Methodology for Information Management and Data Assessment in Cloud Environments

Mariam Kiran, Department of Computer Science, University of Sheffield, Sheffield, UK

Gregory Katsaros, FZI Research Center for Information Technology, Karlsruhe Institute of Technology, Karlsruhe, Germany

Jordi Guitart, Barcelona Supercomputing Center (BSC), Barcelona, Spain

Juan Luis Prieto, Atos Research and Innovation, Barcelona, Spain

ABSTRACT

The emergence of cloud technologies has affected the service computing ecosystem introducing new roles and relationships as well as new architectural and business models. Along with the increase of capabilities and potentials of the service providers comes the increase of the information available and issues to efficiently manage it. In this paper, an architectural approach is presented that involves a combination of a cloud-enabled data model, the monitoring infrastructure and the establishment of assessment mechanisms which are based on factors such as trust, risk, energy and cost (TREC factors). This architectural model discusses the monitoring features as well as the how the assessment functionalities can work together with other components to produce a self-reliant cloud ecosystem preventing any fails during the service lifecycle. This paper elaborates on how the self-management, by using decision-making processes, can maximise business level objectives of the providers. The results presented in the paper show how the suggested architecture can help develop efficient cloud architecture.

Keywords: *Architectural Model, Cloud Computing, Cost, Data Model, Energy, Information Assessment, Information Management, Monitoring, Risk, Trust*

1. INTRODUCTION

Cloud computing is continuously evolving and becoming one of the most challenging paradigms of Information Technology with various business models to target the needs of users and enterprises (Buyya et al, 2008). Cloud providers are forming broad cloud ecosystem to meet the rising demands. In this paper, two types of cloud providers are discussed particularly the

DOI: 10.4018/IJGHPC.2014100104

Service Providers (SPs) which ask for a service to be executed, and the Infrastructure Providers (IPs) which actually execute the service on their infrastructures.

With the rapid evolution of the cloud, together with the new emerging needs of customers, cloud environments are becoming very complex in terms of dimension and management. These systems are composed of various entities, such as stake-holders with different interests such as users or providers, Service Level Agreements (SLAs) contracts agreed between two parties, virtualised resources, to name a few. Thus cloud service models, whether being Platform as a Service (PaaS) or Infrastructure as a Service (IaaS), have huge amounts of information that needs to be collected, managed and evaluated for successful completion of the services. Therefore there is a need to define a consistent cloud-enabled data model which represents multiple entities and their interrelationships managing these efficiently. The virtualisation technology, in cloud environments, offers a manner to autonomously manage IT (cloud) entities allowing dynamic resource allocation that, based on accurate monitoring information, enables the appropriate enactment of cloud-related features like elasticity of services or high availability of resources.

Monitoring information can range from application-related metrics, such as web-based service response time, to infrastructure-related metrics like power consumption or resource capacity and utilisation. This information should then be assessed in order to provide the business-level parameters (BLPs) like trust, risk, ecological efficiency and cost metrics related to the service as a decision-making process.

Self-management of cloud systems should be governed by certain business-driven management policies which embrace the needs of both stakeholders: (1) users, who typically specify several constraints in SLAs, also known as Service-Level Objectives (SLOs); and (2) providers, who have their own Business-Level Objectives (BLOs) to fulfill such as saving costs and energy. This multi-purpose approach may lead to significant trade-offs, which must be solved by proper management policies. For instance, maximising the eco-efficiency of a cloud provider's infrastructure is possible through the consolidation of several virtual machines in the same physical host. However, the performance of the services running in those virtualised resources may be diminished leading to the agreed SLAs being violated.

The assessment of high-level parameters, such as trust, risk, ecological efficiency and cost, are crucial inputs for these management policies driven by the business aspects in order to efficiently guide the operation of cloud providers in terms of the BLOs fulfillment.

To address the research challenges mentioned, this work contributes in several aspects:

- A generic cloud-enabled data model, which represents all involved entities in typical complex cloud environments and their interrelationships, presenting an efficient way to collect, aggregate and store monitoring information from cloud infrastructures.
- Several assessment tools that process monitoring information and are aimed to assist in the self-management of the cloud focusing on trust, risk, ecological efficiency and cost factors (Ferrer et al, 2012).
- Management entities for service (SP) and infrastructure (IP) cloud providers guided by decisions based on assessment tools.
- A policy-related approach driven by both customer needs and provider BLOs. This involves efficient information management and determining the most convenient management action(s) when IT-level events (expected or unexpected) take place.

Addressing the above mentioned points, this paper is organised in the following manner: in Section 2 the related work is presented with the technologies regarding information management in cloud environments. Section 3 introduces the cloud data model of the approach discussed in this paper, with Section 4 discussing the collection process of the information. Section 5 elaborates on the assessment processes within the cloud environment analysing the four factors trust, risk, eco-efficiency and cost (TREC) in terms of the service and infrastructure providers and their interactions with the other components. This paper discusses the use of SPs and IPs as the main actors involved in a cloud environment as a means to show how data assessment can be automated for these entities for a general discussion. The end-user has been ignored for reducing research complexities and will be considered in the future research work. Finally, Section 6 summarises the conclusions and further future work needed to be carried out in this research.

2. RELATED WORK AND TECHNOLOGIES

2.1. Monitoring Data

Considering service oriented architectures (SOA) and Grid computing as the immediate "siblings" of the cloud computing paradigm, the trends and developments of these technologies in the field of information management have been investigated extensively (Mell and Grance, 2009). This section analyses the current technologies and related work existing in the two major areas which this paper addresses, namely (i) management of monitoring information in cloud environments, including collection, aggregation and storage, and the (ii) assessment of that monitoring data in order to assist in autonomous decision-making processes of cloud providers leading to self-managed cloud environments.

Firstly, regarding management of monitoring information, different tools exist for resource monitoring which have been traditionally used in clusters and grid computing environments and are now being adopted by cloud computing. Some of those tools are open source and potentially scalable and flexible. One of the most popular monitoring tools is Nagios (1996). Due to its plug-in architecture, Nagios allows the monitoring metrics related to any kind of resource, from physical to virtual and application-related, with the development of further specific plug-ins (Nagios Plugins, 2000). Regarding aggregation and storage of the collected information, each of the Nagios plug-in stores the data in a log file, in the form of the "service checks". This allows Nagios, through NDOUtils plugin, to export both historical and currently collected data to a MySql database. Regardless, of these collection processes, the information is stored in a simple format, without any further processing or aggregation. Therefore, to subsequently use this data for assessment, it is not possible to perform calculations for the current cloud computing environment where multiple metrics at the different levels must be considered to take actions in limited time.

Another popular solution for monitoring data is presented by Ganglia (Sacerdoti et al, 2003), widely used on high performance computing systems such as clusters and grid environments. Its hierarchical architecture allows monitoring of clusters with high number of computer nodes (up to 2000). Ganglia uses monitoring metrics which belong to two main groups, namely the built-in (capturing computing node information) and the user-defined metrics (representing application-specific states), allowing an adaption and extension of the monitored data set. Management of data is done by means of XML for data provision, XDR for data transport, while monitoring results are stored and provided in a graph format by means of the RRD tool. One of the main advantages of Ganglia is that it introduces very low overhead per node, thus allowing collection and storage of large amounts of data. However, even if Ganglia is suitable for monitoring of large distributed systems, it does not address the monitoring requirements of rapidly changing and dynamic infrastructures in clouds services.

The work performed within the IRMOS project (2008) offers a monitoring solution combining the different monitoring technologies at different levels of the cloud environment. The Monitoring and Discovery Service (MDS), included with the Globus Toolkit is used to manage information at the application level (Katsaros et al, 2010), while virtual and physical infrastructure related metrics are managed by the monitoring system within the ISONI (Intelligent Service Oriented Network Infrastructure). The project uses different techniques to measure different parameters in the Infrastructure as a Service layer, including networking, storage and computing resources (Voith et al, 2010) (Narasimhamurthy et al, 2011) (Voith et al, 2009). However, even if it provides with the aggregation of both virtual and physical level data, it does not provide a generic cloud-enabled model and also does not provide efficient assessment functionality on various business level objective metrics.

2.2. Assessing Data

Regarding data assessment, Nagios provides a simple mechanism to evaluate monitoring data collected by plug-ins comparing them with defined thresholds. Based on the results alert levels - OK, Warning, Critical or Unknown - are supported triggering necessary actions, such as initiating a corrective action or sending a notification. A more sophisticated solution for the assessment of monitoring data in SLAs based on semantic technologies is presented by Ejarque et al. (2010). In particular, the authors use semantics to describe tasks and resources, as well as link them to different objects of ontology. Based on this semantic ontology and a rule engine, inferences could be made to assign resources to tasks. However, this approach takes a long time to make decisions because of the overhead introduced by the semantics.

The Business-Driven IT Management (BDIM) discipline (Sauve et al, 2006b) has been widely used to manage IT systems from the business point of view. IT self-management processes driven by business-level aspects have been proposed in several works. For instance, Aiber et al. (2004) presented a general architecture, with a set of technologies and methodologies enabling autonomous self-optimisation according to Business Level Objectives (BLOs). However, these only considered the goal of maximising the income or costs. Hence, such approaches must be extended to be used by autonomous cloud providers driven by several (disparate) BLOs. Moreover, some research efforts have used BDIM methodologies to also increase the business value of e-commerce applications, such as (Sauve et al, 2006a) and (Marques et al, 2006). However, these approaches do not provide dynamism when allocating resources to services, which is required to deal with the typical changes in the environment such as demand variations.

As described above, existing work in the area does not take advantage of the complete potential of existing off-the-shelf tools, due to two main reasons, (i) traditionally because it was usually not required by the concrete application problem. For example, in HPC clusters and grid, the main information required is related to monitoring of physical IT resources and, (ii) in recent cloud computing deployments, major research challenges are still being identified. Therefore, this paper introduces an additional level above the pure collection of "raw" monitoring data, performing a more sophisticated aggregation, classification and storage according to a well-defined data model.

The second major goal of this paper is to describe how the proposed solution enables assessment of the different types of collected monitoring data, allowing the appropriate reaction in the form of alerts and corrective actions, leading to a self-managed cloud environment. Assessment of monitoring data becomes a major challenge when the needs of providers and consumers must be fulfilled with satisfied completion of Service Level Agreements. These assessments should ideally result in corresponding alerts and triggering of corrective actions. These issues become more complex with the involvement of great amount of metrics of different natures and levels of abstraction (physical infrastructure, virtual infrastructure, application and business related such as trust, risk, cost or eco-efficiency, to name a few).

