Renegotiation in Service Level Agreement Management for a Cloud-Based System

AHMAD FADZIL M. HANI, Centre for Intelligent Signal & Imaging Research (CISIR), Universiti Teknologi PETRONAS
IRVING VITRA PAPUTUNGAN, Department of Electrical & Electronic Engineering, Universiti Teknologi PETRONAS
MOHD FADZIL HASSAN, Department of Computer Information Science, Universiti Teknologi PETRONAS

Managing Service Level Agreement (SLA) within a cloud-based system is important to maintain service continuity and improve trust due to cloud flexibility and scalability. We conduct a general review on cloud-based systems to understand how service continuity and trust are addressed in cloud SLA management. The review shows that SLA renegotiation is necessary to improve trust and maintain service continuity; however, research on SLA renegotiation is limited. Of the two key approaches in renegotiation, namely bargaining-based negotiation and offer generation–based negotiation, the latter approach is the most promising due to its ability to generate optimized multiple-offer SLA parameters within one round during renegotiation.

51

Categories and Subject Descriptors: C.2.4 [**Cloud Computing**]

General Terms: Legal Aspect, Management

Additional Key Words and Phrases: Negotiation, renegotiation, SLA, trust

ACM Reference Format:
Ahmad Fadzil M. Hani, Irving Vitra Paputungan, and Mohd Fadzil Hassan. 2015. Renegotiation in service level agreement management for a cloud-based system. ACM Comput. Surv. 47, 3, Article 51 (April 2015), 21 pages.
DOI: http://dx.doi.org/10.1145/2716319

1. INTRODUCTION

A cloud-based system essentially consists of virtual servers created from a combination of hardware, networks, storage, and interfaces that provide computing services over the Internet. The computing services can be in the form of software delivery or infrastructure or storage (either as separate components or as a complete platform) based on user demand. The objective of cloud computing is to provide virtual computing services for consumer-oriented applications such as financial or medical data, platforms for development, and software application services.

The essential characteristics of cloud computing are the ability to scale up and down and to perform self-service provisioning and deprovisioning based on user needs at low cost. This flexibility attracts individuals and businesses to move to the cloud-based system. However, in the case of users who want to store and govern their sensitive

This work is supported by the Ministry of Higher Education of Malaysia, under grant ERGS 0153AB-117.
Authors' address: A. F. M. Hani, I. V. Paputungan, and M. F. Hassan. Universiti Teknologi PETRONAS, 31750 Tronoh, Perak, Malaysia; emails: fadzmo@petronas.com.my, irving@uii.ac.id, mfadzil_hassan@petronas.com.my.

DOI: http://dx.doi.org/10.1145/2716319

data (e.g., financial data and medical data), a guaranteed service level from the service provider becomes important to ensure the continuity of the service [Buyya et al. 2009]. The service level is documented in the Service Level Agreement (SLA).

SLA between the service provider and the service consumer specifies the expectations of service provisioning, including penalties that should be applied when a violation occurs [Greenwood et al. 2006; Kandukuri et al. 2009; Patel et al. 2009]. It contains certain measurable conditions for the service, objectively called *service level objectives* (SLOs), such as availability, performance (throughput and response time), and scalability (number of users and request) [Kim 2013]. SLA is commonly used to address not only issues of problem management, legal compliance, security, and confidential information but also resolution of disputes with regard to customer duties and termination [Yan et al. 2007]. However, it can vary depending on the applications or data that are outsourced [Ahronovitz et al. 2010]. Moreover, the nature of cloud-based environment in which network infrastructure plays an important role, network conditions that changes frequently might affect the level of service given to the customers.

Hence, the contents in SLA are important in the adoption of cloud computing. Customers have to place the data (even critical data) and depend on the service of any cloud infrastructures, and thus the SLA also represents evidence of trustworthiness that cloud consumers have in the cloud provider [Patel et al. 2009; Alhamad et al. 2010]. Customers will not want to outsource their data without strong assurances that their requirements will be met. Therefore, an SLA within a cloud-based system must be well managed.

The objective of this article is to present a literature review on current Service Level Agreement management (SLAM) in the area of cloud computing with a purpose to motivate future work in developing better SLAM to support the flexibility and enhance the trust level of a cloud-based system. Section 3 discusses details on how each SLAM step deals with the dynamic nature of cloud. The main issues with regard to SLAM are presented at the end of the section. Based on these issues, SLA renegotiation is proposed and described further in Section 4. In Section 5, a discussion of how SLA renegotiation can support SLAM is presented. We conclude in Section 6.

2. MATERIAL AND METHODS

The published works that we reviewed were collected from IEEE Explore and Google Scholar. First, we searched for papers describing SLAM in general (Section 3), and second, we searched for papers mentioning the SLA negotiation and renegotiation approach (Section 4). The keywords “service level agreement management cloud computing” were used for the former search, whereas “service level agreement negotiation renegotiation cloud computing” was used in the subsequent search.

As the cloud computing trend began to emerge in 2008, our review is limited to works published between 2008 and 2014. Some of the service-based or service-oriented systems were also retrieved in the search. The articles yielded were then refined to ensure that only the related ones were selected for the review. In general, the search yielded articles that contained information related to SLAM, SLA negotiation, or SLA renegotiation, including renegotiation approaches for cloud-based systems and other service-based systems.

In Section 4, the review initially focuses on a high-level description of SLA negotiation that has an impact on renegotiation. It covers aspects such as specification language, framework, architecture, annotations, and protocols. By reviewing such works, the requirements for SLA renegotiation are then derived. Strategies that cover several decision support systems or techniques from several works on contract negotiation are subsequently reviewed to find the most suitable one for SLA renegotiation. Both reviews are sorted based on relevancy and publication year.

Fig. 1. SLAM life cycle for a service-based system.

3. SLAM

The management of SLA within a cloud-based system comprises five main states: negotiation of the contract, resource provisioning, service delivery monitoring, service performance assessment, and service termination, as depicted in Figure 1.

In this section, we present a general review of how each process in SLAM deals with the dynamic nature of the cloud environment with regard to the level of service delivered. Such a dynamic situation is caused by three factors: (1) service requirements of the service consumer that may change over time, (2) unpredictable network performance, and (3) regular system maintenance that the provider needs to perform. In addition, some available services that focus on one or more SLAM phase are investigated to summarize the state of the art.

3.1. SLA Negotiation

SLA is established by a negotiation process between two parties prior to service provisioning. Such negotiation is normally aimed at maximizing revenue while minimizing the service cost to both parties. Thus, establishing an SLA in business is not a simple task, especially when the situation is frequently changing in the cloud computing environment. Several works on SLA negotiation within cloud-based systems dealing with such changing environments from different points of view have been reported.

Buyya et al. [2009] proposed broker-based negotiation to deal with the various demands of cloud users. Typically, the broker will select suitable providers and negotiate on behalf of users. The broker-based approach has been also reported by Cuomo et al. [2013], Badidi [2013], Wu et al. [2013], and Amato et al. [2013]. In dealing with the same problem, Stantchev and Schropfer [2009] and Macias et al. [2010] proposed to differentiate the customers into several levels based on their requirements of the service, such as Platinum, Gold, and Standard. Alsrheed et al. [2013] tried to solve the various preferences of all parties in the cloud by developing automated negotiation and customizable SLA. Xu and Li [2013] and Henzinger et al. [2010] proposed a dynamic service pricing with different service requirements to obtain better profit. Both focus on the use of idle computing resources that may speed up the computing process. Customers will choose the most suitable service price based on their requirements, and providers will then finish the jobs as expected. However, all of the works mentioned focus more on dealing with different user requirements.

In a different fashion, runtime SLA negotiation for dealing with the changes within a cloud environment is proposed [Mahbub and Spanoudakis 2011; Mehdi et al. 2011; Groleat and Pouyllau 2011; Mach and Schikuta 2012]. Mahbub and Spanoudakis [2011] argue that by having a proactive runtime negotiation with other service providers, service failure is avoidable, whereas Mehdi et al. [2011] map an urgent job to the cloud services with fewer negotiation rounds. Groleat and Pouyllau [2011] utilized reinforcement learning (RL) to learn the dynamic customer demand during service runtime and in turn maximize the cloud revenue. Mach and Schikuta [2012] proposed an automatic negotiation and renegotiation framework to accommodate changing demands from cloud users. Runtime SLA negotiation and the SLA negotiation approach are believed to be capable of supporting all cases in the dynamic nature of the cloud environment.

