ABSTRACT

Cloud Computing as a service on demand architecture has become a topic of interest in the last few years. The outsourcing of duties and infrastructure to external parties enables new services to be established quickly and with low financial risk. These services also can be scaled on demand. Nevertheless, several issues such as security and legality should be considered before entering the cloud. The financial benefits of cloud services conflict with the need to secure and control the access to outsourced information. Companies have to comply with diverse laws across jurisdictions and are accountable to various national regulators. Security requirements may not be compatible with those offered by existing providers.

In this paper, we propose an architecture to facilitate the integration of these security requirements in the cloud environment and to address the legal issues attached. Our approach customizes the selection of a service provider based on the companies preference. We also define a trusted third party to handle the monitoring and auditing processes over different service providers.

1. INTRODUCTION

Cloud Computing is a concept of utilizing computing as an on-demand service. It fosters operating and economic efficiencies and promises to cause an unanticipated change in business. Numerous authors argue for the benefits of cloud computing focusing on the economic benefits [8] [4]. Using computing resources as pay-as-you-go model enables companies to convert the fixed IT cost into a variable cost based on actual consumption. However, despite of the non-contentious financial advantages cloud computing raises questions about privacy, security, reliability, and legislation.

Beyond the self-motivated interest in securing organization’s data pool, several laws demand public and private organizations to protect the security of their information systems [27]. The European Union’s Data Protection Directive (EU DPD), for instance, has clear restrictions for the movement, processing, and access of specific types of data across political borders. Some of these laws cover only specific markets such as the financial markets or health care industry. However, any organization that does business in countries with existing legal restrictions regarding the information security is subject to these laws. These obligations that protect the security of data apply no matter where these information assets are located. Furthermore, legal practice in the US and in the EU hold organizations liable for the activities of their subcontractors such as cloud service providers. Therefore, each organization must understand the relative legal requirements as well as the consequent implications when considering moving data to the cloud [20].

For each potential cloud customer, it is both expensive and time consuming to handle these legal concerns. They will have to perform a scrutiny on the security capabilities of the service provider independently. This has to include: studying particular security policies and service level agreements (SLA) that are usually written in a plain natural language1 and inspecting the facilities of the service provider. Carrying on these two tasks is indeed inefficient as well as time consuming. Therefore, it is only logical to have a third party who is specialized in legal and security matters to monitor and audit such tasks. In this paper, we propose an architecture that uses a trusted third party to supervise and attest compliance of these legal requirements.

We also take into consideration that it is expected in the future an abundance of cloud service providers competing for the favor of customers by providing similar kinds of services. Which creates the problem of proper selection for the needed provider in such competitive market. IDC2 has issued its second forecast for cloud services, and estimates that the technology will grow by an average of 26 percent over the next four years. According to this study, worldwide cloud IT services are currently worth 17.2bn dollars, but will rise to 44.2bn by the year 2013. Further industry analysts estimate great future potentials in cloud computing, e.g. Morgan Stanley [28] expects this technology to be a 160$ billion market opportunity.

In this environment, factors for selecting from multiple providers who offer similar services are: the quality and nature of the offer-

---

1Amazon. Amazon ec2 service level agreement, 2009.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 2010 ACM 978-1-4503-0380-4 ...$10.00.
2. MOVING TO THE CLOUD

In this section we illustrate by example the aspects involved in a company’s decision to use cloud services.

2.1 Motivating Example

Our example is of a financial consulting company which has several branches spread over European countries. The company is considering to take advantage of the economic benefits offered by maintaining parts of its IT-resources by a cloud service provider. The company, therefore, performed an extensive study on its data stocks to determine the appropriate resources to be transformed into the on-demand cloud computing model. The study showed that there is a need for a collaboration platform that facilitates file sharing and activity management. Some of the corporate data can be shared among employees and customers and is expected to be available for them. Hence, it needs to be protected by strong access measures. However, some data contains personal information that cannot be made public (e.g. information about employees and customers), which require protection from any unauthorized access.