3. SETTING THE SCENE: THE CLOUD DATA MODEL

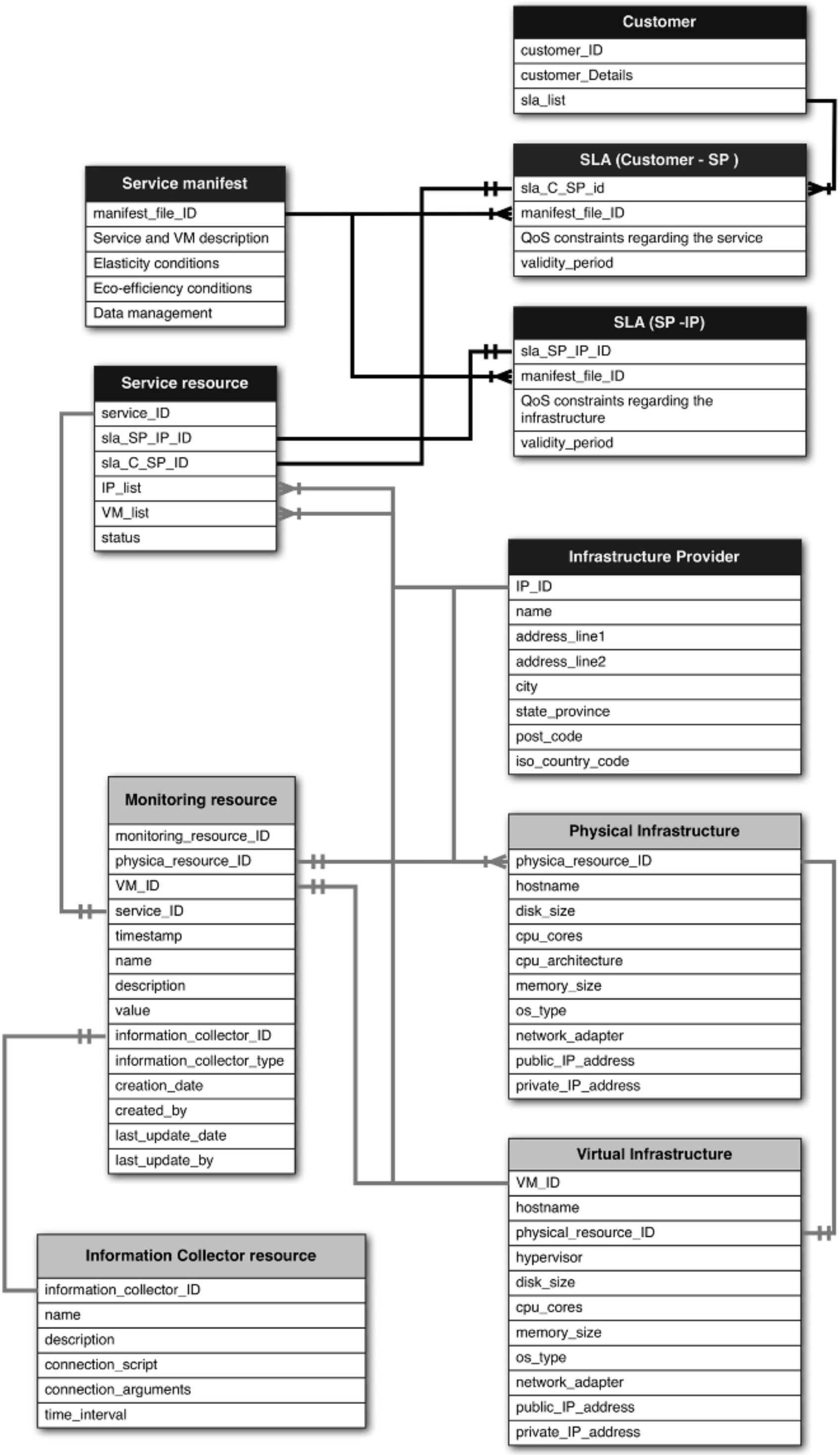
The collection, management and assessment of information of a system are always done under the “umbrella” of a pre-defined data model. In the service oriented, as well as grid computing domains, research work has been identified as a need and raises the issue of interoperability between the different systems (cf. (Pfoser et al, 2003) (Field et al, 2008) (Andreozzi et al, 2008)). For example, the Open Grid Forum (OGF) has suggested several data model specifications such as GLUE (OGF, Glue Working Group, 2009) and Activity Instance (AID) (OGF, JSDL Working Group, 2009). GLUE is a conceptual information model for Grid entities, independent from Grid implementations in order to enforce interoperability. It has been quite successfully adopted by various projects and initiatives (such as (D-GRID Project Consortium, 2009) (NORDU GRID, 2011)). AID tries to catch information related to “activities” such as resource usage, security data, state or monitoring data. It acts more as a container format and less as a specific format for monitoring.

Additionally, as presented before, the cloud ecosystem includes various roles and entities. Different types of services are offered within the cloud paradigm (SaaS, IaaS, PaaS, NaaS etc.) and as the related technologies mature, more types will flood the marketplace. Research as well as business activities related with cloud computing, grow in importance with the production of information and the demand for better management. The adoption of models and techniques designed for SOA and Grid computing can be a solution to one extent, but the innovations and differences that cloud computing brings, usually causes inconsistencies. For the introduction of the virtualisation layer, different business models are applied with new identified roles such as infrastructure provider, service provider, platform provider, storage provider, etc. These are just some of the cloud specific requirements that drive the need for a cloud-enabled data model. To this end, a data model is designed that applies to different cloud scenarios and use cases. For example, in a Service Provider (SP) centric model, we consider the service (or application) as the main entity in the cloud ecosystem but we also incorporate the roles of the Customer as the end user of the service. In the Infrastructure Provider (IP) centric model, the data model can be populated with more physical layer data and can have relevant data for fault tolerance and mitigation. More types of providers can be managed in a similar manner.

Following is an analysis of each entity defined and presented in Figure 1. The tables presented with black color are associated with components and capabilities residing on the SP side, while the tables in grey are the entities on the IP side.

- **Service Manifest:** Is the entity that represents the template of a service to be deployed in a cloud infrastructure. Each service template consists of several pieces of information: the description of the service, the elasticity requirements (rules and policies that define the elasticity of the service at runtime), eco-efficiency conditions (policies and values related to the energy consumption and the eco-efficiency of the service execution in total), data management (details about the storage of data and access to it from the service). In this paper, the OVF standard (, DMTF) is used for expressing a service as a combination of virtual machines. Even though this specification might be limited in terms of topology representation, it captures the basic characteristics of Cloud applications. In a similar manner, the service manifest could be expressed using TOSCA (DMTF), a high level specification for Cloud applications just to define some aspects of the service.
- **SLA (Customer - SP):** Is the Service Level Agreement between a potential Customer and the Service Provider (SP) that offers access to several services and applications. A signed SLA document includes a reference to the manifest file that the customer selected as the template for

Figure 1. Cloud enabled data model



their chosen service or application. In addition, it includes the solid terms regarding the Quality of Service (QoS) defined for the specific service or application usage. The level of detail of the described terms is related with the capabilities expressed from the service description specification (OVF or TOSCA).

- **SLA (SP - IP):** Is the Service Level Agreement between the Service Provider (SP) and the Infrastructure Provider (IP). The SP selects an appropriate IP, based on the manifest file and the requested QoS parameters of a service deployment. The signed SLA among these actors defines the solid QoS constraints regarding the infrastructure, physical as well as virtual, provided by the IP.
- **Customer:** Is the entity that represents the end user of the service provided by the SP. The specific data structure includes the details of the customer (name, address, contact and account information, etc.) as well as references to the SLAs signed with the SP. That entity is consumed by the accounting and billing components of the cloud ecosystem.
- **Infrastructure Provider (IP):** Is the representation of an Infrastructure Provider (IP) that offers cloud services on top of its physical infrastructure. Details about the ownership and location of the provider (city, country, address etc.) are incorporated in that entity.
- **Physical Infrastructure:** Is the entity that describes the physical nodes of an infrastructure. Each record of that structure includes technical details about a host, such as disk space, CPU cores, memory, operating system and more associated with an IP.
- **Virtual Infrastructure:** Represents the virtual environment that a provider maintains. Each record of this entity describes a Virtual Machine (VM), deployed on top of a physical resource, including all the technical specifications of that deployment (hypervisor, disk space, memory, CPU cores, IP addresses etc.).
- **Information Collector Resource:** The actual collection of data is achieved through various software elements and systems depending on the type of infrastructure monitored. For example, the physical resource could be monitored by the frameworks such as Nagios, Ganglia or any other monitoring toolkit. For the virtual infrastructure one could use the Libvirt API or any other interface that cloud management software provides. In this data model, the entity information collector resource is defined that keeps the necessary information for accessing any kind of system. By defining in this, the structure the connection details (connection script, connection arguments), the monitoring system can perform a "pull" operation and gather data. The only requirement is that the output of the connection script should thus present a structure compatible with the Monitoring resource.
- **Monitoring Resource:** This entity represents the monitoring information collected from the service, the physical and the virtual infrastructure during the execution of a service. Every record includes the necessary identifiers referring to the respective entities (service resources, physical and virtual infrastructure) in order to set the appropriate relationships. The information collector type field is used in order to validate the consistency of the monitoring report based on the following rule: if the type is "physical", the physical resource ID must be NOT NULL, in order to be able to relate that report with the physical infrastructure. Likewise, when set to "virtual", the VM ID must be filled in and set to "service" with the respective service ID.
- **Service Resource or Instance:** This entity is the glue that relates the service, SLA, physical host and the virtual infrastructure to one another. It describes the actual deployment of a service, with signed SLAs on an IP's infrastructure. It is a very important entity, which keeps the SP centric data model together and allows the SP related

components to keep track of the deployment and execution of a service.

This data model represents the core entities and interactions of the cloud paradigm and is not exhaustive in the list. Each entity could include more detailed fields that could further characterise the actors and interactions. Figure 1 only represents the basic components and how they are interrelated.

4. COLLECTION OF MONITORING INFORMATION IN THE CLOUD

Deployment and execution of applications in highly dynamic infrastructures, such as clouds, introduces a new set of requirements with respect to monitoring that need to be addressed by the developers and providers of the related services. In addition to the user-derived, there are other requirements driven by the constraints and characteristics that new technological trends present. The introduction of the virtualization layer along with the energy efficiency directives applied in the data center operation field have increased the amount of data that must be collected and processed. Furthermore, the elasticity characteristic of the infrastructure requires the monitoring system to be able to keep up when an application or infrastructure scaling it up or down dynamically.

Apart from scalability, such systems should adopt the service-oriented design pattern, which is the keystone of the cloud computing paradigm. The existence of multiple layers such as the physical infrastructure, virtual environment or application layers, each is ultimately associated and dependent on each other. This results in the need of collecting and aggregating all information to lay the foundation of an effective decision taking mechanism. In addition, the storage of data must be performed in an efficient way so that it can be reused by other components of the platform layer such as the performance estimation mechanisms.

