Local Reputation Management in Cloud Computing

Jiuyun Xu, Dan Jiang, Bin Wang, Dan Yang
School of Computer and Communitcation Engineering
China University of Petroleum(China East)
Qingdao, China
jyxu@upc.edu.cn,Jiangdan363@163.com,
wb_upc@163.com, yangdan918@yeah.net

Stephan Reiff-Marganiec
Department of Computer Science
University of Leicester
Leicester,UK
srm13@le.ac.uk

***Abstract*—In the Cloud computing community, the calculation of the reputation using the feedback of cloud customers is widely adopted to address the issue of trustworthiness of cloud services. Currently, most methods pursue a global reputation score essentially assuming that the value of a cloud service's reputation is the same for every consumer. However depending on the expectations and needs of a consumer, there can be significant deviation of perceived reputation for the same cloud service. In this paper we propose a trust management framework that differentiates reputation for various user groups thus providing what we term local reputation. To achieve this we compute the similarity of consumers based a decision-tree model which is used to cluster feedback into localised scores. To refine the result, a time decay factor applicable to feedback is also to be considered. The simulation results illustrate that our approach is feasible and also effective for consumers to choose reputable cloud service.**

***Keywords*-Cloud Computing; Trust Management; Reputation; Similarity; Decision Tree;**

I. Introduction

Over the last few years, cloud computing is gaining prevalence due to its advantages such as on-demand availability of computing resources and software services. With cloud computing, computer resources are designed and governed in the form of services using virtualization techniques. They are used to automate business logic in both public and private sectors. Except those benefits, data confidentiality, data privacy and trust establishment are considered to be the main security concerns for an organization moving its data to the cloud platform [1], [2].

Currently, to address the issue of trustworthiness, one popular method is the computation of the reputation [2]. Reputation is the topic of a lot of interdisciplinary research. In [5] the concept of reputation is defined as "the collected and processed information about one entity's former behaviour as experienced by others". In cloud computing terms, the reputation value of a cloud service represents the extent to which consumers can trust the service, based on aggregation of feedback given by previous consumers. The use of reputation systems is common in commercial setting, such as eBay's feedback mechanism or Epinions [4]. In existing methods for cloud computing, the common trait is that a global value of the reputation is calculated through these reputation systems meaning that the value is the same for all consumers.

However, in general the reputation is potential subjective. Feedback scores are based on the individual experience of a service, which always needs to be seen in the context of the users experience and expectations of a service. Consider for example two cloud service consumers A (denoted as C_A) and B (denoted as C_B). They are both looking for a cloud storage service. C_A is an expert in the field of IT and has very specific requirements for confidentiality. C_B is an amateur whose main aim is to obtain cheap storage. Now consider a free storage service – C_B is likely to be very happy as the cost is minimal and if usability is ok he will rank this service very highly. C_A might read the small print regrading data policies and is likely to find issues that are not to her liking; so they would rank the service somewhat lower.

Both are of course right in their respective rankings. A global reputation score would not reflect the service users accurately. To address this issue, in this paper, we introduce the concept of local reputation. In other words, the value of the cloud service reputation is differentiated for different cloud service requestors. Returning to our consumers above, our trust management model would produce two different scores for C_A and C_B for the same storage service but reflecting their respective perspectives.

This paper makes two main contributions. First, we propose the preference similarity to divide consumers who gave feedback to the cloud service into several classes allowing to match requestors to a group of similar users. Unlike the current method of calculating the preference similarity, we utilize a decision tree to establish the relationship between the personal information which is recorded in our local reputation model and the similarity among consumers allowing to predict the similarity degree between the requestor and the feedback providers. Second, time decay is deeply embedded into our method as a factor which reflects the reduction in reliability of feedback over time.