In summary, there are some benefits to having a runtime SLA renegotiation. First, it is a better way to realign the delivered service with the current business strategies. Second, it is a win–win solution prior to service failure due to unexpected situations in the cloud environment. Third, it is potentially able to reduce the cost of service in the case of resource underutilization. Fourth, it presents an opportunity to extend the relationship between the provider and customer by adding some values to the renegotiated contract.

3.2. Resource Provisioning

Another key process in SLAM for cloud computing is resource provisioning. The process provisions and allocates resources per customer request, such as creating virtual machines (VMs) and storage space based on the negotiated price. Here the focus is to minimize the infrastructure and operational cost while maintaining the agreed SLA from the provider's point of view. However, resource provisioning and allocation are challenging due to the dynamic changes in the cloud environment, in which SLA violations can occur and there may be diverse customer requests with regard to resources.

Boloor et al. [2010] suggested the idea of context awareness to adapt the changing on-demand usage within automated cloud management instead of providing static resources. User context with regard to different situations will provide smarter and more tailored responses to enhance customer service. A novel approach that allocates user requests dynamically based on different classes of users is proposed. Related to this, Zaman and Grosu [2013], Zhang et al. [2014], and Chang et al. [2010] set the problem as an NP-problem and developed an optimal solution for allocated resources in the cloud from the users' perspective. In addition, Wu et al. [2011] worked on resource allocation for Storage as a Service (SaaS) providers to minimize infrastructure cost and SLA violations based on customer quality of service (QoS) parameters (e.g., response time).

On the other hand, to ensure the availability of cloud service, Addis et al. [2010] developed resource allocation policies based on the adaptation to the cloud infrastructure in terms of performance and energy trade-offs. Such an SLA-driven approach for resource provisioning is also being used in Bonvin et al. [2011]. The same approach is used by Ledoux and Kouki [2012] to optimize the resource capacity planning in the cloud SaaS model. This approach is meant to maximize revenue and minimize cost from a provider perspective. In addition, self-capacity planning is also designed to meet changing demands of the user. Therefore, resource self-reconfiguration can be performed automatically and seamlessly.

Based on the pricing model, Rogers et al. [2010] described a framework that will optimize system cost while satisfying QoS expectation for database management systems on the Infrastructure as a Service (IaaS) cloud. The framework that contains a resource consumption prediction and a coarse-grained workload profiling is formalized as constraint problem programming. Added with execution duration consideration, Henzinger et al. [2010] designed a flexible framework called *FlexPRICE* to give cloud users different job scheduling for their submitted job to a cloud provider. However, both works have difficulty in creating a precise user workload profile.

To avoid SLA violation, Yazir et al. [2010] and Maurer et al. [2010] manage the resources in a distributed architecture to avoid wasting resources and overutilization. The service reconfiguration in this work is done in an event-triggered manner, based on the occurrence of any SLA violation. Furthermore, Hedwig et al. [2011] developed a framework that harmonizes economic and operational goals within an elastic information system to help dynamic on-demand cloud resource provisioning. The framework benefits by using historical data and the performance model to derive an optimal operational strategy.

Zhu et al. [2011] proposed a dynamic resource provisioning technique and a flexible hybrid queuing model within cloud data center architecture to improve the overall performance through effective cost reduction of extra resource usage and maximizing the global profit of cloud infrastructure providers. The architecture includes a resource adjustment process due to dynamic variations in workload.

Above all, most of the resource provisioning works were carried out to find the optimum allocation while avoiding service interruption. Violation of the agreed service level will undeniably reduce the trust level of a service provider from the customer's perspective. Hence, the understanding of customer context using historical and performance data becomes important to provide the resource optimally in a cloud-based system.

3.3. Service Monitoring and Assessment

The need for efficient monitoring and evaluation on the delivered service becomes important due to the dynamic situation in the cloud environment. The monitoring process will continuously capture the performance of physical and virtual servers, network performance, the shared resources, and the applications running on them. Such data is then evaluated for postprocessing (e.g., SLA violation prediction and detection). Both SLA monitoring and assessment are central to the development of trust level from a customer's perspective.

Bouchenak [2010] discussed the issues of performance, availability, and economical cost of the cloud with regard to SLAM. SLA as a service is introduced to assist such issues and is applicable to IaaS, Platform as a Service (Paas), or SaaS. The paper proposed ways to manage the SLA within a cloud environment automatically to address its elasticity. A monitoring tool that can capture variations in cloud usage and workload, detect SLA violation, and trigger cloud reconfiguration automatically is mentioned in the proposal. Romano et al. [2011] presented a novel seamless QoS monitoring facility for cloud computing to increase the level of trust of potential cloud customers. The gist of this work is "quality of service monitoring as a service." However, if SLA is violated, there is no recommended action to resolve it.

On cloud usage, Jamkhedkar et al. [2011] introduced the notion and importance of usage management in cloud computing, as existing SLA frameworks are not designed for usage policies. The ability to express usage policies in detail provides cloud customers with greater trust and confidence to employ cloud services. The developed framework also offers the possibility to track the usage with customer data.

Faniyi and Bahsoon [2011] considered the challenge of providing more dependable, transparent, and trustworthy provision of services in cloud service providers through the development of an innovative decentralized cloud SLAM framework. Monitoring and responding to SLA are the main items addressed with regard to those challenges. The framework presented self-awareness and self-expression toward meeting SLAs. SLA renegotiation is highlighted as one of the framework requirements to address the possibility of resource reprovisioning in real time.

Aceto et al. [2012] recently provided an analysis of monitoring in cloud computing. It is mentioned that trusted cloud systems can be justified if they can persistently perform service delivery while facing changes. Cloud monitoring, on the provider side, must be able to prevent and recover any violation without adding any performance burden. Ghosh and Ghosh [2012] added that monitoring on the service customer side is also important for monitoring any provider outages. In addition, the SLA parameter of security is not well addressed in current practice. The work done is focused on Storage-as-a-Service.

To estimate service response time, Salah and Boutaba [2012] proposed a model that is able to predict the size and number of computer instances to provide minimum VMs and avoid service violation. Violation in this work refers to either overprovisioning or poor utilization of resources. Moghaddam et al. [2012] developed a fault prediction framework in cloud computing based on the behavior of clients. This event-based mechanism profiles the clients by analyzing their behavior to increase system resilience. The ability of the provider to determine, estimate, and measure the level of service is arguably able to assist the SLA negotiation process.

Recently, Jehangiri et al. [2013] presented a framework for monitoring and evaluating cloud service to detect the root cause of performance problems. The authors argue that the current solutions have some limitations, such as limited scalability and an error-prone diagnosis procedure, when monitoring large-scale virtual platforms. One of the main requirements mentioned is the ability to perform proactive resource management to evolve the relevant metrics during service delivery.

It is concluded that monitoring and assessment of service delivery is important for developing a certain level of trust from a cloud customer perspective. The monitoring will not only acquire the service performance data but also the resource usage during the service lifetime to be able to detect or predict any violation. Both mechanisms need to be performed in transparent way. In case of any violation, service reconfiguration must be considerably executed based on historical data and the initial agreement.

3.4. Service Termination and Penalty Management

Either when the service lifetime is expiring or any unacceptable violation has occurred, the contract of service termination would be initiated by both parties. In the latter case, a the service provider incurs a penalty. Kotsokalis et al. [2009] developed penalty management in formal definition that takes into account the fairness, business value, and QoS parameters to solve ambiguity in expressing the penalty. A failure ratio is formulated to determine the relationship between planned quality and achieved quality; this formulation can also be used to reason any possible reward.

On a different note, Seibold et al. [2011] reported an idea to solve the lack of stringent SLOs and significant penalties in SLA for SaaS business applications. MobiDB was developed to provide strict guarantees on low response time at high throughput rates while minimizing space overhead in an operational business intelligence scenario. With stricter SLAs, customers find it easier to compare different cloud offerings.

3.5. SLAM Service

Several readily available online or offline tools or services support SLAM within large-scale systems, such as the cloud-based system. Table I illustrates the various SLAM phases, corresponding service characteristics, and SLA manager tools. It can be seen that some tools address one or more SLAM phases. Most of the tools deal with monitoring and assessment phases of delivered service either proactively or reactively. This is because monitoring and assessment play an important role in ensuring that service requirements are being met as stated in the SLA. Furthermore, service providers continually have to monitor and assess their service levels for real-time adjustments and plan for future services. However, only a small number of tools are available for these phases.