Since the online storage is a competitive market, the company has to decide between a long list of service providers: Box.net, Live Mesh, Dropbox, Amazon S3 storage service, etc.. All of which have the same functionality and practically the same usability. They integrate with the working environment and allow customers to effortlessly upload and download files. The financial company -the customer- is of course interested in choosing the most reliable service, so it compares each offered solution independently which is, sufficient to say, a cumbersome task. This task includes studying: physical locations, legal provisions, security policies, and service level agreements (SLA).

2.2 Problems to Consider

Some problems already occurred from moving to the cloud. Part of these problems are (but not limited to):

- Legal Issues: In our example, the company is working through different countries, hence, it is susceptible to various laws. The decision of using cloud services includes matching the offers with legal proceedings since the company still have to comply with diverse laws across jurisdictions and is accountable to several national regulators. Unfortunately, there is no clear guidelines or regulations that would help to classify and define an appropriate security level for companies data. SLAs act in this case as the binding electronic contracts. Since various SLAs are defined in a plain natural language, they have to be analyzed and validated manually, which is both expansive and slow task. Moreover, different organizations have various definitions for SLA’s parameters such as availability, response time, bandwidth. In the European Union, data protection laws establish a number of very specific requirements and compliance will be overseen by the data protection authorities of particular member state. Dealing with personal information is regulated in slightly different ways across the EU. In the event that legal provisions of any involved state change, the consumer-company is responsible for ordering the provider to apply the necessary adjustments. That implies that the company has to spend resources on being au courant with current legislation.

- Security Issues: There are a lot of security and trust questions in the cloud environment: data encryption possibility, data integrity checks, ensuring consumer’s privacy, the existence of detailed access log (trails) to the data, staff background checks, data failure and disaster management, ensuring correct working security policy, identity of who access the data and the metadata, authentication methods, intrusion detection and so on. In order to ensure that the agreed requirements are continuously met during the period of a contract, the company has to invest in inspecting the service provider’s security capabilities and perform ongoing detailed audits.
- **Missing QoS standardization** The problem of handling service level management in inter-domain scenarios is not entirely solved up to present. The quality and nature of the offerings with Quality of Service (QoS) attributes should be formalized. For example, there is no guarantee that the best vendor today will be the most reliable partner in the near future.

Therefore, to make the decision of moving data to a public cloud, a company has to:

1. understand the legal requirements and stay au courant with current legislation and service providers offers
2. identify assets which are suitable for keeping beyond organizational boundaries
3. identify a reliable cloud service provider and audit his compliance with the security policies and legislative requirements
4. understand information management mechanisms, including the security capabilities of the provider

It is easily expected from the previous requirements that most companies will consider using cloud services a security and legal hassle. It will cost them time and a great deal of money to avoid such hassle. Therefore, they may skip the idea altogether. In the next sections we describe our framework that semi-automizes most of these processes to spare the company the cumbersome efforts as well as to help realizing the cloud vision.

### 3. ARCHITECTURE

The ground of our approach is to find a balance between benefiting from the cloud nature of pay-per-use and ensuring the safety of the company’s data as well as the legality of the transactions performed over this data. The goal is to achieve such balance by automating the process of selecting a cloud provider and removing the auditing responsibility from the customer’s side. Selecting a cloud provider involves: the definition, negotiation, monitoring, and enforcement of mutual expectations and agreements. Our proposed architecture (figure 2) is based on three components:

1. **Trusted Authority Center**: Handling the legal and security concerns is both expensive and time consuming for companies. Therefore, we propose using a trusted third party in the cloud that is responsible for:
   - supervising and attesting compliance of the legal requirements
   - studying related security policies and service level agreements
   - inspecting the facilities of the service provider

A cloud can have one or more TACs that can share one or more service registries. As a result, moving a big part of the tasks required to ensure the security of the data to a specialized trusted party. We are planning to realize part of this architecture by expanding security patterns to include public legal patterns as an extension to our work in [25]. Discussing the related business model (e.g. cost of using this service) is out of this paper scope.