The remainder of the paper is organized as follows: Section 2 introduces the trust management model in detail

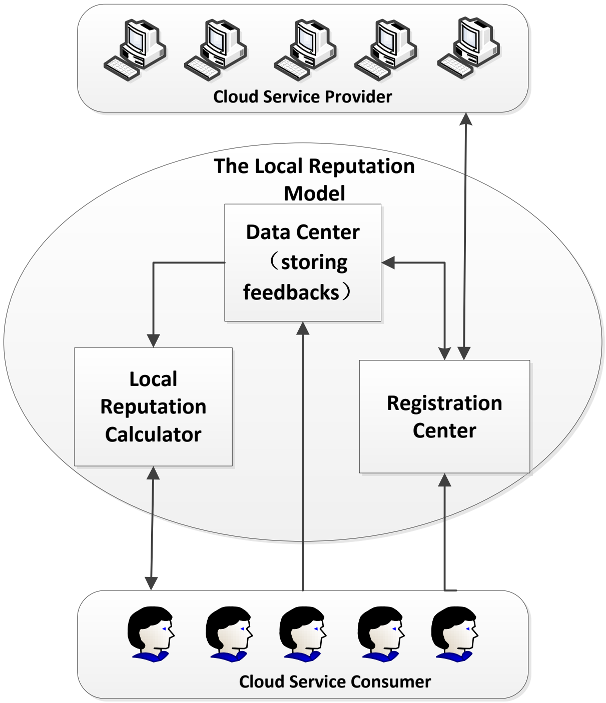
Cloud Service Provider
The Local Reputation Model
Data Center （storing feedbacks）
Local Reputation Calculator
Registration Center
Cloud Service Consumer

Figure 1. Overview of Local Reputation Management Approach

and presents the method to calculate the value of the local reputation. Section 3 describes how to construct the decision tree based on consumers' attributes as well as how the decision tree predicts the similarity. Time decay is also discussed in Section 3. An experiment to verify the feasibility and effectiveness of our approach is conducted and the performance is analysed in Section 4. In the end of the paper, we discuss the related work and make a conclusion of our work in Section 5 and Section 6 respectively.

II. Local Reputation Management

The overall structure of the proposed local reputation management approach is shown in Fig. 1. The cloud service providers and the cloud consumers make up the cloud environment with the trust management center playing the central role. Trust management consists of three roles: the Registration Center, the Data Center and the Local Reputation Calculator.

1) Registration Center.: The cloud Service providers and the cloud service consumers are all required to register in the registration center before they enter the cloud market. Registration requires provision of essential information which is stored in the date center. For cloud service providers, the essential information includes the cloud service's ID, the function that the service provides and the offered performance of the function. For cloud consumers, we need personal information typical for user profiles, which we assume is confidential in our model.

2) Data Center.: The Data Center stores the essential information mentioned afore and also records feedback given by cloud service consumers who experienced a service. Feedback is denoted as a tuple $F = (S_{id}, U_{id}, R, T)$, where the S_{id} represents the identity of the cloud service provider whose service is invoked by the consumer and U_{id} is the identity of the cloud consumer who used the S_{id}. R is the score that the consumer gave to the service with integer values in the interval $[0, 5]$. T is a timestamp which records when the feedback was given by the consumer.

3) Local Reputation Calculator.: The Local Reputation Calculator is the core feature of our approach, which is able to aggregate the feedback for a cloud service into a local reputation score for the service requestor based on the preference similarity between the service requestor and feedback givers. The mechanism used will be a decision tree introdced in detail in Section III. In general, we expect that users with higher similarity in their attributes would have a higher probability for reaching a consensus on an identical thing. Hence in the cloud environment, we assume that the similar consumers' feedbacks are more trustworthy than the global view.

In our model,the local reputation of a service is calculated as follows:

$$R(s) = \frac{\sum_{i=1}^{n} f(i,s) * C_i}{n} \tag{1}$$

where $R(s)$ is the value of the local reputation of cloud service s and $f(i,s)$ is the feedback to the s^{th} cloud service from the i^{th} cloud consumer. C_i is the credibility of the i^{th} cloud consumer, which is computed based on the similarity between the requestor for the s^{th} cloud service and consumers who have experienced the s^{th} cloud service and given the feedback.

