Scientific Research Publishing

Open Journal of Applied Sciences, 2022, 12, 1898-1914
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Shared Resource Quality Monitoring and Dynamic Trust Management in a Community Cloud

Rodrigue N'goran[1*], Linda N. Vallee[1], Grâce Y. E. Johnson[1], Jean-Louis Tetchueng[2], Yvon Kermarrec[3], Olivier Asseu[1]

[1]Lastic, ESATIC, Abidjan, Ivory Coast
[2]University of Rennes 1, Rennes, France
[3]Lab-STICC, IMT-Atlantique, Brest, France
Email: *rodrigue.ngoran@esatic.edu.ci

How to cite this paper: N'goran, R., Vallee, L.N., Johnson, G.Y. E., Tetchueng, J.-L., Kermarrec, Y. and Asseu, O. (2022) Shared Resource Quality Monitoring and Dynamic Trust Management in a Community Cloud. *Open Journal of Applied Sciences*, **12**, 1898-1914.
https://doi.org/10.4236/ojapps.2022.1211131

Received: October 21, 2022
Accepted: November 20, 2022
Published: November 23, 2022

Abstract

The collaboration tools offered by Cloud Computing have increased the need to share data and services within companies or between autonomous organizations. This has led to the deployment of community cloud infrastructures. However, several challenges will arise from this grouping of heterogeneous organizations. One of the main challenges is the management of trust between the actors of the community. Trust issues arise from the uncertainty about the quality of the resources and entities involved. The quality of a resource can be examined from a security or functional perspective. Therefore, ensuring security and monitoring the quality of resources is to ensure a high level of trust. Therefore, we propose in this paper a technique for dynamic trust management and quality monitoring of resources shared between organizations. Our approach consists, on the one hand, in evaluating the quality of resources based on quality of service measurement attributes and, on the other hand, in updating the trust values according to the information deduced from these measurements. The proposed framework is evaluated in terms of resource sharing success rate and execution time. Experimental results and comparison with TNA-SL and InterTrust models show that the framework can identify and track the behavior of malicious organizations with relatively low execution time.

Keywords

SMI (Service Measure Index), Trust Management, SLA, QoS, Community Cloud

1. Introduction

According to the American consulting and research firm Gartner, 85% of companies will use Cloud Computing services by 2025 with 95% of their workload stored in the cloud compared to 30% in 2021 [1]. Cloud Computing promotes collaboration and sharing within and between companies.

Several organizations with common needs and interests come together around community cloud infrastructure to minimize investment costs and promote resource sharing [2]. Organizations offer services or make their excess or unused resources available to the community. However, several challenges will emerge from this gathering of different organizations. Among these challenges, trust between entities is one of the major obstacles to sharing and collaboration.

These trust issues in this type of environment can be seen from several aspects including the quality of shared resources. The Cloud Services Measurement Initiatives Consortium (CSMIC) has defined a set of qualitative and quantitative attributes for measuring the quality of Cloud Services (SMI) [3]. The SMI framework is used to characterize a resource based on several criteria including ease of use, affordability, security and privacy.

In environments such as the Community Cloud, the ability of organizations to guarantee and deliver quality resources over the long term conditions the sustainability of the infrastructure and its productivity. It is therefore essential to have mechanisms to track quality throughout the resource usage cycle. In this paper, we propose a framework for resource quality monitoring and dynamic trust management in a community cloud. We address the quality assessment of shared resources based on four (4) SMI attributes: response time, availability, threat and vulnerability management, and billing compliance. The main contributions of this paper are:

-A framework for identifying, defining and monitoring resource quality;

-A dynamic trust management mechanism derived from the results of resource sharing and usage in accordance with the established SLA.

The remainder of this article is organized as follows. Section 2 reviews related work. Section 3 describes the proposed framework. Section 4 presents the experiments and associated results. Finally, Section 5 concludes the paper and proposes perspectives for future work.

2. Related Work

Managing resource quality and trust within distributed architectures remains a major concern for enterprises. Several works presenting techniques for the evaluation and comparison of services and their owners have been carried out.

K. Papadakis *et al.*, proposed in [4], Reputation-based Trust Management (RTM), a collaborative SLA and trust management platform for service providers in a cloud federation. The system allows to evaluate services based on service level agreements (SLAs) and key performance indicators. It is combined with a reputation-based trust management system to help select future providers.

In [5], the authors proposed a Cloud service recommendation system using a clustering-based trust degree computation algorithm. This algorithm relies on quality of service (QoS) parameters, and offers a time saving in the calculation of the degree of trust.

A secure resource allocation system (MSMC) among multiple organizations within a community cloud is proposed in [6]. This system consists of algorithms for resource allocation and workflow execution. The model offers time and cost saving benefits.

S. Garg *et al.* presented SMICloud in [7], a service quality assessment model based on the CSMIC consortium's Service Measurement Indices (SMI). SMIcloud is based on attributes such as response time, availability, reliability, accuracy, transparency and security. This mechanism based on the Analytical Hierarchy Process (AHP) allows services to be ranked based on QoS requirements.

In [8], the authors proposed a resource allocation algorithm considering SLA requirements in a community cloud. They introduce the concept of social pricing to optimize profits and better manage failures.

The literature studied shows that most of the work highlights the impact of service level agreements in the choice of resources or service providers. Furthermore, these works were carried out in federated or public cloud deployment environments that do not take into account certain requirements and governance modes of a community cloud. In addition, these proposals do not highlight the monitoring of trust and quality during the entire usage cycle of a resource. Therefore, it is necessary to propose a framework for trust monitoring and quality assessment of resources based on measurement attributes that are consistent with the social and sustainability characteristics of a community cloud environment.

3. Method, Model, and Material

3.1. General Idea of the Work

The Community Cloud aims to enable organizations to share resources in order to reduce capital costs, create business opportunities without disregarding the quality of shared resources. We propose a framework for resource quality monitoring and dynamic trust management between organizations in the community. Resource quality monitoring is done through performance indicators offered by the CSMIC SMI attributes. A community cloud is characterized by organizations with specific needs or common interests. Thus, the active participation of members in the life of the community and the existence of lasting relationships are assets for the infrastructure. Furthermore, the social character of the collaborations between organizations is one of the fundamental reasons for a community cloud. Based on these facts, our framework is based on four (4) SMI attributes: availability, vulnerability level, response time and billing mode. The attributes of availability, vulnerability, and response time measure and ensure active participation of members and security of resources. In addition, the attribute of the billing mode of the resources brings into play the social aspect in this case the gratuity in the exchanges.

