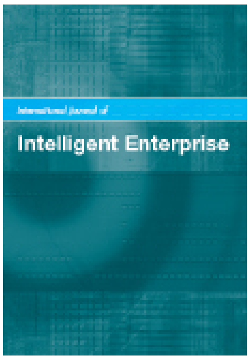


International Journal of Intelligent Enterprise

ISSN online: 1745-3240 - ISSN print: 1745-3232
https://www.inderscience.com/ijie

Ensuring prompt cloud service provider based on service level agreement using fuzzy logics and decision support system

P. Karthikeyan, E. Sathiyamoorthy

DOI: 10.1504/IJIE.2023.10051666

Article History:

Received:	18 July 2019
Last revised:	15 November 2019
Accepted:	17 February 2020
Published online:	30 November 2022

Ensuring prompt cloud service provider based on service level agreement using fuzzy logics and decision support system

P. Karthikeyan* and E. Sathiyamoorthy

School of Information Technology and Engineering,
Vellore Institute of Technology,
Tamilnadu, 632014, India
Email: pkarthikeyan@vit.ac.in
Email: esathiyamoorthy@vit.ac.in
*Corresponding author

Abstract: Cloud service users (CSU) subscribe services from a specific cloud service provider (CSP). Several cloud service providers may offer the same as well as different services. Hence various factors have to be considered while selecting a CSP which offers the same type of services. An intelligent third party (ITP) helps the CSU in choosing a best suited CSP by capturing the requirements from CSU using fuzzy logics. These are matched with service offerings of the CSPs using multi criteria decision making process called the analytic hierarchy process (AHP) and the best suited CSPs are suggested by ITP among the registered CSPs to the CSU. A negotiation process follows where the CSU can make their choice in choosing the suggested CSPs. A partial SLA is generated between the CSU and preferred CSP defining the various services selected attributes provided by the CSP, other requirements of CSU and violation details.

Keywords: cloud service provider; CSP; service level agreement; fuzzy logic; decision support system; analytic hierarchy process; AHP; intelligent third party; ITP.

Reference to this paper should be made as follows: Karthikeyan, P. and Sathiyamoorthy, E. (2023) 'Ensuring prompt cloud service provider based on service level agreement using fuzzy logics and decision support system', *Int. J. Intelligent Enterprise*, Vol. 10, No. 1, pp.1–13.

Biographical notes: P. Karthikeyan is Assistant Professor at SITE, VIT University, His main thrust research areas are big data, network security and cloud computing. He is pioneer researcher in the areas of big data and IoT and has published more than 70 papers in various top international journals.

E. Sathiyamoorthy is an Associate Professor, at the School of Information Technology and Engineering, Working in Vellore Institute of Technology, Vellore, Tamilnadu, India. He is having 19 years of experience in teaching and ten years of research experience as a guide. His areas of interest and research domain are in cloud computing, web services, and data mining.

1 Introduction

A cloud service user (CSU) is the one who is in need of cloud services for their business activities and subscribes for cloud services and uses them. A cloud service provider (CSP) offers various types of services to the CSUs. These cloud services are SAAS based. As cloud computing is taking centre stage for different IT enabled business enterprises it is highly essential to define policies, procedures and service level agreements (SLA) in order to maximise the value of cloud for both the consumer and the service provider. SLA statements written must be measurable, achievable, relevant and timely and should remain specific to cloud services aimed at minimising ambiguities for both the cloud consumer and the CSPs.

The cloud service models (IaaS, PaaS, SaaS, etc.) offer new paradigms of computing resources and IT enabled capabilities for all types of organisations. IT industry experts claim that over 80% of enterprises have adopted some cloud service in their organisation. The key term 'service' in cloud computing creates the need to develop contracts named SLA between the client organisation and the CSP. SLAs are used by companies for a long time, especially when the company hires third party service provider to manage some of their business operations. SLAs will ensure the consumer receives all the services availed as agreed by the provider and of course ensure money's worth for the client.

Likewise, when an organisation decides to hire cloud services for their IT needs, SLAs come into play to make sure the services offered by the CSP are delivered as promised. SLA has become a pre-requisite due to cloud business strategy and provides series of rules and directives that must be taken by cloud consumers to evaluate and negotiate terms with CSP. It describes a set of non-functional requirements of cloud services.