III. Decision tree-based Similarity Computation

A. Calculation of similarity among cloud consumers

The core of our method is based on calculating the similarity between consumers. [11] used consumers' similarity based on Pearson Correlation Coefficients to reduce the subjective influence on the reputation – in the end all opinions are subjective. However, to the contrary we believe that the reputation should reflect the preference of the consumer and hence that such subjective opinions are a positive aspect. Hence, we take advantage of the subjective personal experience to calculate the local reputation. We make an **assumption** that the individual essential information (including the information of the location, the level of education, the job, the age and the gender) affects the preference of a person. There could be further factors and these can easily be added into our method. A decision tree then helps to predict the extent that consumer's preferences are similar.

Decision trees are one machine learning method to solve classification and prediction problems and are mainly utilized in the data mining area. A decision tree is a flowchart-like tree structure, where each internal node of the tree represents a property of the test, namely in the form of $(a_i = v_i)$, where a_i is a property and v_i is one possible value for the property. Each branch then represents a test result. Leaf nodes specify a category. Employed in our context, the decision tree can help us find the relationship between attributes and compute the preference similarity.

We adopt the C4.5 algorithm to construct up the decision tree. In the training step, we first define the attributes of a consumer available from registration as $U = (L, E, J, A, G)$ meaning the location of the consumer, the level of education that a consumer has, the job that a consumer engages in, the age and the gender of the consumer (as stated above there could be other factors in addition or replacing some of these). Also, V will be the set of all consumers of a service.

Each training sample in the training set is a six-tuple, denoted as $T = (L', E', J', A', G', \Delta R)$ and the first five dimensions in T are depicted as internal nodes of the decision tree, whereas the ΔR is represented as the leaf node of the tree. T is calculated as follows: we randomly select a service S and then choose a consumer C_r randomly from S's consumer set V_s. C_r's attribute $U_c = (L, E, J, A, G)$ and also C_r's feedback $F_c = (S_id, C_id, R, T)$. are obtained from the data store. Then, we compare C_r's attribute with the rest of the $member$'s attributes in V one by one. When their attributes are the same in one dimension, the corresponding dimension of T is 1, others are set to 0. For example, when C_r and another consumer are both from Hebei province, then $L' = 1$; but one of them is female while the other is male would mean that $G' = 0$. Once this is done for all attributes, ΔR can be calculated.

The value of ΔR is an absolute value capturing the similarity. $\Delta R = |R_c - R_o|$ with values ranging from 0 to 5. R_c is C_r's rating of service S, and R_o is one $consumer$'s rating in the set V_s.

Eventually, the leaf node of the decision tree that is built based on the training set contains a value for ΔR which stands for the degree of similarity. The smaller value of ΔR, the more similar those two consumers are. And the remaining dimensions in T are internal nodes of this tree. Algorithm 1 describes how to construct the decision tree.

After the training step, the decision tree is ready for application in the service request process, where it is able to compute the degree of the similarity between the requestor and the feedback giver based on their attributes by dividing feedback givers into different classes. Figure 2 depicts the whole process.

It is worth mentioning that the decision tree is available for every request for all kinds of cloud services, since the target of the decision tree is identifying the similarity of a consumer to other consumers. The decision tree is an universal approach.

Algorithm 1 The Decision Tree

```
Input: Service $S$'s consumers set $V_s$ with $consumers$'
    attributes $U$s and feedbacks $F$s;
    a random consumer with his attribute $U_c$ and feedback
    $F_c$.
Output: A Decision Tree
$Attribute\_list \leftarrow U$
while $V_s$ do
    $U'_i \leftarrow |U_c - U_i|$
    $F'_i \leftarrow |F_c - F_i|$
    $Sample_i \leftarrow U'_i \cup F'_i$; $\cup$ is the connection operation
end while
C4.5($Samples$, $Attribute\_list$)
return A Decision Tree
```

B. Time Decay of the Feedback

Generally the performance of the cloud service is not stable, and the service might be subject to other changes (for the better or worse). For example providers with good quality might be very popular and their quality reduces due to high loads or services starting quite poorly improve based on feedback (see for example [6]). So a rating given some time back might not be accurate anymore, despite having been right and when it was made. Hence we need to factor time decay into our approach to adjust the credibility of the feedback. In brief, feedback given a long time ago ought to be assigned a lower weight. The relationship between feedback and time is depicted as follows [7], [8]:

$$p(t) = e^{-k*\Delta t} = e^{-k*(t_n - t_i)} \tag{2}$$

where t_i means the time when the feedback was given and t_n stands for the time when the requester asked for the reputation of the cloud server. The parameter k decides the speed of decay.