2. **Requirements Formalization Service**: Plenty of potential cloud customers are non-IT specialists. Though most of them do have IT departments, it is difficult for them to adjust the way they formalize the company’s requirements to fit the generic nature of cloud environment. Especially if it involves new parameters that weren’t of issue in other environments. We propose a new component service that takes from the customers their requirements, formalizes them, then translates them into formal electronic contracts (i.e. SLAs). The service also transforms service provider’s capabilities into a standardized form that is used in the e-contract.

3. **Reputation Service**: After matching user requirements and provider capabilities, we use the reputation of the providers to produce the final list of potential providers. A provider’s reputation holds the details of his performance plus his ratings in the service registries and saved in a Reputation Object (introduced in our previous work [2] [1]). By reading this object, we know his reputation concerning each performance parameter (e.g. has high response time, low price). Therefore, a set of potential providers is given to the consumer extracted based on the order of the performance parameters in the consumer’s requirement list (e.g. a consumer cares about the response time more than the price).

Once the requirements are formalized, the center asks the set of registered cloud providers for their formalized offers and uses the matching service to match formal requirements with provider’s capabilities. The resulted list of providers is passed, along with the consumer’s priority list, to the reputation service to be filtered according to providers’ reputations. The consumer then receives the final list of potential providers. We discuss these components in details in the next sections.

### 4. REQUIREMENTS FORMALIZATION SERVICE

In this section we discuss one of the architecture’s components: the Requirements Formalization Service (RFS). RFS works for both the consumer side and the provider side. It formalizes consumer expectations (on data retention and security measures) into a formal set of requirements and also formalizes provider’s capabilities (a non-formal description of his offers and policies). The main goal of the RFS is to facilitate the formalization of the electronic contract’s description and components to enable the process of automatic service provider selection.

At the consumer side, the company/consumer gives the RFS a set of prioritized requirements to be fulfilled by the potential provider. The consumer is asked to categorize his data into protection levels based on its appropriate access conditions to decide later which data to reside on the cloud. This is done by domain experts in the company since they have detailed knowledge on existing processes within the organization. In our example, the company needs to distinguish between: assets with the most stringent protection requirements, less sensitive data, and the one that doesn’t need protection. This distinction depends on the company’s mission, legislative requirements, and internal security policy. The categorization should

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extreme</td>
<td>Threatening enterprise existence</td>
</tr>
<tr>
<td>Very High</td>
<td>Severe financial or security consequences</td>
</tr>
<tr>
<td>High</td>
<td>Impact on customer’s services or reputation</td>
</tr>
<tr>
<td>Medium</td>
<td>Affects the enterprises mission</td>
</tr>
<tr>
<td>Low</td>
<td>Minor financial damage</td>
</tr>
<tr>
<td>Negligible</td>
<td>No security risks</td>
</tr>
</tbody>
</table>

Table 1: General Asset Value Classification
be performed in two steps: Security Level classification (identifying the data access security level) and Asset Type categorization (additional description for the data in each security level). The security level classification step defines three levels of data security: high, medium, and low, defined as following:

- **High**: critical data not applicable to be kept beyond organizational boundaries and should be protected by high security measures, e.g. ongoing research projects.

- **Medium**: less critical data which still requires protection, but the benefits of keeping it in the cloud outweigh the associated security risks (e.g. project management data accessed by staff in multiple locations in large companies). This way the storage of data in a cloud is a balance between security and convenience.

- **Low**: the remaining data that is of importance to the company but does not require over-protective security measures as long as it remains accessible to company’s employees or customers such as stock information, business reports, and price histories of various shares.