These SAAS-based services follows any of the below given pricing models.

- *Subscription model* – A specific set of services were subscribed by the user for a specific period. The client must pay the entire fee for subscription in order to start and utilise the service for the given period of time.
- *Fee-based model* – In this type of model the client will be charged whenever the CSU accesses the service.
- *Businesses model* – In Business model CSP will collect certain percentage of revenue gained as an outcome of their business process for the usage of the services provided by them. The percentage will be decided through SLA as a predefined agreement.

The above given are the general pricing models. Several other pricing practices can be found based on the business process.

SLA can be defined as a legal agreement between the users who are in need of cloud services and the providers who will be providing the services which the user needs. The formal contract between the cloud user and CSP will be managed by intelligent third party (ITP) who acts as a bridge between the two set of entities.

SLA may be a service-based SLA where the SLA covers a generalised service for all clients. In customer-based SLA an agreement will be formed between CSP and CSU-based on the needs and usage of the services which the customer wants to use. The final type of SLA is multi-level SLA where it's a combination of service based and customer-based model.

In Ibrahim et al. (2016) the service users provide the functional and non-functional requirements for various services. However, some consumers may find it hard to provide the exact requirements that are suitable to their environment. The following method helps the service users in providing value-added requirements.

In Boolean logic, the truth values of the variables may only be 0 or 1. But in fuzzy logic, the truth values may be any real number between 0 and 1. The degrees of these variables (linguistic) are handled by membership functions. Fuzzification and defuzzification are done to the various attributes of interest.

The AHP is designed to solve complex problems which involve multiple criteria. The purpose of the AHP is to facilitate making choices from among a number of different alternatives and criteria are defined by formulating priorities. The output of AHP is a prioritised ranking, indicating the overall preference for each of the decision alternatives. In Garusinghe et al. (2016) AHP is used in decision making for cell phone service providers. Similar concept is used here. Dan et al. (2004) also uses the AHP to select a SaaS product.

2 Literature survey

In this section, survey done on concepts of generating a SLA and the work related to service attributes, service metrics are expressed with cloud applications.

Cloud services requires some fundamental security requirements as follows: access control, data privacy, confidentiality and data integrity. But still some security factors are under research. They are: recovery and prosecution and physical protection with attack/harm detection, non-repudiation, security, security auditing which has been indicated (Xia et al., 2011).

It is important to keep up the agreement steady as promised by the service provider, though it is difficult to promise the service assurance legally. There is no such metrics used in generating the SLA. A framework is proposed where performance metrics are well defined based on the applications (Garcıa et al., 2017). Here the proposed methodology is used to assess the applications performance in different testing environment to assure good services quality as mentioned in SLA. For better utilisation of services, the time correlations between the metrics measures is reviewed. If the service contracts are verified before deployment of SLA, some invalid service attributes which default the agreement on either side (Zulkernine and Martin, 2011) may be avoided. This motivates the designer to encode a contract for model checking directly as event-condition-action rules, in terms of contract entities like business operations, role players with their rights, obligations and prohibitions.

An SLA framework was proposed for large scale resource management in modular networking to treat robustness, resource utilisation and other general aspects by interacting through application programming interfaces (API), between cloud management system (CMS) and SLA subsystem (Dan et al., 2004). Moving forward, the cloud computing technology like as – a – service (XaaS) will create a bird's eye view to increase the spotlight on cloud service engineering and management in related concepts (Keller et al., 2003).

These developments had made strong impression on non-functional service aspects and quality-of-service (QoS) and become more appropriate to include these in SLAs (Firesmith, 2004; de Miguel et al., 2002), whereas QoS promises are typically defined as legally binding SLA. The description of SLAs are developed using languages like WSLA (Wu et al., 2013; de Miguel et al., 2002) or OGFsWS-agreement (Abdelsadiq et al., 2011) where the service level objectives (SLOs), numerical QoS objectives (Firesmith, 2004) are included and are monitored based on a particular time interval. Usually SLAs define monetary penalties like when the provider fails to satisfy the requirement agreed as well as when the user fails to abide by the terms and conditions mentioned (Xia et al., 2011). High performance computing service provider has been introduced to formulate the job submission and to schedule control through long term SLA which was proposed to cover the above said backlogs (Rosanty et al., 2012).

