Towards Reputation-as-a-Service

Channel Hillebrand[1] and Marijke Coetzee[2]
Academy of Computer Science and Software Engineering
University of Johannesburg,
South Africa
channel.hillebrand@gmail.com[1] marijkec@uj.ac.za[2]

***Abstract*—Reputation is used to regulate relationships of trust in online communities. When deploying a reputation system, it needs to accommodate the requirements and constrains of the specific community in order to assist the community to reach their goals. This paper identifies that there is a need for a framework to define a configurable reputation system with the ability to accommodate the requirements of a variety of online communities. Such a reputation system can be defined as a service on the Cloud, to be composed with the application environment of the online community. This paper introduces the concept of RaaS (Reputation-as-a-Service) and discusses a potential framework for creating a RaaS. In order to achieve such a framework, research is conducted into features of SaaS (Software-as-a-Service) components, user requirements for trust and reputation, and features of current reputation frameworks that can be configured in order to support a reputation service on the Cloud.**

Keywords: RaaS, trust, reputation, SaaS, reputation framework

I. INTRODUCTION

Online shopping has grown significantly in the past years and it is predicted that such sales will increase annually by 10% for the next 4 years [1] [2]. People are influenced by product reviews to make purchasing decisions and tend to buy from online stores with a good reputation [1]. As online shopping is characterized by insecurity, anonymity, lack of control and potential opportunism, online communities should take the necessary steps to ensure that participants are trustworthy.

In such environments, a reputation system can be used to compute and publish reputation scores for service providers, services, products or entities such as customers within a community. The calculation aggregates the collection of opinions or ratings that entities have about the objects. The ratings are given to a reputation system that uses a specific reputation algorithm to compute the reputation scores [3]. In order to be effective, such reputation managers need to accommodate the specific needs of the communities where they are deployed.

Consider a scenario of organization ABC, an online store for a start-up company that sells products to consumers over the mobile web. As trust and reputation is a major component to enable m-commerce, the online store of organization ABC need to deploy a reputation system to control trust relationships between consumers, suppliers and their portal. As there is no off-the-shelf reputation system to integrate into their application environment, and it is expensive to custom develop, the m-commerce web site may initially be implemented without it. Ideally, organization ABC needs a reputation system that is simple to use with easy to understand ratings between 0 and 5 to ensure the growth of the community. In an online community that provides a platform to post and record crime incidents with mobile phones, a reputation system is needed to ensure that no malicious or false incidents are reported. The requirements for this reputation system may be very different than those from the online store of organization ABC, indicating that a configurable or customizable reputation system is needed to support multiple online communities that will be cost-effective and efficient.

Recently, a business model for software applications namely SaaS (Software-as-a-Service) has emerged which lowers the cost of development, customization, deployment and operation of applications [4]. As SaaS applications generally support the concept of software application configuration and customization, this research proposes to present a configurable reputation system as a SaaS solution. This implies that the multi-tenant architecture is followed where organizations only pay for features they access, and are able to configure or customize the reputation system to suit their community's needs.

The contribution of this paper is to identify requirements and challenges in order to define a RaaS (Reputation-as-a-Service) framework. As trust and reputation systems can be very complex, the focus of this research is the definition of a service framework that provides similar but configurable functionality currently supported by central online reputation systems.

In the next section, trust and reputation is defined for this research. Five general components of reputations systems are given which is referred to throughout the paper. The requirements for a RaaS component is identified by considering SaaS configuration aspects, user requirements for trust and reputation and finally requirements from reputation frameworks. A RaaS framework is presented and the paper is concluded.

II. TRUST AND REPUTATION

Trust and reputation is present in a variety of online communities. Trust is the individual's perspective on a particular service or product and reputation is a group's perspective on a particular service or product [5]. As trust and reputation are concepts that are often used interchangeably, they are now defined for the purposes of this research.

A. *Trust*

Trust is challenging to define as it manifests itself in many different ways in varying contexts. Almost every aspect of daily life is supported by some form of trust. For example, in Figure 1, consumer X, the trustor, orders products from organisation ABC, the trustee. For this research, the following definition of trust is adopted. The trust of consumer X in organisation ABC is defined as the level of subjective probability that organisation ABC will deliver high quality products on time [6].

