Chapter 14
Using Obstacles for Systematically Modeling, Analysing, and Mitigating Risks in Cloud Adoption

Shehnila Zardari
University of Birmingham, UK

Funmilade Faniyi
University of Birmingham, UK

Rami Bahsoon
University of Birmingham, UK

ABSTRACT

In this chapter, the authors motivate the need for a systematic approach to cloud adoption from the risk perspective. The enormous potential of cloud computing for improved and cost-effective service delivery for commercial and academic purposes has generated unprecedented interest in its adoption. However, a potential cloud user faces numerous risks regarding service requirements, cost implications of failure, and uncertainty about cloud providers' ability to meet service level agreements. Hence, the authors consider two perspectives of a case study to identify risks associated with cloud adoption. They propose a risk management framework based on the principle of GORE (Goal-Oriented Requirements Engineering). In this approach, they liken risks to obstacles encountered while realising cloud user goals, therefore proposing cloud-specific obstacle resolution tactics for mitigating identified risks. The proposed framework shows benefits by providing a principled engineering approach to cloud adoption and empowering stakeholders with tactics for resolving risks when adopting the cloud.

DOI: 10.4018/978-1-4666-2199-2.ch014

1. INTRODUCTION

The ever increasing need for data processing, storage, elastic and unbounded scale of computing infrastructure has provided great thrust for shifting the data and computing operations to the cloud. IBM advocates cloud computing as a cost efficient model for service provision (IBM, 2008). The adoption of cloud computing is gaining momentum because most of the services provided by the cloud are low cost and readily available. The pay- as-you- go structure of the cloud is particularly suited to Small and Medium Enterprises (SME) who have little or no resources for IT services (Biggs, Vidalis,2009).

The growing trend of cloud computing has led many organisations and even individuals to move their computing operations, data, and/or commissioning their e-services to the cloud. Moving to the cloud has reduced the cost of computing and operations due to resource sharing, virtualization, less maintenance cost, lower IT infrastructure cost, lower software cost, expertise utilization and sharing etc. (Miller, 2008). For example, the New York Times managed to convert 4TB of scanned images containing 11 million articles into PDF files, which took 24 hours for conversion and used 100 Amazon EC2 Instances (Gottfrid, 2007). Such relatively quick conversion would be very expensive if done in-house. The term cloud computing may simply refer to different applications over the Internet or the hardware shared between different users (Armburst et al, 2010). Buyya et al have defined cloud:

A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service level agreements established through negotiation between the service provider and consumer (Buyya et al, 2008).

In a cloud, hardware/software are shared and utilized as services at lower cost. Many services are now offered in the realm of cloud computing. These are:

- **Infrastructure as a Service (IaaS):** A model in which an organization outsources the equipment required to perform operations like storage, hardware, servers etc. Cloud service provider provides all the hardware needed for operations and is responsible for maintaining it. The client pays for what he uses. Amazon's Elastic Compute is an example of such a service.
- **Platform as a Service (PaaS):** Cloud Service Provider provides a platform to the user on which a user can develop an application. The applications are delivered to the users through cloud service provider's infrastructure. Coghead and Google App Engine are examples of PaaS.
- **Software as a Service (SaaS):** Delivers a single application through the browser to thousands of users. Users are not required to invest on purchasing servers or software licensing. Payment is made on the basis of the data transferred and some fixed rent. Google App Engine is a representative example of SaaS.

This chapter is structured as follows. We motivate the need for a requirements engineering framework for cloud adoption in Section 2. Risks that were identified from different cloud service providers' SLAs are presented in Section 3. Section 4 introduces goal-oriented requirements engineering for the process of cloud adoption. We define obstacles in Section 5 and argue that obstacle analysis should be part of the lifecycle for cloud adoption process. We have modelled a case study with two different perspectives (user and cloud service provider) using goal-oriented approach in Section 6. We have proposed some obstacle resolution tactics which could help in mitigating the risks associated with cloud adoption in Section 7. In Section 8, method for prioritizing goals using the utility theory is explained. Section 9 provides some insight into the related work. We discuss the open problems in Section 10. Section 11concludes the chapter.

2. BACKGROUND AND MOTIVATION

We looked at a case study of a leading cloud service provider- referred to as *Indus* throughout this paper. The case study revealed that there are many risks associated with cloud adoption. Since the cloud is perceived as a “black box”, a user has little or no control over the promises set by cloud’s SLAs. For instance, the user mostly cannot negotiate the SLAs with the cloud service provider and hence has to agree with the set terms and conditions.

As an example, Indus sufficiently describes their security mechanisms in the agreement accompanying their SLAs “Indus Web Services Customer Agreement.” Interestingly enough, Indus mentions that the nature of communication over the Internet is unpredictable and largely insecure. Given the vulnerability of the Internet, Indus cannot guarantee the security of user’s content. While Indus strives for a secure environment, the security responsibility and accountability lie solely on the users and the organisation using the services. In the event of any breach in security requirements, Indus is not entirely liable to the user for any unauthorized access, use, deletion, corruption or destruction of user’s contents. The service provider has attempted to win the confidence and trust of the users by publishing the agreement, yet it has failed to ascertain the individual needs of different users. Indus has a shared responsibility environment for the safety of user’s contents, where one of the inherent problems with such “joint” responsibility is that it makes accountability difficult. Referring to SLAs terms and conditions, Indus absolves itself of any responsibility in the wake of anything goes wrong. For instance, Indus recommends customers to encrypt data transferred over the network. There are tradeoffs involved with large data encrypted over the network: this will lead to higher processing time, which might affect cloud performance and consequently violate the promises set in the SLA. In fact, sending a disk containing the encrypted data by post has been stated to be more effective when the data volume is large (Ambrust et al, 2009). Such transfer of disks through post involves a lot of human efforts and therefore the system cannot be termed as automated. Despite the promised dynamic elasticity of cloud architecture, resources continue to be scarce. E.g. enhancing security provision may cause performance bottleneck. As a result, the promised Quality of Service (QoS) can’t be often met as the SLA terms and conditions stipulate.

We call for a novel requirements engineering methodology for cloud adoption, which could assist businesses in screening, selecting cloud service providers and negotiating their services and qualities of provision. The framework aims at helping businesses screen, match, and negotiate their requirements against cloud services’ provision. The framework will also assist in the problem of managing the tradeoffs associated with matches and mismatches of users’ requirements against cloud’s provision. Such provision aims at objectively evaluating the strategic decisions, satisfaction of the technical and operational goals, cost and value of such decisions and the tradeoffs involved in moving to the cloud. Despite rapid growth of cloud use, there is a general lack of systematic methodologies aiming at its adoption. Decisions regarding the selection of cloud service providers are made on ad hoc basis based on recommendations or on the reputation of the service provider. The lack of such methodologies exposes businesses considering the cloud to unpredictable risks. It would be expensive to get “locked in" with a wrong cloud. Evaluating pre adoption choices at early stages is cost-effective strategy to mitigate risks of probable losses due to wrong or unjustified selection decisions. Furthermore, the framework aims at assisting users in assessing their requirements against cloud provision. Due to the dynamic evolution of the cloud, mismatches may occur between what is required by the user and what is provided by the cloud provider. The framework will assess the suitability of cloud service providers by exploring mismatches, managing risks, and suggesting possible tradeoffs. We use goal-oriented approach for modelling cloud provision. The expected beneficiaries of the work are small to large businesses, educational institutes and even individuals, who wish to exploit the cloud. Such work is novel and bridges an important gap in making the process of cloud adoption more transparent, systematic and user oriented. It would be worth noting that ongoing research on cloud selection has addressed the problem of dynamic selection of the cloud services with respect to QoS (Goscinski, Brock, 2010). (Khajeh-Hosseini et al, 2010) is a recent research to facilitate users in cloud migration but this research doesn't involve refining and elaborating user requirements. We proposed a lifecycle based on goal-oriented requirements engineering for cloud adoption in (Zardari, Bahsoon, 2011). This book chapter is an extension of our previous work. We have refined the cloud adoption lifecycle and have introduced the notion of obstacles. We have also defined some concrete and abstract obstacle resolution tactics.

3. RISKS IDENTIFIED FROM DIFFERENT CLOUDS AND THEIR IMPLICATIONS

Risks that were identified by looking at the SLAs of different cloud service providers are shown in Table 1. The information presented in Table 1 has been collected after inspection of SLAs and white papers of several leading cloud providers such as Amazon Web Services, Google App Engine and Microsoft Azure. Risks could be classified into various areas of concern (e.g. network, environment etc). For each identified risk, we provide an impact assessment of the occurrence of the risk. We believe that the identified areas, potential risks and impact assessment is representative of the state-of-the-art in cloud service provision.