Finally, the credibility of the feedback will be expressed as follows:

$$C_i = \alpha S_i + \beta P_t \qquad \alpha + \beta = 1 \tag{3}$$

Where α and β denote the similarity among cloud consumers and the time decay factors' normalized weight, respectively. S_i is the degree of the similarity obtained from the decision tree. Algorithm 2 depicts the overall process of our approach.

IV. Experiments and Analysis

Due to the lack of a benchmark dataset for this kind of work, we used the social network dataset on Weibo[1] which includes more than ten million $users$' individual information

[1] http://www.datatang.com/data/46324

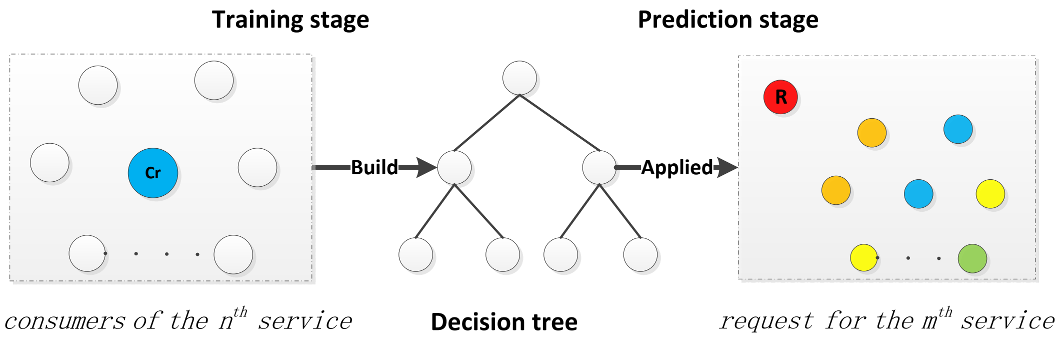
Training stage
Prediction stage
Cr
Build
Applied
R
consumers of the n^{th} service
Decision tree
request for the m^{th} service

Figure 2. Workflow involving the Decision Tree

Algorithm 2 Calculation of a local reputation value

```
Input: A Decision Tree;
    Service $M\prime$s consumer set $V_m$ with $consumers\prime$ attributes $U$s and feedbacks $F$s;
    a $requestor\prime$ s attribute $U_r$ who requests for service $M$.
Output: A local reputation value of a service
while $V_m$ do
    $Tree\_input \leftarrow |U_r - U_i|$
    $S_i \leftarrow Decisiontree$(Tree_input); $S_i$ means the similar degree between the requestor and the consumer $i$ in $V_m$
    $t_i \leftarrow F_i$; $t_i$ means the time that the feedback gave by the consumer $i$
    $$p(t_i) = e^{-k*\Delta t} = e^{-k*(t_n - t_i)}$$
    $$C_i = \alpha S_i + \beta P_t$$
end while
$$R(s) = \frac{\sum_{i=1}^{n} f(i,s) * C_i}{n}$$
return R(s)
```

and combined this with the Epinion[2] rating data set which is a real-life trust data set. Then we simulated the rate given by various consumers. In addition, we assume that there are six kinds of job and they are skill-relevant, research-relevant, art-relevant, economic administration-relevant, and social-relevant, public relation-relevant.

A. Similarity among Consumers

We applied the machine learning and data mining tool WEKA using the C4.5 algorithm to draw the decision tree based on the training set. Figure 3 shows the result of the decision tree with the 'Job' attribute as the root. That is to say, the information of job is seen to be the major contributor for the similarity degree and the information of education and location are in the second place, with a lighter weight on age and gender information.