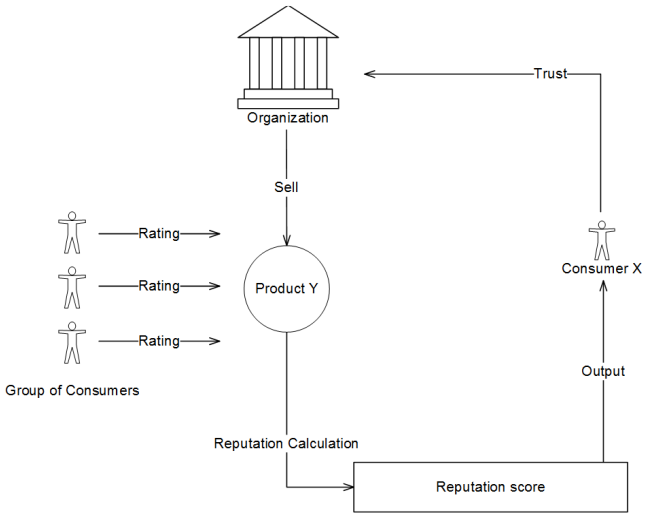

Figure 1. Trust and Reputation

The trust of consumer X in organisation ABC is affected by trust properties such as transitivity, subjectivity and the asymmetric nature of trust [7]. If organization ABC has the reputation of delivering high quality products, consumers automatically assume that any product of organization ABC is also of high quality due to the property of transitivity, suggesting that trust is transferable. But, as both consumer X and Z can have different levels of trust towards the same organisation ABC, trust is subjective. The asymmetric property of trust is defined by the fact that consumer X needs to trust that organisation ABC will deliver the necessary services, but organisation ABC needs to trust consumer X to pay on time.

Closely related to trust, is reputation. In the next section the concept of reputation is addressed in order to identify elements that it consists of.

B. *Reputation*

Reputation can be considered as a collective measure of trustworthiness [16]. In order to better regulate relationships in online communities, opinions about interacting parties' past behaviour can be collect and aggregated in order to define a summary evaluation, or reputation. In Figure 1, reputation is illustrated by a group of consumers' opinion on a specific product. The group of consumers in Figure 1 gives product Y a good rating over time; this ensures that the product will have a good reputation score [8]. In this paper the term "rater" is used to represent a participant or consumer who assigns ratings for others. Reputation is calculated by incorporating past experiences, direct experiences and recommendations using various algorithms and models for this purpose [9]. One party can trust another based on their "good" or "bad" reputation.

The five main components found in reputation systems and models are [10]:

1. *Gathering behavioural information* where direct experiences, and experiences of acquaintances of consumers, recommendations from others, transaction history, pre-trusted entities and raters reliability are collected.
2. *Scoring and ranking* of entities are done next resulting in a reputation score, computed using averages, fuzzy logic, or Bayesian networks.
3. *Entity selection* is done next using the reputation score and other utility functions as specified.
4. *Transaction* is carried out with the selected entity.
5. *Reward and punishment* is finally given by assessing the transaction and giving a rating.

Most current reputation systems are built using these common components, but for a specific context and application domain, using proprietary vocabularies [11]. Each defines its own method to query, store, aggregate, infer, interpret and represent reputation information. The next section investigates requirements for a RaaS framework to suit the needs of different communities.

III. REQUIREMENTS FOR A RAAS FRAMEWORK

In order to determine general reputation system requirements to define a RaaS framework, this research now reports on general SaaS on configuration requirements of RaaS applications to direct focus of further analysis, then requirements elicited by users for trust management [9], and finally an analysis of components of reputation systems to determine a comprehensive list of requirements for a RaaS component.

A. *SaaS application requirements for RaaS*

The focus of a RaaS model is to deliver software functions to many clients over the Web with a single instance of a software application running on a multi-tenancy platform [12]. This model lowers the cost of ownership and also supports higher levels of mobility for users over a web-based delivery model. However, every tenant that needs to use a RaaS supported with this model can be unique, requiring changes to the reputation system.

Tenants may have a different industry focus, their customer may behave differently, they may support diverse product offerings and have different regulations, organizational culture and operational strategy. These features require RaaS to be tailored, by leveraging two major approaches namely configuration and customization [12]. Configuration does not involve source code change of the RaaS application and

support differences through setting pre-defined parameters, or leveraging tools to change application functions within pre-defined scope, such as adding data fields, changing field names, modifying drop-down lists, adding buttons, and changing business rules. On the other hand, customization involves RaaS application source code changes to create functionality, leading to a more costly approach for both SaaS vendors and clients.