Although, cloud service providers have tried to win the confidence of the users by either publishing their whitepapers, web service agreements or by providing the details regarding their cloud service on their website yet most of the cloud service providers have failed to take the responsibility of the users' contents in case of provider's inability to live up to promises to the user. SLAs of most of the cloud service providers absolve them from any responsibility should anything happen to users' contents. Signing a contract with such cloud service providers can be a risky business. It is evident from Table 1 that many risks will eventually lead to financial losses for a business. To minimise the risk of being locked-in with a cloud provider who is unable to meet the needs of the users; the proposed framework informs cloud provider selection on a systematic and sound framework.

4. CLOUD- BASED GOAL ORIENTED REQUIREMENTS ENGINEERING

The specification of users' requirements is the first step towards selecting an appropriate cloud service provider. Boehm estimates that the late corrections of requirements errors could cost 200 times as much as corrections during requirements engineering phases (Boehm, 1981). Though the Boehm's argument is related to the case of projects and software developed for relatively defined set of requirements, it would be argued that late rectification of unwise selection may lead

Table 1. Risks identified from cloud service providers SLAs

Area	Description of Risk	Impact Assessment
Network	IP spoofing Port scanning Packet sniffing	Data leakage Financial losses Customer trust
Environmental	Fire Electrical Failure Temperature inside the data centre	Hardware damage Service delivery Customer trust Business reputation Huge financial losses
Technical	Integration Risk Insecure interfaces and APIs Shared technology issue Weak encryption of data in transit Incomplete data deletion Resource exhaustion	More investment on company's infrastructure Huge financial losses Data leakage Company's reputation Customer's trust Service delivery
Physical Access	Unauthorized access to the data centre Malicious insider	Financial losses Customer's trust Business reputation Data leakage Hardware damage
Business Risks	Compliance challenges Natural disasters Data Loss or leakage Unknown risk profile Malicious insiders SLA clauses containing business risks Incomplete data deletion Long term viability Interoperability Resource exhaustion	Industry certifications Company reputation Customer trust Closure of business High financial losses Going bankrupt Company's secret made public Financial losses Service delivery
Legal	Risks from change of jurisdiction SLA clauses containing business risks Loss of governance Unknown risk profile Data Recovery and investigation	Financial Losses Customer trust Business reputation Service delivery

to higher cost, which could be disproportional to the benefits of using the cloud and will place such investment into risk.

Our approach starts with high-level goals. Initially when any organization decides to move to the cloud, the goals are general expression of the requirements, whether these are strategic requirements, functional and/or non-functional. The higher level goals (such as secure payment transactions) are more stable than low level ones (such as Secure Socket Layer (SSL) Technology). Defining goals is the first step of requirements elicitation in software engineering. Pamela Zave in her research defines requirements engineering as something which is "concerned with the real world goals for functions of and constraints on software systems" (Zave, 1997). A goal is a target that a system must achieve. Goals are generally more stable than the requirements to achieve them (Anton et al, 1994). Goals cover different types of issues - both functional and non-functional. The functional goals can be associated with the services to be provided whereas non-functional goals can be associated with the quality of service e.g. security, reliability, availability etc (Lamsweerde, 2001). Goals represent the roots for detecting conflicts among requirements and for resolving them in time (Lamsweerde et al, 1998). GORE uses goals for eliciting, elaborating, analysing, documenting, modifying and refining the requirements (Lamsweerde, 2004). We specify the requirements of the stakeholders in the form of goals.

To make a sensible decision for selecting a cloud service provider, businesses require a methodology which could help them make a choice. In conventional software development, the requirements engineering basically consists of eliciting stakeholders' needs, refining the goals into non-conflicting requirements, followed by validating these requirements with stakeholders (Nuseibeh, Easterbrook, 2000). The main objective of the requirements engineer is to ensure that the requirements specifications meet stakeholders' needs and represent a clear description of the cloud services that are to be adopted. Stakeholders' requirements play a deciding / leading role in cloud service provider's selection.

5. OBSTACLES FOR MITIGATING RISKS IN CLOUD ADOPTION PROCESS

We recently proposed a methodology which could help users to adopt cloud services (Zardari, Bahsoon, 2011). The methodology is grounded in Goal-Oriented Requirements Engineering. The initial work did not provide the means to resolve obstacles for achieving goals. Requirements engineering is concerned with elicitation of high-level goals that are to be achieved by the system-to-be. The high-level goals are refined into low level goals which can be operationalised. Classically in GORE we operationalise goals by assigning them to the agents such as humans, devices and software (in our case, the agent can be either cloud service provider or any particular service). Sometimes the requirements engineering process results in goals and assumptions about the agent behaviour which is too ideal; some are unlikely to be satisfied by the system. We present techniques to resolve the obstacles in achieving goals. We are defining a number of abstract and concrete techniques for resolving obstacles which could mitigate risks in the cloud adoption process.

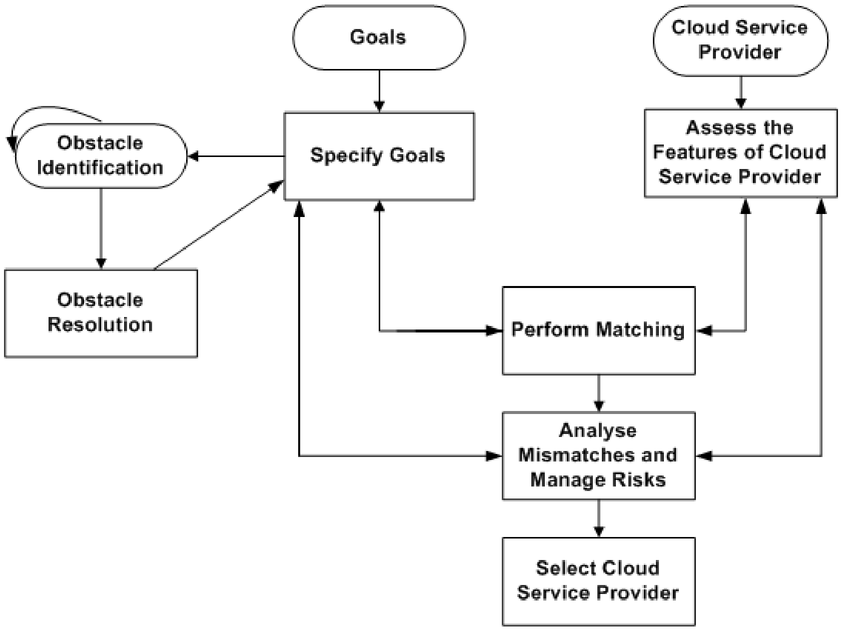
5.1 Obstacle Analysis

First sketch of goals tend to be too ideal (Letier, Lamsweerde, 2004). Such over ideal goals can be violated from time to time due to unexpected behaviour of agents. In KAOS, such exceptional behaviours are called obstacles to goal satisfaction. Obstacles were first proposed by Potts in (Potts, 1995). He identified obstacles for a particular goal by asking different questions. For example: "Can this goal be obstructed, if so, how?" According to Anton (Anton, 1996) while obstacle denote the reason why a goal failed, scenarios determine concrete circumstances under which the goal may fail. KAOS provides well-developed methods for identifying and resolving obstacles (Lamsweerde, Leiter, 2000). Lamsweerde et al recommend that obstacles be identified from leaf-level goals Once identified obstacles can be refined like goals (by AND/OR decomposition). Obstacles can lead to risks. Obstacle resolution will eventually result in minimising potential risks.

5.2 Obstacles in Requirements Engineering Process for Cloud Adoption

We found that there are some obstacles in achieving goals. We have therefore refined the lifecycle for cloud adoption (Zardari, Bahsoon, 2011) by introducing obstacle identification and resolution in the lifecycle (see Figure 1). During the Specify Goals phase the goals are refined and obstacles are generated. Obstacles can be repeatedly refined. The generated obstacles are resolved which result in updating goals. New goal specification may result in new iteration of goal refinement and obstacle identification. It can therefore be said that obstacles analysis and resolution has to be a continual process. Figure 1 shows the life cycle for

Figure 1. Lifecycle for cloud adoption


Goals
Cloud Service Provider
Obstacle Identification
Specify Goals
Assess the Features of Cloud Service Provider
Obstacle Resolution
Perform Matching
Analyse Mismatches and Manage Risks
Select Cloud Service Provider

cloud adoption. An obstacle from the perspective of a cloud user can be defined as any hindrance in achieving any goal or the desired properties as set out by the cloud user. Obstacles obstruct the goal such that when the obstacle is true then the goal may not be achieved (Lamsweerde, Leiter, 2000). Suppose a social networking website hosted on cloud wants 24/7 availability. If there is a power outage at the data center of the cloud service provider then the service may not be available and hence power outage will be an obstacle in achieving the goal of 24/7 availability.