Figure 1 below shows the architecture of our framework. This architecture is subdivided into three layers:

- **Expression of needs and definition of SLAs**: This layer houses the directory of resources and associated suppliers. In addition, it manages the service level agreements between the resource requester and the provider. The agreements are established on the basis of the four (4) SMI attributes mentioned above. In addition, it hosts the trust value manager and the inventory management module responsible for updating the availability status of resources.
- **SLA monitoring**: This consists of monitoring the state of the resources throughout the period of use in order to verify the conformity of the SLAs with the initial agreements established. This operation makes it possible to collect and consolidate the qualitative and quantitative contractual parameters related to the shared resource. This ensures that the commitments of each entity are respected.
- **Update Manager**: The role of the update manager is to update the trust information based on the information provided after the SLA monitoring. Each time the contractual parameters are violated during the period of use of the resource, the trust value update mechanism is triggered. In addition, SLA monitoring provides updated data for resource inventory management.

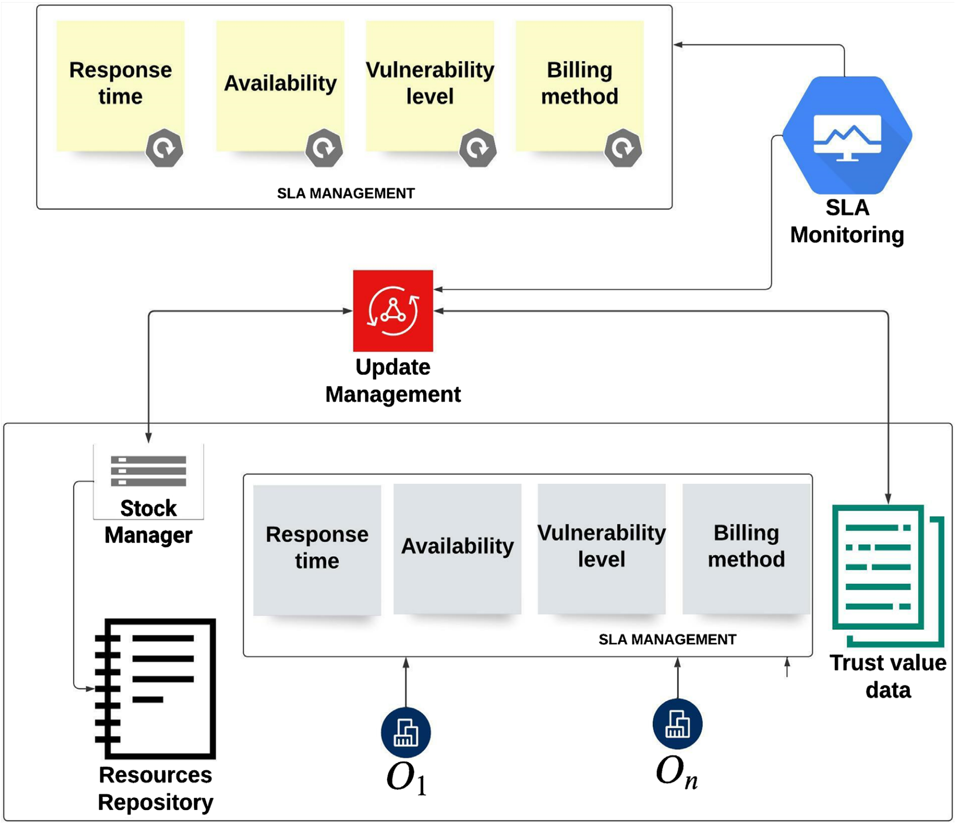

Figure 1. Overall architecture of the resource quality and trust monitoring system.

3.2. Description of Resource Quality Measurement Attributes

The CSMIC recommends a clear and simple definition for each attribute. Thus it proposes definitions centered around the following fields: measure name, related attribute, context, purpose, measure audience, measure definition, and data collection [3]. **Tables 1-4** describe in detail the billing mode, response time, availability, and vulnerability management attributes of the proposed model, respectively.

Table 1. Billing method attribute.

Name of the measure	**Billing method**
Related attribute	Billing process
Context	When negotiating for resource sharing, it is important to It is necessary to specify the billing method (free of charge, sharing in exchange for a resource, exchange by means traditional fiduciary, exchange by virtual currency). The billing mode will have an impact on the willingness to share or not, on the quantity shared, on the time given to the use of the resource and on the extension capacities of the resource.
Measure audience	Organizations (supplier and applicant)
objective	It is a mutual consent measure of both parties. involved and which will allow or not the feasibility of a transaction. The commercial objective is the selection of a furnace provider that meets the financial needs of an applicant and vice versa.
Definition of the measure	Assign a value to each billing method. The values range from 1 to 4: 1) for the fiduciary mode-. 2) for a virtual currency, 3) for an electronic currency exchange for a resource and 4) free. Resources may be acquired free of charge depending on the reputation of the of the applicant or in exchange for a resource in order to encourage sharing within the community.
Data collection	The billing method is collected from each actor.

Table 2. Response time attribute.

Name of the measure	**Response time**
Related attribute	Service Response Time
Context	A date on which the requester would like to have the response is associated with each request for a resource. The response time represents the time taken by the provider to respond to this request.
Measure audience	Organizations (supplier and applicant)
objective	Ensure the efficiency and timeliness of the implementation of the provision of a resource by a supplier.
Definition of the measure	The evaluation criteria are as follows: the date desired by the applicant t_{ui}, the response date promised by the t_{pj} and the actual delivery date t_d. The time of response will be evaluated by three different scores: • The score (1) for an exchange made on the scheduled date; • The score (0.5) for an exchange that is scheduled before the desired date but is provided at a date greater than the scheduled period and less than the desired date; • The score (−1) for a delivery beyond the promised date and with a desired date less than the promised date.
Data collection	The dates (desired and promised) are defined respectively by the applicant and the supplier. The delivery date is deducted at the time of delivery of the resource.

Table 3. Availability attribute.

Name of the measure	**Availability**
Related attribute	Availability
Context	A resource is provided for a defined period. It must have the capacity to maintain an agreed level of availability among the actors during the period of use of the resource
Measure audience	Organizations (supplier and applicant)
objective	Maintain an efficient level of resource availability provided within the community.
Definition of the measure	It is expressed as a proportion of time during which resources are available in relation to time during which they should be. $\text{Availablity}(r)=\frac{T_w}{T_f} \quad (1)$ with T_w the service availability time and T_f the time total use of the service. This value will be compared to the promised availability rates by the supplier to assess the quality of availability of the resource
Data collection	The availability rate is obtained thanks to the returns of information from both actors

Table 4. Vulnerability attribute.