There are seven fundamental configuration and customization requirements that can be tailored, to make the RaaS component as flexible as possible [12] namely:

- Support for different organization structures require the ability to add, delete and changes roles.
- Support for different types of data can be made possible by adding custom fields and types, and deleting data not needed.
- Support for different processes requires tasks to be switched, added and reordered and their roles to be changed.
- Business rules can be modified by changing or setting rules and the rule triggers.
- Reputation computations can be made more generic by adding or changing actions or triggering actions at different points.
- The user interface can be changed with respect to the look and feel, the data presented and the addition of data.
- Reporting can be changed with respect to style, dataset used and query rules.

In summary, the RaaS should be developed to have standardized software features to serve as many clients as possible using a configuration approach. The RaaS developer needs a strategy to enable self-defined configuration by their tenants without changing the SaaS application source code for any individual tenant [12]. The RaaS environment needs to be thoroughly analyzed to determine the common configuration requirements. In conjunction, a sophisticated web based tool is needed to allow clients to configure the RaaS service themselves.

The next step is to investigate user requirements for trust and reputation systems.

B. User trust and reputation requirements for RaaS

Previous research [9] collected formal user requirements for trust and reputation systems from system developers. It was found that these users required a clear, layered and pluggable architecture for representing the calculation process of the trust score. Categorized user requirements were found to be closely coupled with the previously discussed five components found in reputation systems [10]. User needs for each of the components were identified as follows:

1) Information Gathering

- The success of each interaction needs to be rated and quality parameters continuously monitored.
- Simple and intuitive rating scales should be used.
- Services from providers with good reputation should be consumed.
- The quality parameters of a service should be controlled and certified by a trusted party; ratings of such a party can be used as a starting point for trust computation.
- Raters reliability must be controlled as they could provide dishonest ratings.
- An initial rating should not influence or bias subsequent votes.
- Use recommendations if recommenders preferences are similar to own preferences.
- Over time trust values decay and become invalid.

2) Scoring and Ranking

- The need is for a single trust rating calculated by taking into account different service aspects and their weights.
- The computation should be an aggregation of all weighted aspects, similar to an "average".

3) Entity Selection

- When services are selected, they should be sorted according to their trust rank and providers should be made comparable to each other.

By considering such a user-centered design approach, the proposed RaaS component can be created to fulfill the needs of users as far as possible.

RaaS framework requirements are discussed next.

C. Reputation system framework requirements for RaaS

The focus of this section is to identify major characteristics of reputations systems to identify requirements for the RaaS component. In order to achieve this, an adapted framework is defined from the work of others [13] [14] [15]. There are 8 elements which are discussed following the phases of reputation management components from information gathering, to scoring and ranking and entity selection. The elements, shown in Figure 2 include:

1) Network architecture
2) Information gathering
3) Inputs
4) Rating approaches
5) Incentives
6) Reputation measurement parameters
7) Reputation computation engines
8) Rating score

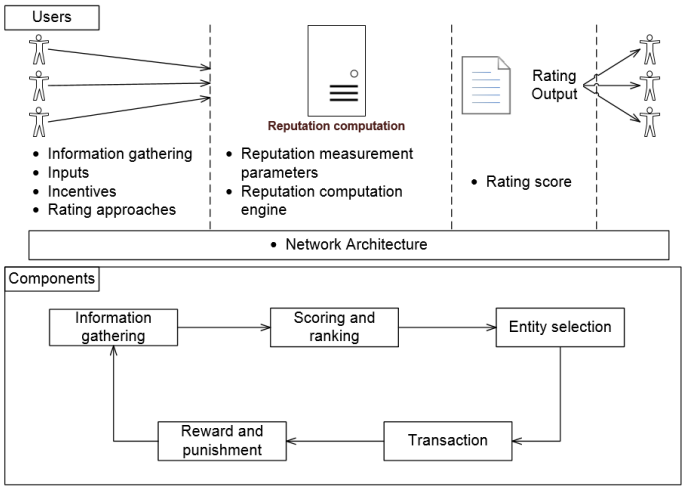

Figure 2. Reputation manager framework

Figure 2 shows how the 8 elements at the top of the diagram fit in with the 5 components at the bottom.