In telecom SLA customer required additional resilience where it is missed on most of the SLAs. But this factor may prove to be highly expensive and sometimes unnecessary. Fortunately, there is a possible approach proposed to require the customers to specify their time-differentiated demand for resilience through the critical windows (CWs) (Keller and Ludwig, 2003) provided in SLA framework. This results in a high-level resource utilisation that can be increased efficiently and protected with high resilience in terms of availability.

SLAs are commitments made to customers by companies, or SLOs, which are agreed upon by departments of a company, have been the starting point for Exchange designs in the past. It is also likely that not all users are equal. You can provide different SLAs for different user groups, such as hosting VIPs or mailboxes used by business-critical applications in smaller databases that can be recovered faster. You have to define the priority in which the applications used by your business will be recovered after a disaster. Although exchange may have high priority, there are likely other business-critical applications for your company. SAP or Oracle databases used by a business-critical application, are two examples.

2.1 SLA requirements

- Security: A consumer must understand his security requirements and what controls them. Federation patterns are necessary to meet those requirements. A provider must understand what they must deliver to the consumer to enable the appropriate controls and federation patterns.
- Data encryption: Data must be encrypted while it is in motion and while it is at rest. The details of the encryption algorithms and access control policies should be specified.
- Privacy: Basic privacy concerns are addressed by requirements such as data encryption, retention, and deletion. An SLA should make it clear how the cloud provider (CP) isolates data and applications in a multi-tenant environment.

It is very important to have a clear understanding about the policies, standards and terms between the service provider and users or clients. A high-level model has been proposed to solve that which empowers a service level management to encode a service contract model checking directly as event-condition-action rules (Zulkernine and Martin, 2011) in terms of contract entities: business operations, role players with their rights, obligations and prohibitions.

Negotiations are one of the challenging tasks in SLM as it involves so many parameters to be considered. Though group decision support system (DSS) is suitable for multi criteria decision making system, so many authors had implemented fuzzy AHP which lacks in voting for decision makers. To compensate this issue social choice (Federal Office for Information Security, www.bsi.bund.de) is added to this GDSS with fuzzy analytical hierarchy process (AHP) which involves the decision makers to directly involve in decision making system.

3 Existing system

An SLA fundamentally represents a documented agreement between a service provider and a service consumer for the provision of one or more services that have specific characteristics and target levels of quantity, quality, and performance. The tricky parts are: How are the services defined? How are service levels specified? How does the SLA relate to other parts of a customer relationship agreement? To date, SLA development has been largely provider-driven, with SLAs being designed to suit each provider needed and circumstances and service capabilities. To be fair, since many services may be provided to millions of users, it would be unreasonable to expect custom-designed SLAs for each customer.

Some examples of cloud SLAs re: Microsoft Azure, Amazon AWS EC2, Google Compute Engine, HP and Rackspace. As is obvious, there is no standard format or presentation for cloud SLAs (yet).

Current cloud SLAs do not cover security requirements. Some consortiums have proposed standards for the evaluation of security offered by the CPs. Cloud brokers (CB) can then generate security level agreement (SecLA) contracts between customers and providers to fit users' requirements. However, the SecLAs do not provide enough details for complex customers' situations, such as sharing resources with other users/companies, or set up specific access controls and security properties (ACSP) which are required for access control management.

In the existing system, this issue is tackled by introducing a general requirement specification language (ACSP-RSL) to allow the customers to express their needs in terms of ACSP. The client will generate the ACSP requirements based on RSL which are sent to the CB. AC requirements in RSL are expressed through the following request: Req (roles, objects, actions, cloudLevels, targetRoles, targetActions, aontexts, permissions). As AC rules only describe direct access, SP rules are used which also describe indirect accesses. Properties in SP are as follows: confidentiality, integrity of information, race conditions, privilege separation, domain integrity, trusted path execution. In addition to ACSP requirements, other general security requirements are also mentioned in requirements specifications. Each CP will list their security offerings to the CB. The CB will perform the negotiation process and link the client to a suitable CP. After collecting the client requirements, the CB matches the client requirements to each CSP security offerings and finds the best match. This is called negotiation phase.

