS (18)	Open-Nebula (20)	Amazon S3 (71)	Rack-space (72)	O.P.	Total (209)
Management Related (Anti)patterns							
Query Interface Support	0	0	0	0	0	0%	0 (0%)
Missing Query Interface (5/161)	1	1	1	1	1	3%	5 (2%)
No Detection (0/161)	0	0	0	0	0	0%	0 (0%)
Compliant Create (6/31)	4	1	0	1	0	19%	6 (2%)
Non-Compliant Create (24/31)	0	1	3	11	9	77%	24 (11%)
No Detection (1/31)	1	0	0	0	0	3%	1 (0.4%)
Compliant Update (15/24)	3	2	3	0	7	62%	15 (7%)
Non-Compliant Update (8/24)	0	0	0	8	0	8%	8 (3%)
No Detection (1/24)	0	0	0	1	0	4%	1 (0%)
Compliant Delete (32/32)	3	2	3	14	10	100%	32 (15%)
Non-Compliant Delete (0/32)	0	0	0	0	0	0%	0 (0%)
No Detection (0/32)	0	0	0	0	0	0%	0 (0%)
Compliant Retrieve (26/90)	10	5	11	0	0	28%	26 (12%)
Non-Compliant Retrieve (64/90)	0	0	0	31	33	71%	70 (30%)
No Detection (0/90)	0	0	0	0	0	0%	0 (0%)
Compliant Trigger Action (0/21)	0	0	0	0	0	0%	0 (0%)
Non-CompliantTriggerAction (21/21)	0	7	0	3	11	100%	21 (10%)
No Detection (0/21)	0	0	0	0	0	0%	0 (0%)
Cloud Structure (Anti)patterns							
Compliant Link between Resources(4/5)	2	0	2	0	0	80%	4 (1%)
Non-Compliant Link between Re- sources (1/5)	0	0	0	0	1	20%	1 (0.4%)
No Detection (0/5)	0	0	0	0	0	0%	0 (0%)
Compliant Association of Resourcewith Mixin	-	-	-	-	-	-	-
Non-Compliant Association of Re- source with Mixin	-	-	-	-	-	-	-
No Detection	-	-	-	-	-	-	-
Compliant Dissociation of Resourcefrom Mixin	-	-	-	-	-	-	-
Non-Compliant Dissociation of Re- source from Mixin	-	-	-	-	-	-	-
No Detection	-	-	-	-	-	-	-
REST Related (Anti) Patterns							
Compliant URL (209/209)	28	18	20	71	72	100%	209(100%)
Non-Compliant URL (0/209)	0	0	0	0	0	0%	0 (0%)
No Detection (0/209)	0	0	0	0	0	0%	0 (0%)
Compliant Req.H (0/209)	0	0	0	71	72	68%	143 (68%)
Non-Compliant Req.H (209/209)	28	18	20	0	0	66%	66 (31%)
No Detection (0/209)	0	0	0	0	0	0%	0 (0%)
Compliant R.H (143/209)	0	0	0	71	72	68%	143 (68%)
Non-Compliant R.H (66/209)	28	18	20	0	0	100%	66 (31%)
No Detection (0/209)	0	0	0	0	0	0%	0 (0%)

Alteration of Terraform from TOSCA

As its name suggests, terraform's primary purpose is to facilitate the provisioning of infrastructure and the setup of suitable middleware environments, both of which are necessary for the effective deployment of services on this infrastructure. The TOSCA topology shown in Figure 7 partly reflects our motivating case and serves to highlight how our transformation may enable these two features. A pair of TOSCA nodes form the basis of the topology. One node, called vote, stands in for a voting service that must be hosted on another, called app-server1, which stands in for the computing resource that would be supplied by the AWS provider with those capabilities. In order to convert TOSCA into Terraform-specific Artifacts, this topology model will serve as the source model.

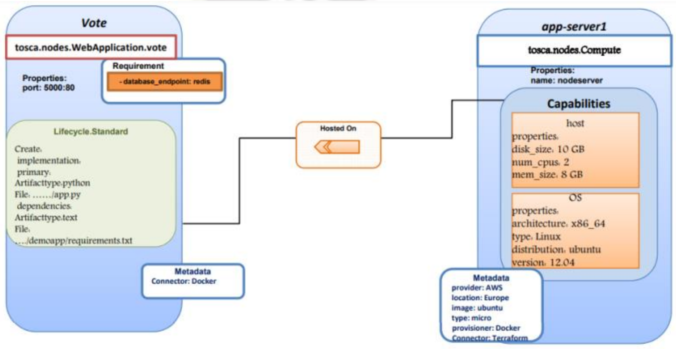

Figure 7: TOSCA topology

Prior to beginning the transformation, it is necessary to gather the provider-specific data. By way of illustration, if a user chooses Amazon Web Services (AWS) as their preferred cloud provider, one of the available selection algorithms could determine the resource size (e.g., Large, Micro, Medium, etc.)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6339
***Article Received**: 28 December 2022 **Revised**: 20 January 2023 **Accepted**: 10 February 2023*

and image (e.g., Ubuntu 64) based on the properties related to the computing node's capabilities (CPU, RAM, OS, etc.). Take note that the consumer has the option of having the supplier picked automatically. Further, as these data are not essential to the TOSCA topology and may run counter to its technology-agnostic nature, they are added to the metadata section (see app-metadata server1's in Figure 7) associated with any compute node, and their processing is completely invisible to the user. API developers and cloud providers would do well to pay close attention to both the OCCI and REST best practices that we have codified as patterns and anti-patterns. Further, we looked at both the literature and the OCCI standard to determine which of these were indeed trends to avoid.