Table I
SUMMARY OF RESULTS FOR THE DECISION TREE

Correctly Classified Rate	71.8519%
Incorrectly Classified Rate	28.1481%
Kappa statistic	0.6552
Mean absolute error	0.1123
Root mean squared error	0.2735

The leaf node stands for the deviation of the rate given by different consumers, therefore the smaller the value of the leaf node is, the more similar the two consumers are. It is reasonable that people who are working in the same area have high chances of focusing on the same performance of the cloud service and then the main factors affecting the preference are education and location, because again they make users more similar.

According to the tree, we can see that, if the $requestor$'s attribute and the $consumer$'s attribute are the same in all dimensions, that is to say values of all internal nodes are 1, they will reach a zero leaf node showing that there is no difference between the users (or that they are perfectly similar with regard to the compared values). We would then assume that their service ranking will be in consensus as they would have similar expectations. Table 1 illustrates the accurateness of the decision tree approach: 71% of cases were correctly classified with very small statistical error margins.

B. Time Decay of the Feedback

We use formula 2 to calculate the time decay. We assumed that the cloud service will update once in ten days, so we set the parameter k to 0.5. As Fig. 4 shows, the degree of trustworthiness declines over days.

In order to verify the effectiveness of our method, we assume two requestors asking for the reputation of the same cloud service; let the service