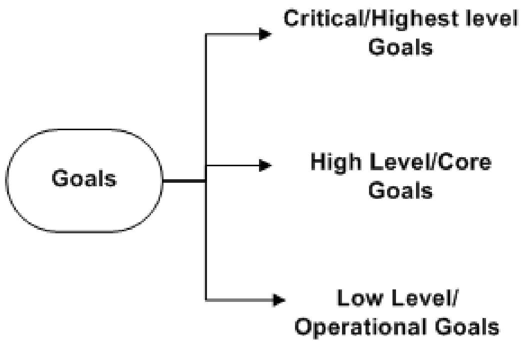
We are briefly describing the steps involved in the lifecycle shown in Figure 1. For details read (Zardari, Bahsoon, 2011).

Acquire and Specify Goals

Requirements act as criteria to evaluate cloud service providers. The goals can be divided into three categories (i) strategic or business goals, (ii) high level or core goals, (iii) and low level or operational goals. Strategic goals are concerned with the survival of the enterprise/business. Assume that University of Birmingham (UoB) wants to outsource its email services to the cloud in order to reduce cost by 20% in next five years; this is the strategic/business goal of UoB. High Level goals are nothing but core goals described in requirements engineering phase for cloud adoption. Suppose UoB wants that the cloud should be able to handle 500 users during the peak time and provide high level of security; these are high-level goals to be achieved. Operational goals can be encryption of data, password protection, and backup mechanisms etc. Sometimes, operational goals are required to be assessed and evaluated against high level core goals in certain cases. See Figure 2.

We will have to define the acceptance level for each goal as these cannot be completely met; instead they are satisfied to a certain degree within acceptable limits (Mylopoulos et al, 1992). The acceptance interval ranges from target level i.e. the optimum value that users consider to be fully satisfied, to the worst level, i.e. the lowest value of the acceptable interval in which the goal starts to be considered unsatisfied. The goals need

Figure 2. Goal categories



to be prioritized once they are specified, users have to engage in an extensive prioritization process in order to distinguish core goals (i.e. critical needs that should always be satisfied) from desirable goals (i.e. the ones that could be traded off). Section 8 describes the prioritization process for goals.

Obstacle Identification

During the Specify Goals phase obstacles are generated. The obstacles can be recursively refined. Obstacles are identified from the goal graph. It is suggested that obstacles be identified from the terminal goals that are assigned to agents. It is essential that the set of identified obstacles is complete for every goal (at least for high-priority goals). To resolve obstacles we would like to identify as many obstacles as possible for the goals- particularly for high-priority goals. A set of obstacles $O_1, \ldots, O_n$ is complete for goal G if the following condition is true:

$$\{ \neg O_1,\ldots,\neg O_n\} \models G$$

This condition means that if none of the identified obstacles occur than the goal is satisfied. Figure 3 and Figure 4 show the obstacles identified from the case study.

Obstacle Resolution

Obstacle resolution tactics are applied to resolve the identified obstacles. The selection of a specific obstacle resolution tactic depends on the likelihood of the obstacle occurrence and the impact of such occurrence. The goal/obstacle loop may terminate once the obstacle that remain are thought to be acceptable without applying any resolution tactics. Table 2 shows some obstacle resolution tactics in the process of cloud adoption.

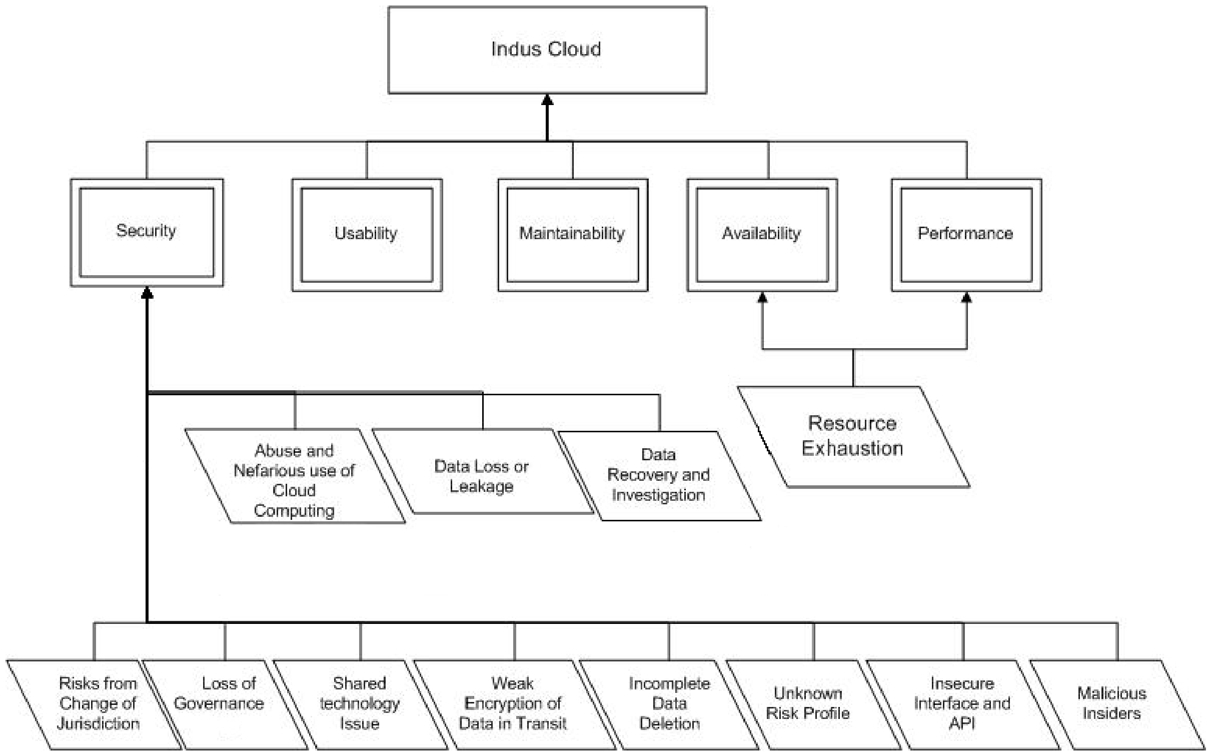
Cloud Service Provider

Once the generic goals have been defined the search for potential cloud service provider will start. At initial stage goals are kept generic so that the search for a cloud service provider is not limited by unnecessary constraints. At initial phase of cloud service provider identification it should be ensured whether the cloud satisfies the critical goal; total cost of ownership match the available budget, reputation of service provider with other users is compared and that the service provider is willing to participate in collaborative partnership etc.

Perform Matching

This phase involves gathering sufficient information about the cloud service provider and its services in order to assign the satisfaction scores to operational goals. Based on the results of matching the evaluation team can aggregate individual satisfaction score (i.e. how the cloud satisfies each operational goal) into global satisfaction scores (i.e. how the cloud satisfies the set of operational goals). Therefore, it is possible to compare cloud service providers and then, inform the decision making process. The matching process involves analysis of cloud service provider satisfaction and further discussion with users to determine whether or not a particular cloud sufficiently satisfies their needs. The matching of a particu-

Figure 3. Goal tree of obstacles to achieve security in an Indus Cloud



lar cloud is considered satisfactory if the cloud satisfies the operational goals within acceptable range. The information regarding features and service provisions of different clouds can be accessed through various sources like cloud white papers, SLAs, available benchmarks, the Internet, reviews, evaluations and recommendations of users, experiences etc. After looking at the features provided by the cloud service providers, the user may come up with new set of requirements that were not identified during the initial step. The examination of the cloud service providers' features is a better technique to refine and understand how high-level requirements of the stakeholders can be actually satisfied. As cloud services and SLAs are designed to satisfy the generic requirements of the market, some users' requirements may not be satisfied and therefore users should be prepared to engage in extensive process of requirements prioritisation and negotiation.