4 Proposed system

The proposed architecture illustrated in the Figure 1 Targets the subscription fees-based SAAS billing model with a customer-based SLA. The CSU approaches the ITP to select the services they want to subscribe and specify their requirements. The ITP captures the requirements and processes them using fuzzy logics. Thus, the requirements are précised. The ITP contains the data of various CSPs' service offerings along with the ratings on particular aspects such as scope, quality and responsibilities. Based on these data of CSPs and the precise requirements of CSU, the analytic hierarchy process (AHP) is carried out which is a DSS that decides which CSP is best suited to the CSU. If the CSU is not satisfied with the result, the next best CSP is suggested. Once, the CSU is satisfied, an SLA is generated which contains the service details and the requirements of the service user. This SLA is incomplete and is forwarded to the CSP. The CSU approaches the CSP for subscribing the services. Several plans exist for each service. The customisation of plans is provided too. When a plan is confirmed, a complete SLA is generated between the CSU and the CSP.

Figure 1 Basic Structure of SLA (see online version for colours)

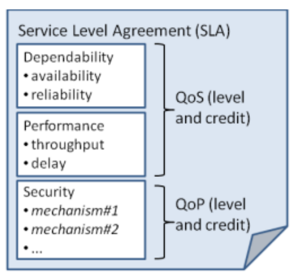
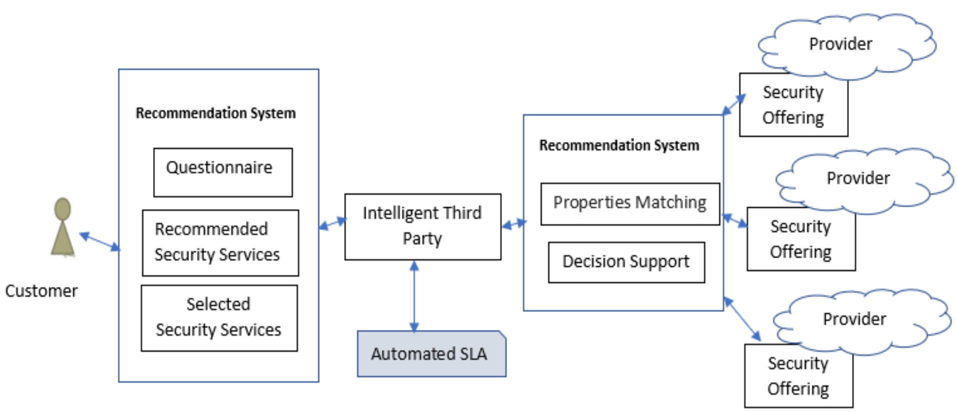


Figure 2 Proposed architecture (see online version for colours)



5 Result and analysis

Consider a scenario where user-X (CSU) approaches the ITP for selecting the services and subscribing them under a CSP. Registered CSPs with the ITP are SP-A, SP-B, SP-C. User-X selects a service, and answers the questionnaire given by ITP. Requirements are captured from the questionnaire and are processed via fuzzy logics. The results obtained are taken as inputs in the AHP. The output is the best CSP suitable to the user-X for the service they opted.

5.1 Selecting services

Several SAAS services offered by various service providers are listed out with their name, and description of the services. The services can be filtered out with the service types or by using keywords. Example of services is quick presentations and creating UML. Screenshots are also provided for every service along with user reviews. User-X selects the services they require from the list of services displayed by the ITP. In our scenario, let the service selected by User-X be quick presentations.

5.2 Collecting requirements using fuzzy logic

A questionnaire will be collected from customers using fuzzy logic. For example, the attribute customisability has been given below.

Customisability

Q1 How much satisfied are you with the look and feel of the service from its screenshots? Range: 0–100%

Q2 How much level of UI (fonts, icons) of the service do you prefer to change? Range: 0–100%

Number of queries for an attribute is the number of linguistic variables. X (satisfaction of look and feel), Y (change in UI) are the two linguistic variables whose values may be 35%, 60% respectively. The sets defined for these variables X, Y are high, medium, low and small, large respectively.

$$\mu\text{satisfaction (X)} = \text{low (35)} = 0.5$$

$$\mu\text{satisfaction (X)} = \text{medium (35)} = 0.2$$

$$\mu\text{satisfaction (X)} = \text{high (35)} = 0.0$$

$$\mu\text{change (Y)} = \text{small (60)} = 0.1$$

$$\mu\text{change (Y)} = \text{large (60)} = 0.$$

The membership functions are generated from trapezoid fuzzy sets. A trapezoid fuzzy set contains 4 boundary values, left, left middle, right middle, right.