Name of the measure	**Threat and vulnerability management**
Related attribute	Proactive threat and vulnerability management
Context	The level of vulnerability of shared resources has an impact on the security of the infrastructure. To protect against threats and attacks, it is important to reduce the level of vulnerability of shared resources
Measure audience	Organizations (supplier and applicant)
objective	The goal is to ensure that shared resources meet the vulnerability levels of the CVSS [9].
Definition of the measure	Assign sensitivity levels to each shared resource. These levels are defined according to the CVSS v2.0 vulnerability level standard
Data collection	The level of sensitivity is defined by each supplier of resources and validated by the applicant.

3.3. Updating the Confidence Values and the Stock of Resources

3.3.1. Inventory Management

The inventory management module updates the availability status of resources. The resource directory is updated when a new member joins, a new service is added, an organization leaves or a resource is no longer in use. An available resource is in an I and B status where applicable.

As in [8], we define the function $\gamma_r(t)$ as the indicator function of the availability state of a resource at time t. Thus:

$$\gamma_r(t)=\begin{cases}1 & \text{if the resource is availlable then in a state I}\\ 0 & \text{else then in a state B}\end{cases} \tag{2}$$

The total quantity of a resource of vulnerability level j available at time t is expressed as follows:

$$q_{aj}(t)=\sum_{n=1}^{Q_{rpj}} r_{sj}(t) \ \text{ with } \ r_{sj}(t)=1 \ \text{if} \ \gamma_{rn}(t)=1 \ \text{else} \ r_{sj}(t)=0 \tag{3}$$

With Q_{rpj} the total quantity of a resource of a vulnerability level *j*, and $r_{sj}(t)$ the availability state of the resource at time *t*. At the beginning of each new exchange, an update operation of the resource availability state is performed.

3.3.2. Updating of Trust and Reputation Values

The selection of a resource provider for an exchange is based on the degree of trust the requester places in the provider. This trust value is calculated based on previous interactions between the organizations and the reputation of the provider. This value is expressed as follows:

$$g_t = \mu rd_t + (1-\mu) rep \tag{4}$$

With *rep* the reputation of the supplier (the initial reputation of an organization $rep_{ini} = 0$), rd_t the trust value resulting from previous interactions between the involved entities. This trust value is computed using the subjective logic (SL) presented in [10] [11]. The SL allows to express the trust between a resource requesting organization $O_{r\ et}$ and a supplier O_p from the following parameters: belief (*b*), distrust (*d*), disbelief (*u*) and base rate *a*. These parameters are formulated as follows [10] [11]:

$$\begin{cases} b = \dfrac{s_s}{s_s + s_f + 2} \\ d = \dfrac{s_f}{s_s + s_f + 2} \\ u = \dfrac{2}{s_s + s_s + 2} \end{cases} \Leftrightarrow \begin{cases} s_s = \dfrac{2b}{u} \\ s_f = \dfrac{2d}{u} \end{cases} \tag{5}$$

s_s the number of successful prior shares between O_r and O_p, and s_f the number of negative exchanges. The base rate α is defined by:

$$\alpha = 0.5 \tag{6}$$

The confidence value between O_r and O_p is expressed as:

$$rd_t = b + (\alpha * u) \tag{7}$$

For managing exchanges with an intermediate organization O_i between O_r and O_p, The SL proposes an update operator (⊗) to determine the transitive trust between organizations O_i and O_p [10] [11]. This trust is expressed as below:

$$\omega_{O_p}^{O_r:O_i} = \omega_{O_i}^{O_r} \otimes \omega_{O_p}^{O_i} \begin{cases} b_{O_p}^{O_r:O_i} = b_{O_i}^{O_r} b_{O_p}^{O_i} \\ d_{O_p}^{O_r:O_i} = b_{O_i}^{O_r} d_{O_p}^{O_i} \\ u_{O_p}^{O_r:O_i} = d_{O_i}^{O_r} + u_{O_i}^{O_r} + b_{O_i}^{O_r} u_{O_p}^{O_i} \\ \alpha_{O_p}^{O_r:O_i} = \alpha_{O_p}^{O_i} \end{cases} \tag{8}$$

Moreover, in the case of an exchange involving several two intermediaries O_{i1} and O_{i2} the trust between O_r and O_p, is deduced thanks to the consensus operator (⊕) and formulated as follows [10] [11]:

$$\omega_{O_p}^{O_{i1}:O_{i2}} = \omega_{O_p}^{O_{i1}} \oplus \omega_{O_p}^{O_{i2}} \begin{cases} b_{O_p}^{O_{i1}:O_{i2}} = b_{O_{pj}}^{O_{tz1}} u_{O_{pj}}^{O_{tz2}} + b_{O_{pj}}^{O_{tz2}} u_{O_{pj}}^{O_{tz1}} \Big/ deno \\ d_{O_p}^{O_{i1}:O_{i2}} = d_{O_p}^{O_{i1}} u_{O_p}^{O_{i2}} + d_{O_p}^{O_{i2}} u_{O_p}^{O_{i1}} \Big/ deno \\ u_{O_p}^{O_{i1}:O_{i2}} = u_{O_p}^{O_{i1}} u_{O_p}^{O_{i2}} \Big/ deno \\ \alpha_{O_p}^{O_{i1}:O_{i2}} = \alpha_{O_p}^{O_{i1}} \end{cases} \tag{9}$$

$$deno = u_{O_p}^{O_{i1}} + u_{O_p}^{O_{i2}} - u_{O_p}^{O_{i1}} u_{O_p}^{O_{i2}}$$

After the delivery of the requested resource, an initial evaluation of the exchange is performed. This evaluation allows an initial update of the trust values and reputation. In addition, during the use of the resource and until the end of the period of use, this process of updating the values is triggered by each violation of the quality of service attributes specified in the SLA.