Analyse Mismatches and Manage Risks

A systematic approach is needed for the evaluation team to understand the effects of mismatches, analyse conflicts between goals, explore tradeoffs and manage risks. In the context of cloud service provider's selection, risks are defined as unacceptable outcomes, generally caused as a result of conflicting goals, mismatches and obstacles. Given that mismatches represent non-adherence of cloud to operational goals and conflict arise when satisfying one goal damages the satisfaction of another goal, we deduce that risks arise when the loss caused by unsatisfied goal is intolerable. A fundamental issue in handling mismatches is the capacity to systematically structure tradeoffs. The tradeoffs analysis forms the basis of risk management strategy. The objectives of risk management strategy are to understand and handle risk events prior to their conversion to threats. Risk

Figure 4. Portion of the goal tree for Smart Bank

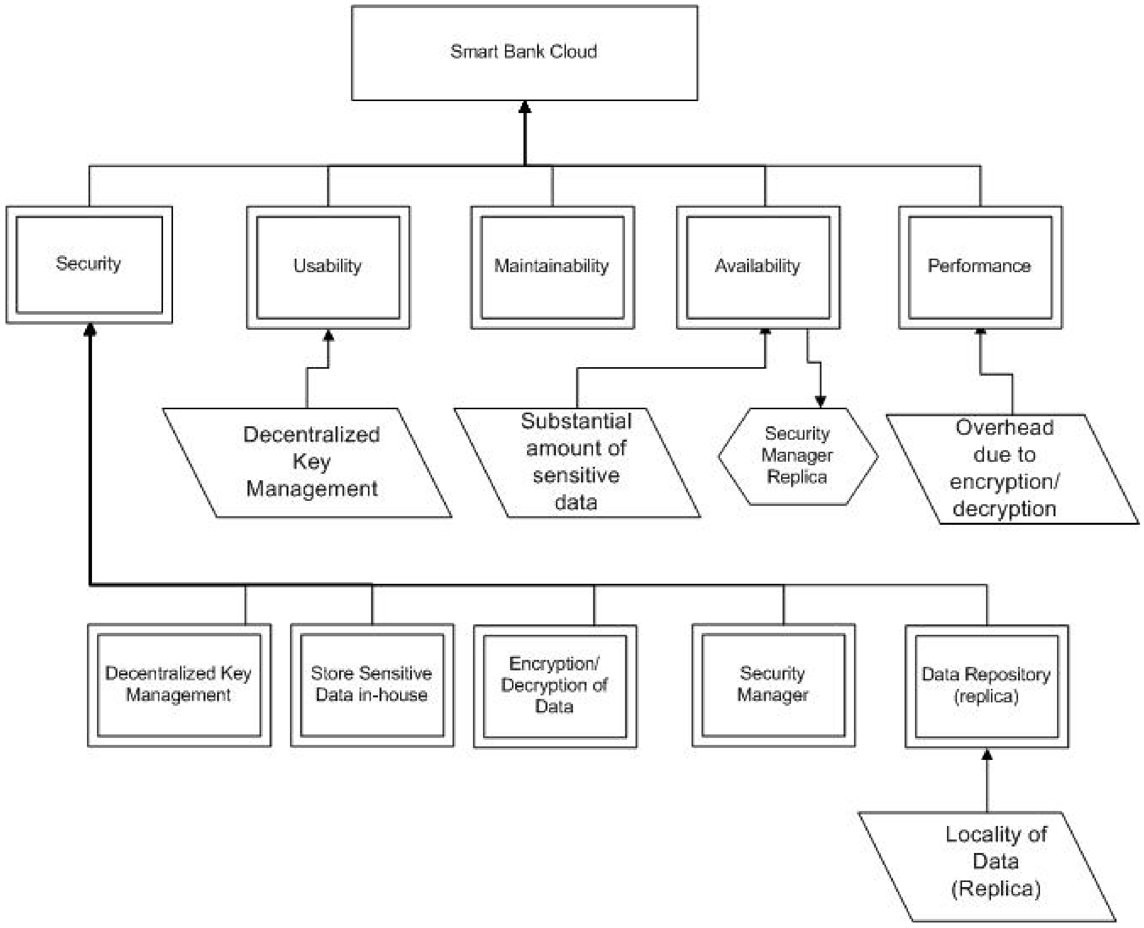

Smart Bank Cloud
Security
Usability
Maintainability
Availability
Performance
Decentralized Key Management
Substantial amount of sensitive data
Security Manager Replica
Overhead due to encryption/ decryption
Decentralized Key Management
Store Sensitive Data in-house
Encryption/ Decryption of Data
Security Manager
Data Repository (replica)
Locality of Data (Replica)

Table 2. Obstacle resolution tactics for cloud adoption

Category	Short Description	Obstacle Resolution Tactic	Example
Obstacle Prevention	Avoid the obstacle	**Tactic1:** Add another goal to prevent obstacle **Tactic2:** Improve/Negotiate SLA.	Figure 5 shows a goal which cannot be achieved if the data is stored outside home country. Figure 7 and Figure 8 show the resolution technique for such obstacles.
Cloud Service Substitution	Assign the responsibility of obstructed goal to another cloud	**Tactic3:** Transfer goal to another cloud **Tactic4:** Split goal among multiple clouds	Figure 9 show the obstacle and Figure 10 shows obstacle resolution by using tactic 3 Figure 11 show the obstacle resolution using tactic 4.
Goal Weakening	Weaken the obstructed goal specification	**Tactic 5:** Weaken goal objective function Relax requirements Relax the degree of satisfaction	Figure 12 shows the obstacle resolution using tactic 5.

management helps in successfully selecting and integrating the chosen cloud service provider. Risk management process has three steps: risk identification, risk analysis and risk mitigation. Risk mitigation action may cover options such as: change goals, negotiate cloud services features, or choose other alternatives. There is no one solution for all the risks; therefore, decision for the best risk mitigation strategy relies on the judgments and experience of the evaluation team.

Cloud Service Provider Selection

The ultimate objective of this phase is to choose the "optimal cloud." By optimal selection we mean that the selected cloud not necessarily needs to be optimal but satisfying. A satisfying cloud is the one that sufficiently satisfies the set of goals defined by stakeholders, where as an optimal cloud aims to maximize the satisfaction of goals at any cost. Factors to assess the worthiness of each cloud service provider for selection are value, cost, and risk. Cost represents the monetary investment, time and effort. The overall merit of a cloud based system is the combined measure of the value, cost, and risk. Value refers to ascertaining level of happiness of the stakeholders if a given alternative successfully satisfies the desired goals. Notably this is related to the priority stakeholders have assigned to goals. The risk is assessed using goal mismatches, conflicting goals and involves risk scenario elements.

6. CASE STUDY

We have modelled two perspectives (the cloud service provider and cloud user) of a case study. Figure 3 and Figure 4 show the goal tree of the obstacles that have been identified the case study. For the purpose of this case study we have adopted the following conventions. Double edged rectangles show the goals, parallelograms are the obstacles in achieving some of the goals and hexagons represent an agent. An AND-refinement is represented by an arrow with a small circle connecting the subgoals contributing to the parent goal. An OR-refinement is graphically represented by multiple arrows pointing to the same goal.

The objective of modelling two different perspectives of a case study is to show obstacles at the service providers' and the user level. Cloud service provider may have claimed about high security in their cloud but a close look at their whitepapers revealed that the risks associated with the adoption of such cloud are imminent. After looking at the Indus Web Services Customer Agreement and Indus white papers we identified many risks associated with adoption of Indus cloud. For the purpose of simplicity we have not modelled all the risks identified. The goal graph of the Indus informs users of the possible risks that may occur due to adoption of this cloud. The goal tree of Smart Bank helps us in introducing some obstacle resolution tactics which may mitigate the risks associated with adopting the wrong cloud service provider.

6.1 Goal Tree of Indus

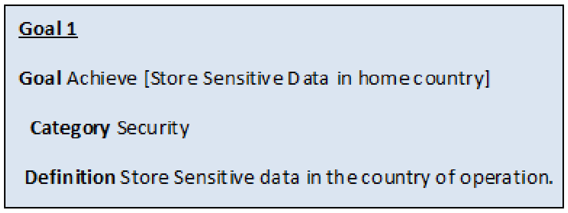
Figure 3 shows the goal tree of Indus cloud. Many obstacles were identified to achieve security while migrating to the Indus cloud. The goal tree of Indus cloud informs users of the potential risks that are involved with Indus cloud adoption.

6.2 Smart Bank Goal Tree

Recently a cloud architecture evaluation method was proposed by Faniyi et al in (Faniyi et al, 2011). A case study of a Smart bank was presented where the bank decides to adopt cloud services to deliver a robust, scalable and on-demand service provisioning for the risk management aspect of its business. The architecture proposed was evaluated using Architectural Tradeoffs Analysis Method

(ATAM) (Clements et al, 2001) incorporating dynamic analysis via implied scenario generation and was named as ATMIS.