For service negotiation, most SLA managers use the service offering or catalogue model for their customers. Customers choose services based on their preferences. The SLA manager then helps the provider allocate the resources as needed. Different approaches have been proposed in Cloudcuity and mOSAIC by providing a cloud services brokerage to find the most profitable service. To improve a service plan, the SLA adjustment is introduced based on history-related data and reports, as in the case of the HP Universal SLA Manager (HP USLAM). In addition, HP USLAM supports penalty calculation in the case of an SLA violation to manage the financial risks.

Table I. SLAM Services

SLAM Phase	Service Characteristics	SLA Manager Tools
Service negotiation	Brokering	mOSAIC (www.mosaic-cloud.eu) Cloudcuity (www.njvc.com)
	Offering	IBM Maximo (www.ibm.com) OTRS (www.otrs.com) LANDesk (www.landesk.com) HP USLAM (www.hp.com)
Resource provisioning and allocation	SLA driven	Kaseya (www.kaseya.com) OTRS LANDesk Oracle EM (www.oracle.com) HP USLAM
Service monitoring and assessment	Proactive	OTRS LANDesk PRTG (www.paessler.com) Kaseya HP USLAM
	Reactive	SolarWinds (www.solarwinds.com) Altiris (www.incit-technology.com) mOSAIC Oracle EM
Service termination and penalty	Event driven	HP USLAM

Although resource provisioning usually is handled by SLA managers based on the agreed SLA, Oracle EM is also able to suggest an optimum resource allocation (e.g., the required amount of memory for VMs).

From the research point of view, mOSAIC solution, an open-source project by European Consortium, is one of the most interesting ones. Using mOSAIC, a cloud-based system is able to perform broker-based SLA negotiation to facilitate finding the most suitable provider for service consumers, especially when they need to change a provider regularly (e.g., monthly). mOSAIC also performs monitoring and evaluation of resource utilization and system performance to provide the effective values of QoS parameters. In addition, the mOSAIC package is open to cloud developers.

3.6. Main Issues in SLAM

Among the studies that have been done on SLAM within the cloud-based system, it is clearly mentioned that cloud service providers continuously need to adapt the network situation and changing demands of customers to avoid violations during service delivery that may affect the agreed SLA. Table II provides some salient points with regard to cloud SLAM to maintain service delivery. Reconfiguring the resources to support service continuity is what was suggested by most reference works. However, considering that available resources are limited, SLA renegotiation is advised as well. Services that are maintained and evolve regularly can lead to fewer service level violations. Such violations may reduce trust with regard to the cloud provider.

As seen in Table II, detecting and predicting SLA violations are central to maintaining the delivery of cloud service. The proper approach must be able to capture the service level performance and resource usage to trigger an alert to any abnormalities. The service provider can subsequently perform service reconfiguration if necessary; otherwise, penalty will be incurred. Only a few works are found in the literature that focuses on penalty in cloud computing, possibly because executing penalty should be avoided, if possible. It is better to retain customer satisfaction by reviewing an ongoing contract to apply certain acceptable levels when a violation takes place.

Table II. Salient Points of Cloud SLAM on Maintaining the Service Delivery

	SLA Negotiation	Resource Provisioning	Monitoring and Assessment	Service Termination
SLA Renegotiation	Mach and Schikuta [2012] Mahbub and Spanoudakis [2011] Mehdi et al. [2011] Groleat and Pouyllau [2011]		Faniyi and Bahsoon [2011]	
Service/Resource Reconfiguration	Buyya et al. [2009] Stantchev and Schropfer [2009] Macias et al. [2010]	Boloor et al. [2010] Chang et al. [2010] Zaman and Grosu [2013] Addis et al. [2010] Ledoux and Kouki [2012] Yazir et al. [2010] Maurer et al. [2010] Zhu et al. [2011]	Bouchenak [2010] Faniyi and Bahsoon [2011]	
SLA Violation Detection and Prediction		Rogers et al. [2010] Yazir et al. [2010]	Bouchenak [2010] Romano et al. [2011] Aceto et al. [2012] Salah and Boutaba [2012] Moghaddam et al. [2012]	
Penalty		Wu et al. [2011]		Seibold et al. [2011] Kotsokalis et al. [2009]

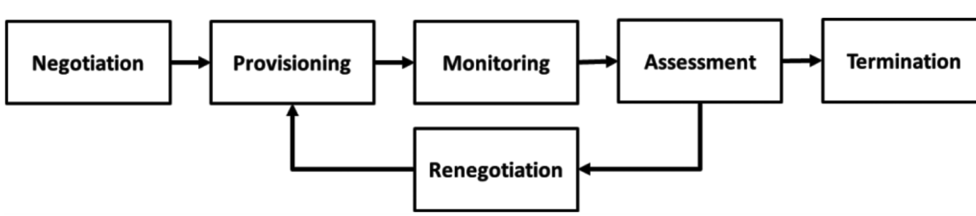

Fig. 2. SLAM life cycle with a renegotiation state.

4. SLA RENEGOTIATION

Renegotiation is defined as second or subsequent negotiations to alter the terms of an existing agreement [Parkin et al. 2008]. Renegotiation allows customers and providers to initiate changes in an established agreement prior to service/resource reconfiguration, such as if storage size need to be resized with growing data and bandwidth since the capacity channel tends to be reduced at certain periods of time. It is evident that changing circumstances lead to the development of an SLA renegotiation. Therefore, it is necessary to add an SLA renegotiation process in the SLAM [Sharaf and Djemame 2014; Patel et al. 2014] for the cloud-based system, as shown in Figure 2. The SLA renegotiation mechanism should be carefully designed and, more importantly, made to achieve optimum benefit out of the cloud service for the parties involved. In addition, increased flexibility in the SLA will enhance trustworthiness of the cloud service [Spoorthy and Sreedhar 2012]. Thus, besides having good security, reputation, and a transparency mechanism [Uusitalo et al. 2010], SLA renegotiation can also play an important role in increasing levels of trust in cloud technology.

In 2005, a Rouse survey of IT managers and directors of the largest 1,600 sites in Australia engaged in IT outsourcing reported that SLA renegotiation will lead to greater outsourcing success and suggested that, at the very least, SLA contracts should be renegotiated annually. However, renegotiation is not a trivial task [Sharaf and Djemame 2010], as it requires specific protocols for changing the SLA parameters, such as availability, response time, and throughput [Venticinque et al. 2011]. The renegotiation protocol includes revoking the previous SLA, SLO redefinition, negotiation the new one, and assessing the negotiation boundaries in which the SLO can be renegotiated with regard to its price. Furthermore, if an important SLO is violated, renegotiation can be difficult, because typically it can affect business profit [Venticinque et al. 2011]. Above all, it is necessary to have a mechanism to help SLA renegotiation in the cloud-based architecture.

4.1. High-Level View on SLA Negotiation and Renegotiation

There are several reported significant works on SLA negotiation and renegotiation that developed frameworks and protocols with corresponding specifications, such as Web Service-Agreement (WS-A) [Parkin et al. 2008], Proactive Service Discovery and Negotiation (PROSDIN) [Mahbub and Spanoudakis 2011], and Simple Bilateral Negotiation Protocol (SBNP) [Yaqub et al. 2012].

WS-A is the most-referenced negotiation framework; it contains several protocols that are widely used in service-based systems. It is a XML-based protocol developed initially by the Open Grid Forum for establishing SLAs between service providers and consumers in grid computing. It not only specifies the agreement but also provides a template that describes the requirements and assurances on service quality. Due to limitations in negotiation specifications, it can only offer the basic negotiation; in particular, the offer can only be accepted or rejected and does not cater to any changes from either party during the service lifetime. This basic WS-A negotiation has been extended in many ways, as discussed in the following section.

Another SLA negotiation framework called *PROSDIN*, which proactively reduces the extent of interruptions during service maintenance, is proposed by Mahbub and Spanoudakis [2011]. The framework can discover candidate constituent services during runtime. The SLA negotiation in such a framework is built as part of the service discovery to make a preagreement with other providers to ensure that sufficient resources are provisioned during the service lifetime. However, the framework does not resolve a situation where an agreed-upon resource is somehow not activated.

Yaqub et al. [2012] proposed a negotiation protocol, namely SBNP, in which the negotiation is expressed by declarative rules and implemented by two finite state machines: sender and receiver. Thus, the negotiation is an automated process and can be used either for negotiation or renegotiation purposes. However, the framework does not consider the existing agreement when performing the renegotiation procedure.