1) Network architecture

The network architecture of a reputation system can either be a centralized, decentralized or hybrid architecture [16]. The network architecture determines how information is gathered and stored. For this research, the RaaS component needs to follow the cloud architecture, thereby limiting the scope of architecture choice. In a centralized architecture, all data is stored in a central repository with all reputations scores publicly available to participants. For distributed or hybrid reputation systems, there is no central point where ratings are submitted or feedback can be obtained. Instead, each participant is given the responsibility to collect ratings from others. For a RaaS component this is not a viable option as the cost and complexity level would be too high. A centralized architecture is simple and cost-efficient, and conforms to the RaaS and user requirements identified previously. This choice directly influences the discussion of the next elements, as they need to comply with this requirement.

Next the information gathering stage is discussed. For this research, a transaction between two participants is the basis of a rating. Generally, a participant cannot rate another one without having had a transaction with him. After a transaction, participants usually have no direct incentive for providing rating about the other party. The information gathering phase should be carefully designed to address this issue.

2) Information gathering

The information gathering phase collects rating inputs over a period of time. There are a numbers of important aspects to consider such as the collection channel, the information sources, the type of reputation information and collection costs.

Collection channels can be direct or indirect. Direct channels collect information from raters just after the transaction, by sending emails asking them to do a rating or by using a 3rd party for rating collection. Indirect channels collect information from other reputations systems, increasing complexity of information gathering.

Most online reputation systems consider reputation ratings from a global perspective. For example, eBay's feedback forum provides feedback profiles of sellers and buyers publicly to the all users. The shortcoming of such a global view is that these values lack personalization [5].

Information can be gathered from past experiences, direct experiences and recommendations [19]. The gathered information plays a major role to calculate a reputation score for a particular user or product. This information might come from several sources such as direct experiences with the targeting entity, neighbors of participants, acquaintances, the group the participant belongs to or organizations. In this regard it is important to consider the set of raters, their expertise and credibility [23].

A sufficient number of raters who rate transactions can help a reputation system to avoid personal bias whereas a restriction on the number of raters may influence level of detail between raters and objects being rated. A reputation system can be defined to have no restrictions on the number of raters leaving ratings, which means anyone can rate; or only registered participants can provide a rating; or only some registered raters can provide a rating after a transaction has finished such as eBay allows. It should not be allowed to rate a transaction or object more than once, for example, in eBay if buyers and sellers transact, the reputation system will only allow one rating per transaction to avoid the manipulation of the reputation score.

Directly related to the number of raters who rated objects is granularity [24], which indicates if the model is context-dependant or not. As raters may have a good reputation for their expertise in one domain, and a low reputation for another, granularity identifies how information sources associates to the reputation object. When a system allows any raters to do a rating, the granularity is usually very loose. If a reputation system requires information sources to have a good credibility to leave reviews this increases the cost for a rater to provide a rating which in turn reduces the number of invalid ratings.

The reputation of the rater should be considered by having other participants to give feedback on those ratings. Some reputation systems have a ranking mechanism for their users, called the 'Karma' mechanism that records every action of a user and gives points to it [15].

Finally, the input collection costs should be considered. This is the cost that indicates how much time it takes to collect a single unit of reputation information, where collection channels can have an important effect on this cost [23].

Next, the type of information source is described.

3) Inputs

Different information formats can be chosen based on the way in which they will be used in a reputation system. Some reputation systems support arithmetic operations and other evidence where numeric quantification is more appropriate. It

can also be possible to provide a mapping from qualitative to numeric labels. For example, ratings such as a score between 0 and 10 can easily be aggregated to an overall score, to give a comparable value between reputation objects. On the other hand, text reviews contain detailed information which can be very useful.

Generally, a rating can be expressed as either a quantitative or Boolean format [13]. A quantitative metric is a measurable input such as a value between 0 and 10 whereas a Boolean format is either 0 or a 1 to represent "like" or "dislike". As it is important that the reputation score is useful to the community where it will be used, the RaaS can be configured for this purpose.

In order to ensure the completeness of ratings collected, rating approaches are discussed next.

4) Rating approaches

A larger variety of rating information can give a better view of a reputation object as it provides a more complete picture. For example, travel reputation systems can allow participants to rate hotels for their value, rooms, location, cleanliness and service separately [23].

In single-criterion rating systems or binary rating systems, participants reveal their general opinion with regards to a reputation object, resulting in reputation information that is not too reliable and accurate.

In systems where multiple-criteria can be used, better quality reputation scores can be defined. A set of criteria needs to be defined and a rating is provided for each. This can allow a participant to choose a partner based on specific criteria that matches his own. On the down side, many rating criteria may reduce the evaluators' motivation on leaving ratings. This can be overcome by making some criteria optional to rate.