Smart Bank policies prohibit some of the sensitive data from being hosted on any server outside its country of operation. As cloud acts as a black box and the user has little or no knowledge about the location of the data centre; this pose a threat to Smart Bank's contents. SLA clauses often make no mention of any sub-contracting of data centres. Data from Linode, a leading Cloud service Provider was unavailable to the users for about 20 minutes due to a power outage at Hurricane Electric which owns the data centre where Linode stores data (CloudHarmony, 2011). The outage at Hurricane Electric was completely outside the control of Linode. Such sub-contracting poses a grave threat to Smart Bank's policies. See Figure 5.



Goal 1

Goal Achieve [Store Sensitive Data in home country]

Category Security

Definition Store Sensitive data in the country of operation.

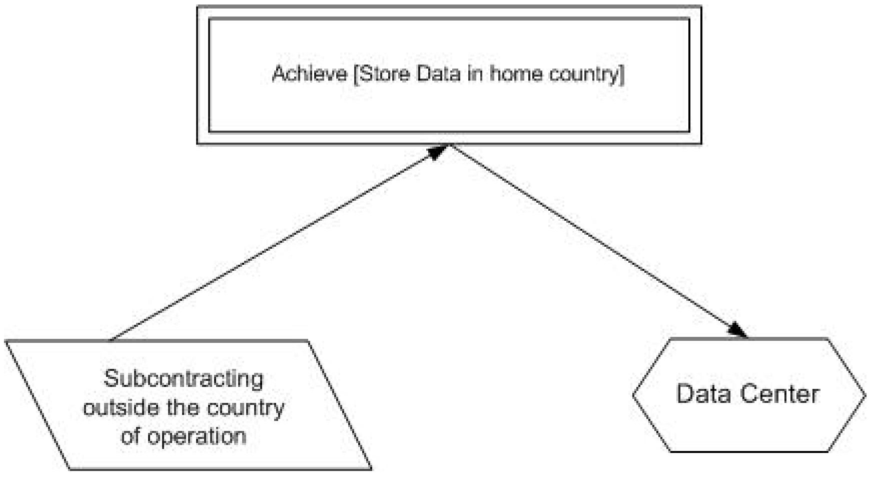
7. RESOLVING OBSTACLES IN THE PROCESS OF CLOUD ADOPTION

By Looking at both the perspectives of the case study i.e. Indus (see Section 6.1) and Smart Bank (see Section 6.2) we will try to map the requirements of the user with the features of Indus. We are proposing some concrete and abstract obstacle resolution tactics which could help the user in minimizing the risks associated with cloud adoption. Table 2 shows the obstacle resolution tactics. Some of the obstacles cannot be fully resolved. We can set the priority of resolving the obstacles by quantifying the value of the obstacles. Every obstacle has a value. The value of an obstacle is the product of the likelihood of the obstacle occurrence and the consequence of that obstacle. As obstacles lead to risks therefore they should be quantified in order to know the damage that may be caused due to the resulting risks.

$$V(O) = L(O) * C(O) \tag{1}$$

where L(O) is the likelihood that obstacle O will occur, C(O) is the cost/consequence, and V(O) is the value of obstacle O. For simplicity we will rate each on 1 to 4 scales. The larger the number

Figure 5. Obstacle for achieving goal 1


Achieve [Store Data in home country]
Subcontracting outside the country of operation
Data Center

the larger would be the consequence or likelihood. Priority for resolving the obstacle can be established by using the matrix shown in Figure 6. Equation 1 is based on Boehm et al equation for quantifying the risks in a software development process (Boehm, DeMarco, 1997).

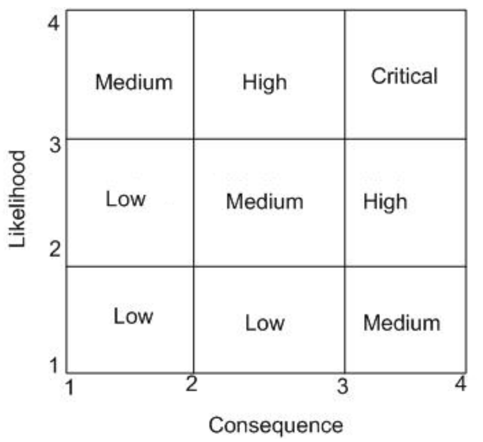
It may be possible that the obstacle resolution for a particular obstacle is more expensive than the value of the damage that may be caused due to the obstacle itself. Some obstacle may be resolved to a certain degree but may not be fully resolved. Table 2 defines some obstacle resolution tactics in the process of cloud adoption.

7.1 Obstacle Prevention

Obstacle prevention strategy resolves the obstacle by adding a new goal. By adding a new goal obstacle occurrence can be avoided.

Tactic 1: Goal tree of Indus revealed an obstacle of resource exhaustion. The same obstacle was identified in Smart Bank case study where the sensitive information is stored in-house. If the amount of sensitive information is significant than we have to move sensitive data to the cloud. Smart Bank can store the sensitive data in the cloud in encrypted form using Trusted Platform Module (TPM) (Santos et al, 2009). Using TPM is a new goal which has been added to the system for avoiding the obstacle. This will prevent Smart Bank from the maintenance of data repository in-house. See Figure 7.

Figure 6. Matrix for quantifying the value of obstacles



Tactic 2: Let's take Linode example where the data centre has been subcontracted to Hurricane Electric. In this situation if Smart Bank wants to exploit the services of Linode then the SLA should clearly state that the data center is not owned by Linode. SLA should also mention the location of the data center as Smart Bank does not want its data to be stored outside its country of operation. This will allow the user to make decision by keeping their requirements satisfied. Negotiation of SLA is another goal which has been added to the goal tree to prevent the obstacle shown in Figure 5. See Figure 8 for the resolution to this obstacle and Figure 9 for the goal tree of this tactic.

7.2 Cloud Service Substitution

Tactic 3: Smart Bank wants its system to be available 24/7. We assume that Bank also wants to scale up when there is increased traffic and scale down when the demand is low. Nallur et al have proposed a self-optimising architecture (Nallur et al, 2009) where the website can scale up and scale down according to the traffic. Assume that when the traffic is low the website is using cloud C1 and when the traffic is high the website starts using cloud C2. This strategy resolves the obstacle of non-availability of the website when the traffic is high. Smart Bank can sign up to the services of different clouds for different time. See Figure 10.

Goal 2

Goal Achieve [24/7 Availability of Smart Bank]

Category Availability

Definition The website should always be available to the users.

Tactic 4: This tactic consists of splitting a goal assigned into subgoals, so that every subgoal can be assigned to a different cloud to satisfy it. Application of this tactic may lead to the introduction of some new clouds to satisfy the subgoals. As web services hosted on the cloud evolve with the passage of time, their requirements evolve accordingly. Smart Bank is a new bank with fewer customers. With the passage of time the bank will become more popular and it will be critical for Smart Bank to be highly stable, secure and available. Downtime of Smart Bank is unacceptable to its customers. Such web applications which have become popular

Figure 7. Obstacle resolution tactic1

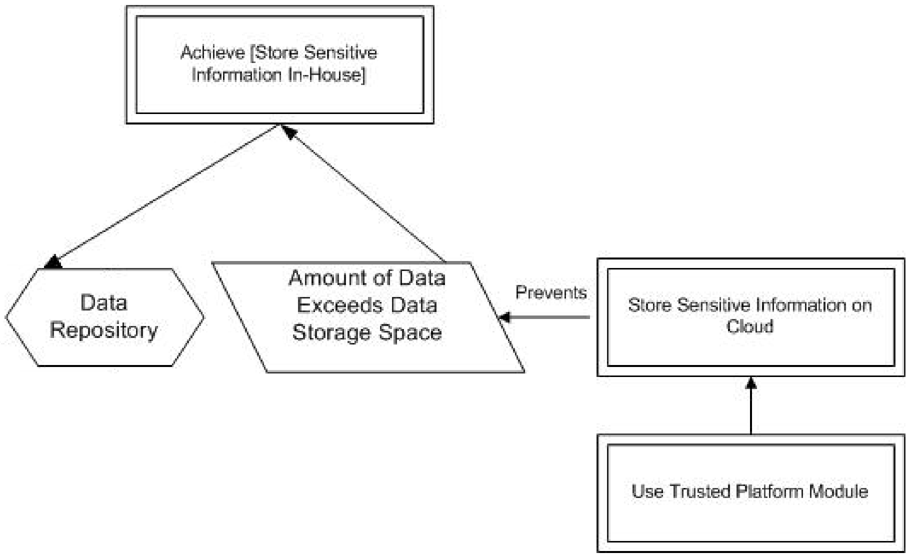

Achieve [Store Sensitive Information In-House]
Data Repository
Amount of Data Exceeds Data Storage Space
Prevents
Store Sensitive Information on Cloud
Use Trusted Platform Module