In the following section, WS-A, PROSDIN, SBNP, and other models are further analyzed to understand features of SLA renegotiation, such as proactive/reactive and bilateral/multilateral. Proactive/reactive negotiation allows us to determine how SLA renegotiation can be invoked. Bilateral/multilateral negotiation motivates us on providing the resources optimally during renegotiation.

4.1.1. Proactive and Reactive Renegotiation. In the case of proactive renegotiation, there are two ways in which renegotiation can happen. In the first case, the customer can initiate the renegotiation to support the changing requirements during the service lifetime

Table III. Related Works on Proactive and Reactive Renegotiation

Renegotiation	Approach
Proactive	Add renegotiation state in current protocol that supports renegotiation [Parkin et al. 2008; Ziegler et al. 2008; Battre et al. 2010; Langguth and Schuldt 2010; Yaqub et al. 2011, 2012]
	Extend the WS-A by integrating runtime renegotiation function [Di Modica et al. 2009; Sharaf and Djemame 2010]
	Offer renegotiation proposal based on resource constraint, temporal restrictions, and previous offer [Galati et al. 2013]
	Discover candidate constituent services and make a preagreement [Mahbub and Spanoudakis 2010, 2011]
Reactive	Add renegotiation state in current protocol that supports renegotiation as another negotiation [Parkin et al. 2008; Mach and Schikuta 2012; Yaqub et al. 2011, 2012]
	Modify the SLA template in WS-A [Brandic et al. 2009]
	Renegotiate the contract to certain acceptable boundaries [Venticinque et al. 2011]

before service termination. In such circumstances, a multiround renegotiation is then added into the current WS-A negotiation protocol to accept a new proposal from the customer. The new proposal contains a different service level to which the provider must comply [Parkin et al. 2008; Ziegler et al. 2008; Battre et al. 2010; Langguth and Schuldt 2010; Yaqub et al. 2011, 2012]. Di Modica et al. [2009] proposed a "modifiable SLO" as opposed to the normally fixed SLO to be incorporated into the WS-A for such renegotiation. In the second case, the provider can initiate the renegotiation to prevent any violation on the agreed-on service level due to unexpected changes in the cloud environment. Sharaf and Djemame [2010] and Di Modica et al. [2009] proposed runtime renegotiation within WS-A in such cases to avoid service suspension. Resource constraint, temporal restrictions, and previous SLOs need to be reconsidered prior to presenting a new proposal to a customer [Galati et al. 2013]. To reduce the extent of interruptions during service maintenance, PROSDIN incorporates a runtime negotiation to allow preagreement for a composite service with other constituent services [Mahbub and Spanoudakis 2010, 2011].

In the reactive case, when any SLA violation is detected, renegotiation becomes an option instead of terminating the service [Brandic et al. 2009; Mach and Schikuta 2012]. The negotiation frameworks mentioned in the proactive case can also be used to support this kind of renegotiation [Parkin et al. 2008; Yaqub et al. 2011, 2012]. In addition, Venticinque et al. [2011] stated that the content in the contract could be renegotiated to certain acceptable boundaries. Table III summarizes the proactive and reactive renegotiation approaches.

Proactive SLA renegotiation is preferable to maintain certain levels of trust on the provider side. This is because if any SLO violation occurs, it may affect business operations [Wu and Buyya 2010] and consequently the credibility of the provider [Di Modica et al. 2009]

In order to perform proactive renegotiation of the the SLA, the provider must monitor and assess the service delivery to forecast the possibility of any violation. The QoS historical data obtained in the monitoring process plays an important role in predicting the occurrence of violations. When an SLA violation is predicted, acceptable values of QoS must be redefined prior to renegotiation. From a customer point of view, this option is arguably better than terminating the service.

4.1.2. Bilateral and Multilateral Renegotiation. There are two different types of negotiation based on the number of parties involved: bilateral (one-to-one) and multilateral (one-to-many). Bilateral negotiation is one of the most widely used negotiation style between

Table IV. Salient Features and Challenges of Bilateral and Multilateral Negotiation

Renegotiation	Main Features	Challenges
Bilateral Parkin et al. [2008] Ziegler et al. [2008] Yaqub et al. [2012]	The deal is often optimum for both parties Direct communication	How to reach the agreement quickly and optimally
Multilateral Vigne et al. [2012] Siebenhaar et al. [2012]	The deal is done on the highest utility value Broker-based communication	How to deal with resource availability when performing concurrent multilateral negotiation

the service provider and service consumer, which is done directly. The WS-A initially specifies such negotiation within a service-based system with a take-or-leave-it model. In a subsequent model, a renegotiation protocol is included under the negotiation type instance to assist bilateral renegotiation [Parkin et al. 2008; Ziegler et al. 2008; Yaqub et al. 2012]. Such renegotiation of the previous contract is considered the same as negotiation [Battre et al. 2010] in a multiround model.

With the emergence of cloud computing technology, multilateral negotiation becomes the alternative protocol as several cloud providers compete with one other to attract customers. It can be either a one-to-many or many-to-many relationship model. One-to-many negotiation takes place when a customer sends service requirements to a negotiation agent to find the most suitable service from providers at a low cost [Vigne et al. 2012]. An auction style will normally be used for one-to-many negotiation [Yaqub et al. 2011; Mach and Schikuta 2012]. In the many-to-many model, Siebenhaar et al. [2012] proposed that multilateral negotiation should also support a provider in performing concurrent negotiation with several consumers to determine which consumer generates the highest profit.

Table IV describes the salient points and challenges between bilateral and multilateral negotiation. Bilateral negotiation, on the one hand, is able to approach an optimum agreement either at the early stage or after several rounds of direct offer and counteroffer. On the other hand, in multilateral negotiation, the deal will be carried out on the preferences that can give the highest utility value. However, in the case of several service consumers performing concurrent negotiation, the provider must consider the resource availability.

In both types of negotiation relationship, to maintain the credibility of the service delivered where violation normally occurs, bilateral renegotiation should be conducted in timely manner to achieve an optimum resource allocation.

4.1.3. SLA Renegotiation Requirements. In summary, the design and analysis of a high-level view of SLA negotiation and renegotiation mechanisms provide a clear perspective on how SLA renegotiation should be performed. Based on the preceding discussion on SLA renegotiation mechanisms in service-based systems, the following requirements are proposed to perform SLA renegotiation and evaluate the quality of the renegotiation strategy used:

(1) The renegotiated agreement is optimized for both parties, especially after considering the previous agreement.
(2) Renegotiation can be initiated by either the service provider or consumer.
(3) Renegotiation is able to deal with any violation that is predicted.
(4) Renegotiation should be performed in a timely manner so that the new agreement can be reached quickly, especially when any SLA violation is predicted or detected by the monitoring process. It can be achieved by minimizing the number of renegotiation rounds or giving several offers in one round.

4.2. Decision-Making Strategies Used in SLA Negotiation for Consideration in SLA Renegotiation

Having a strategy during SLA renegotiation helps the service provider and consumer achieve their service objectives. In a service-based system, such objectives are normally related to prices that have to be paid by the service consumer in the agreement. To determine how to obtain on an optimal contract pricing, Wilkes [2008] introduced the application of utility theory in SLAM. Using the utility function, preferences from both parties can be weighted and calculated. Several decision-making strategies that cover decision support system techniques for SLA negotiation can be considered in SLA renegotiation.

In the following sections, strategies that have been applied to support negotiation within service-based systems are reviewed based on the aforementioned renegotiation requirements to obtain the most suitable one to perform second or subsequent SLA negotiation. The strategies are categorized into bargaining-based negotiation and offer generation–based negotiation. Such categorization is derived from the way the deal is reached between the service provider and service consumer during negotiation.

4.2.1. Bargaining-Based Negotiation. Conventional SLA negotiation is normally about accepting or rejecting an offer. In bargaining-based negotiation, the notion of counteroffer is introduced. Both parties will make offer and counteroffer until an agreement reached. The machine learning method is applied in some cases to learn the opponent's preferences or previous successful negotiations. This negotiation approach focuses more on the goal; therefore, a negotiation agent will normally calculate the option that will maximize the final gain.

Game theory. Figueroa et al. [2008] introduced a mathematical model for reaching an agreement in the negotiation process. The model is similar to the bargaining process in game theory, where the client is given not only a take-it-or-leave-it contract but also the possibility to make a counteroffer. Kaminski and Perry [2008] presented a framework to create SLA autonomously, based on agreed-on SLOs that are evaluated using utility function. Resource prioritization is also realized in the framework to generate possible offers to the customers. A similar idea is presented by Xiaotai and Xiaoyan [2009], who developed a new method called the *bargaining algorithm* to evaluate the utility value of the offer from the opponent (counterpart) and create its own subsequent offer–counteroffer. Zheng et al. [2010] then considered the Web service SLA establishment negotiation as a bargaining game.