If value <= left, µvalue = 0.
If left <= value < left_middle, µvalue = (value – left) / (left_middle – left).
If left_middle <= value < right_middle, µvalue = 1.
If right_middle <= value < right, µvalue = (right – value) / (right – right_middle).

If right <= value, μvalue = 0.

Figure 3 Visual representation of satisfaction of look and feel (see online version for colours)

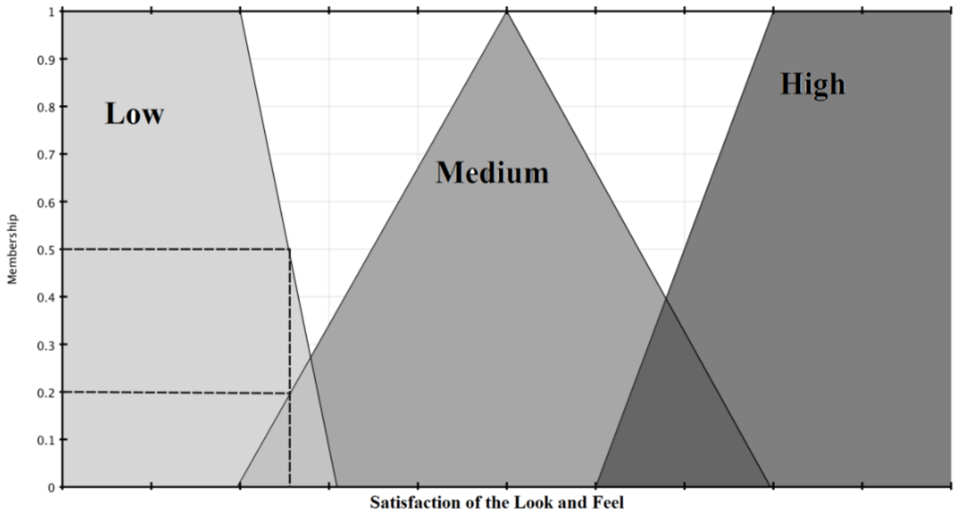
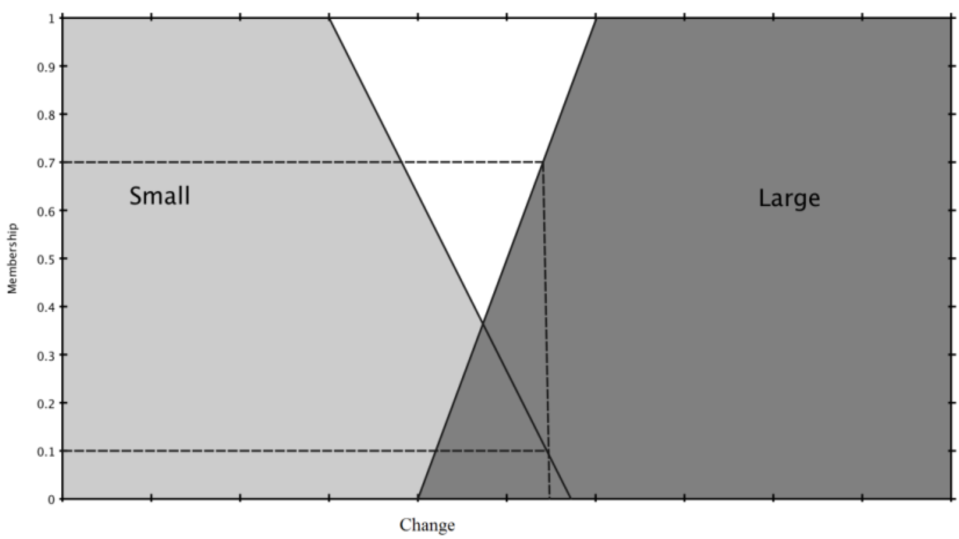


Figure 4 Visual representation of change



The fuzzy rules

If X is low and Y is large, customisability is extremely preferred.

If X is medium, customisability is strongly preferred.

If X is low and Y is small, customisability is moderately preferred.

If X is high and Y is large, customisability is lightly preferred.