Next the role of incentives in information gathering is discussed.

5) Incentives

Raters of a reputation system may have different motivations for providing ratings. Incentives are important as their absence drives only some of the users to voice their opinions and report feedback where those with a moderate outlook are unlikely to provide ratings [24]. This results in an unrepresentative sample of ratings and opinions. For example, reputation systems have incentives for raters such as sellers to behave honestly in order to be chosen by buyers as this can increase their profit through the increased amount of transactions. These incentives are necessary because fabricated ratings can promote specific sellers or to discredit others - e.g. authors can write fake reviews on Amazon in order to boost the sale of their own books. In order for RaaS to be implemented successfully, the motivations for providing a rating should be identified.

There are various types of motivations [15] such as altruistic motivation which is in favour of doing good to users being rated and can be classified as tit-for-tat, friendship and exploiting opinionated incentives. Commercial motivation, is used to generate revenue and is categorized as direct revenue incentives and branding incentives. Egocentric motivation is used for self-gratification and is categorized as fulfilment incentives and recognition incentives.

By explicitly rewarding participants for reporting feedback, rewards made by the reputation systems must cover the cost of reporting feedback to encourage more participants to report, giving a more representative set of ratings. In addition, rewards must be designed so that selfish participants are convinced to rate truthfully to advance themselves [24].

The next section now considers the next reputation component namely the scoring and ranking of ratings. Here, the reputation computation engine and rating approaches are discussed.

6) Reputation computation engines

One of the most critical features of a reputation system is the reputation computation aggregation algorithm. Such an algorithm integrates ratings into one score, and at the same time needs to ensure that bad raters are identified and removed to obtain accurate ratings. There are many complex aggregating algorithms that have been proposed such as fuzzy models and Bayesian systems.

Currently, most online reputation systems as eBay and Amazon choose to use simple algorithms [16], such as summation, average or percentage. Simple summation adds all of the ratings, regardless if it is positive, neutral or negative and the calculation is easily understood and adopted by users [13] [16]. Unfortunately, this feedback metric is flawed, for example, if a user has 10 positive feedback points out of 10 transactions and another has 20 positive and 10 negative feedback points out of 30 transactions, they would have the same reputation score [5].

Average rating is based on the same principle as simple summation, however average rating is perceived as more accurate. Ratings can also be calculated by means of weighted average ratings. This infers that each user has a credibility score that determines their weight ratio [5]. Many interesting aggregating algorithms have been proposed that can be classified into five categories [25].

- By averaging ratings, simplicity in algorithm design is ensured and low cost in system execution.
- Weightings are introduced by weighting the ratings of acquaintances but those of strangers are averaged.
- Only ratings from witnesses are used, who have interacted with the entity being rated. In such a weighted majority algorithm only the ratings from witnesses are aggregated, and the weight of witnesses is decreased if it differs from self own recognition.
- Here, the weight of ratings is based on the similarity of the experience between the rater and the other participant to improve accuracy.
- Ratings can be aggregated and weights of raters can be updated through deriving the expectation of the Beta distribution.

In Simulation it was found that most complex algorithms will have better results. However, in several circumstances the simple algorithm can outperform the complicated algorithms. In particular, the first average algorithm is found to be more resistant to different type of bad raters [25].

To configure the reputation aggregation algorithm for a RaaS, one of these aggregation algorithms can be chosen as they may be able to accommodate a variety of communities and would be understood and adopted by users [5].

7) Reputation measurement parameters

There are crucial parameters which may increase the accuracy of the expected reputation score namely transitivity rate and time [13].

Transitivity rate represents the fact that recommendations from third-hand ratings with a transitivity degree of three may have the least influence on the trustworthiness measurement. Therefore, in a recommendation chain, recommendations from known participants who already have had interaction with the requested party should have more weight as first-hand recommendations, than those who are known but have not had any previous interactions with the requested party or those who are unknown.

Time influences the effect of ratings on the computation as the most recent rating will have a higher weight ratio than ratings that are older. Thus ratings decay over time. The advantage is that users benefit from having a rating value that reflects how the most recent services performed. These parameters attempt to ensure that ratings are more accurate as weight ratios are an effective way to counteract "bad" raters.

Finally, the entity selection component is considered where the resultant reputations is now used.

8) Rating score

The reputation system finally reports its results to users in two different formats namely aggregated reputation scores and individual ratings and opinions [23]. Reputation scores are the result of the scoring and ranking component, whereas the individual ratings are collected through the information gathering component.