4.1. Information Providers and Data Collection

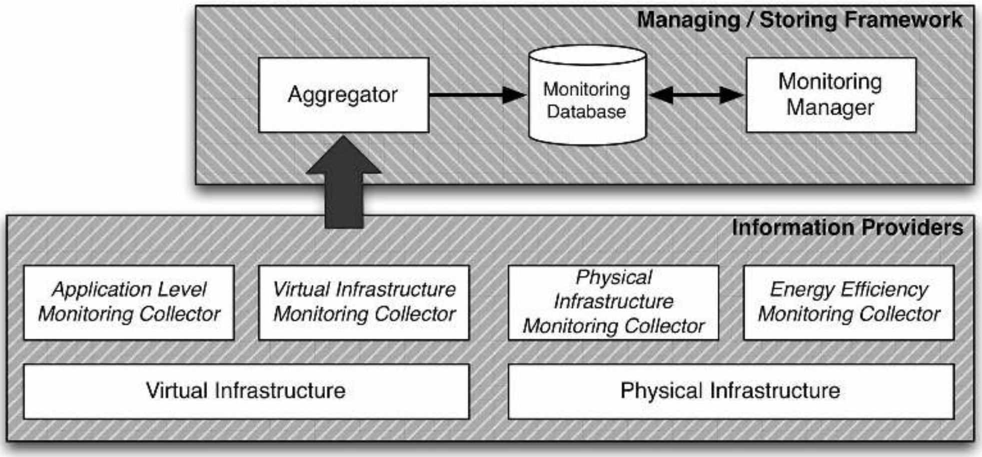
For the monitoring infrastructure, the proposed architectural model incorporates the conceptual layer of the Information Provider (Figure 2). These are the different sources from where the monitoring data can be collected. The Information Providers are considered as the main entities producing information, including the physical infrastructure, the virtual infrastructure and the application specific metrics. In addition, the physical infrastructure data relating to performance metrics can be distinguished from the energy efficiency measurements. The actual collection of information is performed by components named Collectors, which include Energy Efficiency Monitoring Collector, Physical Infrastructure Monitoring Collector, Virtual Infrastructure Monitoring Collector and the Application Level Monitoring Collector. Figure 2 describes the design of the solution, where the layer containing the collectors is scalable. This allows the incorporation of additional sources through their corresponding collectors.

The operation of data collection can be realised either with a "push" model, where the Collectors gather data and push the monitoring report to the Aggregator components in the higher conceptual layer, or a "pull" model, where the Aggregator pulls the data by invoking the respective collector. In addition, a combination of both can be used as well, as proposed in (Huang and Wang, 2010). More details about the operation and implementation of the monitoring collection mechanism can be found in (Katsaros et al, 2011) as well as in (Katsaros et al, 2012) where specific monitoring of energy efficiency is discussed.

4.2. Aggregation and Storage

The second hierarchical level of the proposed approach shown in Figure 2 deals with managing and storing of data. This layer includes the components in charge of man- aging the data collected from the monitoring information providers and storing them in an aggregated way.

Figure 2. Architecture of monitoring infrastructure on the infrastructure provider side


Managing / Storing Framework
Aggregator
Monitoring Database
Monitoring Manager
Information Providers
Application Level Monitoring Collector
Virtual Infrastructure Monitoring Collector
Physical Infrastructure Monitoring Collector
Energy Efficiency Monitoring Collector
Virtual Infrastructure
Physical Infrastructure

The aggregation is done according to data model consistency principles. These state that each monitoring report must contain one identifier that relates it to the sources of information. The structure of the report and the specific identifiers are represented in the data model (Figure 1). Based on these and in combination with the relationships supplied by the proposed data model, it can be ensured that a consumer of the monitoring information can effectively track down any useful datasets from every aspect of the service execution.

Additionally, the Monitoring Manager component serves as the orchestrator of the whole process. The role of this component is actually twofold involving:

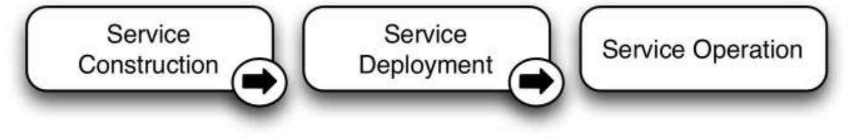
- The capability of controlling the process (start/stop actions) and providing the necessary interfaces to other components of the cloud ecosystem (e.g. evaluators and assessment tools, as discussed later in Section 5).
- As well as to other external consumers of monitoring information (e.g. a Graphical User Interfaces, administrators etc.).

Finally, the collected and aggregated data is stored in a local database for storing the historical information of the deployments and executions. In addition, the Aggregator offers access to monitored data via an interface that provides the last aggregated record, cached within the component for quick access to the status of the infrastructure. This functionality was developed in order to minimise interactions with the database whenever possible to reduce overhead.

5. ASSESSMENT OF MONITORING INFORMATION: ENABLING THE CLOUD SELF-MANAGEMENT

The evaluation of monitoring data is a crucial step towards an efficient and proper self-management of cloud entities, for the services and virtualised resources for both SPs and IPs. Once such information is obtained, a monitoring manager provides it to the several assessment tools. Based on the results of these assessment tools, a business-driven policy management framework can determine and trigger the most suitable management actions in terms of fulfilling both the provider's BLOs and the users' requirements. The assessment tools are based on specific business requirements. This section discusses assessment based on Trust,

Figure 4. Service lifecycle



Risk, Eco-efficiency and Cost for the BLOs of the entities involved.

Virtualisation technology also offers a number of useful actuators, such as dynamic resource allocation to virtual machines, as well as check-pointing and migration of virtualised environments. All of them are indispensable to enact specific cloud-related features, such as elasticity of services and fault tolerance of resources. In addition, evaluation of constraints specified in the SLAs and business-level parameters important to the providers need to be monitored and assessed continuously to ensure compliance with the agreements (Lawrence et al, 2010). In terms of the business-level parameters, this research focuses on the four main business aspects of trust, risk, ecological efficiency and cost (known as TREC parameters). These high-level metrics are mainly used to determine the fulfillment or not of the business-level objectives coming from executives of cloud providers and sometimes from typical BLOs such as to maximise user satisfaction or service availability.

5.1. Evaluation of TREC Parameters: Assessment Tools

There can be various assessment tools to assist in the self-management of cloud environments that can be divided based on evaluation of SLA-related parameters and high-level TREC parameters. Figure 4 depicts the three stages of a typical service from construction, deployment and operation. Optimisation of a complete service lifecycle starts from the service construction, which includes evaluation based on trust, risk, eco-efficiency and cost, through till the end when the service is in operation. This will then present an optimised cloud ecosystem based on the trust among the consumers and providers and the risk of not accomplishing the ecological and economical goals. These are known as the TREC factors for Trust, Risk, Eco-Efficiency and Cost. Based on these evaluations, the cloud can self-preserve itself, predict problems in the future and make use of self-adapting solutions. The definitions and scope of these factors can be dependent on the interpretation of the users. The definitions used in this paper are as follows:

- **Trust:** The trustworthiness of the SP and the IP is measured by considering the past performance track record and the legal profile of the provider being assessed. This involves checking historical data about the past transactions and social recommendations recorded about the providers by others.
- **Risk:** The risk factor assesses how risky it would be for the SP or the IP to execute the service. This is measured with regards to various factors such as past service performance, hardware and virtual machine performance, security certifications, maintenance data, legal issues and historical data.
- **Ecological Efficiency:** The eco-efficiency factor takes into account the constraints of minimising energy usage during the operation of a service. This takes into account the renewable green and brown energy profiles of the providers.
- **Cost:** This refers to the cost of ownership of a service during its lifecycle including the resource acquisition cost, usage costs and penalty costs on service failure.

Figure 3. Assessment framework

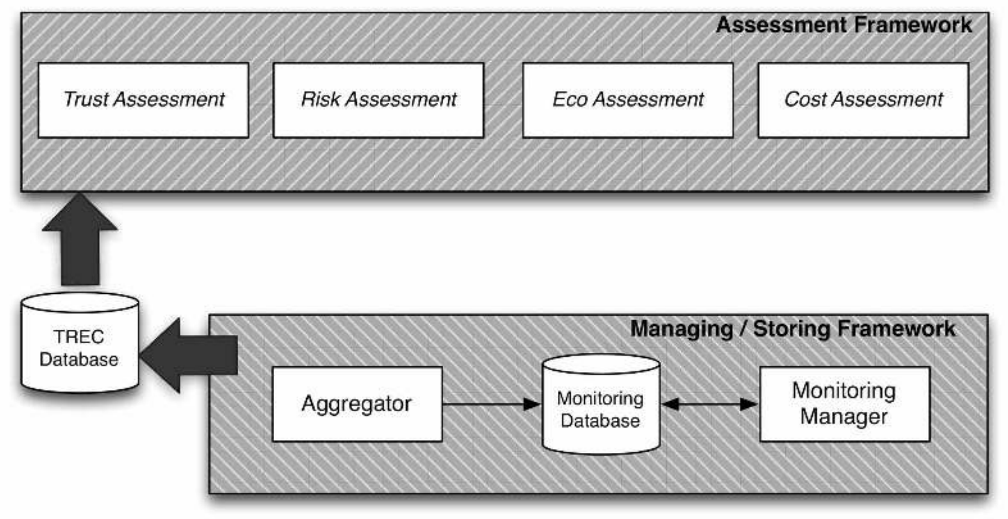


Figure 3 shows the assessment framework, which contains a dedicated TREC database interacting with the monitoring infrastructure. The monitoring infrastructure collects data from its collectors and feeds it into the TREC database. The TREC components, at the assessment level, can then continuously use this TREC database to calculate the values for their trust, risk eco-efficiency and cost factors. This dedicated filtering reduces the overhead involved if the complete monitoring database has to be looked up which involves traversing unnecessary information. The TREC database keeps an up-to date catalogue of data fields which are repeatedly used by the TREC components reducing the latency between the TREC components constantly querying the monitoring database for the same information, allowing this to be pooled only once.

5.1.1. TREC Assessment System

The various parties involved in the cloud ecosystem have different interests reflected in their business level objectives (BLOs) and use different strategies implemented to satisfy these. These policies determine the actions taken and the monitoring information that must be gathered. For instance, from the point of view of the end-user, the primary interest lies in guaranteeing that a service will adhere to the Service Level Agreement made with the specified functional and non-functional parameters defined. This requirement is satisfied by a continuous monitoring of the service state during the deployment and the operation phases.