As in proposal [12], a discount factor θ is defined to determine the reputation of the supplier O_p according to the equation below:

$$rep_{O_p} = \begin{cases} rep_{O_p}^{ct} + \theta_n \text{ if positive result} \\ rep_{O_p}^{ct} + \theta_n/2 \text{ if positive result with minor violation} \\ rep_{O_p}^{ct} - \theta_n \text{ if negative result} \\ \text{with } \theta_n(k) = \dfrac{\left(V_{\text{lmin}}(k) + V_{\text{lmax}}(k)\right)/2}{V_{\text{lmax}}} \end{cases} \tag{10}$$

$rep_{O_p}^{ct}$ the current reputation of the provider, $V_{\text{lmin}}(k)$ the minimum value of vulnerability degree of the resource for a level k, $V_{\text{lmax}}(k)$ the maximum value of vulnerability degree, V_{lmax} the maximum value of vulnerability degree. The resource vulnerability levels are defined according to the CVSS 2 standard [9]. Thus:

$$\begin{aligned} &k = 1 \text{ if } V_l \in [7,10] \\ &k = 2 \text{ if } V_l \in [4,7[\\ &k = 3 \text{ if } I_l \in [0,4[\end{aligned} \tag{11}$$

3.4. Algorithm of the System

Figure 2 below illustrates the operating mechanism of the proposed framework. The process is characterized on the one hand by the validation, initialization and sharing phase. On the other hand, the second phase consists of monitoring the quality of the resources and updating the trust values in case of SLA violation. Algorithm 1 and Algorithm 2 in the **Appendix** describe these different phases.

4. Experiments and Results

4.1. Experimentation Environment

The performance of our proposal was evaluated through simulations of resource sharing between organizations in a community cloud. Data was generated to model the organizations and resources in the community. A MacBook Pro computer (Retina, 15-inch, mid-2015), 2.2 GHz Intel Core i7 quad-core processor,

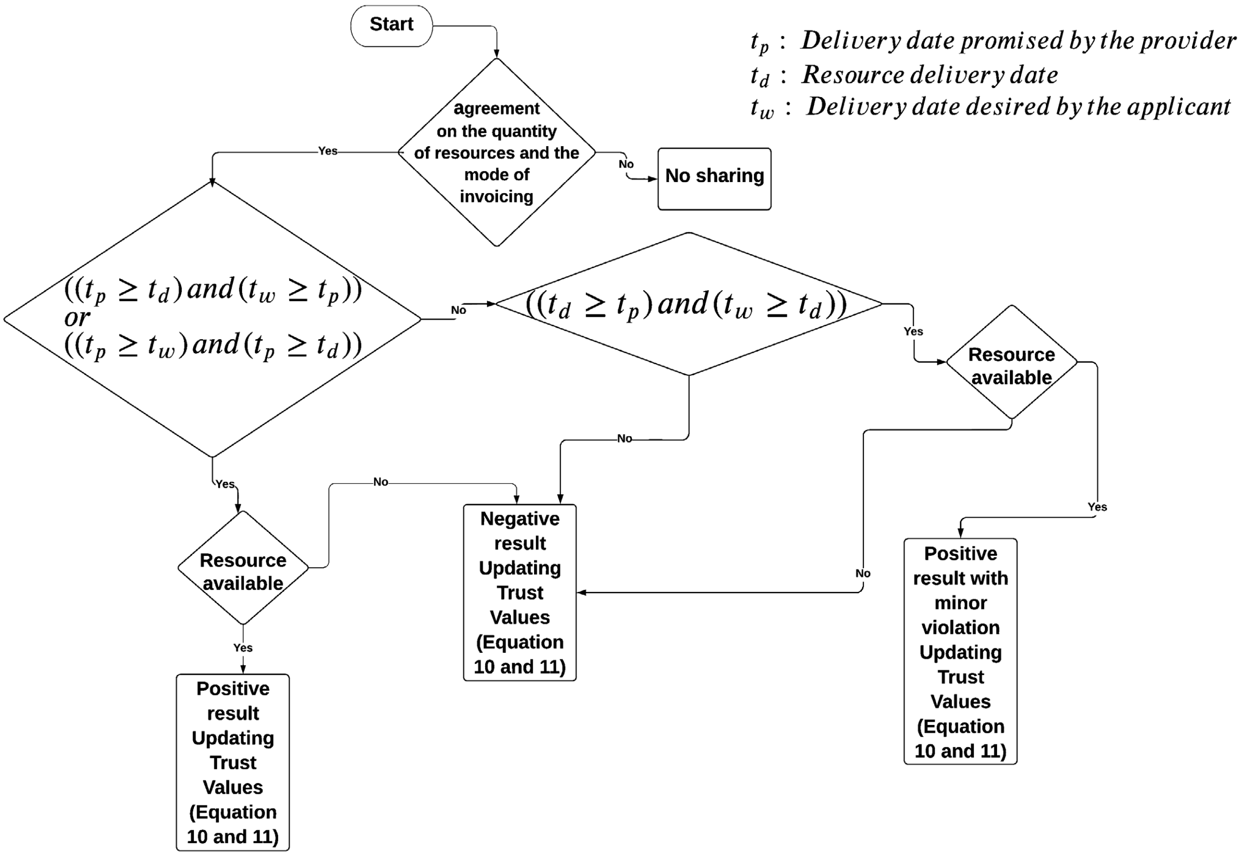
Start
t_p : Delivery date promised by the provider
t_d : Resource delivery date
t_w : Delivery date desired by the applicant
agreement on the quantity of resources and the mode of invoicing
Yes
No
No sharing
$((t_p \geq t_d) and (t_w \geq t_p))$ or $((t_p \geq t_w) and (t_p \geq t_d))$
No
$((t_d \geq t_p) and (t_w \geq t_d))$
Yes
Resource available
No
Yes
No
Resource available
Yes
Negative result Updating Trust Values (Equation 10 and 11)
No
Yes
Positive result with minor violation Updating Trust Values (Equation 10 and 11)
Positive result Updating Trust Values (Equation 10 and 11)

Figure 2. Workflow of framework.

16 GB 1600 MHz DDR3 memory was used to perform our experiments. The python language was used for programming with a Pycharm editor. Organizations providing SLA compliant resources are qualified as good providers. Those providing non-compliant or violating resources are called malicious. To examine the scalability of our framework the simulations are conducted on organization groups of 80,120, 180,220 and 250 members. Furthermore to measure the attack resistance of our model, the number of malicious providers is varied from 20%, 40%, 60%, 80%. The performance of our model is compared with two other models namely TNA-SL [11] [13] and InterTrust [14]. The framework has been evaluated based on two metrics namely the success rate of exchanging quality resources by good providers (SSR) and the execution time. In order to verify the execution time, the different algorithms were run on the same computer with identical loads.

$$\text{SSR} = \frac{\text{number of resources provides by good organizations}}{\text{total number of resources provided by good organizations}} \tag{12}$$

4.2. Results and Discussion

4.2.1. Resistance to Attacks

In order to evaluate the attack resistance capacity of our model, the rate of malicious providers is varied in a pool of 80 organizations. The following rates of malicious providers were used: 20%, 40%, 60% and 80%. 15 rounds of 500 transactions were used to obtain the results presented in Figure 3(a), Figure 4(a), Figure 5(a) and Figure 6(a). The different graphs present the evolution of the SSR values for the different proportions of malicious entities. The SSR of our model is constantly increasing and significantly higher than those of the InterTrust and TNA-SL algorithms until reaching the maximum for a rate of 20% of malicious entities. This observation is explained by the fact that our approach proposes the calculation of the trust value of the supplier by combining the previous direct or recommended interactions and the reputation.