If X is high and Y is small, customisability is not preferred.

$$\mu\text{customizability} = \text{extreme} = \min\left[\mu x(a), \mu y(a)\right] = \min[0.5, 0.7] = 0.7$$

$$\mu\text{customizability} = \text{strong} = \mu x(a) = 0.2$$

$$\mu\text{customizability} = \text{moderate} = \min\left[\mu x(a), \mu y(a)\right] = \min[0.5, 0.1] = 0.1$$

$$\mu\text{customizability} = \text{light} = \min\left[\mu x(a), \mu y(a)\right] = \min[0.0, 0.7] = 0.0$$

$$\mu\text{customizability} = \text{no} = \min\left[\mu x(a), \mu y(a)\right] = \min[0.0, 0.1] = 0.0$$

From the above, customisability is extremely preferred and given a ranking of 5. Thus, the rankings which denote the preference level are assigned to each of the attributes of every service selected by user-X.

The performance of this fuzzy logic technique based on membership functions is evaluated and concluded as a reliable technique. Several methods are available to perform defuzzification. Centroid method is the most popular one.

$$\text{Result} = \frac{(30 + 40 * 0.1) + ((50 + 60) * 0.2) + ((70 + 80 + 90 + 100) * 0.7)}{(0.1 * 2) + (0.2 * 2) + (0.7 * 4)}$$

Customisability preference value = 78.5%

Similar process is done for the rest of the attributes. The following are preference values obtained from the above attributes.

Figure 5 SLA workflow (see online version for colours)

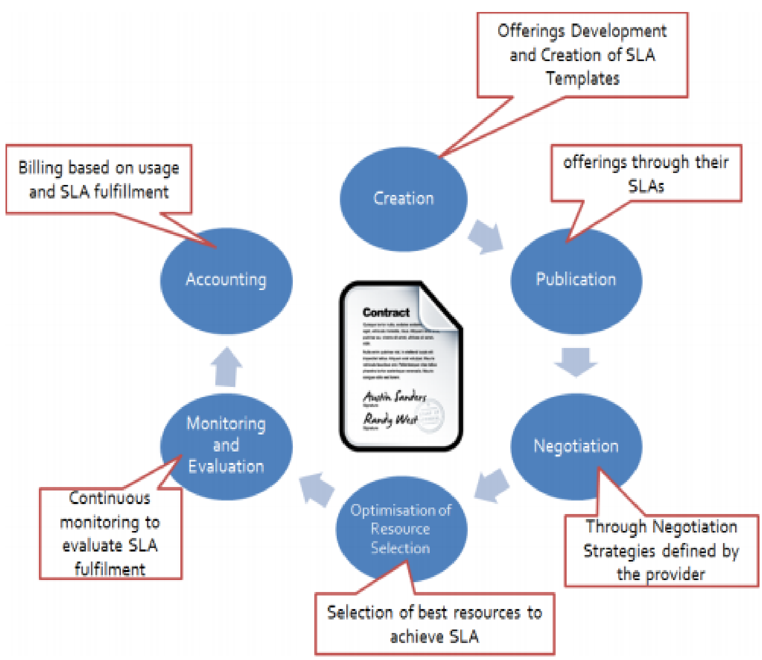


Table 1 Preference table for attributes

Attribute	*Preference value*	*Preference range*	*Preference ranking*
Customisability	78.5%	Extremely preferred	5
Performance	60%	Strongly willing to perform	4
Efficiency	99%	Willing to achieve extreme efficiency	5
Suitability	75%	Strongly preferred	4
Low cost	65%	Strongly preferred	4
Availability	99%	Extremely preferred	5
Reliability	70%	Strongly preferred	4
Service response time	88%	Extremely preferred	5
Scalability	54%	Moderately preferred	3
Efficiency (of provider)	70%	Strongly preferred	4
Performance	90%	Extremely preferred	5
Data integrity	99%	Extremely preferred	5
Data privacy	10%	Not preferred	1
Maintainability	30%	Lightly preferred	2

5.3 Service offering by various CSP's

Service offerings by various CSP's are listed out. For example, offerings of CSA – A.