Before deploying a service, a service provider is responsible for selecting the appropriate infrastructure provider for the given service execution. Both the SP and the IP use monitoring information at both stages of deployment and operation, for example the IP's physical and virtual infrastructure details. An IP would then use the monitoring information to optimise the usage and consolidation of its physical infrastructure dedicated to executing this particular service and maximise its business level objectives like profitability as well.

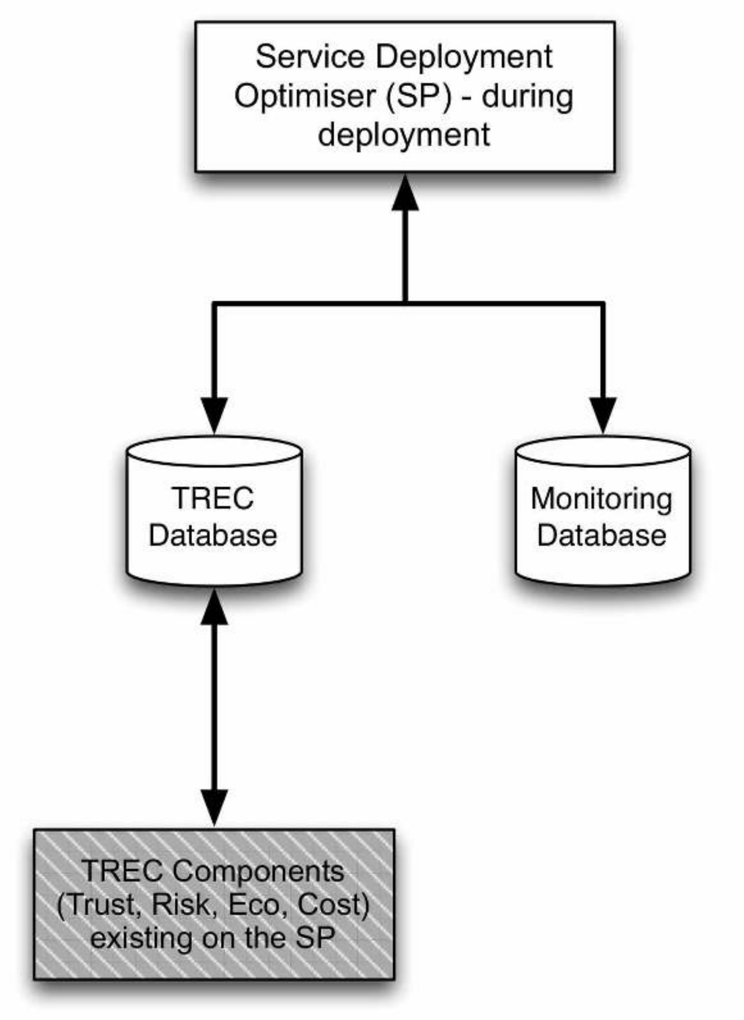
The TREC Assessment tools act as a filter mechanism at service deployment and during service execution. These tools provide suggestions to other components, so that an SP or IP can make improved informed decisions that optimise the use of a cloud in the context of these factors. In essence, monitoring information can be used to prevent SLAs from being breached, by taking appropriate actions if some measurements are nearing or have reached the Quality of Service (QoS) thresholds. To optimise the cloud using predefined business level objectives, monitoring information can be used to enhance global decisions that are in the interest to the actor.

The consumers of this monitoring information could be various components such as the TREC Assessment Tools, the SLA Manager, Cloud Optimiser, the Elasticity Engine and more (Optimis Consortium, 2011). The result of assessing this information causes these components to react to the changes in the environment and optimize the operation of the Cloud. Depending on the thresholds, these components can take or advise other components to take action to minimise failure risks. They can also act differently during the various stages of a service life cycle. For instance, during the deployment phase, the components such as the TREC components would use in- formation which is prerecorded as in the historical database to make their decisions for deploying the service to the IPs. In another instance, during the operation phase, the TREC components will use the live data accessed, to make deductions on the live service performance to reveal failure risks.

Figure 5 describes the components on the service provider accessing the monitoring and TREC components. The main component is the service deployment optimizer (SDO), which during deployment uses the historical knowledge from the TREC components; TREC database and the monitoring database to assess which infrastructure to deploy to. This operation is only involved during deployment of a service, and may in the future; monitor the live data during the operation phase. But this is highly dependent on the kind of data being accessed at that stage. Figure 6 describes the components interacting with the TREC database and Monitoring database on the infrastructure provider. During deployment, the admission controller will use the historical data from these databases to make a decision to accept the service request from the service provider. Once accepted and deployed, the other components will use the live data in different ways to monitor the service. The virtual machine manager and data manager will manage the virtual machines and the data associated with the service and their placements. The cloud optimiser will try to find optimum ways in which the service can be optimised on the available resources. The fault tolerance engine will use the live data to find situations which may be a threat to the service or the infrastructure putting in place fault tolerance mechanisms.

The TREC and monitoring database exist as two separate entities on the service provider or infrastructure provider. This depicts these entities as independent acting as a complete system replicating how various industries either provide service or infrastructure provider facilities. In addition to the placement of these components on the entities, the components can have different natures - reactive or proactive. A reactive component, would read the data from the environment, and if something wrong has occurred, it would then take appropriate actions to resolve it. However a proactive component would monitor data and predict if something wrong is about to happen and prevent it. In the case of Cloud computing, both kinds of natures would allow a self-healing system to be developed. In the scenarios, described in Figures 5 and 6, the components present a more proactive nature, as the TREC components can monitor data from the associated databases and calculate values for each trust, risk, eco-efficiency and cost. These values are calculated based on their own mathematical models and can help other components to make more informed decisions to prevent failure of services. These components would thus reflect on the perspectives of either the SP or the IP and make decisions to optimise their own BLOs. The decisions made could be in the form of the if-then activities, which can either treat the four factors of TREC independently or as dependent on each other. Examples of such situations could use relationships between trust and risk or eco-efficiency and cost. However this is the subject of further research and out of the scope of the current research.

Figure 5. High level diagram of TREC interacting with other components on the SP


Service Deployment Optimiser (SP) - during deployment
TREC Database
Monitoring Database
TREC Components (Trust, Risk, Eco, Cost) existing on the SP

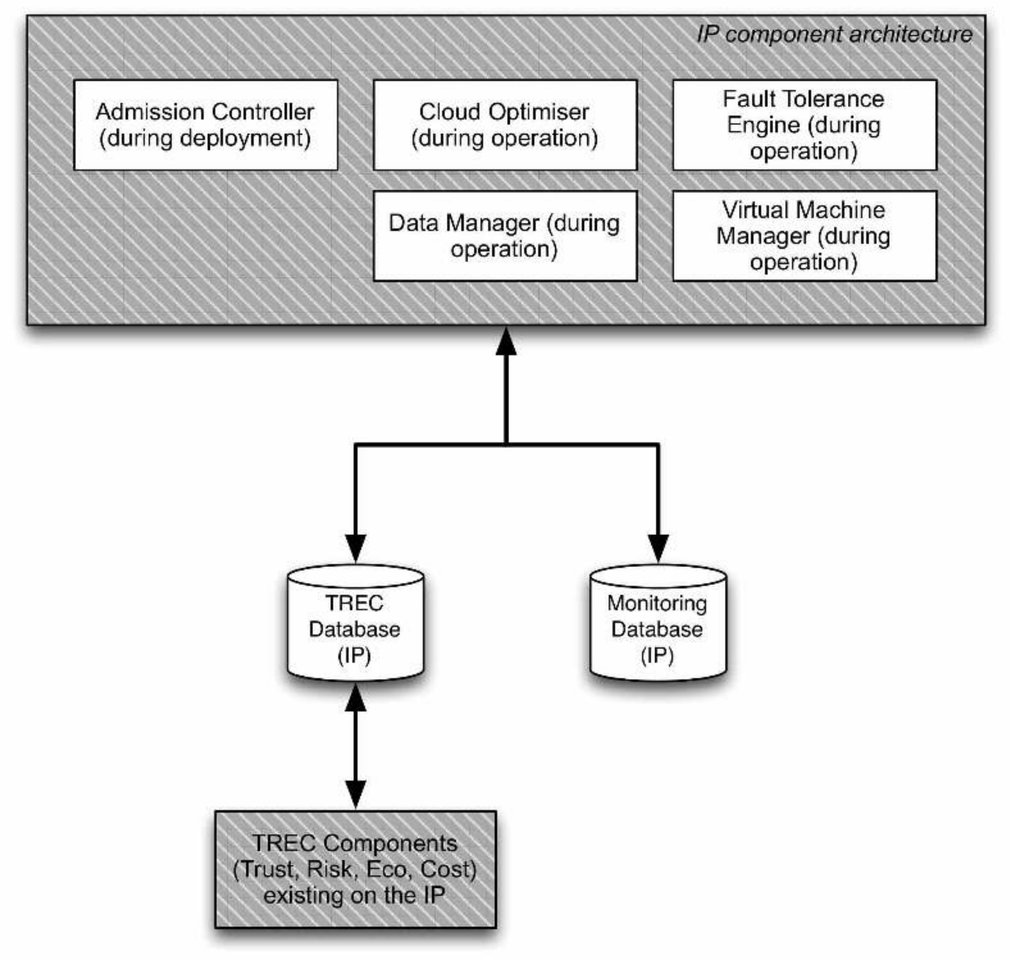
5.1.2. Trust Component

The concept of trust can have a direct relationship to reputation and security in the context of cloud computing. Choosing a trustworthy service or infrastructure provider is quite important because it harbours, in some cases, confidential data which may be located outside the end-users possession.

As part of the TREC parameters, the Trust Framework needs to perform different actions from both perspectives of the actors. From the point of view of the service provider, the framework will gather information on any other service running on the infrastructure which may be a potential threat to its service. From the infrastructure's point of view, the information will be managed and stored so that it allows the framework to generate a historical asset to calculate the trust value. This also involves the social network between the SPs and IPs being saved along with historical data.