In addition, the dynamic management of trust through the updating of trust values based on the respect of SLAs, allows to increase the reputation values of the organizations contrary to the two other models which do not integrate this aspect. The SSR of the InterTrust and TNA-SL models also experience a relatively lower evolution for rates of 20%, 40% and 60% of malicious illustrated by

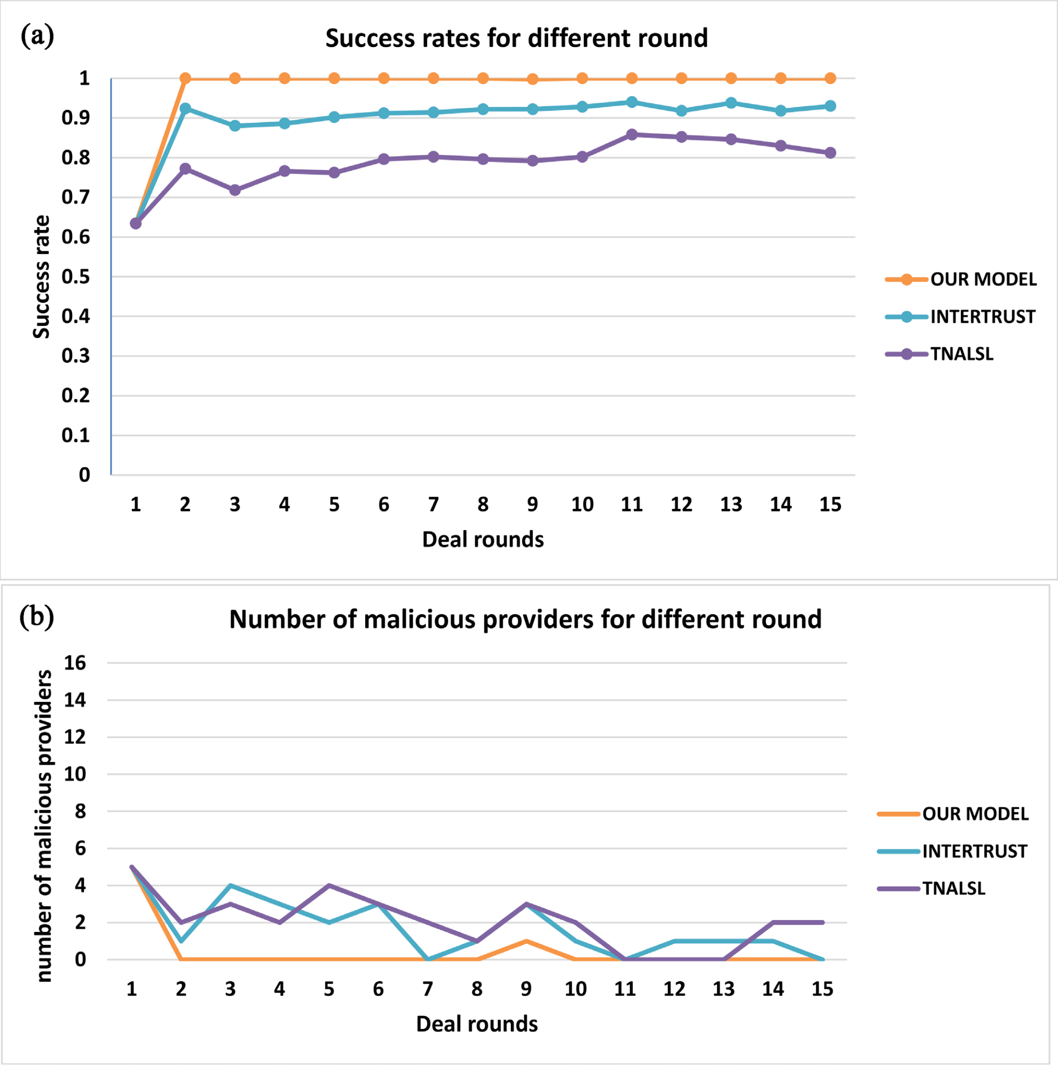

Figure 3. (a) Success rates for different numbers of resource providers of which 20% are malicious providers; (b) Numbers of malicious providers for different roud (20% are malicious providers).