Table 2 Preference table for CSP-A

Attribute	*Offered value*	*Preference range*	*Preference ranking*
Customisability	10%	Not offered	1
Performance (of users)	70% (average performance of other users)	Strongly performed	4
Efficiency (of users)	50% (average efficiency of other users)	Moderate efficiency	3
Suitability	30% (average user feedback rating)	Lightly suitable	2
Low cost	85%	Extremely offered	5
Availability	90%	Extremely offered	5
Reliability	60%	Strongly offered	4
Service response time	54%	Moderately offered	3
Scalability	35%	Lightly offered	2
Efficiency	95%	Extremely offered	5
Performance	52%	Moderately offered	3
Data integrity	50%	Moderately offered	3
Data privacy	30%	Lightly offered	2
Maintainability	80%	Extremely offered	5

Similarly, the offerings of all the service providers CSA-B and CSA-3 are calculated.

5.4 Analytic hierarchy process

Analytic table was created based on the preference table with various attributes.

Table 3 Analytic table based on preferences (from A1 to A8)

	A1	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*
A1	1	5/4	1	5/4	5/4	1	5/4	1
A2	4/5	1	4/5	1	1	4/5	1	4/5
A3	1	5/4	1	5/4	5/4	1	5/4	1
A4	4/5	1	4/5	1	1	4/5	1	4/5
A5	4/5	1	4/5	1	1	4/5	1	4/5
A6	1	5/4	1	5/4	5/4	1	5/4	1
A7	4/5	1	4/5	1	1	4/5	1	4/5
A8	1	5/4	1	5/4	5/4	1	5/4	1
A9	3/5	3/4	3/5	3/4	¾	3/5	3/4	3/5
A10	4/5	1	4/5	1	1	4/5	1	4/5
A11	1	5/4	1	5/4	5/4	1	5/4	1
A12	1	5/4	1	5/4	5/4	1	5/4	1
A13	1/5	1/4	1/5	1/4	¼	1/5	1/4	1/5
A14	2/5	2/4	2/5	2/4	2/4	2/5	2/4	2/5
VS	11.2	14	11.2	14	14	11.2	14	11.2

Table 4 Analytic table based on preferences (from A9 to A14)

	A9	*A10*	*A11*	*A12*	*A13*	*A14*	*RP*
A1	5/3	5/4	1	1	5	5/2	0.089
A2	4/3	1	4/5	4/5	4	2	0.07
A3	5/3	5/4	1	1	5	5/2	0.089
A4	4/3	1	4/5	4/5	4	2	0.07
A5	4/3	1	4/5	4/5	4	2	0.07
A6	5/3	5/4	1	1	5	5/2	0.089
A7	4/3	1	4/5	4/5	4	2	0.07
A8	5/3	5/4	1	1	5	5/2	0.089
A9	1	3/4	3/5	3/5	3	3/2	0.053
A10	4/3	1	4/5	4/5	4	2	0.07
A11	5/3	5/4	1	1	5	5/2	0.089
A12	5/3	5/4	1	1	5	5/2	0.089
A13	1/3	1/4	1/5	1/5	1	1/2	0.018
A14	2/3	2/4	2/5	2/5	2	1	0.035
VS	18.7	14	11.2	11.2	56	28	1

5.5 Calculate relative priorities of Service providers with respect to each attribute

Table 5 Relative priority table

Performance (of user)	*SP-A*	*SP-B*	*SP-C*	*Relative priority*
SP-A	1	1	4/3	0.364
SP-B	1	1	4/3	0.364
SP-C	3/4	3/4	1	0.272
Vertical Sum	2.75	2.75	3.667	1
Customisability	*SP-A*	*SP-B*	*SP-C*	*Relative priority*
SP-A	1	1/5	1/3	0.111
SP-B	5	1	5/3	0.556
SP-C	3	3/5	1	0.333
Vertical sum	9	1.8	3	1

Similarly, the relative priority will be calculated for all the attributes.

5.6 Calculating weight values of each service provider

$$\text{Weight value of a service provider} = \text{P} = \sum_{i=1}^{n} \text{rci} * \text{rai}$$

where

- rci, represents the relative priority of an attribute.
- rai, represents the relative priority of the service provider with respect to that attribute.