Figure 7 depicts the various steps involved in using the framework. The TREC database

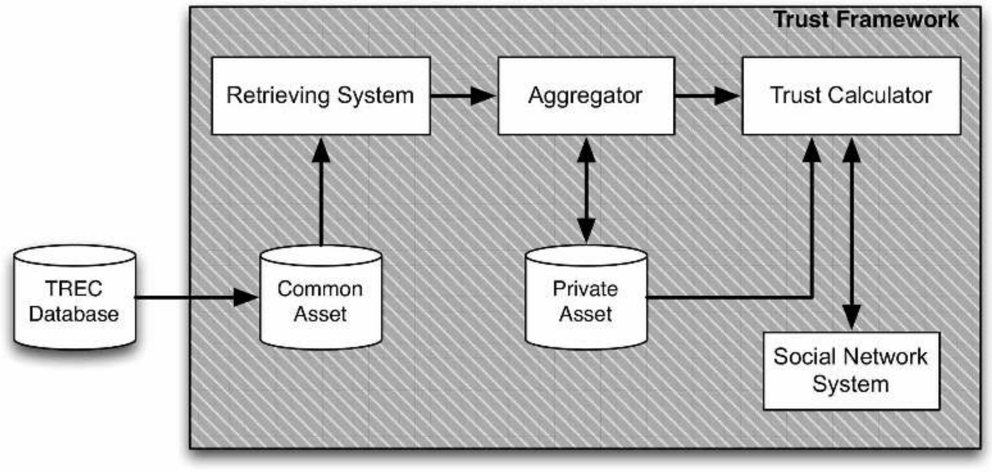
Figure 6. High level diagram of TREC interacting with other components on the IP


IP component architecture
Admission Controller (during deployment)
Cloud Optimiser (during operation)
Fault Tolerance Engine (during operation)
Data Manager (during operation)
Virtual Machine Manager (during operation)
TREC Database (IP)
Monitoring Database (IP)
TREC Components (Trust, Risk, Eco, Cost) existing on the IP

has already filtered all relevant data from the monitoring infrastructure with timestamps. The historical raw data is then accessed directly by calling the common asset. This common asset is a raw data asset that contains data about the lifecycle of a service deployed in the cloud, from the Deploy operation through to the Undeploy or end of its life. Aggregated values like reliability, age, robustness are calculated based on other values collected within the trust framework. These are used by the Trust Calculator, along with the social network, to generate and manage the relationships to produce a trust value.

The Trust calculator uses the trust model based on subjective (Josang, 2011), fuzzy logic (Stepnicka et al, 2010) and different techniques of data comparison. The data coming from the social network is used as a controller on the trust calculation, such that it controls big changes originating in the common monitoring system (such as system or network failures). More details about the principles and the techniques used for the implementation of the trust assessment tool are presented in (Nieto, 2011) (Josang et al, 2007).

Figure 7. Trust framework using the TREC database



5.1.3. Risk Component

In the most general sense, risk can be defined as a combination of the probability of an event occurring and its consequences. This constitutes both the opportunities for benefit (upside) and the threats to success (downside) to the entity. Although risk and trust can be related in concept, the risk in the Cloud is more focused towards the actors BLOs being satisfied. Here the risk component utilises the monitoring information to enable self-management of the cloud. Risk can also be assessed at various stages of the service lifecycle (deployment (Figure 8) and operation (Figure 9) and the actors involved. For instance, the service provider needs to perform a risk assessment on the various infrastructure providers available in order to choose one of them for its purpose. Once this is done, the service provider can only monitor the performance of the service on the infrastructure depending on the data available to it. However, from the infrastructure provider's level, the risk can be assessed at deployment stage, depending on historical data to work with the suggested service provider and at operation stage, a continual risk assessment is needed on the service being executed, for assessing the risk of it failing.

Risk analysis requires data to be collected in two forms: static and dynamic. The static data is accessed using the historical database collected over a time period of past dealings between the service and infrastructure providers. Dynamic data is collected via the monitoring component and continual risk analysis is performed on it. Monitoring information is thus used by several components to perform assessment on the dynamic environment of a cloud.

Figure 8 depicts the risk architecture that exists on the SP. The monitoring database feeds information to the TREC database, which is then read by the risk assessment tools. These comprise of the assessors, historical database and the risk inventory which work collectively to produce a risk value at the deployment and operation. At operation this architecture is quite limited as it depends on the data coming from the IP, which may not be released due to legal or business issues. Figure 9 depicts the risk architecture existing on the IP. Unlike the previous figure, this architecture is quite detailed as it has access to more information. The monitoring tools feed information to the TREC database where it can be accessed by the assessors, historical database and the risk inventory. Examples of such information used are the current workload, system outages,

Figure 8. Service provider - risk assessment components

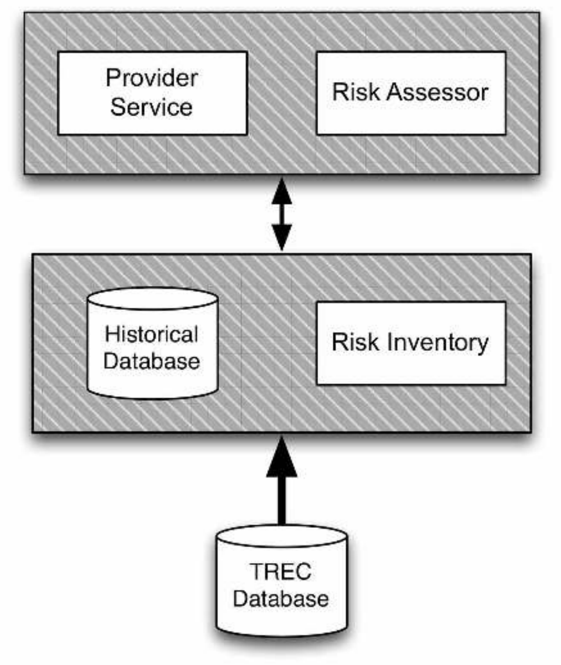


Figure 9. Infrastructure provider - risk assessment components

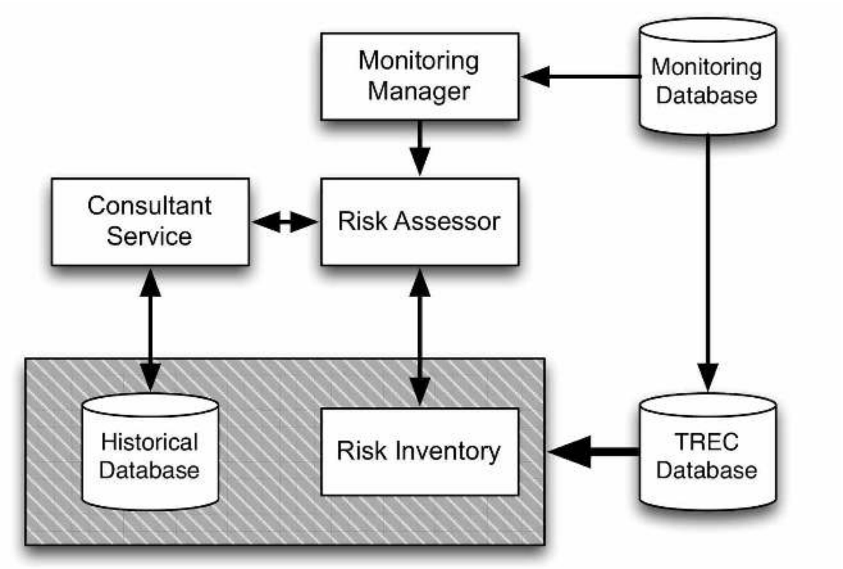


Table 1. Dynamic data - data snippet for risk associated with redeployment of data

Asset identified: SLA
Vulnerability of asset: Lack of jurisdiction information
Threat to asset: Breach in data confidentiality
Resulting risk item: Changes in jurisdiction
Risk Category item belongs to: Policy
Risk Likelihood: Very high (5) [Range 1-5]
Risk Impact: High (4) [Range 1-5]
Resulting Risk level: Product of risk likelihood and risk impact [Range 1-25]
Risk event: Redeployment of data
Resulting risk mitigation: Seek legal advice

temporary performance shortages, monitored network traffic, expert availability or general information regarding the number of services to operate. The monitored data helps to determine bottlenecks in the IP's infrastructure such that the provider can improve its capacity planning, administration and management of its resources.

During the risk analysis process, the risk tools assess the risk based on certain categories focused for the fulfillment of the SLA. These categories help simplify the different kinds of risks being assessed for the IP. The various risk categories identified are as follows with the example risk items:

1. **Legal:** SLA issues collecting data parses from SLAs and infrastructure details.
2. **Technical:** Hardware, VM failure collecting live data on downtime and uptime of services.
3. **Policy:** Data jurisdiction policies collecting live data from the data management tools.
4. **General:** Various issues such as security collecting active antivirus and login logs and parsing them to determine discrepancies.

Some of these risk items are static in nature, but some such as technical or legal issues must be constantly monitored using dynamic data from the monitoring tool to assess their risk. Table 1 shows the example of such dynamic data. Each risk category can then adopt its own risk algorithm which can be used to calculate a value. A collection of all of these can then be used to determine an aggregated risk for the IP or service failing.

The calculation of the risk probability can be dependent on the assets being assessed. These can be based on user or expert data to predict the likelihood and impact of failure of particular assets. Usually the risk factor is a product of the likelihood and impact in certain models. But these can be based on historical assessment depending on the business model used.

The various risk models can allows associated mitigation solutions where possible. Appropriate mitigation strategies can be suggested by the risk assessment tool to the respective actors (Gary Stoneburner et al, 2002). These mitigation strategies can also be assessed by the data from the monitoring tool to see how they affected the performance of the service.

5.1.4. Ecological (Eco) Efficiency Component

Energy consumption is one of the major concerns for today's data centers. Especially for cloud providers, an energy-efficient management of cloud services and infrastructures is imperative. A number of management policies and actuators have focused on reducing the energy consumption of data centers (Goiri et al, 2010). Because of virtualisation, several services can run on the same physical host without affecting its performance and security. This consolidation is also a very common habitual practice for minimising energy consumption of data centers (Srikantaiah et al, 2008). In addition, there are complementary techniques, such as power on/

off nodes and Dynamic Voltage/Frequency Scaling (DVFS) (Chen et al, 2005), which also help greatly in achieving high energy efficiency objective. However, these energy-aware actions have been used by considering only IT-level metrics such as power consumption. Further management policies can be proposed which consider high-level aspects such as the source of energy used, like renewable (green) or non-renewable (brown).

Therefore, ecological efficiency can be used in decision-making processes, which should be defined as a combination of many energy-related facets. The eco-efficiency tool is responsible for evaluating and forecasting eco-related aspects, such as energy efficiency and carbon emission levels. In particular, it performs the monitoring and assessment of energy efficiency through different metrics known as Key Performance Indicators (KPIs) and Green Performance Indicators (GPIs). These indicators are derived from variables that are measured and monitored in the cloud environment at different levels (facilities, server hardware, and virtual systems) by means of physical sensors. The assessment is then performed based on the relation between the measured KPIs and GPIs (or a combination), or other generic parameters such as minimum acceptable levels of other TREC parameters.