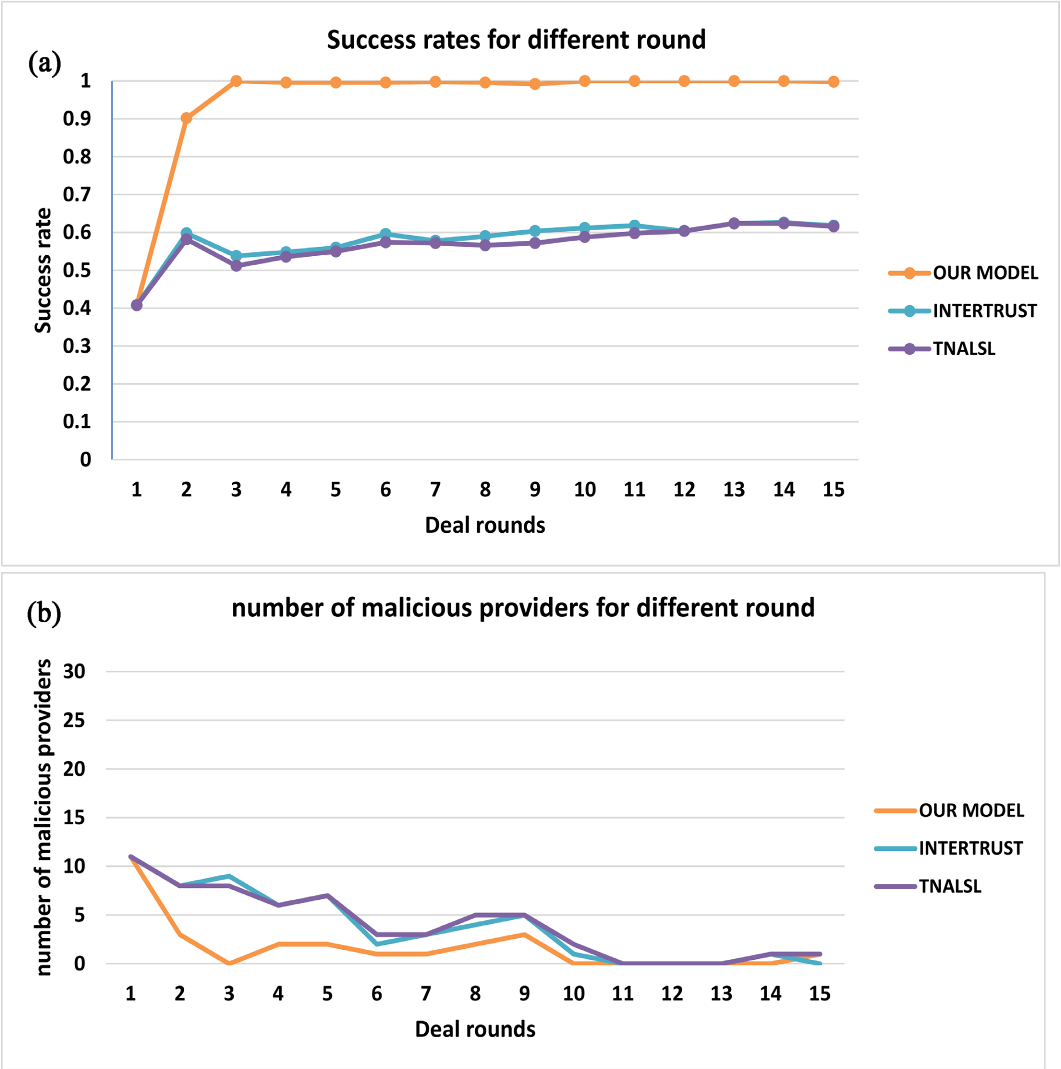

Figure 4. (a) Success rates for different numbers of resource providers of which 40% are malicious providers; (b) Numbers of malicious providers for different roud (40% are malicious providers).

Figure 3(a), Figure 4(a) and Figure 5(a). However, with a rate of 80% of malicious on Figure 6(a), a decrease of the SSR is observed for these two algorithms until reaching the zero value for the TNA-SL. Moreover, Figure 3(b), Figure 4(b), Figure 5(b) and Figure 6(b) show the variation of the number of malicious organizations according to the rate of injected malicious organizations. It can be seen that our model is able to reduce or even eliminate malicious organizations within the community. The resource quality monitoring through SMI attributes in case of SLA violation proposed by our model allows to identify quality resources and classify good providers. In contrast to the TNA-SL and InterTrust contributions, our technique increases the trust value and reputation of good providers faster with each transaction round and significantly reduces that of bad providers due to the discount factor θ introduced in Equation 10. The probability of selecting malicious organizations for long-term trading is thus limited.

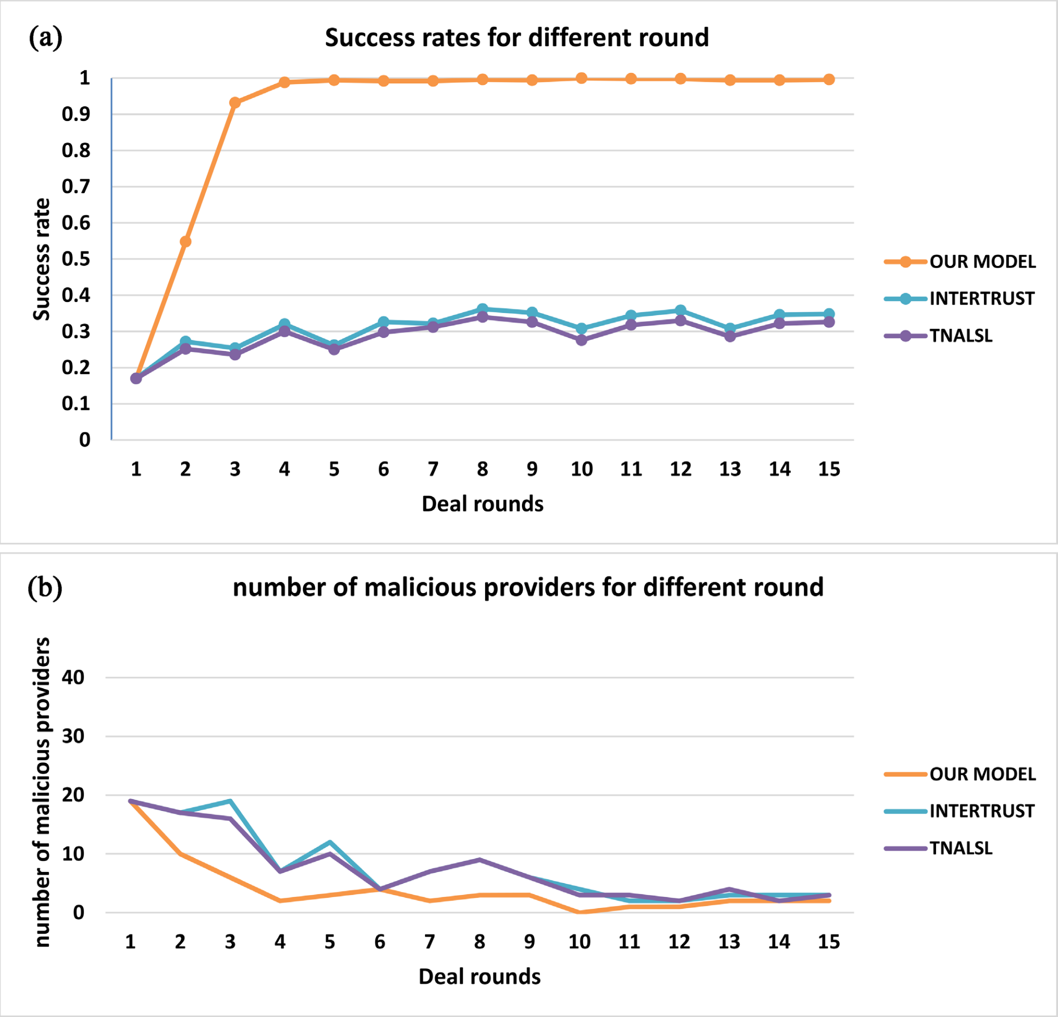

Figure 5. (a) Success rates for different numbers of resource providers of which 60% are malicious providers; (b) Numbers of malicious providers for different roud (60% are malicious providers).