Figure 8. Obstacle resolution tactic 2

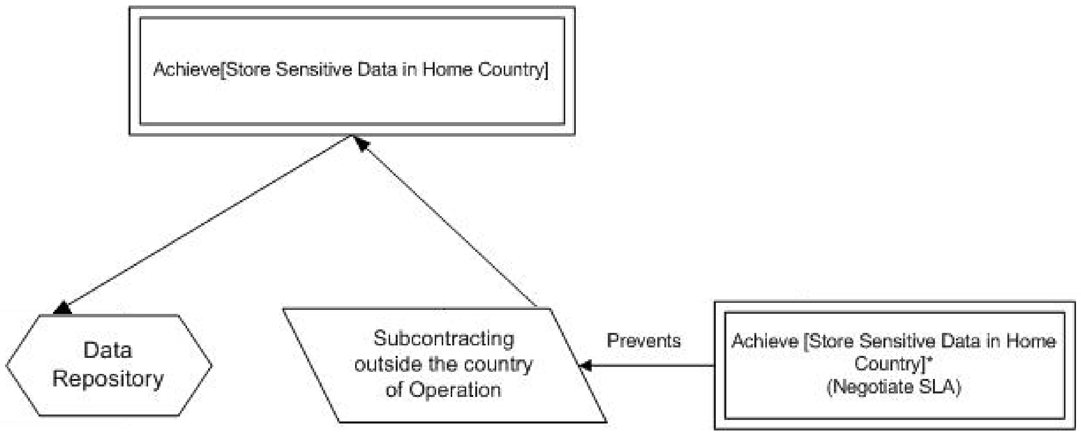

Achieve[Store Sensitive Data in Home Country]
Data Repository
Subcontracting outside the country of Operation
Prevents
Achieve [Store Sensitive Data in Home Country]*
(Negotiate SLA)

Figure 9. Goal tree of tactic 2

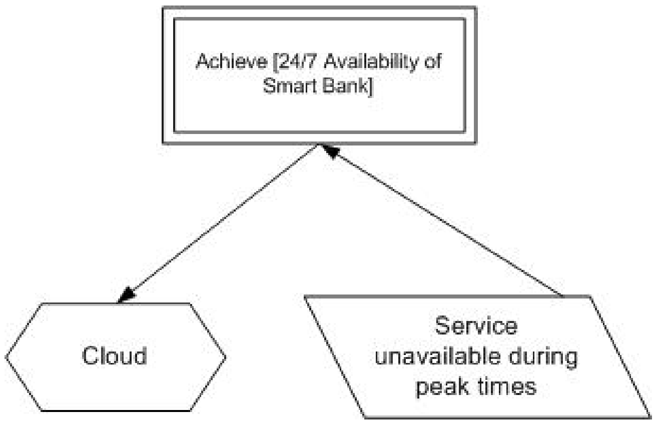
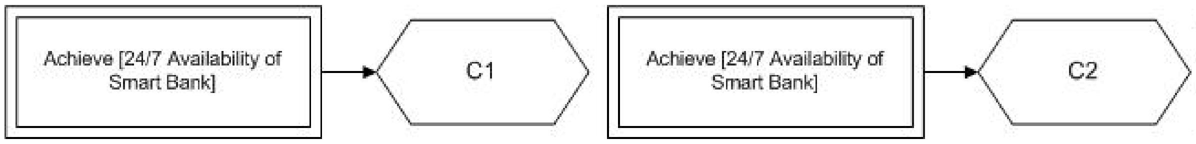
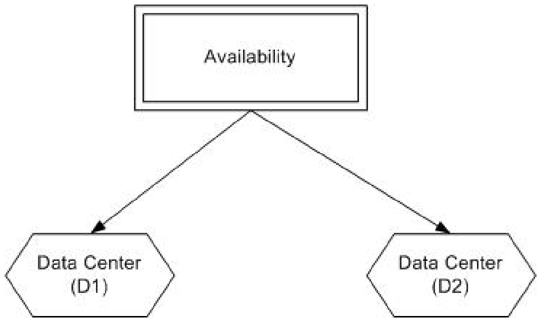

Achieve [24/7 Availability of Smart Bank]
Cloud
Service unavailable during peak times

Figure 10. Obstacle resolution tactic 3


Achieve [24/7 Availability of Smart Bank]
C1
Achieve [24/7 Availability of Smart Bank]
C2

among the users need to adapt their architecture with the growing demand. Assume that the data centre D1 was used by Smart Bank but as the demand grew Smart Bank has to lease another data centre to avoid any chances of unavailability to the user. The SLA can be negotiated to lease another data centre from a cloud different than the current one so that data centres D1 and D2 are owned by two different cloud service providers. See Figure 11.

Figure 11. Obstacle resolution tactic 4



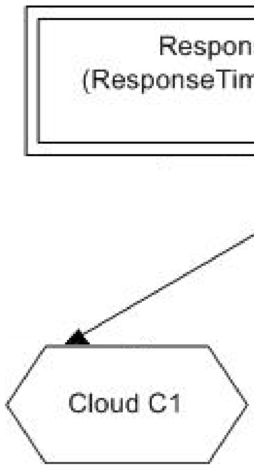
7.3 Goal Weakening

Tactic 5: This tactic involves weakening the goal definition or the degree of satisfaction. As an example UoB wants the response time for its email service to be 10 seconds but agrees to settle for 12 seconds if the cloud service provider C1 is unable to guarantee a response time of 10 seconds. The satisfaction degree of the goal was relaxed to avoid the elimination of a potential cloud service provider for service provision. See Figure 12.

8. GOAL PRIORITIZATION

Goal prioritization is a core activity for resolving the obstacles in the requirements engineering process for cloud adoption. Stakeholders engage in extensive prioritization process to distinguish core goals (i.e. critical goals that should always be satisfied) from the desirable goals (i.e. goals that

Figure 12. Obstacle resolution tactic 5



can be traded off). There might be some obstacles which could be avoided due to their significantly lower impact on the system. However, obstacles that can critically affect the system should be resolved with high priority. In the goal prioritization process goals are assigned weights according to the stakeholders' preferences. We describe an approach which helps in prioritizing the goals and obstacles. The highest level of goal refinement tree is level 0; the next level of subgoals is level 1 and so on. We assign the weights to goals in each level, so that the sum of the weights of goals in each level is 1.

Step 1: Obtain the relative importance of the goals. Consider the goal tree of smart bank. In level 1 we have five goals g_1, g_2, g_3, g_4, g_5 and now we want to know their relative value. Rank the importance of each goal. Smart Bank prefers security over everything else. After comparing and obtaining the relative importance of the level 1 goals, we have that:

$$g_1 > g_4 \approx g_5 > g_3 > g_2$$

This implies that g_1 is the most important goal followed by g_4 and g_5 (g_4 and g_5 are of equal importance), g_3, and g_2. The weight μ of each goal is:

$$\mu_{g1} > \mu_{g4} \approx \mu_{g5} > \mu_{g3} > \mu_{g2}$$

In a similar way stakeholders compare the relative importance of goals of the same parent in different levels.

Step 2: Assign each goal its weight based on the obtained relative importance.
Stakeholders' engage in brain storming sessions to discuss how the satisfaction of each goal can contribute to the achievement of strategic objectives. These sessions lead to an agreement among stakeholders about their priorities and assigning weights to goals. We continue the process of assigning weights to the goals belonging to the same parent. For level 0 goals (see Figure 13) we assign 0.3 to g_1 as security was deemed most important by the stakeholders. Since g_4 and g_5 have an equal importance therefore they have been assigned the same weight. The weights have been assigned to the goals keeping in view their relative importance.