Issues related to the preceding bilateral bargaining include incomplete information, time constraint, and multiple attributes. Incomplete information means that both parties will not know each other's preferences. Time constraint is a finite duration that has been set for the bargaining process to terminate. Multiple attributes are intended to distinguish each parameter in the SLA that must be normalized to a certain associated utility value. Although the bargaining process can resolve conflict between the customer and provider within a negotiation, it normally takes several rounds to reach the deal. This is due to the incomplete information that is possessed by both parties involved in the negotiation. In addition, an occurrence of a situational update during negotiation may result in a less accurate estimation of the counterpart's offer.

In order to perform fast and simple negotiation, Zulkernine and Martin [2011] proposed time-based decision functions to help in bilateral bargaining. With this kind of function, an agreement can be reached within the maximum negotiation time if the interest of both parties intersects after several rounds of negotiation. However, there is a possibility for nonoptimal agreement to be reached at the end of the negotiation period. To reach for a quick agreement, Al-Aaidroos et al. [2011] suggested the need to learn from the previous successful negotiation and to consider the counterpart's behavior in generating a proposal.

Badidi [2013] proposed a cloud service broker (CSB) to overcome the multiattributes issue. The broker helps the service consumers select the right provider to fulfill their requirements. The provider will be ranked using the multiattributes model by matching the QoS. To reach an agreement, both parties will negotiate the SLA through several rounds of offers and counteroffers or until a predefined maximum number of rounds.

Reinforcement learning. One of the machine learning approaches to support agreement negotiation is the RL algorithm. It works by (1) learning the incomplete information from either the provider or customer [Jian 2008] and (2) learning the negotiation policy, especially when there is any situational update [Groleat and Pouyllau 2011; Pouyllau and Carofiglio 2011]. The objective of RL is to maximize the final revenue from the provider point of view during negotiation. The RL technique performs better in the initial negotiation, as it works faster without remembering the negotiation information. A major challenge in RL is the low runtime performance due to the tasks involved in readapting and relearning any changing situation if performing SLA renegotiation.

Bayesian learning. Bayesian theory has been applied in the contract negotiation domain to discover the previous negotiation history decision by comparing the offer probability. It can also learn the information given during negotiation until the deal is reached. Lau [2008] tried to discover nonparametric negotiation knowledge during automated negotiation. It learns from vital information such as negotiator preferences without making any assumption based on previous series of offer exchanges. Hindriks and Tykhonov [2008] then modeled the opponent preferences using the Bayesian learning approach. The proposed algorithm can effectively learn from the counterpart during negotiation by making some assumptions about the preferred structure and bidding process rationality. However, there are issues that (1) the successful rate of the Bayesian agent normally depends on the first step of negotiation and (2) it has more possibility to finish the negotiation at the early stage, which results in a less optimal outcome.

Silaghi et al. [2010] then developed an intelligent SLA negotiation framework based on the time constraint strategy to overcome the preceding issues. They proved that the time-constrained negotiation strategy could exceed the baseline learning scheme in lengthy negotiation by extending the Bayesian learning agent. They argued that time is important in a negotiation because resource allocation must be immediately performed in a finite time. Silaghi et al. [2011] then refined the framework to be applied in computational grids as to compete the available resources for the allocation of available resources. However, the way in which a resource is allocated within a cloud environment is completely different from grid computing. Thus, performing SLA renegotiation using the Bayesian learning theory is less feasible.

Rule based. Rules play an important role in agreement negotiation. Mahan et al. [2011] developed negotiation agents based on Event-Condition-Action (ECA) rules based in e-commerce. Using combined and fairness protocols, the agents negotiate on behalf of the sellers and buyers to reach the best deal. The agents in this work are also equipped with learning properties based on a fuzzy decision tree for updating the knowledge base to support the ongoing negotiation. However, determining which rules should be used during negotiation is the challenge in this approach.

Above all, the authors maintain that negotiation is a proposal-exchanging process. Both parties can gradually learn how their counterparts place an offer until a deal is reached, which is normally a win–lose situation. However, timing during negotiation becomes important issue in this approach. Table V summarizes the previously mentioned bargaining-based negotiation techniques.

Table V. Bargaining-Based Negotiation Techniques

Techniques	Description	Advantages	Challenges
Game theory [Figueroa et al. 2008; Kaminski and Perry 2008; Xiaotai and Xiaoyan 2009; Zheng et al. 2010; Zulkernine and Martin 2011; Al-Aaidroos et al. 2011; Badidi 2013]	Assist the offer–counteroffer process in each negotiation round	Able to resolve conflict without complete information about the opponent	A nonoptimal deal is possible to reach at the end of negotiation time
Reinforcement earning [Jian 2008; Groleat and Pouyllau 2011; Pouyllau and Carofiglio 2011]	Learns and changes the negotiation policy during the negotiation	Able to learn in incomplete information settings	Computationally expensive during learning process
Bayesian learning [Lau 2008; Hindriks and Tykhonov 2008; Silaghi et al. 2010, 2011]	Learns and predicts vital information regarding negotiator preferences	Can learn opponent's preference	Computationally expensive on a large number of parameters
Rule based [Mahan et al. 2011]	Providing knowledge base to make a counteroffer in the negotiation	Counteroffer can easily be generated based on the rule	Determining which rules will be used is tricky

4.2.2. Offer Generation–Based Negotiation. The offer generation approach in negotiation is defined as providing some offers with various SLO levels to achieve the expected utility value of the requirement. Such variation gives the service consumer more understanding about what they probably take as the agreement. The customer chooses one of the offers as the agreed-on SLO. This approach frames the negotiation as an interaction for a win–win solution.

Placement technique. Ismail et al. [2010] employs the placement technique to support SLA negotiation in service-oriented computing, such as the cloud. Such technique considers temporal, resource, and business-related factors. The placement technique will find the optimal placement for a set of requirements into a set of timeslots after several cycles. The offers will be generated with differentiated levels of service to comply with the requirements. However, there is a possibility that the approach fails to find a solution in the early cycle of offer generation.

Genetic algorithm. The genetic algorithm (GA) is another technique used to support negotiation that works as a heuristic agent to find the best negotiation offer. Niu and Wang [2008] presented the idea of using the GA in agent-based automatic negotiation for an e-commerce environment. To obtain a certain level of satisfactory plans, the GA generates an optimum offer; this can be more than once based on certain criteria from both parties. This work shows how well the GA constructs the solution space within one cycle. However, the GA is computationally expensive; careful consideration is needed when applying the GA in renegotiation.

Multicriteria decision making. SLA negotiation can involve SLA properties based on several criteria. Multicriteria decision making (MCDM) techniques can be employed to facilitate SLA negotiation, such as simple additive weighting (SAW) and the analytical hierarchy process (AHP), to rank all possible alternatives for obtaining the best decisions according to the the agreed-on SLA. Utomo et al. [2009] used AHP for multicriteria decision making to provide ranked agreement options before engaging into negotiation within a group of stakeholders. In this work, the game theory model (with complete information) that involves forming coalitions is used to determine the best solution from the options. Gomes and Madeira [2011] proposed a fully automatic SLA negotiation protocol to allow the client to negotiate network resources and the protocol

Table VI. Offer Generation–Based Negotiation Techniques

Techniques	Description	Advantages	Challenges
Genetic algorithm [Niu and Wang 2008]	Generate offers based on utility values from customer preferences	Can give multiple best negotiation plans with multiple attributes Can creates several possible solutions	Computationally expensive during optimization
Multicriteria decision making [Utomo et al. 2009; Gomes and Madeira 2011; Amato et al. 2013]	Provides ranked agreement alternatives	Can work with multiple attributes	Difficult to assess the optimum solution when performing bilateral negotiation
Placement technique [Ismail et al. 2010]	Places the requests into timeslots based on resource availability	Simple implementation Can represent the problem of finding the proper request assignment	Computationally expensive on a large number of parameters

Table VII. Comparison between Bargaining and Offer Generation Approaches

Approach	Features	Challenges
Bargaining	Conflict resolution Win–lose solution	No or little information (mainly service price) on preferences is available during negotiation. Offer and counteroffer is a time- consuming process in reaching a deal
‘Offer generation	Multiple offers within one negotiation round Potential win–win solution	Requires understanding on the current situation within the cloud environment Requires a suitable learning technique prior to generating the offer

stack used in the virtual network. The negotiation strategy used is MCDM: weighted sum model, weighted product model, and AHP. However, MCDM in most cases is less accurate due to the difficulty in giving an exact value for each criterion. A similar approach is proposed by Amato et al. [2013], who applied the MCDM concept to rank the offers within an agent from different cloud providers that suit the most customer preferences.