Moreover, there can be three different use cases for this assessment tool, depending on the entity that is being assessed. These could be a complete IP infrastructure, a given node or a particular cloud service. An SP can only request eco-efficiency data of a service already deployed and running on top of the virtualised infrastructure. However the IP is capable of knowing the eco-efficiency of its infrastructure or of the particular physical node in order to determine the best one for virtual machine (VM) consolidation. However in multi-cloud scenarios, such as in cloud federations, providers consider the eco-efficiency of the third-party providers only when outsourcing virtual machines to them. Assessments and predictions produced by the eco-efficiency tool can be used (directly or indirectly) by several management entities to optimise their solutions. One example of these components could be the Elasticity Engine that would take into account the energy-related issues, when setting management policies or rules for performing appropriate virtualisation-level management actions.

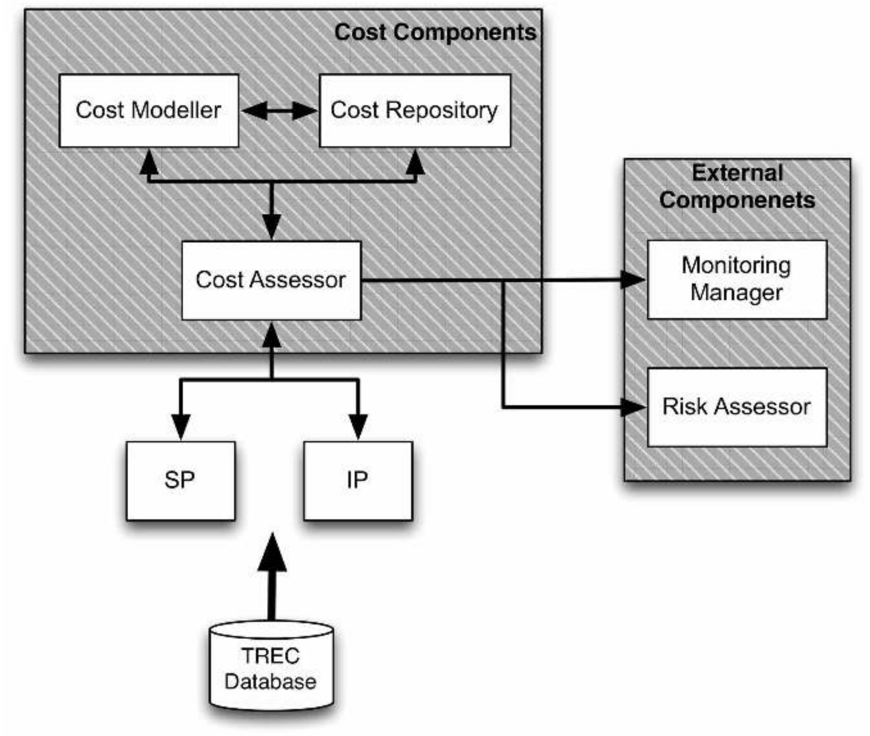
5.1.5. Cost Component

Along with agility, cost is one of the key drivers in the move towards cloud computing. There are two critical components for cost management which include the current costs and predicting the future costs. Both of these capabilities are data driven activities requiring both historical and real time information for their calculations. Figure 10 depicts the architecture of the cost module. The cost module can be deployed independently by both the SP and IP. It is necessary to accurately assess the cost of providing a service, whether at the SP or the IP level, to enable transparent billing. This capability allows organisations to accurately predict and manage their costs to ensure their profitability.

The cost module is composed of three components - the Cost Assessor, the Cost Modeller and the Cost Repository (Figure 10). The Cost Assessor is the central communication hub managing all requests from external components and the TREC database. This then feeds to the Cost Modeller which is responsible for defining and executing the mathematical model to accurately record and predict the cost of service deployment and operation. In order to build this model, the modeller utilizes statistical techniques on the stored data in the cost repository and the data provided through the TREC database. Examples of such data include, data for the SP, the hosting charges (coming from IP), the service usage or SLA defined levels. In case of the IP, data on hosting income (from SP), service usage, SLA levels, infrastructure utilisation and the energy usage can be examples of data needed.

The Cost Repository acts as a persistent store for all previous cost analysis (predicted or realised) and the relevant cost data gathered from the TREC, monitoring and SLA/SLO managers.

Figure 10. Cost assessment component



The described assessment and prediction functions provided by the cost tool can then be used to support the decision making process of both the SPs and IPs. In combination with suitable management policies, the cost component can enable the optimised economic management of a service.

5.2. Self-Management of Cloud Entities

Cloud environments are dynamic and constantly subjected to changes and events that directly or indirectly affect their operation. Among others, unpredictable workload surges and resource failures are typical examples of such (unexpected) circumstances. Therefore, having a self-management system for cloud environments is essential for the success of cloud providers. Successful self-management solutions need tools to monitor and assess the cloud status, such as management engines that can take decisions or low-level actuators that can carry out the decisions.

Low-level actuators can be easily enabled by means of the virtualisation technology as they can dynamically allocate or deallocate resources from or to virtual machines when necessary, particularly due to time-varying workloads. These can also migrate virtual machines when resource failures are foreseen. However, there is a need to build tools that are able to gather monitoring information in an efficient way because of the large amounts of information being considered.

In addition, this information should also be processed and aggregated to assess current status and foreseen impact of potential management actions. This aggregation should be considered from the low-level infrastructure parameters as well as the high-level business ones, all of them linked together to the four factors - risk, trust, cost or eco-efficiency. The results provided by these assessment tools will be crucial inputs for autonomous decision-making processes in cloud providers, leading to the self- management of cloud entities, such as services and infrastructures. This autonomous management can also be aligned with the high-level expectations of providers, their interests and goals (BLOs), as well as with the service-related requirements specified by the owners.

According to this, the self-management middleware for cloud providers should include the Business-Driven IT Management (BDIM) discipline (Sauve et al, 2006b). The main goal would thus be to consider the business impact of IT (cloud) management actions prior to their implementations. This business-driven approach is encapsulated in a governing management component, such as a provider optimiser. It comprises of a policy management framework that incorporates BLOs in a global decision-making process for the cloud providers, where management policies need to be fully guided by business-level TREC parameters and metrics. Actually, cloud providers would be composed of several interrelated management components, which have to be configurable by the policy framework present.

There can be different BLOs desirable in cloud providers, such as the maximisation of their profits, the minimisation of the ecological impact or the maximisation of their customer satisfaction. In order to accomplish objectives like these, management policies must take into account several parameters such as performance, reliability, SLA requirements and infrastructure usage. However, the business-level parameters which are the results of the TREC tools are also involved. In the end, an optimized self-management of cloud entities is achieved for fulfilling the provider BLOs.

5.2.1. Using the TREC Assessments in Decision-Making: Example Case

TREC guided assessment can aid the business-driven self-management of the cloud for determining the fulfillment of the provider's BLOs. In this section, several examples of how these TREC parameters can be used in cloud providers of services or infrastructures are presented:

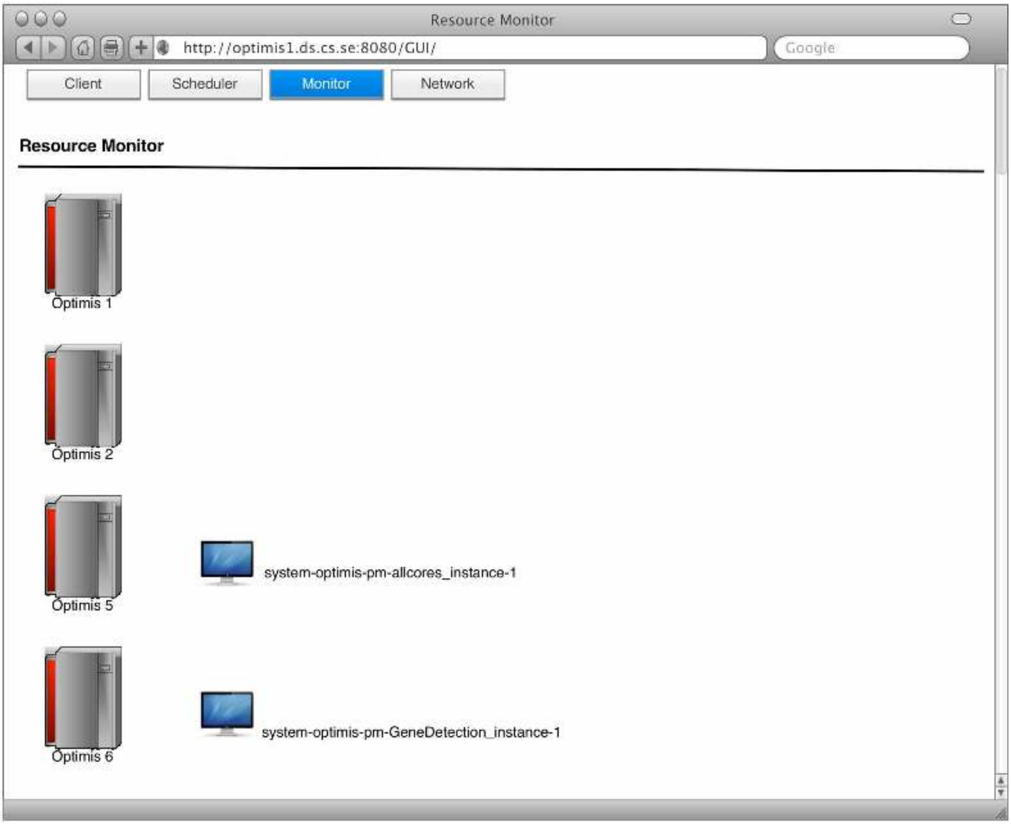
Consider the case of the service provider (SP), which upon receiving a service manifest from the service developer (user), sends this manifest as a request for offers to one or more IPs (or to a broker). After potentially multiple rounds of negotiation, it selects which provider(s) to deploy the service based on the constraints and objectives of the service (TREC-factors, data location restrictions, provider price offers and more). This also includes the SP's high-level objectives such as the trust factor of the different IPs calculated from the historical data and the past dealings with the IPs. The price (cost) offered by IPs is another common parameter taken into consideration by business executives.

Once the service is deployed, contextualised and operating at an IP, the SP could adapt its operation to face changes in the underlying cloud environment. For instance, it can consider the cost-benefit of adding more VMs to a service so that it does not violate the SLA. If the cost of adding a new VM is greater than the penalty to be paid, due to the violation of the SLA, the operation would not be performed, as the manager aims to maximise the provider's profitability.

The TREC assessment tools can also assist in the decision of the best IP(s) to redeploy services, in case of unacceptable low performance of the IP. This approach can be used during the whole service lifecycle, such as deployment and operation, in order to fulfill expectations of SPs and their customers.