Step 3: Obtain composed weights for offspring goals.
The final weight of the goal is called a composed weight. Composed weight is the product of the weight of each goal with the weights of all the parent goals. We can obtain the composed weight for the obstacles $\neg g_2$, $\neg g_4$, $\neg g_5$ and $\neg g_{1.5}$, so that:

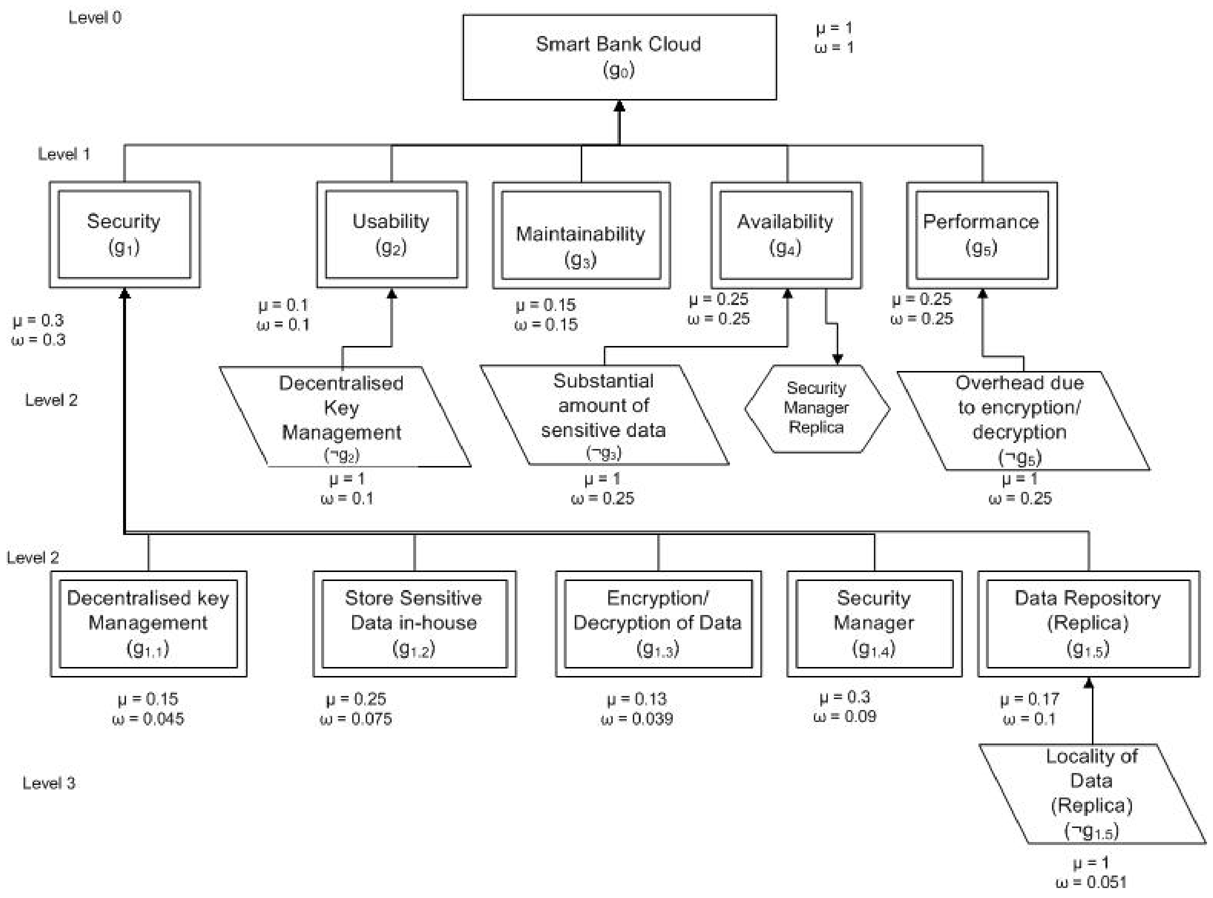
$$\mu g_2 = 0.1$$

$$\mu g_0 = 1$$

$$\omega g_2 = \mu g_2 \text{ x } \mu g_0$$

$$\omega g_2 = 0.1 \text{ x } 1 = 0.1$$

Figure 13. Goal prioritization: goal weights and composed weights



Now calculating the composed weight ω for obstacle $\neg g_2$:

$\omega\neg g_2 = \mu\neg g_2 \times \mu g_2 \times \mu g_0$

$\omega\neg g_2 = 1 \times 0.1 \times 1 = 0.1$

Similarly the composed weights for other obstacles can be calculated:

$\omega\ \neg g_4 = 0.25$

$\omega\ \neg g_5 = 0.25$

$\omega\ \neg g_{1.5} = 0.051$

This technique of prioritizing goals by comparing the offspring of the same parent allow decision makers to reason about the importance of a particular goal in comparison to other goals. The calculated composed weight helps in prioritizing the obstacles. The greater the value of composed weight the higher would be the priority of the obstacle/goal. The prioritized list of obstacles provides the basis for the obstacle resolution phase.

9. RELATED WORK

A comprehensive use of goal-oriented requirements engineering can be found in Lamsweerde (2001). Requirements are represented in the form of goals in GORE. The advantage of using this approach is that a goal graph provides vertical traceability from high-level strategic concerns to low-level technical details; it allows evolving versions of the system under consideration to be integrated as alternatives into one single framework. Goal-oriented approaches have received significant attention over the years, where goals were used for modelling functional requirements (Lamsweerde, Letier, 2004), non-functional requirements, and Agent-oriented systems (Bresciani et al, 2004). For example, Mylopoulos et al used goal oriented approach for eliciting, specifying and refining the non-functional requirements (Mylopoulos et al, 1992; Bresciani et al, 2004) demonstrates another use of Goal-oriented approaches in Agent Oriented Programming (AOP) for open architecture that need to change and evolve due to changing requirements. GORE was also used to model the system architecture to meet changing business goals and for evolving systems (Anton, Potts, 1998; Gross, Yu, 2001). One interesting application of GORE, which has inspired our work is that of (Alves, Finkelstein, 2002; Alves et al, 2005; Chung, 2002), where GORE was used to inform the process of selecting Commercial off the Shelf (CotS) products matching user's requirements. Though the fundamental use of GORE exhibits resemblance with that of (Alves, Finkelstein, 2002; Alves et al, 2005; Chung, Cooper, 2002) the problem of cloud adoption is by far more challenging as we are dealing with "open loop" systems, with dynamic, unbounded and elastic scale where continuous service evolution is the norm. Due to continuous evolution of the cloud there may be numerous obstacles for adopting cloud services.(Lamsweerde, Leiter, 2000) present a detailed view of handling obstacles in goal-oriented requirements engineering.

There is a well cited work in software risk management (Boehm, DeMarco, 1997; Boehm, 1991). Most of the work carried out in this field is related to software projects (Fairley, 1994; Schmidt et al, 2001) where it is assumed that project manager knows almost everything related to the system. In case of the cloud where large number of users is concerned and little is known about the cloud service provider it is quite difficult to devise a risk management plan. So far, no strategy has been formulated which could help users in mitigating the risks involved with cloud adoption.

Research efforts over the years have looked at the problem of service discovery with runtime mechanisms to inform and optimize the selection e.g. self-managed applications in the cloud (Nallur, Bahsoon, 2010; Nallur et al, 2009) and self-optimizing architecture (Nallur et al, 2009). There is a recent research on cloud migration (Khajeh-Hosseini et al, 2010) which does not involve user requirements for cloud adoption. Up to the authors' knowledge, there has been no research on cloud procurement and adoption from requirements engineering perspective using the notion of obstacles. The need for such research is timely as there is complete lack of systematic methodologies, which could help stakeholders screen, match, negotiate their requirements against cloud services' provision and manage the tradeoffs associated with matches/mismatches of users' requirements and mitigating risks.

10. DISCUSSION AND FUTURE WORK

Cloud computing has gained popularity among businesses in recent years. Users are as excited as nervous about using the cloud. They are excited as most of the services provided by the cloud are low cost and readily available. At the same time, in spite of many promises by the cloud service providers, users remain much concerned about the general risk associated with the adoption of the cloud such as security and privacy of the data held, processed and exploited. We have proposed a framework which would help in addressing the genuine concerns of the cloud users. We looked at the case study of a cloud service provider and found that there were indeed many risks associated with cloud adoption. We advocated the necessity of a systematic methodology, for cloud adoption. The methodology can help the users in refining their requirements and negotiating with the cloud service provider while using goal oriented approach for our research. Our approach will help the users to identify the conflicts between the requirements and to reduce them. There might be certain obstacles involved in achieving some goals. Obstacles can be any hindrance in achieving any goal or the undesired properties (Potts, 1995). Obstacle analysis needs to be done in the initial phase of requirements engineering i.e. at the goal level (Lamsweerde, Leiter, 2000). Obstacles can also be regarded as expression of risks. We have proposed obstacle resolution tactics for cloud adoption process inspired from Axel et al's work (Lamsweerde, Leiter, 1998). Up till now no work has modelled the obstacles for cloud adoption process. We have also defined a framework for prioritizing user requirements. The framework is based on utility theory. Prioritizing the requirements help users to resolve the most critical obstacles first and to achieve goals that are most important. We intend to complement our methodology by using economic-driven approaches for evaluating the cost effectiveness of the decisions and the tradeoffs concerned. As the cloud services keep evolving it is imperative for the user to adapt to the changes made in the cloud. A requirements engineering framework could help users in elaborating and specifying user requirements and later matching it to the cloud service providers' features. Such dynamic selection of cloud with respect to user requirements is challenging but is outside the scope of this paper.