$$\begin{aligned}\text{Weight value of SP} - \text{A} = {} & (0.089*0.111) + (0.07*0.364) + (0.089*0.25) \\ & + (0.07*0.1818) + (0.07*0.385) + (0.089*0.333) \\ & + (0.07*0.307) + (0.089*0.23) + (0.053*0.2) \\ & + (0.07*0.357) + (0.089*0.23) + (0.089*0.23) \\ & + (0.018*0.166) + (0.035*0.333) = 0.259\end{aligned}$$

$$\begin{aligned}\text{Weight value of SP} - \text{B} = {} & (0.089*0.556) + (0.07*0.364) + (0.089*0.416) \\ & + (0.07*0.454545) + (0.07*0.229) + (0.089*0.333) \\ & + (0.07*0.307) + (0.089*0.384) + (0.053*0.4) \\ & + (0.07*0.285) + (0.089*0.384) + (0.089*0.384) \\ & + (0.018*0.416) + (0.035*0.333) = 0.3738\end{aligned}$$

$$\begin{aligned}\text{Weight value of SP} - \text{C} = {} & (0.089*0.333) + (0.07*0.272) + (0.089*0.333) \\ & + (0.07*0.363636) + (0.07*0.385) + (0.089*0.333) \\ & + (0.07*0.384) + (0.089*0.384) + (0.053*0.4) \\ & + +(0.07*0.357) + (0.089*0.384) + (0.089*0.384) \\ & + (0.018*0.416) + (0.035*0.333) = 0.355\end{aligned}$$

So, SP-B has the highest weight value. Thus, SP-B is the best suitable CSP for user-X.

In this work we have taken an existing system where cloud customers can specify their security requirements manually. In our proposed system, we have shown a way to use these requirements and negotiate effectively. The negotiation process is carried by an ITP which contains a store of CSPs security offerings. Security requirements of cloud users are matched with CSPs' security offerings. Among the matched CSPs to find the best suitable CSP, an AHP model of DSS is used. Finally, an SLA is generated based on customer interest. Monitoring of SLA phase is out of the scope of this research work and is most likely an extension work.

References

Abdelsadiq, A., Molina-Jimenez, C. and Shrivastava, S. (2011) 'A high–level model–checking tool for verifying service agreements', *Proceedings of The 6th IEEE International Symposium on Service Oriented System Engineering* (*SOSE 2011*).

Dan, A., Davis, D., Kearney, R., Keller, A., King, R.P., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M. and Youssef, A. (2004) 'Web services on demand: WSLA-driven automated management', *IBM Systems Journal*, Vol. 43, pp.136–158, doi:10.1147/sj.431.0136.

de Miguel, M.A., Ruiz, J.F. and Garcia, A. (2002) 'QoS-aware component frameworks', *2002 IEEE*.

Federal Office for Information Security, *Security Recommendations for Cloud Computing Providers*, White Paper, Article Number: BSI-Bro11/314e [online] www.bsi.bund.de.

Firesmith, D. (2004) 'Specifying reusable security requirements', *Journal of Object Technology*, Vol. 3, No. 1, pp.61–75.

Garcıa, J.M., Fernandez, P., Pedrinaci, C., Resinas, M., Cardoso, J. and Ruiz-Cortes, A. (2017) 'Modeling service level agreements with linked USDL agreement', *IEEE Transaction on Services Computing*, January/February, Vol. 10, No. 1.

Garusinghe, A., Perera, I. and Meedeniya, D. (2016) 'Managing service level agreements in service oriented product lines', *2016 International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp.274–280.

IBRAHIM, A.A.Z.A., Kliazovich, D. and Bouvry, P. (2016) 'Service level agreement assurance between cloud services providers and cloud customers', *16th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*.

Keller, A. and Ludwig, H. (2003) 'The WSLA framework: specifying and monitoring service level agreements for web services', *Journal on Network and Systems Management*, Vol. 11, No. 1, ,pp.57–81.

Rosanty, E.S., Dahlan, H.M. and Hussin, A.R.C (2012) 'Multi-criteria decision making for group decision support system', *2012 IEEE*.

Wu, L., Garg, S.K., Buyya, R., Chen, C. and Versteeg, S. (2013) 'Automated SLA negotiation framework for cloud computing', *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*.

Xia, M., Tornatore, M., Sevilla, S., Shi, L., Martel, C.U. and Mukherjee, B. (2011) 'A novel SLA framework for time-differentiated resilience in optical mesh networks', *J. Opt. Commun. Netw.*, Optical Society Of America, April, Vol. 3, No. 4.

Zulkernine, F.H. and Martin, P. (2011) 'An adaptive and intelligent SLA negotiation system for web services', *IEEE Transactions on Services Computing*, Vol. 4, No. 1, January–March 2011.