When reputation scores are presented, the time line it represents should be provided to assist users with decision-making. Reputation information is disseminated to end users via different access methods such as web sites, emails or RSS (Rich Site Summary) feeds. Certain information may be made publicly available, whereas others may require a subscription fee.

Next a summary is given of the requirements for RaaS.

D. Summary of requirements for RaaS

Table 1 briefly summarizes the most relevant requirements for a RaaS framework. The requirements are given according to the first three components of reputation systems. Where a high cost is associated with a set of requirements, it has been indicated.

The first column addresses the input to the system or information gathering. From the three sets of requirements, the design of the RaaS can be driven by considering the configuration of input data such as roles and data types, the rating of transaction by considering who can rate in which circumstances and aspects such as collection channels and context and incentives. A high cost factor is the collection channels used to source ratings and feedback.

In the scoring and ranking component reputation computations are performed and can be configured with respect to aspects such as tasks executed, choice of algorithms, rules and weightings of criteria. This is not a trivial to apply and it will be associated with a high cost as it is very complex in nature.

The result of the reputation computation is used in the entity selection component, where aspects such as reports and ranks are provided to end users.

	INFORMATION GATHERING	**SCORING AND RANKING**	**ENTITY SELECTION**
SAAS APPLICATION REQUIREMENTS	• add, delete and changes roles • custom fields and types, and deleting data not needed • user interface look and feel, data presented.	•tasks switched, added and reordered and task roles to be changed •business rules modified by changing or setting rules and the rule triggers. •adding or changing actions or triggering actions at different points. **(HIGH COST)**	• reports changed with respect to style, dataset used and query rules.
USER TRUST AND REPUTATION REQUIREMENTS	• each interaction needs to be rated and quality parameters continuously monitored • simple and intuitive rating scales • initial ratings should be treated differently • the quality parameters of a service should be controlled and certified by a trusted party • raters reliability must be controlled • similarity between recommenders is important. • trust values decay and become invalid over time.	•a single trust rating calculated by taking into account different service aspects and their weights. •computation should be an aggregated of all weighted aspects, similar to an "average". **(HIGH COST)**	• services should be sorted according to their trust rank and providers should be made comparable to each other.
REPUTATION SYSTEM FRAMEWORK REQUIREMENTS	• direct or indirect collection channels • restrictions on who can rate • context-dependant ratings • collection costs • simple rating format • single/multiple criteria • incentives **(HIGH COST)**	• complexity of aggregation algorithm • weighting of recommendations • decay of trust **(HIGH COST)**	• Reputation score / individual ratings • time reported

Table 1. Summary of requirements for RaaS

Next RaaS architecture is discussed in light of the identified requirements.

IV. RAAS ARCHITECTURE MODEL

In this section a RaaS architecture model is discussed to illustrate how RaaS can be deployed as a cloud application. The architecture is discussed by only considering features of reputation services, and no SaaS business components such as billing and metering.

This research assumes that the RaaS will be integrated with the applications of the tenant. Users will not directly interface with the RaaS, but all interaction will be over standardised protocols such as SOAP (Simple Object Access Protocol) and REST (Representational state transfer) between applications. Such integration can be underestimated in complexity and effort as it is a very similar endeavour as integrating externally hosted systems.

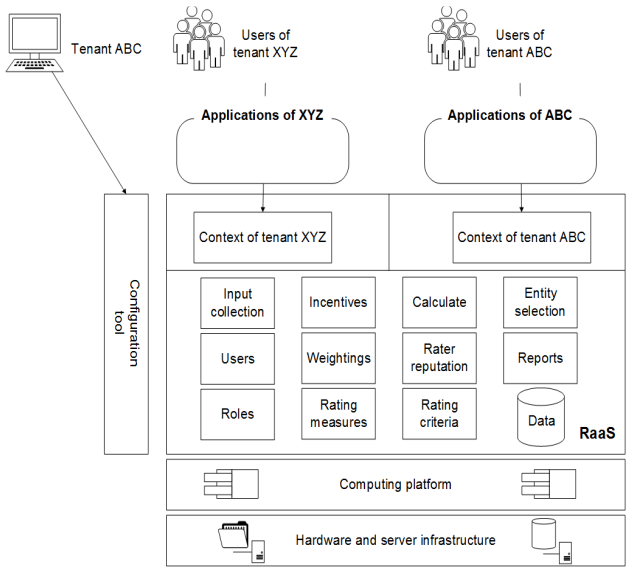

Figure 3. RaaS architecture

Figure 3 gives a basic architecture diagram of a RaaS component. The RaaS is defined over a basic cloud architecture found in data centres where hardware and software are used to define virtual machines that are provided to tenants to run their applications on.