4.2.2. Execution Time

Figure 7 below shows the execution times of our model, the InterTrust algorithm and the TNA-SL model. These experiments were performed for each round of 500 transactions with different groups consisting of 80, 120, 180, 220, 250 members. The results show that the execution time of our model is significantly lower than those of the other two models. The selection of the resource provider in our model is done first based on the previous transaction list of the requester and then based on the reputation list. This mechanism speeds up the selection process. Our model guarantees a high SSR while maintaining a low execution time.

5. Conclusion

The community cloud deployment model promotes collaboration and resource sharing (data and services) between organizations with specific requirements and common needs. However, ensuring trust between members of this community

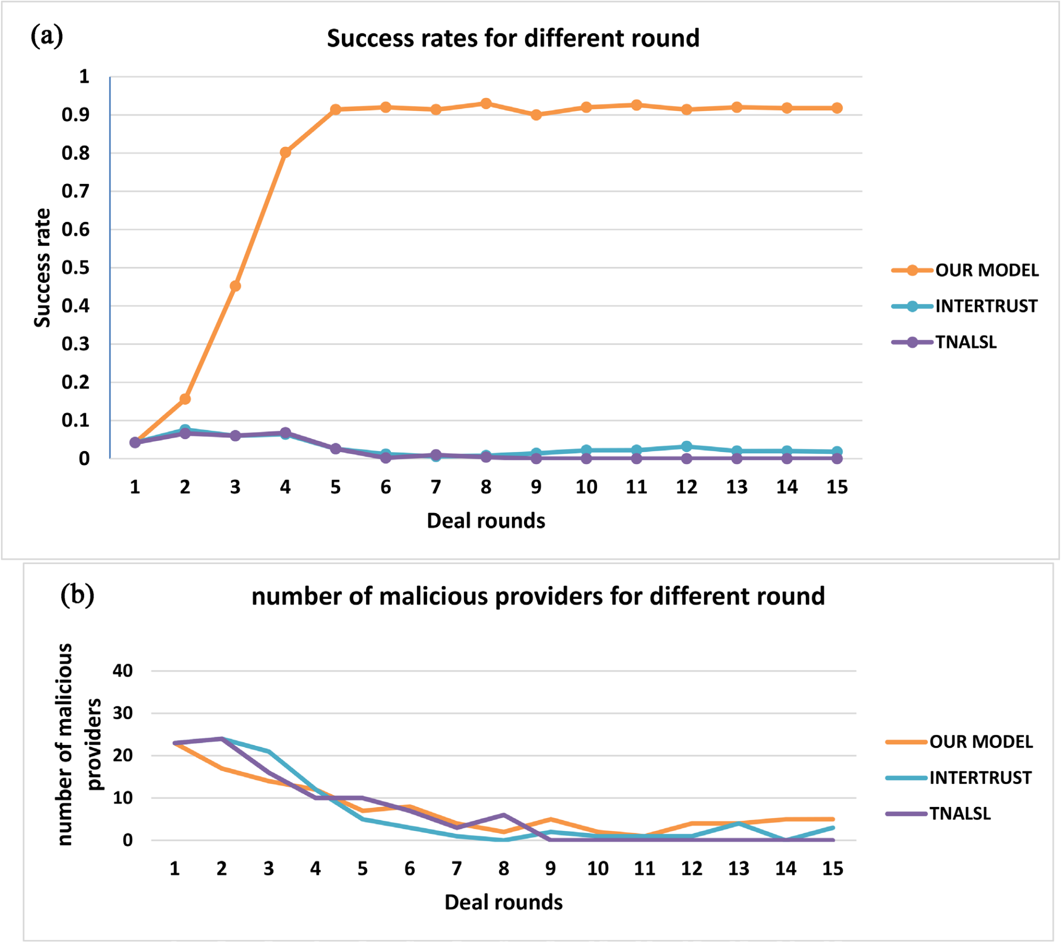

Figure 6. (a) Success rates for different numbers of resource providers of which 80% are malicious providers; (b) Numbers of malicious providers for different roud (80% are malicious providers).

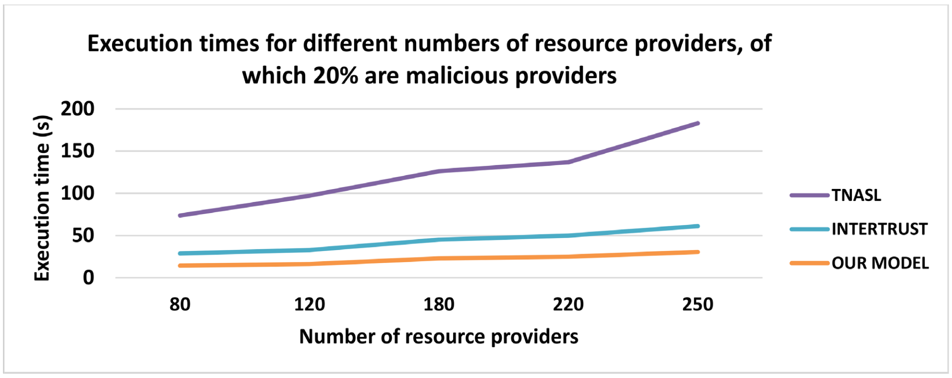

Figure 7. Execution times for different numbers of resource providers of which 80% are malicious providers.

remains a major challenge. Monitoring the quality of shared resources in this environment is an answer to this problem. Therefore, we propose in this paper a framework for resource quality evaluation and trust monitoring based on quality of service measurement attributes. The model is based on the definition of a service level agreement based on four SMI (Service Measurement Index) attributes:

billing mode, availability, threat and vulnerability management and response time. Moreover, the monitoring of SLA attributes allows to propose a dynamic management mechanism and to update the trust and reputation values. Through experiments, our proposal has been compared to the TNA-SL and InterTrust models. The results of the experiments show that our proposal allows the deployment of a community cloud that is resistant to attacks. On the other hand, it allows distinguishing good providers from malicious ones and to eliminate or reduce the participation of malicious ones in the community exchanges. Finally, our model offers better execution times compared to TNA-SL and Intertrust models. In future work, we will propose an agent-based collaboration framework. The aim is to present a model characterized by explicit communication and mutual commitment of the actors for successful exchanges.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Coret, S. (2021) Gartner. https://cloud-computing.developpez.com/actu/328766/Gartner-Les-revenus-mondiaux-du-cloud-s-eleveront-a-474-milliards-de-dollars-en-2022-contre-408-milliards-de-dollars-en-2021-le-cloud-sera-la-piece-maitresse-des-nouvelles-experiences-numeriques/