Such a technique is attractive, as it may reduce the duration of renegotiation by providing ranked SLA alternatives to the customer. However, MCDM fits better in multilateral renegotiation when a customer wants to find the best provider compared to bilateral SLA renegotiation cases.

In summary, the offer generation mechanism that is performed within SLA negotiation is able to provide various proposals to customers concurrently. Such an approach uses objective criteria to find a mutual deal condition after exchanging information. However, the successful rate of this approach may increases if parties are willing to share information openly. The comparison of the techniques used within offer generation–based negotiation is depicted in Table VI.

5. DISCUSSION

Table VII broadly summarizes key points of the two main SLA negotiation approaches, namely bargaining and offer generation, used in renegotiation. However, even though a systematic review has been done on these works, it is not possible to make a clear distinction among them, and the key points may overlap.

Bargaining-based negotiation works based on no or little information regarding both parties' preferences. It will perform the offer and counteroffer process until the deal is reached based on the utility value of the preferences. The customer makes an offer to the provider. Game theory or other machine learning methods will then assess the utility value to provide a price based on such preferences or previous successful negotiation. If the offer is accepted by the customer, the negotiation is over. If it is refused, the customer will make a counteroffer. The process will be repeated until one of them accepts the offer or until the deadline. It normally takes a considerable amount of time to perform such negotiation rounds to reach the deal. This becomes one of the challenges of bargaining-based negotiation. Yet prolonged negotiation rounds will only give a small concession to the other side to accept an unavoidable offer.

Bargaining-based negotiation is also regarded as a nonzero sum negotiation. Even though such negotiation is able to resolve the conflict between the service provider and service consumer, the common result is a win–lose solution because the objective is to maximize self-utility value. One side will obtain more utility values than the other. In addition, win–lose negotiation requires a good knowledge-based system to provide beneficial offers to both parties. Such pros and cons show that bargaining-based negotiation is less suitable for implementation in cloud-based SLA renegotiation.

The offer generation approach is a more advantageous approach in the renegotiation stage. A win–win solution reflects the ability to provide optimized SLA-negotiated parameters by both parties. In addition, by giving multidifferentiated offers at the same time for the customer to choose, it will reduce the renegotiation time. Yet by giving several options, it represents possible ways of meeting as many of both parties' interests as possible [Alfredson and Cungu 2008].

To generate an offer efficiently and optimally during renegotiation, a range of values is set within an upper and a lower bound. An upper-bound level is needed by the service provider to avoid overallocation of resources, whereas a lower-bound level is required by the service consumer to avoid service violation that may affect the business. Such bounds can be obtained from learning the history of the service level, assessing previous agreement during the service lifetime, and assessing the current situation within the cloud environment.

The initial agreed-on service level is important during renegotiation. The generated offer must consider the minimum acceptable level delivered as stated in the previous term. The provider must also be aware of the maximum delivered level that is still profitable and without creating negative consequences to other customers. Choosing the most suitable technique to learn the service level history and the current context will provide an accurate service usage trend and level.

Even though offer generation–based negotiation is able to generate optimum multiple offers, determining the offer to select as the new objective becomes another open issue from the customer side, as the offers generated using this approach will have the same utility value. Customers needs to wisely choose the best one suitable to their current requirements.

SLA renegotiation is also expected to deal with any violation on the agreed-on service level either proactively or reactively. The machine learning technique used not only will provide a precise value as the constraint but also will play an important role in predicting and detecting any SLA violation. For proactive and reactive renegotiation, both the service provider and service consumer can initiate the renegotiation. In the proactive one, when the customer wants to adjust the established SLA due to a change in circumstances, a new SLA parameter proposal may then be sent to the provider. On the provider's side, if any SLA violation is forecast, the provider will immediately start to renegotiate the SLA. In such cases, the notion of SLA violation prediction must be taken into account during SLAM. Meanwhile, reactive renegotiation is executed following any detected violation. Although service termination is normally chosen within the nature of a service-based system, renegotiating the SLA with a decreased level and some penalty charge is arguably a better option.

The strengths of offer generation techniques can be exploited to develop SLA renegotiation. In general, the negotiation preferences will first be evaluated using utility function to get acceptable weighted values. The solutions given by the computation are then sorted using certain criteria to simplify selection of the most suitable one.

The practicality of SLA renegotiation has been shown in the Condition Monitoring on a Cloud (CMAC) Project [Galati et al. 2013]. CMAC is a monitoring platform that enables implementation of the SLA protocol for negotiation and renegotiation. Here, SLA renegotiation is implemented to prevent violations on guaranteed terms and to maximize revenues for the provider. For renegotiation, CMAC proposes a modification offer to the client based on historical data and current situations within the cloud environment to provide optimum service delivery.

6. CONCLUSION

Cloud computing offers the ability to scale self-service provisioning and deprovisioning up or down based on user needs. Nevertheless, this flexibility creates anxiety problems with regard to the promised service level from the cloud provider to the customer. However, service level management can address the anxiety issue and consequently raise the trust level of a cloud customer if handled well.

From the literature, it has been shown that researchers have studied problems associated with all phases of cloud SLAM. Nonetheless, maintaining cloud-based services under changing circumstances remains an outstanding issue. SLA renegotiation can provide a solution in such situations, especially in relation to enhancing trust in a cloud-based system.

As the outsourcing relationship between the provider and customer grows, the changing technology delivery requirements and market forces often lead to continual updating of contracts with amendments. Therefore, SLA renegotiation as an additional mechanism becomes crucial to support such requirements in a cloud-based system. It can provide advantages to different types of users, most importantly to users who have critical application or data to be put in the cloud. In this article, the authors propose yet another perspective of SLA renegotiation, which applies context awareness by learning the system and the user-updated situation to assist in accurate offer generation.

Compared to the bargaining approach, which results in a single offer during the negotiation round, the offer generation approach uses suitable techniques after taking some input from a learning mechanism that can provide multiple offers in a single negotiation round, resulting in a win–win outcome. Here the objective is to create a solution space and provide some best alternatives within one cycle. This approach is expected to generate a different combination of SLO values that is mutually acceptable for both the service provider and customer based on the past, current, and future situation. Thus, the renegotiated SLA will be considerably adaptive to the changing context or situations within a cloud environment.

Generally, the preceding approaches can provide a more flexible way to maintain the trust level among cloud customers and thus are beneficial to customers who wish to have long-term service or lifetime service.

There are many directions to further refine and extend the approach presented in this article. First, adjusting SLOs to a new level in an SLA may cause violations of the other SLA, and thus the consideration of the SLA parameter that is renegotiable between the service provider and customers within the cloud context becomes important. Aspects such as security and reliability can also be considered. Second, the proposed approach mentioned in this article suggests a renegotiation within a certain range of acceptable SLO values. Further investigation is needed to find a technique to determine such a range for each customer when renegotiation is initiated.

REFERENCES

G. Aceto, A. Botta, W. de Donato, and A. Pescapè. 2012. Cloud monitoring: Definitions, issues and future directions. In *Proceedings of the IEEE 1st International Conference on Cloud Networking (CLOUDNET)*. 63–67.

B. Addis, D. Ardagna, B. Panicucci, and L. Zhang. 2010. Autonomic management of cloud service centers with availability guarantees. In *Proceedings of the IEEE 3rd International Conference on Cloud Computing.* IEEE, Los Alamitos, CA, 220–227.

Ahronovitz et al. 2010. *Cloud Computing Use Cases: A White Paper Produced by the Cloud Computing Use Case Discussion Group, Version 4.0*. National Institute of Standards and Technology.

M. Al-Aaidroos, N. Jailani, and M. Mukhtar. 2011. Agent-based negotiation framework for Web service's SLA. In *Proceedings of the 7th International Conference on IT in Asia (CITA)*.

M. Alhamad, T. Dillon, and E. Chang. 2010. Conceptual SLA framework for cloud computing. In *Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technologies*. 606–610.

T. Alfredson and A. Cungu. 2008. Negotiation theory and practice: A review of the literature. In *EASYPol Module 179, Food and Agriculture Organization of the United Nations (FAO'08)*. http://www.fao.org/docs/up/easypol/550/4-5_negotiation_background_paper_179en.pdf, p. 6.