General security requirements can be specified at further assessment layers (as extension to the above three levels). Schumacher et al. [26] defined six security levels (table 1) that are used later by Menzel [21] to specify the security requirements and determine risks at the business process layer. We use three levels only for the consumer’s convenience. Nevertheless, it is applicable to map between both classifications as following:

- values extreme & very high to the value high (not suitable for keeping beyond organizational boundaries)

- values high to low to the value medium (potential security risks- requires security guarantees from the cloud provider)

- values negligible to the value low (does not have security requirements), due to the absence of security risks associated with this type of data. However, further requirements can be defined like costs or availability.

Next, the consumer side should include further description for the data in each level. This description is required so the RFS can determine the geographical and legal restrictions on the data in each level. The consumer needs only to include the categorization confidential, personal, and neutral (discussed in details in the next section).

4.1 Modeling Enhancements for Assets Categorization

After receiving the user requirements classified into three security levels and each categorized into three types, the RFS attempts to formalize these requirements. In [21], the author formalizes security and access control requirements for data transmission using

<table>
<thead>
<tr>
<th>Artifact</th>
<th>Name</th>
<th>Asset Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Data Categorization</td>
<td></td>
</tr>
<tr>
<td>Choreography</td>
<td>Number of connected objects: [1..*]</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>String</td>
<td>&quot;confidential&quot;, &quot;personal&quot;, &quot;neutral&quot;, &quot;unspecified&quot;</td>
</tr>
</tbody>
</table>

Table 2: BPMN notation of attribute Asset Type
we decided to use three values for the one or more pools. Although various types of data are conceivable, the artifact represents process related data and can be connected to an attribute at the first abstraction layer. The usage of the attribute is indicated that the entity encloses more details in the lower layers. We define general assessment values by specifying the Asset Type attribute in the first abstraction layer. The usage of the attribute is illustrated in figure 3 and table 2. The "magnifier" sign of the shape indicates that the entity encloses more details in the lower layers. The artifact represents process related data and can be connected to one or more pools. Although various types of data are conceivable, we decided to use three values for the Asset Type attribute:

1. **confidential**: data that should be protected by high security measures and located in a server within the allowed geographical boundaries.

2. **personal**: data that contains customers' and employees' information and does not require explicitly high security measures, however, its storage location is restricted by law (e.g. within the European Union).

3. **neutral**: data that has no restriction over the geographical location or security measures.

The value unspecified is to be used if the user is not able to give an appropriate description of the data.

The outcome of the performed data categorization represents a set of basic information about organization’s assets. These formalized customer service requirements (CSR) serve as basis for the identification of appropriate service providers. Figure 4 is an example of a CSR showing some categorized assets. The given description is highly simplified. However, it gives an impression of how assets are categorized and described in XML-based syntax. These restrictions limit the range of potential cloud providers. The user also passes to the RFS information on his budget boundaries which also narrows down the range of providers selection. Since quality plays a critical role for the user, he also states his expectations of the QoS parameters such as bandwidth and availability.

After this categorization, the RFS maps this structured data into a formal language that describes the e-contract between the provider and the consumer. The QoS values provided by the consumer are used in the selection and negotiation processes. In the next section we discuss the requirements for a framework to realize such mapping.

### 4.2 Managing Electronic Contracts in Cloud Computing

In the previous section we introduced our enhancement for BPMN aimed to express high level expectations of data retention at the business process layer. In this section, we discuss the requirements of a language needed to define the agreements or the electronic contracts. These contracts identify the parties involved and specify how the mutual expectations are carried out. An important aspect of such contracts is the QoS guarantees. In the scope of Service-oriented-Architecture (SOA), this is commonly referred to as a Service Level Agreement (SLA) [19]. An SLA can include a classical technical QoS matrix defined in network technologies (e.g. delay, packet loss rate, throughput, etc.) or other characteristics (e.g. reliability, availability, processing time, or security).