[2] Nauges, L. (2011). https://nauges.typepad.com/my_weblog/2011/04/index.html

[3] Siegel, J. and Perdue, J. (2012) Cloud Services Measures for Global Use: The Service Measurement Index (SMI). 2012 *Annual SRII Global Conference*, San Jose, 24 September 2012, 411-415. https://doi.org/10.1109/SRII.2012.51

[4] Papadakis-Vlachopapadopoulos, K., González, R.S., Dimolitsas, I., Dechouniotis, D., Ferrer, A.J. and Papavassiliou, S. (2019) Collaborative SLA and Reputation-Based Trust Management in Cloud Federations. *Future Generation Computer Systems*, **100**, 498-512. https://doi.org/10.1016/j.future.2019.05.030

[5] Priya, A.S.B. and Bhuvaneswaran, R.S. (2020) Cloud Service Recommendation System Based on Clustering Trust Measures in Multi-Cloud Environment. *Journal of Ambient Intelligence and Humanized Computing*, **12**, 7029-7038. https://doi.org/10.1007/s12652-020-02368-2

[6] Dubey, K., Shams, M.Y., Sharma, S.C., Alarifi, A., Amoon, M. and Nasr, A.A. (2019) Management System for Servicing Multi-Organizations on Community Cloud Model in Secure Cloud Environment. *IEEE Access*, **7**, 159535-159546. https://doi.org/10.1109/ACCESS.2019.2950110

[7] Garg, S.K., Versteeg, S. and Buyya, R. (2013) A Framework for Ranking of Cloud Computing Services. *Future Generation Computer Systems*, **4**, 1012-1023. https://doi.org/10.1016/j.future.2012.06.006

[8] Saovapakhiran, B. and Devetsikiotis, M. (2011) Enhancing Computing Power by Exploiting Underutilized Resources in the Community Cloud. *IEEE International Conference on Communications*, Kyoto, 28 July 2011. https://doi.org/10.1109/icc.2011.5962544

[9] Common Vulnerability Scoring System SIG (2020). https://www.first.org/cvss/

[10] Jøsang, A. (2009) Subjective Logic. *Representations*, **171**, 1-8.

[11] Jøsang, A. and Bhuiyan, T. (2008) Optimal Trust Network Analysis with Subjective Logic. *Proceedings of the* 2*nd International Conference on Emerging Security Information*, *Systems and Technologies*, Cap Esterel, 25-31 August 2008, 179-184. https://doi.org/10.1109/SECURWARE.2008.64

[12] Guo, L., Yang, H., Luan, K., Luo, Y., Sun, L. and Zheng, X. (2021) Trust Management Model Based on Mutual Trust and a Reward-with-Punishment Mechanism for Cloud Environments. *Concurrency and Computation*: *Practice and Experience*, **33**, e6283. https://doi.org/10.1002/cpe.6283

[13] Jøsang, A. and Simon, R. (2006) Trust Network Analysis with Subjective Logic. *Angewandte Chemie International Edition*, **6**, 951-952.

[14] Kurdi, H., Alfaries, A., Al-Anazi, A., Alkharji, S., Addegaither, M., Altoaimy, L. and Ahmed, S.H. (2019) A Lightweight Trust Management Algorithm Based on Subjective Logic for Interconnected Cloud Computing. *Journal of Supercomputing*, **75**, 3534-3554. https://doi.org/10.1007/s11227-018-2669-y

Appendix

Algorithm 1 Validation and initialization of the transaction

Input: the resource r of vulnerability level v_l, quantity requested q_{rpj}, quantity supplied q_{rui}, bm_{Oui} the desired invoicing method, bm_{Opj} the invoicing method accepted by the provider, the date on which the client wishes to have the resource t_w, the date on which the supplier promises to provide the resource t_p, the delivery date of the resource t_d, Initial resource availability A_{rinit},the value of the result of sharing R_s the type of the result of sharing T_{rs}.

```
procedure valInitTram ($r, v_l, q_{rpj}, q_{rui}, O_u, t_{ui}, t_{pj}, t_d, t_{ui}, bm_{Oui}, bm_{Opj}$)
    if ($q_{rui} == q_{rpj}$) and ($bm_{Oui} == bm_{Opj}$) then
        if (($t_p \geq t_d$) and ($t_w \geq t_p$)) || (($t_p \geq t_w$) and ($t_p \geq t_d$)) then
            if ($A_{rinit} == 1$) then
                $R_s \leftarrow 1$
                $T_{rs} \leftarrow 1$
                $updTrustValues(R_s, T_{rs})$
            else
                $R_s \leftarrow 0$
                $T_{rs} \leftarrow 0$
                $updTrustValues(R_s, T_{rs})$
            end if
        else if ($t_d \geq t_p$) and $t_w \geq t_d$) then // minor violation
            if ($A_{rinit} == 1$) then
                $R_s \leftarrow 1$
                $T_{rs} \leftarrow 0.5$
                $updTrustValues(R_s, T_{rs})$
            else
                $R_s \leftarrow 0$
                $T_{rs} \leftarrow 0$
                $updTrustValues(R_s, T_{rs})$
            end if
        else// violation
            if ($A_{rinit} == 1$) then
                $R_s \leftarrow 1$
                $T_{rs} \leftarrow 0$
                $updTrustValues(R_s, T_{rs})$
            else
                $R_s \leftarrow 0$
                $T_{rs} \leftarrow 0$
                $updTrustValues(R_s, T_{rs})$
            end if
        end if
    else
        Unable to share
    end if
end procedure
```

Algorithm 2 update trust values

```
Pre-condition: The result of sharing $R_s$ or violating quality settings
Input: The value of the result of sharing $R_s$, The type of the result of sharing $T_{rs}$
procedure updTrustValues ($R_s$, $T_{rs}$)
    if $R_s == 1$ and $T_{rs} == 1$ then // Positive result without violation
        $reputation\ update\ use\ Equation\ 10\ and\ 11\ (positive\ result)$
    else if $R_s == 1$ and $T_{rs} == 0.5$ then
        $reputation\ update\ use\ Equation\ 10\ and\ 11\ (positive\ result\ with\ violation)$
    else
        $reputation\ update\ use\ Equation\ 10\ and\ 11\ (negative\ result)$
    end if
end procedure
```