Consequently, regarding the infrastructure provider (IP), the optimiser component (known as Cloud Optimiser) has the main goal of optimising the use of the underlying cloud infrastructure in terms of the IP's BLO fulfillment. The decision-making process, governed by the business-driven policy management framework, uses parameters such as performance, SLAs, infrastructure usage and the TREC factors. The decisions taken by this component consider all kinds of high and low-level parameters in a synergistic way depending

Figure 11. Usecase showing two virtual machine instances running on machines optimis5 and optimis6


Resource Monitor
http://optimis1.ds.cs.se:8080/GUI/
Google
Client
Scheduler
Monitor
Network
Resource Monitor
Optimis 1
Optimis 2
Optimis 5
system-optimis-pm-allcores_instance-1
Optimis 6
system-optimis-pm-GeneDetection_instance-1

on the provider's interests such as assessing the revenue loss incurred due to poor performance of services running on the virtualized resources.

Another example is that for each request to start a VM, resulting from either the service deployment or requests for enacting elasticity and fault tolerance corrective actions, the Cloud Optimiser can use the internal policies and TREC assessments to decide whether this VM can be executed on local resources. In case these are outsourced, additional resources to an external provider can be accessed as a profitable decision.

In the case the new VM is accepted, it is forwarded to another management component, the VM Manager, which is in charge of scheduling the VM by using the virtualisation actuators. In particular, given a set of VMs running in a physical infrastructure, the VM Manager's main task is to optimise how these VMs are placed on the physical resources so that the IP's internal goals are maximised. Thus the virtualisation-level manager can be aimed to optimise the placement of a set of VMs during the whole lifecycle. At any given moment the VM Manager is capable of re- organising the mapping of VMs to physical resources according to the IP's BLOs. The VM manager will also need to request new assessments and predictions from the TREC tools, in order to find the best action when scheduling VMs over an infrastructure. For instance, it could try to consolidate all the VMs running in the provider using the minimum number of nodes (necessary to fulfill the SLAs) and shut down the rest of nodes, reducing power consumption. It can also choose to migrate VMs running on a node where its risk of failure is reaching unacceptable levels.

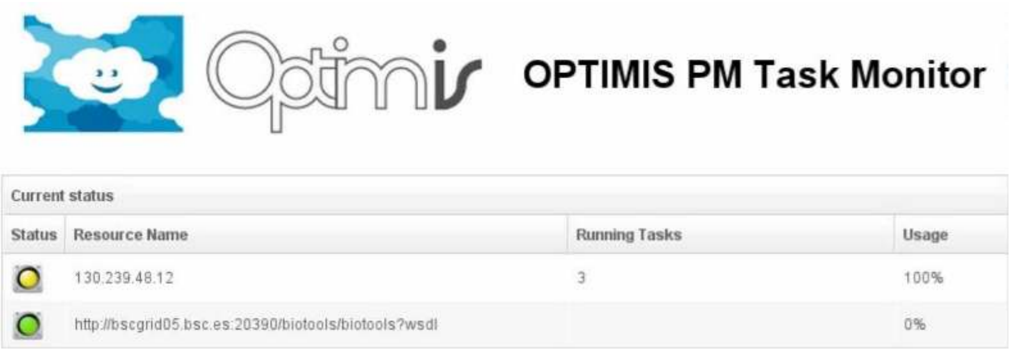
There are other management components that can predict their operations directly on the TREC and monitoring information. For instance, the Fault Tolerance Engine, which aims to ensure the self-healing (fault tolerance) of a cloud infrastructure, can ask for periodic updates from the monitoring system about the state of physical hardware devices as well as the virtual IT infrastructure (or the VMs). Based on the internal fault tolerance rules, this engine then decides whether any corrective action is required, such as booting a new VM to replace a crashed one, replicating data objects, or migrating a VM from one node to another. The Fault Tolerance Engine can communicate to the Cloud Optimiser to perform these actions. A resource monitor can be used to track the instances running on the machines.

An example of a fault tolerance execution is shown in Figures 11 - 13. The case study describes the monitoring on an IP machine during the execution of a service. There are two instances of virtual machines running on two IP machines, *optimis5* and *optimis6*. Figure 12 shows that during execution one of the machine's usages has increased to 100% which may eventually cause it to fail. The risk assessment framework on this IP, during operation, is collecting information through the monitoring framework and feeding through to the risk tools to calculate the risk level on the machine. The graph (Figure 14) shows the risk associated with the service running on the virtual machines on the particular IP machine is going up at time *t = T -14*. *T* denotes the current time. The risk units are based on a scale of (0-1) which depicts the risk probability. The risk model used here, calculates risk based on the availability of disk space, increasing as time progresses due to data being written out during the service execution.

- $\text{Probability}_{\text{Risk}}$ =β x $\text{disk}_{\text{space}}$, where β is a constant used by the provider to estimate the relationship of disk space with risk.
- If ($\text{Probability}_{\text{Risk}}$ >threshold), then generate a mitigation strategy from the risk inventory database, such as suggesting to move the virtual machine to another physical resource.

Figure 14 depicts an increase of risk level from 0.2 to 0.6 units at time *t =T-11*. At this stage, the fault tolerance engine, which is a proactive component, signals that the risk is going up, and if it has goes above a certain threshold, the risk component can suggest mitigation strategies to overcome this problem. In this case it asks for the virtual machine to be hosted to another IP machine due to more space requirements (*optimis1*). Once this is moved, the risk is seen to start reducing as more space is now available for data write out. Therefore the proactive assessment can use the monitoring tools to make

Figure 12. Snapshot showing memory usage of the two machines optimis5 and optimis6

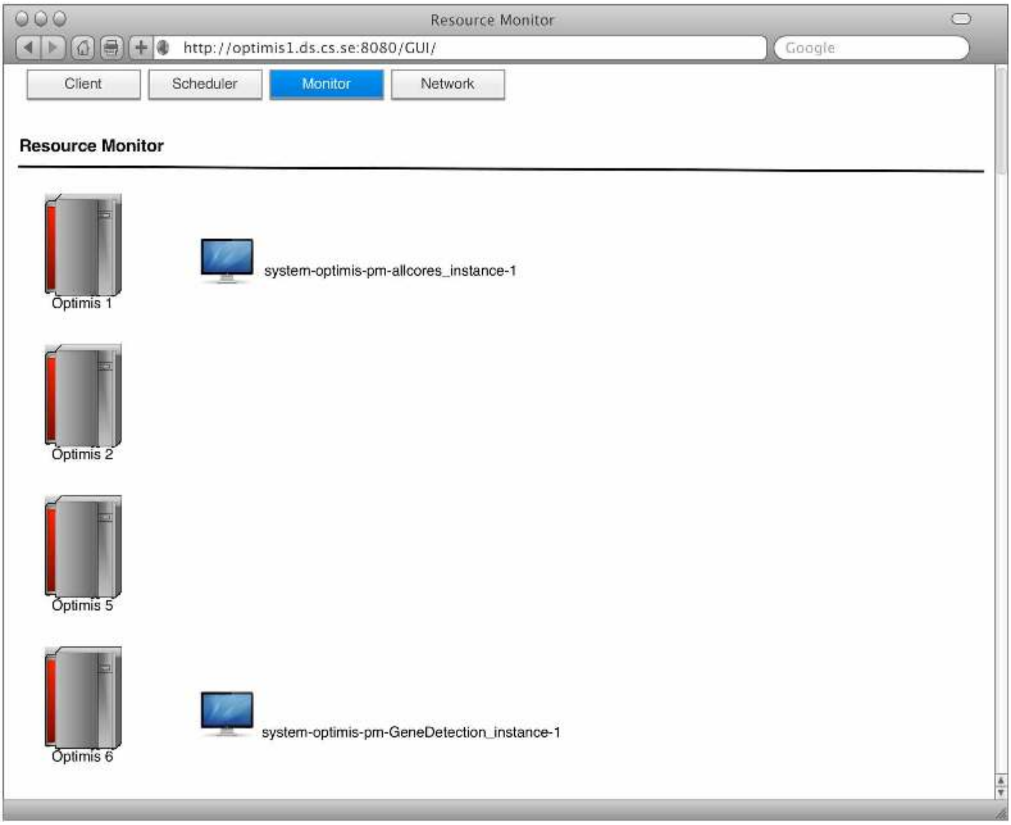


OPTIMIS PM Task Monitor

Current status

Status	Resource Name	Running Tasks	Usage
●	130.239.48.12	3	100%
●	http://bscgrid05.bsc.es:20390/biotools/biotools?wsdl		0%

Figure 13. One of the virtual machine instance is moved from optimis5 to optimis1 because it was about to fail


Resource Monitor
http://optimis1.ds.cs.se:8080/GUI/
Google
Client
Scheduler
Monitor
Network
Resource Monitor
Optimis 1
system-optimis-pm-allcores_instance-1
Optimis 2
Optimis 5
Optimis 6
system-optimis-pm-GeneDetection_instance-1

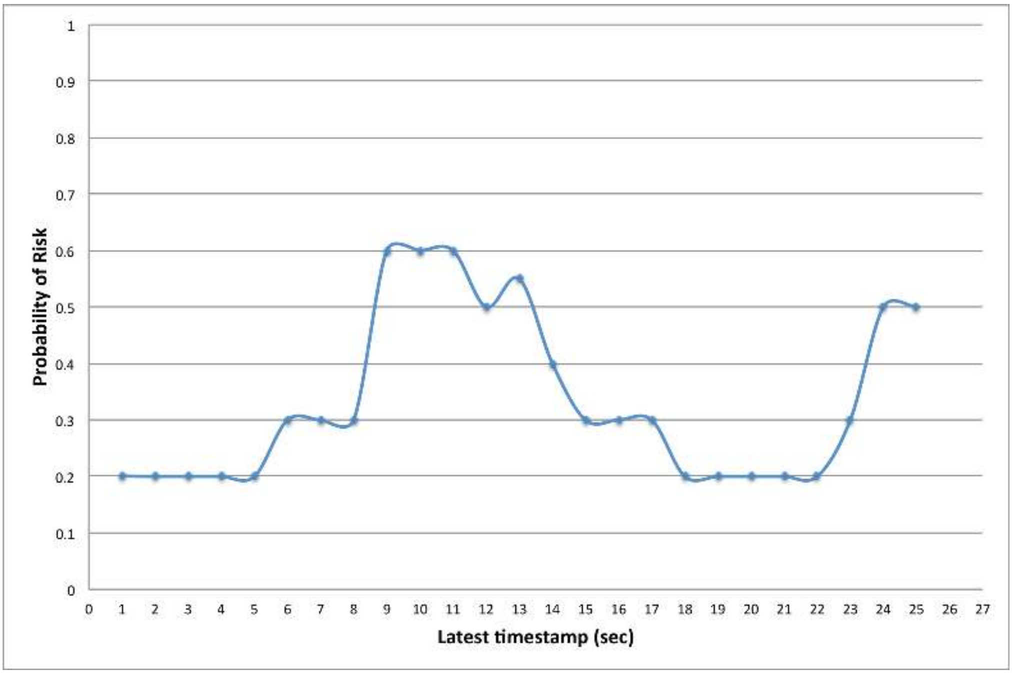
more informed decisions to ensure complete and successful execution of services.