F. Alsrheed, A. El Rhalibi, M. Randles, and M. Merabti. 2013. Rubinstein's alternating offers protocol for automated cloud computing negotiation. In *Proceedings of the 14th Annual PG Symposium on the Convergence of Telecommunications, Networking, and Broadcasting*.

A. Amato, B. Di Martino, and S. Venticique. 2013. Agents based multi-criteria decision-aid. *Journal of Ambient Intelligence and Humanized Computing* 5, 5, 747–758.

E. Badidi. 2013. A cloud service broker for SLA-based SaaS provisioning. In *Proceedings of the 2013 International Conference on Information Society (i-Society)*. 61–66.

D. Battre, F. M. T. Brazier, K. P. Clark, and M. Oey. 2010. A proposal for WS-agreement negotiation. In *Proceedings of the 11th IEEE/ACM International Conference on Grid Computing*. 233–241.

K. Boloor, R. Chirkova, Y. Viniotis, and T. Salo. 2010. Dynamic request allocation and scheduling for context aware applications subject to a percentile response time SLA in a distributed cloud. In *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science.* IEEE, Los Alamitos, CA, 464–472.

N. Bonvin, G. Papaioannou, and K. Aberer. 2011. Autonomic SLA-driven provisioning for cloud applications. In *Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing.* IEEE, Los Alamitos, CA, 434–443.

S. Bouchenak. 2010. Automated control for SLA-aware elastic clouds. In *Proceedings of the 5th International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks.* 27–28.

I. Brandic, D. Music, and S. Dustdar. 2009. Service mediation and negotiation bootstrapping as first achievements towards self-adaptable grid and cloud services. In *Proceedings of the International Conference Industry Session on Grids Meets Autonomic Computing*. 1–8.

R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Journal of Future Generation Computer Systems* 25, 599–616.

F. Chang, J. Ren, and R. Viswanathan. 2010. Optimal resource allocation in clouds. In *Proceedings of the IEEE 3rd International Conference on Cloud Computing.* IEEE, Los Alamitos, CA, 418–425.

A. Cuomo, G. Di Modica, S. Distefano, A. Puliafito, M. Rak, O. Tomarchio, S. Venticique, and U. Villano. 2013. An SLA-based broker for cloud infrastructures. *Journal of Grid Computing* 11, 1, 1–25.

G. Di Modica, O. Tomarchio, and L. Vita. 2009. Dynamic SLAs management in service oriented environments. *Journal of Systems and Software* 82, 759–771.

F. Faniyi and R. Bahsoon. 2011. Engineering proprioception in SLA management for cloud architectures. In *Proceedings of the 9th Working IEEE/IFIP Conference on Software Architecture (WICSA)*. 336–340.

C. Figueroa, N. Figueroa, A. Jofre, A. Sahai, Y. Chen, and S. Iyer. 2008. *A Game Theoretic Framework for SLA Negotiation*. Hewlett-Packard Development Company.

A. Galati, K. Djemame, M. Fletcher, M. Jessop, M. Weeks, S. Hickinbotham, and J. McAvoy. 2013. Designing an SLA protocol with renegotiation to maximize revenues for the CMAC platform. In *Web Information*

Systems Engineering—WISE 2011 and 2012 Workshops. Lecture Notes in Computer Science, Vol. 7652. Springer, 105–117.

N. Ghosh and S. K. Ghosh. 2012. An approach to identify and monitor SLA parameters for storage-as-a-service cloud delivery model. In *Proceedings of the 2012 IEEE Globecom Workshops.* 724–729.

R. L. Gomes and E. Madeira. 2011. An automatic SLA negotiation protocol for a future Internet. In *Proceedings of the IEEE Latin American Conference on Communications (LATINCOM)*. 1–6.

D. Greenwood, G. Vitaglione, L. Keller, and M. Calisti. 2006. Service level agreement management with adaptive coordination. In *Proceedings of the International Conference on Networking and Services (ICNS'06)*.

T. Groleat and H. Pouyllau. 2011. Distributed inter-domain SLA negotiation using reinforcement learning. In *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management.* 33–40.

M. Hedwig, S. Malkowski, and D. Neumann. 2011. Dynamic service level agreement management for efficient operation of elastic information systems. In *Proceedings of the 32nd International Conference on Information Systems.*

T. A. Henzinger, A. V. Singh, V. Singh, T. Wies, and D. Zufferey. 2010. FlexPRICE: Flexible provisioning of resources in a cloud environment. In *Proceedings of the IEEE 3rd International Conference on Cloud Computing.* IEEE, Los Alamitos, CA, 83–90.

K. Hindriks and D. Tykhonov. 2008. Opponent modelling in automated multi-issue negotiation using Bayesian learning. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*.

A. Ismail, J. Yan, and J. Shen. 2010. An offer generation approach to SLA negotiation support in service oriented computing. *Service Oriented Computing and Applications* 4, 4, 277–289.

P. A. Jamkhedkar, C. C. Lamb, and G. L. Heileman. 2011. Usage management in cloud computing. In *Proceedings of the IEEE International Conference on Cloud Computing (CLOUD).* 525–532.

A. I. Jehangiri, E. Yaqub, and R. Yahyapour. 2013. Practical aspects for effective monitoring of SLAs in cloud computing and virtual platforms. In *Proceedings of CLOSER 2013*. 447–454.

L. Jian. 2008. An agent bilateral multi-issue alternate bidding negotiation protocol based on reinforcement learning and its application in e-commerce. In *Proceedings of the International Symposium on Electronic Commerce and Security.* IEEE, Los Alamitos, CA, 217–220.

H. Kaminski and M. Perry. 2008. SLA negotiation system design based on business rules. In *Proceedings of the IEEE International Conference on Services Computing.* IEEE, Los Alamitos, CA, 609–612.

B. R. Kandukuri, V. R. Paturi, and A. Rakshit. 2009. Cloud security issues. In *Proceedings of the International Conference on Services Computing.* IEEE, Los Alamitos, CA, 517–520.

W. Kim. 2013. Cloud computing architecture. *International Journal of Web and Grid Services* 9, 3, 287–303.

C. Kotsokalis, R. Yahyapour, and M. A. R. Gonzalez. 2009. Modeling service level agreements with binary decision diagrams. In *Service-Oriented Computing*. Lecture Notes in Computer Science, Vol. 5900. Springer, 190–204.

R. Y. K. Lau, Y. Li, D. Song, and R. C. W. Kwok. 2008. Knowledge discovery for adaptive negotiation agents in e-marketplaces. *Decision Support Systems* 45, 2, 310–323.

C. Langguth and H. Schuldt. 2010. Extended WS-agreement protocol to support multi-round negotiations and renegotiations. In *Service-Oriented Computing*. Lecture Notes in Computer Science, Vol. 6470. Springer, 563–570.

T. Ledoux and Y. Kouki. 2012. SLA-driven capacity planning for cloud applications. In *Proceedings of the IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom).* 135–140.

W. Mach and E. Schikuta. 2012. A generic negotiation and re-negotiation framework for consumer-provider contracting of Web services. In *Proceedings of the 14th International Conference on Information Integration and Web-Based Applications and Services.* ACM, New York, NY, 348–351.

M. Macias, J. O. Fitó, and J. Guitart. 2010. Rule-based SLA management for revenue maximisation in cloud computing markets. In *Proceedings of the International Conference on Network and Service Management.* 354–357.

F. Mahan, A. Isazadeh, and L. M. Khanli. 2011. Using an active fuzzy ECA rule-based negotiation agent in e-commerce. *International Journal of Electronic Commerce Studies* 2, 2, 127–148.

K. Mahbub and G. Spanoudakis. 2010. Proactive SLA negotiation for service based systems. In *Proceedings of the IEEE 6th World Congress on Services.* IEEE, Los Alamitos, CA, 519–526.

K. Mahbub and G. Spanoudakis. 2011. Proactive SLA negotiation for service based systems: Initial implementation and evaluation experience. In *Proceedings of the IEEE International Conference on Services Computing*. IEEE, Los Alamitos, CA, 16–23.

M. Maurer, I. Brandic, and R. Sakellariou. 2010. Simulating autonomic SLA enactment in clouds using case based reasoning. In *Proceedings of the 3rd European Service Wave Conference: Towards a Service-Based Internet*. 25–36.

N. A. Mehdi, A. Mamat, H. Ibrahim, and S. K. Subramaniam. 2011. On the fly negotiation for urgent service level agreement on intercloud environment. *Journal of Computer Science* 7, 10, 1596–1604.