A tenant such as the online store of organisation ABC needs to integrate the provided functions of the RaaS to support ratings, and display results in the web applications they expose to their customers who are using browsers or mobile applications. Communication between the RaaS and other applications is with REST messages [23].

In the diagram, it is shown how tenant ABC configures their features to suit their needs using the configuration tool. The configuration of features gives a unique experience to the users of tenants of the RaaS application, even though the code base does not change. To be cost-effective, customization of code should be avoided as far as possible. The configuration tool should be designed to be easy and intuitive for the administrators, but at the same time be able to satisfy the needs to tenant requirements. Without this feature, it would be impossible to use the single instance of the software for different tenant applications. The administrator needs to configure the RaaS component and define configuration data for the users of his organisation. This supports the context of interaction when the users of tenant ABC interact with the service, as shown in Figure 3.

The RaaS component is defined by a number of services shown such as input collection, incentives, users and roles. Each service contributes to the operation of the reputation system and is integrated using workflow. The configuration of the collection of ratings, their format and processing is no trivial matter, and much intelligence will be required to ensure that options are set that do not contribute to a true reputation score. Careful consideration should be given to aspects such as the user interface and the type of data that is exposed, the type of rating format is required by specific algorithms, whether weights can be set or not, which groups of raters may be granted the ability to rate, which objects can be rated and the number of criteria to be used.

Considering the above mentioned complexities in configuring reputation computation this research now proposes a two level reputation configuration approach, one for novice users, and one for knowledgeable users who understand the implications of their choices.

For novice users, there may be a few options available to select from such as:

- Simple reputation computation with basic summation of values.
- Reputation computation that encourages strangers by treating them lightly initially.
- Reputation computation that is strict with "bad" behaviour as the risk is high.

An advanced configuration panel may be made available to knowledgeable users to select a variety of options.

In both cases, the RaaS should make available a simulation feature that will illustrate to the administrator what the effect of the choice will be, in order to avoid any misunderstandings.

There are many challeges that stem from the creation of a RaaS framework:

- Configuring reputation computation and behaviour is complex. As workflows allow automation of processes involving human and machine-based activities it may be important to apply it in this context.
- Each tenant will have specific needs with respect to their data requirements. To address this, a template for storing data can be provided that meets most requirements, with options to add fields to tables.
- As tenants of the RaaS component have a large variety of users, and the responsibility for creating individual accounts for end users, and granting access to resources lies with the tenant, a well-developed access control component should be provided.

- The management of raters and other identities is complex. In most cases, user accounts are managed and stored independently by each tenant and authentication occurs within the organizational boundary. This means that the identity of the user, with any relevant credentials is sent to the RaaS to allow identification and access control. Different types of identities and credentials need to be managed.

V. CONCLUSION

The aim of this paper is to identify requirements for a RaaS framework. Here, this has been done only at a theoretical level. Although a comprehensive set of requirements have been identified, more research and analysis is still needed. To the best of our knowledge, this is the first attempt at identifying requirements for such a framework.

The need to create a reputation system that is configurable to accommodate a variety of online communities is plausible from the given scenarios. This research focussed on elements that are present in a reputation systems and how these elements can be configured. The RaaS framework requirements were identified from considering SaaS application requirements, user requirements and reputation framework requirements. A possible architecture has been defined to expose RaaS services.

Future work will address the creation of a RaaS solution. Simulations will be performed to experiment with the different configurable components to identify which elements are more appropriate to configure.

ACKNOWLEDGMENT

The support of SAP Research Pretoria and the National Research Foundation (NRF) under Grant number 81412 and 81201 towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the authors and not necessarily to be attributed to the companies mentioned in this acknowledgement.

REFERENCES

[1] Biz Community, 2012. MasterCard Worldwide reveals online shopping survey. [Online] Available at: http://www.bizcommunity.com/ Article/196/168/73927.html [Accessed 02 May 2013].

[2] Mulpuru, S., Johnson, C. & Roberge, D., 2013. US Online Retail Forecast, 2012 To 2017. [Online] http://www.forrester.com/US+Online+Retail+Forecast+2012+To+2017/ fulltext/-/E-RES93281?objectid=RES93281 [Accessed 16 May 2013].