Summarising, management components in cloud providers need to be configurable through business-driven internal policies and the decisions taken can be supported by assessments provided by the TREC tools. Thus the provider is able to efficiently manage cloud entities in terms of fulfilling user requirements and provider BLOs.

6. CONCLUSION AND FUTURE WORK

This paper discussed an efficient information management within the cloud ecosystems and its usage in multiple scenarios. A monitoring infrastructure is designed on top of a high level cloud-enabled data model, which incorporates all the main entities of a cloud scenario and glues them together with certain relationships. The monitoring infrastructure is able to collect, aggregate and store information from various sources either the service or infrastructure providers. Through the support of a twofold (push and pull) operation and the distinction between collection and aggregation, a scalable, easy to configure and interoperable monitoring system can be devised. Current research approaches have used collected data mainly for verifying the SLAs and detecting possible violations. This paper has introduced a new set of factors that will consume the monitoring information and assess the service execution in the cloud in terms of Trust, Risk, Energy Efficiency and

Figure 14. Risk probability changing with time



Cost (i.e. TREC parameters). By using the outcomes of such assessment tools, any cloud provider is able to apply the most efficient low-level management policies in terms of fulfilling both its business-level objectives (BLOs) and the user requirements. Therefore, cloud providers are capable of self-managing their own cloud environment by means of TREC assessments and actuators provided by the underlying virtualisation technology, for example the dynamic resource provisioning to services. Another aspect is the relationship among the four TREC components, which should be investigated further to determine if these can be treated as one formula to make a better self informed decision when managing the cloud environment. This will involve studying the smaller data feeding into each of the models and then checking for overlapping dependencies among the factors. Trust and risk can be argued to have an inverse relationship, simply by the fact that more trust means less risk and vice versa. Cost and eco-efficiency can be related in terms of the costs per energy used, but the exact values of their relationships can be investigated further.

The monitoring information on the state of the cloud infrastructure, in connection to software or hardware failures, can also be used to improve fault tolerance by taking appropriate actions on the components. These actions can be suggested by the proactive or reactive assessment tools, to improve the reliability of the cloud for a given service. For example, in order to maintain the elasticity of the cloud and maximize the consolidation of resources, there is a need to access the current state of the virtual and physical infrastructures to determine when it is best to scale an application given a specific work load. In future work, the presented architectural model will be applied to more complex cloud scenarios such as cloud bursting scenarios (where a company owning its own private cloud infrastructure accepts to, for some time and given a set of circumstances, decides to use resources of an external cloud provider), cloud brokerage (enable brokerage based cloud federation) and multi-cloud scenarios (an SP deploys one service to multiple IPs). This will allow verification of the consistency of the data model, extending it, if necessary and improving the architectural approach where needed.

ACKNOWLEDGMENT

This work has been supported by the OPTIMIS project (http://www.optimis-project.eu/) and has been partly funded by the European Commission's IST activity of the 7th Framework Program under contract number 257115. This work expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this work. It is also supported by the Ministry of Science and Technology of Spain and the European Union (FEDER funds) under contract TIN2012-34557, by the Generalitat de Catalunya under contract 2014-SGR-1051.

REFERENCES

Aiber, S., Gilat, D., Landau, A., Razinkov, N., Sela, A., & Wasserkrug, S. (2004) Autonomic self-optimization according to business objectives. In: International Conference on Autonomic Computing, 2004. Proceedings., pp 206–213 doi:10.1109/ICAC.2004.1301365

Andreozzi, S., Canaparo, M., & Carpene, M. (2008) Glueman: a wbem-based framework for information providers in grid services. In: *12th Enterprise Distributed Object Computing Conference*., pp 377–384 doi:10.1109/EDOCW.2008.34

Buyya, R., Yeo, C. S., & Venugopal, S. (2008) Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. In: High Performance Computing and Communications, HPCC '08. 10th IEEE International Conference on, pp 5–13

Chen, Y., Das, A., Qin, W., Sivasubramaniam, A., Wang, Q., & Gautam, N. (2005) Managing server energy and operational costs in hosting centers. SIGMETRICS PerformEval Rev 33:303–314, DOI http://doi.acm.org/10.1145/1071690.1064253

D-GRID Project Consortium. (2009) D-MON Project: http://www.d-grid-gmbh.de

(DMTF) DMTF. (2009) Ovf open virtualization format. URL http://www.dmtf.org/standards/ovf

Ejarque J, de Palol M, Goiri ´I, Juli`a F, Guitart J, Badia RM, Torres J (2010) Exploiting semantics and virtualization for SLA-driven resource allocation in service providers. Concurr Comput: Pract Exper 22:541–572, URL10.1002/cpe.v22:5

Ferrer, A. J., Hernandez, F., Tordsson, J., Elmroth, E., Zsigri, C., Sirvent, R., & Sheridan, C. et al. (2012). Optimis: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, *28*(1), 66–77. doi:10.1016/j.future.2011.05.022

Field, L., Andreozzi, S., & Konya, B. (2008) Grid information system interoperability: The need for a common information model. In: IEEE Fourth International Conference on eScience, eScience '08., pp 501–507

Gary Stoneburner G, Goguen A, Feringa A (2002) Risk management guide for information technology systems

Goiri, I., Juli`a and F, Nou R, Berral J, Guitart J, Torres J (2010) Energy-aware scheduling in virtualized datacenters. In: IEEE International Conference on Cluster Computing (CLUSTER)., pp 58–67 doi:10.1109/CLUSTER.2010.15

Huang, H., & Wang, L. (2010) P and p: A combined push-pull model for resource monitoring in cloud computing environment. In: *IEEE 3rd International Conference on Cloud Computing (CLOUD)*., pp 260–267 doi:10.1109/CLOUD.2010.85

Irmos Project. (2008) Project website: http://www.irmosproject.eu

Josang, A. (2011) Subjective logic. URL http://folk.uio.no/josang/sl/

Josang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems*, *43*(2), 618–644. doi:10.1016/j.dss.2005.05.019

Katsaros, G., Kousiouris, G., Gogouvitis, S., Kyriazis, D., & Varvarigou, T. (2010) A service oriented monitoring framework for soft real-time applications. In: *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, pp 1–4 doi:10.1109/SOCA.2010.5707182

Katsaros, G., K¨ubert, R., & Wang, T. G. G. (2011) A multi-level architecture for collecting and managing monitoring information in cloud environments. In: *Proceedings of the 1st International Conference on Cloud Computing and Services Science, CLOSER 2011*

Katsaros G, Subirats J, Fit JO, Guitart J, Gilet P, Espling D (2012) A service framework for energy-aware monitoring and VM management in clouds. Future Generation Computer Systems (0):–, DOI 10.1016/j.future.2012.12.006

Lawrence, A., Djemame, K. W. O., Wolfgang, Z., & Zsigri, C. (2010) Using service level agreements for optimising cloud infrastructure services. In: Service-Wave 2010 workshops, Ghent Marques F, Sauve J, Moura A (2006) Business-oriented capacity planning of it infrastructure to handle load surges. In: 10th IEEE/IFIP Network Operations and Management Symposium, NOMS '06., pp 1–4

Mell, P., & Grance, T. (2009). *The NIST definition of cloud computing*. National Institute of Standards and Technology.

Nagios (1996) Project website: http://www.nagios.org

Nagios Plugins. (2000) Project Website: http://nagiosplug.sourceforge.net/developer-guidelines.html

Narasimhamurthy, S., Umanesan, G., Morse, J., Golbourn, D., & Muggeridge, M. (2011) Storage quality of service for high performance compute clouds. In: Workshop on Trends, Issues and Challenges in Future High-End Computing, All Hands Meeting 2010, Cardiff, UK

Nieto, F. J. (2011). *A Trust Model for Services in Federated Platforms, Lecture Notes in Business Information Processing* (Vol. 76, pp. 118–131). Springer Berlin Heidelberg.

NORDU GRID. (2011) NORDU Grid Project: http://www.nordugrid.org/

OGF, Glue Working Group. (2009) GLUE Specification v. 2.0: http://www.ogf.org/documents/GFD.147.pdf

OGF, JSDL Working Group (2009) Activity Instance Description Specification v1.0,

Optimis Consortium. (2011). *Optimis architecture d1.2.1.1. Tech. rep*. Optimis.

Pfoser, D., Tryfona, N., & Verykios, V. (2003) Services-based data management in a global computing environment. In: *Fourth International Conference on Web Information Systems Engineering Workshops*., pp 45–53 doi:10.1109/WISEW.2003.1286785

Sacerdoti, F., Katz, M., Massie, M., & Culler, D. (2003) Wide area cluster monitoring with ganglia. In: *IEEE International Conference on Cluster Computing, CLUSTER '03*, pp 289–298 doi:10.1109/CLUSTR.2003.1253327

Sauve, J., Marques, F., Moura, A., Sampaio, M., Jornada, J., & Radziuk, E. (2006a) Optimal design of e-commerce site infrastructure from a business perspective. In: *Proceedings of the 39th Annual Hawaii International Conference on System Sciences, HICSS '06*., p 178c doi:10.1109/HICSS.2006.371

Sauve, J., Moura, A., Sampaio, M., Jornada, J., & Radziuk, E. (2006b) An introductory overview and survey of business-driven it management. In: The First IEEE/IFIP InternationalWorkshop on Business-Driven IT Management, BDIM '06., pp 1–10 doi:10.1109/BDIM.2006.1649205

Srikantaiah, S., Kansal, A., & Zhao, F. (2008) Energy Aware Consolidation for Cloud Computing. In: Proceedings of HotPower '08 Workshop on Power Aware Computing and Systems, USENIX

Stepnicka, M., De Baets, B., & Noskova, L. (2010). *Arithmetic fuzzy models. Fuzzy Systems* (pp. 1058–1069). IEEE Transactions on; doi:10.1109/TFUZZ.2010.2062522

Voith, T., Kessler, M., Oberle, K., Lamp, D., Cuevas, A., Mandic, P., & Reifert, A. (2009) Isoni whitepaper v2.0, http://www.irmosproject.eu/publications

Voith, T., Oberle, K., Stein, M., Oliveros, E., Gallizo, G., & Kubert, R. (2010) A path supervision framework a key for service monitoring in infrastructure as a service (IaaS) platforms. In: *36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, pp 127–130 doi:10.1109/SEAA.2010.42