V. CONCLUSION

We can say that there are a growing number of cloud service companies offering services with almost identical functionality through the Internet. A relatively new technology, c10ud computing offers a wide range of services for essential business applications, as well as reliable and adaptable contract management procedures. In order to guarantee quality standards when doing business in the cloud, it is crucial to define a suitable SLA and keep SLA management systems up and running. QoS and other contextual information describe the business objectives and performance in cloud systems. In this effort, we kept an eye on the service level agreement's metric values and flagged any indications of potential violations. These findings have been useful in controlling the resource allocation process and cutting down on service level agreement (SLA) breaches. The efficiency of resource allocation in the private cloud computing environment built with opennebula was undoubtedly enhanced by the SLA-associated adaptive resource allocation. Our suggested approach is appealing because it takes into account cloud service providers and end users, and it facilitates the maintenance of ongoing business activities. The CSLA paradigm provides an unmistakable example of a formal SLA mechanism. A cloud provider may fulfill consumers' requirements through making optimal use of cloud infrastructure. When users' rights infringe upon cloud service providers' assets, they are able to get compensation and legal redress. The lack of consensus on cloud computing standards is becoming more of an issue as more businesses use cloud services. The standardization of SLAs, which is based on the standardization of cloud services and the safety control to protect them, is still necessary for their successful deployment. If a CP's resources are at capacity, work will be transferred to another CP's computers to prevent service level agreement (SLA) breaches. Standardized agreement ensures rapid implementation and serves as the basis for growth throughout the migration process. So, the standardization of cloud computing is another project for the future. Future research should focus on SLA pricing strategy, SLA dynamic monitoring, and SLA administration in cloud computing. We tested the viability of our method by evaluating both OCCI and REST (anti) patterns on working Cloud RESTful APIs, and found it to be both accurate and helpful. Our method has been shown to be highly accurate at identifying OCCI and REST (anti) patterns in cloud RESTful APIs, which could be useful for both cloud API providers and developers in evaluating the quality of their APIs and making any necessary adjustments to prevent future occurrences of these anti-patterns. Furthermore, we demonstrated the essential detection criteria and given suggestions are beneficial and relevant via the assessment of usefulness.

REFERENCES

[1] Islam, Chadni. (2020). A Multi-Vocal Review of Security Orchestration. 10.1145/3305268".
[2] S., Shilpashree& Patil, Renuka & C, Parvathi. (2018). “Cloud computing an overview”. International Journal of Engineering & Technology. 7. 2743-2746. 10.14419/ijet.v7i4.10904.
[3] Paladi, Nicolae &Michalas, Antonis & Dang, Hai-Van. (2018). Towards Secure Cloud Orchestration for Multi-Cloud Deployments. 1-6. 10.1145/3195870.3195874.
[4] Kalliola, Aapo& Lal, Shankar &Ahola, Kimmo & Oliver, Ian &Miche, Yoan& Aura, Tuomas. (2018). Security Wrapper Orchestration in Cloud. ARES 2018: Proceedings of the 13th International Conference on Availability, Reliability and Security. 1-6. 10.1145/3230833.3232853.
[5] Khan, Imran &Alam, Mahfooz. (2017). Cloud computing: Issues and future direction. Global Sci-Tech. 9. 37. 10.5958/2455-7110.2017.00005.2.
[6] Bajpai, D, Vardhan, M, Gupta, S, Kumar, R, Kushwaha DS 2012, “Security Service Level Agreements Based Authentication and Authorization Model for Accessing Cloud Services”. in Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY), Chennai, India, Vol. 1.
[7] Da silva, DA, & De Gues, PL, 2014, “An Approach to Security-SLA in Cloud Computing Environment”, in the proceedings of the IEEE Latin-American Conference on Communications, pp. 1-6.
[8] Guila, P &Sood, S 2013, “Dynamic Ranking and Selection of Cloud Providers Using Service Level Agreements”, in proceedings of the International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, no. 6.
[9] Nayak,SK, Mohapatra, S, & Majhi, B 2012, “An Improved Mutual Authentication Framework for Cloud Computing”, International Journal of Computer Applications, vol 52, no, 5, pp. 36-41.
[10] Rady, M 2012, “Parameters for Service Level Agreements Generation in Cloud Computing- A Client-Centric Vision”. ER Workshops 2012, pp. 13-22.