[3] Wang, Y. & Vassileva, J., 2007. Toward Trust and Reputation Based Web Service Selection: A Survey. International Transactions on Systems Science and Applications (ITSSA) Journal, 3(2), pp.1-38.

[4] Braithwaite, F. & Woodman, M., 2011. Success Dimensions in Selecting Cloud Software Services. In 37th EUROMICRO Conference on Software Engineering and Advanced Applications., 2011.

[5] Li, H., 2007. A Configurable Online Reputation Aggregation System. Canada: University of Ottawa.

[6] Gambetta, D., 1990. Can We Trust Trust? In, Trust: Making and. Oxford, 1990.

[7] Huynh, T.D., 2009. A Personalized Framework for Trust Assessment. In SAC' 09. Honolulu, Hawaii, U.S.A., 2009. ACM.

[8] Fahrenholtz, D. & Lamersdorf, W., 2002. Transactional Security for a Distributed Reputation Management System. In Proceedings of the Third International Conference on E-Commerce and Web Technologies (EC-Web)., 2002. Springer.

[9] Vinkovits, M., 2012. Towards Requirements for Trust Management. In 2012 Tenth Annual International Conference on Privacy, Security and Trust., 2012. IEEE Computer Society.

[10] Mármol, F.G. & Pérez, G.M., 2010. Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems. Computer Standards & Interfaces, 32(4), pp.185-96.

[11] Alnemr, R., Schnjakin, M. & Meinel, C., 2011. Towards Context-aware Service-oriented Semantic Reputation Framework. In 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11., 2011. IEEE Computer Society.

[12] Sun, W. et al., 2008. Software as a Service: Configuration and Customization Perspectives. In IEEE Congress on Services Part II., 2008. IEEE Computer Society.

[13] Noorian, Z. & Ulieru, M., 2010. The State of the Art in Trust and Reputation Systems: A Framework for Comparison. Journal of Theoretical and Applied Electronic Commerce Research , 5(2), pp.97-117.

[14] Swamynathan, G., 2006. Reputation management in decentralized networks. [Online] Available at: http://www.powershow.com/ view/11e08- Ndu1z/Reputation Management_In_Decentralized Networks_powerpoint_ppt_presentation" [Accessed 18 March 2013].

[15] Farmer, F.R. & Glass, B., 2010. Web Reputation Systems. 1st ed. O'Reilly.

[16] Jøsang, A., Ismail, R. & Boyd, C., 2007. A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2), pp.618-44.

[17] Desouza, K.C., Jayaraman, A. & Evaristo, J.R., 2003. Knowledge Management in Non-Collocated Environments: A Look at Centralized Vs. Distributed Design Approaches. In Proceedings of the 36th Hawaii International Conference on System Sciences., 2003. IEEE Computer Society.

[18] Wang, Y., Hori, Y. & SAKURAI, K., 2006. On securing open networks through trust and reputation – architecture, challenges and solutions. In Proceeding of The 1st Joint Workshop on Information Security (JWIS). Seoul Korea, 2006.

[19] Mármol, F.M. & Pérez, M., 2009. TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks. IEEE.

[20] Liang, Z. & Shi, W., 2005. Performance Evaluation of Rating Aggregation Algorithms in Reputation Systems. In Collaborative Computing: Networking, Applications and Worksharing. San Jose, CA, 2005. IEEE Computer Society.

[21] Mell, P. & Grance, T., 2011. The NIST Definition of Cloud Computing (draft). NIST special publication, 145(800).

Oracle, 2013. http://www.oracle.com/technetwork/java/javase/jdbc/index.html. [Online] Available at: http://www.oracle.com/technetwork/ java/javase/jdbc/index.html [Accessed 19 May 2013].

[22] Fielding, R.T., 2000. Architectural Styles and the Design of Network-based Software Architectures. In Doctoral dissertation. University of California, Irvine, 2000.

[23] Liu L, Munro M, 2012, Systematic analysis of centralized online reputation systems, Decision Support Systems, Volume 52, Issue 2, January 2012, pp 438-449, ISSN 0167-9236

[24] R. Jurca., 2007, Truthful Reputation Mechanisms for Online Systems. PhD thesis, EPFL, 2007.

[25] Z. Liang and W. Shi, "Analysis of Recommendations on Trust Inference in Open Environment", *Journal of Performance Evaluation*, Elsevier 2007.