R. F. Moghaddam, F. F. Moghaddam, V. Asghari, and M. Cheriet. 2012. Cognitive behavior analysis framework for fault prediction in cloud computing. In *Proceedings of the 3rd International Conference on the Network of the Future (NOF)*. 1–8.

X. Niu and S. Wang. 2008. Genetic algorithm for automatic negotiation based on agent. In *Proceedings of the 7th World Congress on Intelligent Control and Automation*. 3834–3838.

M. Parkin, P. Hasselmeyer, B. Koller, and P. Wieder. 2008. An SLA re-negotiation protocol. In *Proceedings of the 2nd Non-Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08)*.

K. V. Patel, A. B. R. Khan, M. M. Sidnale, and D. Babu. 2014. Advance strategies to understand the SLA's (Service Level Agreement) with specific issues on requirement gathering. *International Journal of Advanced Research in Computer Science and Software Engineering* 4, 3, 1199–1207.

P. Patel, A. Ranabahu, and A. Sheth. 2009. *Service Level Agreement in Cloud Computing*. Technical Report. Wright State University.

A. Pichot, P. Wieder, O. Waldrich, and W. Ziegler. 2007. Dynamic SLA-negotiation based on WS-agreement. In *Proceedings of the 4th International Conference on Web Information Systems and Technologies (WEBIST)*.

H. Pouyllau and G. Carofiglio. 2011. Inter-carrier SLA negotiation using Q-learning. *Journal of Telecommunication Systems* 52, 2, 611–622.

J. Rogers, O. Papaemmanouil, and U. Cetintemel. 2010. A generic auto-provisioning framework for cloud databases. In *Proceedings of the IEEE 26th International Conference on Data Engineering Workshops*. 63–68.

L. Romano, D. De Mari, Z. Jerzak, and C. Fetzer. 2011. A novel approach to QoS monitoring in the cloud. In *Proceedings of the 1st International Conference on Data Compression, Communications, and Processing (CCP)*. 45–51.

A. C. Rouse. 2005. The role of benchmarking and service level agreement (SLA) practices in IT outsourcing success. In *Proceedings of the 9th Pacific Asia Conference on Information Systems (PACIS)*. 670–682.

K. Salah and R. Boutaba. 2012. Estimating service response time for elastic cloud applications. In *Proceedings of the IEEE 1st International Conference on Cloud Networking (CLOUDNET)*. 12–16.

M. Seibold, A. Kemper, and D. Jacobs. 2011. Strict SLAs for operational business intelligence. In *Proceedings of the IEEE 4th International Conference on Cloud Computing*. IEEE, Los Alamitos, CA, 25–32.

S. Sharaf and K. Djemame. 2010. Extending WS-agreement to support re-negotiation of dynamic grid SLAs. In *Proceedings of eChallenges, 2010*. 1–8.

S. Sharaf and K. Djemame. 2014. Enabling service-level agreement renegotiation through extending WS-agreement specification. *Service Oriented Computing and Applications Journal*. Available at link.springer.com/article/10.1007%2Fs11761-014-0159-5.

M. Siebenhaar, T. A. B. Nguyen, U. Lampe, D. Schuller, and R. Steinmetz. 2012. Concurrent negotiations in cloud-based systems. In *Economics of Grids, Clouds, Systems, and Services*. Lecture Notes in Computer Science, Vol. 7150. Springer, 17–31.

G. C. Silaghi, D. L. Serban, and C. M. Litan. 2010. A framework for building intelligent SLA negotiation strategies under time constraints. In *Economics of Grids, Clouds, Systems, and Services*. Springer Berlin Heidelberg, 48–61.

G. C. Silaghi, D. L. Serban, and C. M. Litan. 2011. A time-constrained SLA negotiation strategy in competitive computational grids. *Future Generation Computer Systems* 28, 8, 1303–1315.

V. Spoorthy and C. Sreedhar. 2012. Multi-level SLAs with dynamic negotiations for remote sensing data as a service. *International Journal of Scientific and Research Publications* 2, 10, 1–5.

V. Stantchev and C. Schropfer. 2009. Negotiating and enforcing QoS and SLAs in grid and cloud computing. In *Advances in Grid and Pervasive Computing*. Lecture Notes in Computer Science, Vol. 5529, Springer, 25–35.

C. Utomo, A. Idrus, M. Napiah, and M. F. Khamidi. 2009. Agreement options on multi criteria group decision and negotiation. *World Academy of Science, Engineering, and Technology* 3, 2, 359–363.

I. Uusitalo, K. Karppinen, A. Juhola, and R. Savola. 2010. Trust and cloud services—an interview study. In *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science*. 712–720.

S. Venticinque, R. Aversa, B. Di Martino, M. Rak, and D. Petcu. 2011. A cloud agency for SLA negotiation and management. In *Proceedings of Euro-Par 2010 Parallel Processing Workshops*. Lecture Notes in Computer Science, Vol. 6586. Springer, 587–594.

R. Vigne, J. Mangler, E. Schikuta, and S. Rinderle-Ma. 2012. WS-Agreement based service negotiation in a heterogeneous service environment. In *5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA'12)*. 1–8.

J. Wilkes. 2008. *Utility Functions, Prices, and Negotiation*. Hewlett-Packard Development Company.

L. Wu and R. Buyya. 2010. Service Level Agreement (SLA) in Utility Computing Systems, CloudBus, http://www.cloudbus.org/reports/SLA-UtilityComputing2010.pdf.

L. Wu, S. Garg, R. Buyya, C. Chen, and S. Versteeg. 2013. Automated SLA negotiation framework for cloud computing. In *Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid)*. 235–244.

L. Wu, S. Kumar, and R. Buyya. 2011. SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments. In *Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing.* IEEE, Los Alamitos, CA, 195–204.

N. Xiaotai and Z. Xiaoyan. 2009. An auto negotiation algorithm based on agent. In *Proceedings of the ETP International Conference on Future Computer and Communication.* IEEE, Los Alamitos, CA, 146–149.

H. Xu and B. Li. 2013. Dynamic cloud pricing for revenue maximization. *IEEE Transactions on Cloud Computing* 1, 2, 158–171.

J. Yan, R. Kowalczyk, J. Lin, M. B. Chhetri, S. K. Goh, and J. Zhang. 2007. Autonomous service level agreement negotiation for service composition provision. *Future Generation Computer Systems* 23, 6, 748–759.

E. Yaqub, P. Wieder, C. Kotsokalis, V. Mazza, L. Pasquale, J. L. Rueda, S. G. Gomez, and A. E. Chimeno. 2011. A generic platform for conducting SLA negotiations. In Service Level Agreements for Cloud Computing, P. Wieder, J. M. Butler, W. Theilmann, and R. Vahyapour (Eds.). Springer Science + Business Media, 187–206.

E. Yaqub, R. Yahyapour, P. Wieder, and K. Lu. 2012. A protocol development framework for SLA negotiations in cloud and service computing. In *Economics of Grids, Clouds, Systems, and Services.* Lecture Notes in Computer Science, Vol. 7714. Springer, 1–15.

Y. O. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, and Y. Coady. 2010. Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis. In *Proceedings of the IEEE 3rd International Conference on Cloud Computing.* IEEE, Los Alamitos, CA, 91–98.

S. Zaman and D. Grosu. 2013. A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds. *IEEE Transactions on Cloud Computing* 1, 2, 129–141.

L. Zhang, Z. Li, and C. Wu. 2014. Dynamic resource provisioning in cloud computing: A randomized auction approach. In *Proceedings of IEEE INFOCOM*. 433–431.

X. Zheng, P. Martin, W. Powley, and K. Brohman. 2010. Applying bargaining game theory to Web services negotiation. In *Proceedings of the IEEE International Conference on Services Computing.* IEEE, Los Alamitos, CA, 218–225.

Z. Zhu, J. Bi, H. Yuan, and Y. Chen. 2011. SLA based dynamic virtualized resources provisioning for shared cloud data centers. In *Proceedings of the IEEE 4th International Conference on Cloud Computing*. 630–637.

W. Ziegler, P. Wieder, and D. Battre. 2008. *Extending WS-Agreement for Dynamic Negotiation of Service Level Agreements*. Technical Report TR-0172. CoreGRID.

F. Zulkernine and P. Martin. 2011. An adaptive and intelligent SLA negotiation system for Web services. *IEEE Transactions on Services Computing* 4, 1, 31–43.

Received January 2014; revised November 2014; accepted January 2015