[2] http://www.trustlet.org/wiki/Downloaded Epinions dataset

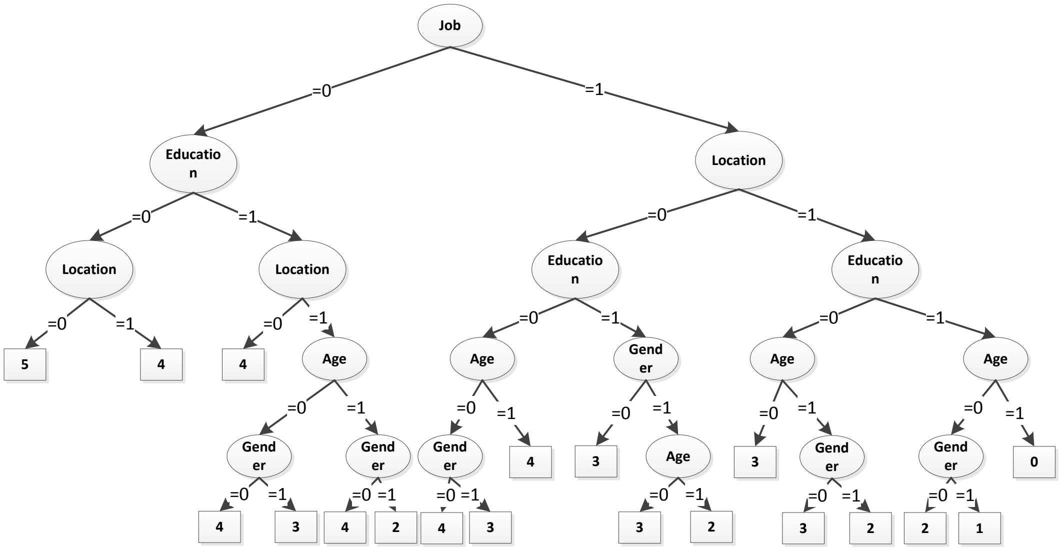

Figure 3. The Decision Tree

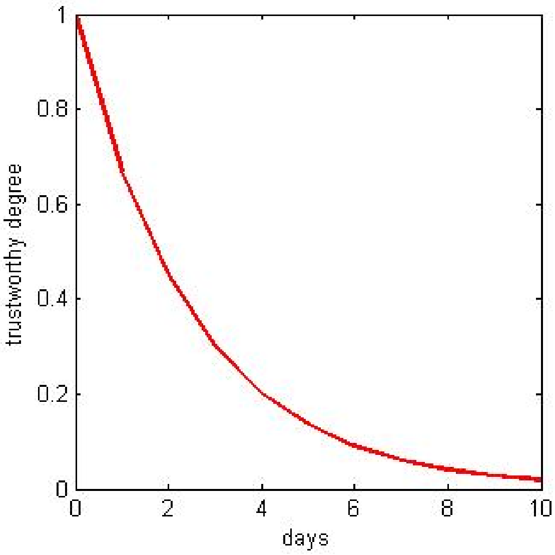

Figure 4. the Result of Time Decay

ID be 4. One user is Alice with the attributes $U_A = \{'hebei','undergraduate','female'.'skill-relevantjob','25'\}$ and the other is Bella with the attributes $U_B = \{'fujian','master','female','social-relevantjob','35'\}$. To make sure that the value of the local reputation falls on integers in the interval $[0,5]$, we defined the set $w = \{5,4,3,2,1,0\}$ which corresponds to the degree of the credibility of the feedback denoted as $D = \{0.1,0.3,0.5,0.7,0.8,0.9\}$.

Figure 5 shows the result for Alice and Bella asking for the local reputation of service 4 within an increasing size of the set of service consumers. The trends for the two values of local reputation is similar for both Alice and Bella because we compare against the same consumer set. However, the results of the values computed are different as expected as Alice and Bella differ.

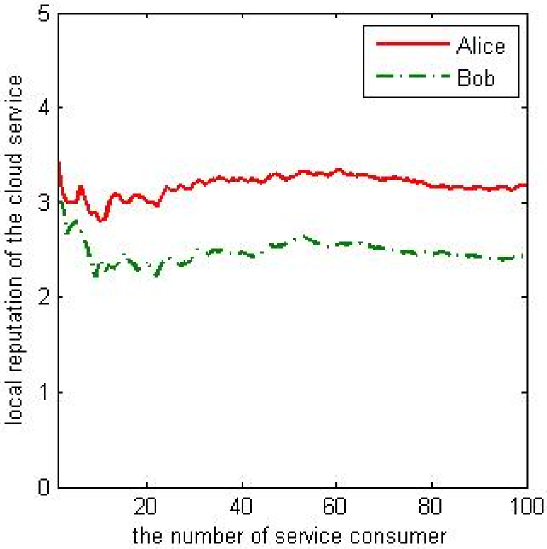

Figure 5. the value of the Local Reputation requsting by Alice and Bob

V. Related Work

Since the trust issue has been identified as a major stumbling block for Cloud service uptake, many solutions have been proposed to solve this problem. [1] and [2] provide good summaries.

Reputation-based trust modeling is one of the most popular ways. In [9], the author proposed an RLM trust model, which employed a linear hidden Markov process to evaluate an accurate reputation. Furthermore, [3] introduced a trust as a service framework where the feedback is given weightdepending on the majority consensus and the cloud consumer's capability. [10] presented a trust framework in a service-oriented environment. The majority consensus and time consistency are considered as main factors which influence the trust. [11] proposed malicious feedback detection based on cumulative sum control charts and then reduces the effect of the subjective feedback. [7] believe that trust and reputation have spatial and temporal dimensions.

[12] emphasize that trust can also be seen from the service providers' perspective, meaning tat providers might not trust customers and hence prefer to avoid providing service to untrustworthy consumers.

[13] proposed a novel polling algorithm to share the reputation in peer-to-peer environment in order to reduce the spreading of malicious contents. [14], use fuzzy techniques for assessing and integration feedbacks.

[15] provided a reputation as a service system, in which the calculation of the reputation is no longer depending on personal experience but rather on the real-time interaction from the cloud itself.

In this paper, considering there are various consumers in cloud environment, we introduced the local reputation mechanism based on decision tree to represent the characteristics of every customer. This breaks with the above works that consider overall rankings applicable to for all customers. At the same time, we adopted a time decay factor to gain a more accurate feedback credibility.