11. CONCLUSION

We have defined the steps involved in the requirements engineering phase for cloud adoption. This paper contributes a lifecycle which is an extension of our previous work (Zardari, Bahsoon, 2011). The lifecycle could help in making a framework for cloud adoption. We have used goal oriented approach for eliciting and modelling the requirements of the user. We have presented a systematic guidance for the identification and evaluation of cloud service providers. The evaluation process concludes in the selection of potentially best cloud service provider available. Key phase of the method is the matching phase where mismatches between the requirements and cloud service providers' features are identified. The analysis of the mismatches may inform the existence of risks. Our approach advocates risk mitigation in early stages of cloud adoption. We have proposed a number of abstract and concrete obstacle resolution tactics. The approach also tries to manage the tradeoffs involved with cloud adoption. The Service Level Agreements presently are too static and non-negotiable. The SLAs do not address the individual needs of every user. This framework would help the cloud service provider and the user to negotiate their requirements and cloud features. It would help making SLAs that are specifically designed for a particular organization. We have also defined a requirement prioritization technique based on utility theory for prioritizing user requirements.

REFERENCES

Alves, C., & Finkelstein, A. (2002). *Challenges in COTS decision-making: A goal-driven requirements engineering perspective.* Paper presented at the 14th International Conference on Software Engineering and Knowledge Engineering.

Alves, C., Franch, X., Carvallo, J. P., & Finkelstein, A. (2005). Lecture Notes in Computer Science: *Vol. 3412*. *Using goals and quality models to support the matching analysis during COTS selection* (pp. 146–156). doi:10.1007/978-3-540-30587-3_25

Amburst, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., & Konwinski, A. … Zaharia, M. (2009). *Above the clouds: A Berkley view of cloud computing*. Technical Report UCB/EECS-2009-28.

Amburst, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., & Konwinski, A. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50–58. doi:10.1145/1721654.1721672

Anton, A. I. (1996). *Goal-based requirements analysis*. Paper presented at the Second IEEE International Conference on Requirements Engineering, Colorado Springs, USA.

Anton, A. I., McCracken, W. M., & Potts, C. (1994). *Goal decomposition and scenario analysis in business process reengineering*. Paper presented at the 6th International Conference on Advanced Information System Engineering.

Anton, A. I., & Potts, C. (1998). *The use of goals to surface requirements for evolving systems*. Paper presented at the 20th International Conference on Software Engineering.

Biggs, S., & Vidalis, S. (2009). *Cloud computing: The impact on digital forensic investigations.* Paper presented at the International Conference for Internet Technology and Secured Transactions, London.

Boehm, B. W. (1981). *Software engineering economics*. Prentice-Hall.

Boehm, B. W. (1991). Software risk management: Principles and practices. *IEEE Software*, *8*(1), 32–41. doi:10.1109/52.62930

Boehm, B. W., & DeMarco, T. (1997). *Software risk management* (pp. 17–19). IEEE Software.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, *8*(3), 203–236. doi:10.1023/B:AGNT.0000018806.20944.ef

Buyya, R., Yeo, C. H., & Venugopal, S. (2008). *Market oriented cloud computing: Vision, Hype, and reality for delivering it services as computing utilities*. Paper presented at The 10th IEEE International Conference on High Performance Computing and Communication.

Chung, L., & Cooper, K. (2002). *A knowledge-based COTS-aware requirements engineering approach.* Paper presented at the 14th International Conference on Software Engineering and Knowledge Engineering.

Clements, P., Kazman, R., & Klein, M. (2001). *Evaluating software architectures: Methods and case studies*. ISBN-13: 978-0-201-70482-2

CloudHarmony. (2011). *Do SLA really matter? A 1 year case study of 38 cloud services.* Retrieved from http://cloudharmony.com/b/2011/01/do-slas-really-matter-1-year-case-study.html

Fairley, R. (1994). Risk management for software projects. *IEEE Software*, *11*(3), 57–67. doi:10.1109/52.281716

Faniyi, F., Bahsoon, R., Evans, A., & Kazman, R. (2011). *Evaluating security of architectures in unpredictable environments: A case for Cloud*. Paper presented at the 9th Working IEEE/IFIP Conference on Software Architecture.

Goscinski, A., & Brock, M. (2010). Toward dynamic and attribute based publication, discovery and selection for cloud computing. *Future Generation Computer Systems*, *26*, 947–970. doi:10.1016/j.future.2010.03.009

Gottfrid, D. (2007, January 11). Protrated supercomputing fun! *The New York Times*.

Gross, D., & Yu, E. (2001). *Evolving system architecture to meet changing business goals: An agent and goal-oriented approach*. Paper presented at the Fifth IEEE International Symposium on Requirements Engineering.

IBM. (2008). *IBM perspective on cloud computing*. Retrieved from ftp://ftp.software.ibm.com/software/tivoli/brochures/IBM_Perspective_on_Cloud_Computing.pdf

Khajeh-Hosseini, A., Greenwood, D., & Sommerville, I. (2010). *Cloud migration: A case study of migrating an enterprise IT system to IaaS*. Paper presented at the 3rd International Conference on Cloud Computing.

Lamsweerde, A. V. (2001). *Goal-oriented requirements engineering: A guided tour*. Paper presented at the 5th IEEE International Symposium on Requirements Engineering, Toronto

Lamsweerde, A. V. (2004). *Goal-oriented requirements engineering: A roundtrip from research to practice*. Paper presented at the 12th IEEE International Requirements Engineering Conference, Kyoto.

Lamsweerde, A. V., Darimont, R., & Letier, E. (1998). Managing conflicts in goal-driven requirements engineering. *IEEE Transactions on Software Engineering*, *24*(11), 908–926. doi:10.1109/32.730542

Lamsweerde, A. V., & Leiter, E. (1998). *Integrating obstacles in goal-driven requirements engineering*. Paper presented at the 20th International Conference on Software Engineering, Kyoto.

Lamsweerde, A. V., & Leiter, E. (2000). Handling obstacles in goal-oriented requirements engineering. *IEEE Transactions on Software Engineering*, *26*(10), 978–1005. doi:10.1109/32.879820

Lamsweerde, A. V., & Letier, E. (2004). Lecture Notes in Computer Science: *Vol. 2941*. *From object orientation to goal orientation: A paradigm shift for requirements engineering* (pp. 153–166). doi:10.1007/978-3-540-24626-8_23

Miller, M. (2008). *Cloud computing: Web based applications that change the way you work and collaborate online*. Que Publishing.

Mylopoulos, J., Chung, L., & Nixon, B. (1992). Representing and using nonfuntional requirements: A process oriented approach. *IEEE Transactions on Software Engineering*, *18*(6), 483–497. doi:10.1109/32.142871

Nallur, V., & Bahsoon, R. (2010). *Design of a market-based mechanism for quality attributes tradeoff of services in the cloud.* Paper presented at the 2010 ACM Symposium on Applied Computing.

Nallur, V., Bahsoon, R., & Yao, X. (2009). *Self-optimizing architecture for ensuring quality attributes in the cloud*. Paper presented at the 2009 IEEE/IFIP WICSA/ECSA.

Nuseibeh, B., & Easterbrook, S. (2000). *Requirements engineering: A roadmap*. Paper presented at the The Future of Software Engineering.

Potts, C. (1995). *Using schematic scenarios to understand user needs*. Paper presented at the of 1st Conference on Designing Interactive Systems: Processes, Methods and Techniques.

Santos, N., Gummadi, K. P., & Rodrigues, R. (2009). *Towards trusted cloud computing*. Paper presented at the 2009 Conference on Hot Topics in Cloud Computing.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. *Journal of Management Information Systems*, *17*(4), 5–36.

Zardari, S., & Bahsoon, R. (2011). *Cloud adoption: A goal-oriented requirements engineering approach*. Paper presented at the IEEE/ACM International Workshop on Cloud Software Engineering, The ACM/IEEE 33rd International Conference on Software Engineering, Hawaii, USA.

Zave, P. (1997). Classification of research efforts in requirements engineering. *ACM Computing Surveys*, *29*(4), 315–321. doi:10.1145/267580.267581