VI. Conclusion

In our increasingly virtualized world, reputation of services is becoming a very important decision criteria for consumers. Since most consumers will not have any direct experience of every service to compare themselves, they must rely on the reputation system. Such reputation systems usually compute a single score expressing the overall satisfaction of all users with a service. However, we feel that different groups of users have different expectations and needs and hence the 'one size fits all' is not appropriate. So we have proposed the local reputation system which employs a decision tree to sort consumers into classes based their similarity. We then use this as input to weigh the credibility of the feedback, and then calculate the local reputation value for each requestor's perspective. In addition, we consider time decay, as this significantly effects the reliability of feedback.

Experimental results show that in the proposed approach, the requestor obtains a reputation value of the service which is in line with the requestor's background.

In the future, we like to pursue some open aspects. At present, it is common for malicious behaviours to occur in reputation systems such as the Sybil attacks and collusion attacks which lead consumers into false safety [16]. Hence, we need to turn to the robustness and efficiency of our method. In addition, our local reputation model is based on the hypothesis that all cloud consumers and cloud service providers are registered with the information that we need and we like to explore whether this reliance can be reduced.

Acknowledgment

The paper is fully supported by a promotive research fund for young and middle-aged scientisits of Shandong Province(No.2014BSE28044) and a grant from the Fundamental Research Funds for the Central Universities (Project No. YCX2014064).

References

[1] Kanwal, Ayesha, et al. ”Assessment Criteria for Trust Models in Cloud Computing.” Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing. IEEE, 2013.

[2] Noor, Talal H., et al. ”Trust management of services in cloud environments: Obstacles and solutions.” ACM Computing Surveys (CSUR) 46.1 (2013): 12.

[3] Noor, Talal H., and Quan Z. Sheng. ”Trust as a service: a framework for trust management in cloud environments.” Web Information System EngineeringCWISE 2011. Springer Berlin Heidelberg, 2011. 314-321.

[4] Schlosser A, Voss M, Brckner L. uoge[J]. Journal of Artificial Societies and Social Simulation, 2006, 9(1).

[5] Dellarocas C. Reputation mechanisms[J]. Handbook on Economics and Information Systems, 2006: 629-660.

[6] Guo, Qiang, et al. ”Modeling and evaluation of trust in cloud computing environments.” Advanced Computer Control (ICACC), 2011 3rd International Conference on. IEEE, 2011.

[7] Bishr, Mohamed, and Werner Kuhn. ”Trust and Reputation Models for Quality Assessment of Human Sensor Observations.” Spatial Information Theory. Springer International Publishing, 2013. 53-73.

[8] Rizvi, Syed, et al. ”A centralized trust model approach for cloud computing.” Wireless and Optical Communication Conference (WOCC), 2014 23rd. IEEE, 2014.

[9] Xiaofeng Wang, Jinshu Su, Xiaofeng Hu, Chunqing Wu and Huan Zhou, Trust Model for Cloud Systems with Self Variance Evaluation, Nepal, Surya, and Mukaddim Pathan, eds. Security, Privacy and Trust in Cloud Systems. Springer, 2014. 297-308

[10] Malik, Zaki, and Athman Bouguettaya. ”Rateweb: Reputation assessment for trust establishment among web services.” The VLDB JournałThe International Journal on Very Large Data Bases 18.4 (2009): 885-911.

[11] Wang, S., et al. ”Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems.” (2014): 1-1.

[12] Conner, William, et al. ”A trust management framework for service-oriented environments.” Proceedings of the 18th international conference on World wide web. ACM, 2009.

[13] Damiani, Ernesto, et al. ”A reputation-based approach for choosing reliable resources in peer-to-peer networks.” Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002.

[14] Aringhieri, Roberto, et al. ”Fuzzy techniques for trust and reputation management in anonymous peertopeer systems.” Journal of the American Society for Information Science and Technology 57.4 (2006): 528-537.

[15] Itani, Wassim, et al. ”Reputation as a Service: A System for Ranking Service Providers in Cloud Systems.” Security, Privacy and Trust in Cloud Systems. Springer Berlin Heidelberg, 2014. 375-406.

[16] Noor T H, Sheng Q Z, Alfazi A. Reputation Attacks Detection for Effective Trust Assessment Among Cloud Services[C]//Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on. IEEE, 2013: 469-476.