QoS issues have been an area of research for several years. There are several academic and industrial approaches addressing QoS management for Web Services such as WSLA, WSOL or SLAang [16] [18]. Though they reflect in the proposed concepts, the general structure remains the same for the SLA: information on the parties involved, the SLA parameters, the resource metrics used to compute the SLA parameters, the algorithm to compute the SLA parameters, the service level objectives (SLO), and the appropriate actions to be taken in case of a violation of the agreement (aka breach management). However, classic QoS parameters that are used to describe the relationships among common web services are not sufficient in a generic cloud environment. Therefore, in the next section we define the requirements for a language that is more suitable for such environment.

#### 4.2.1 Language Requirements

The business case of cloud computing is based on some expected range of availability, scalability, and performance which have to be addressed by the used SLA. We define the cloud provider performance as the sum of network performance, application performance, and cloud infrastructure performance. Hence, a service provider is required to provide additional information on the current system’s configuration as well as the runtime information on the quality metrics to be used. This requires a formal definition of the used SLA in order to be able to automatize the definition, negotiation, and monitoring of contracts.

The first requirement in the language is to extend the SLA from describing only a bilateral agreement, to include the description of a third party duties. The involvement of intermediaries is reasonable if one of the signatory parties does not trust the opposite one. SLA monitoring, for instance, may require the participation of third parties especially if breaches in the agreement are going to be reported. Since the contract may involve multiple services provided by different organizations, it is also important that each party receives only the amount of information needed to perform the agreed task. This requires a mechanism for splitting the contracts and delivering information to individual participants [16]. Protecting the privacy of business partners across the supply chain was addressed in [13] and was followed by our work on trust management distribution along several virtual organizations in [12].

The second requirement is to address generic QoS parameters due to the continuous changes in the business operations and operating conditions of the cloud environment. Therefore, the SLA language should be flexible and capable of specifying various types of requirements as well as the monitoring and measuring mechanisms.
WSLA framework consists of an extensible language based on XML Schema [10] and promotes the idea of individually negotiated and customized SLAs. The framework features wide acceptance and applicability of existing e-Business systems and standards. Furthermore, WSLA is capable of delegating monitoring tasks to third parties and supports configuration of the managed resources, i.e. deriving configuration settings directly from SLAs. According to the WSLA XML schema, an SLA is divided into three sections: parties, service definitions, and service obligations. Parties introduces the signatory (who establish and sign the SLA) and supporting parties (involved third parties). Service definition provides information on the quality of services and the parameters to be observed. Resource metrics are derived from the managed resources as defined in a measurement directive. Finally, obligations defines constraints and guarantees in form of service level objectives (SLOs). It includes the action guarantees entity, which defines the compensating activities in case of SLAs violation.

Since WSLA design goal is formal and flexible XML-based language for SLA in inter-domain environments, we extend the language to include new negotiation mechanisms and parameters. We mainly distinguish between two types of requirements: communication requirements (e.g. availability, bandwidth) and data retention requirements for cloud service provider (e.g. segregation of storage, encryption, processing performance parameters).

We are working on an implementation for specifying the physical location of the cloud resource using watermarks. According to WSLA specification, SLA parameters are properties of a service object. We define a parameter PhysicalResourceLocation as shown in Figure 5 which specify the general geographical location of the data (e.g. within EU). It is assigned the metric Watermark defined independent of the SLA parameter. The assertion service of the cloud provider guarantees to send (Push) the values to trusted third party, which is also authorized to retrieve new watermark values on its own initiative (Pull).

5. REPUTATION SERVICE

Service and service provider’s reputation have been the focus of several studies on the last few years. In SOA, the approaches used are based on collecting sufficient consumers feedbacks to evaluate a service provider by a rating value saved in several service registers. Some approaches require that consumers report only whether the service met the quality constraints set in the SLA [6]. However, these approaches support QoS-based service selection in a very limited manner.

In our previous work [2] [1], we addressed the problem of reputation representation and introduced a model based on using reputation objects rather than reputation values. In summary, the objects hold detailed information about the performance of the service or the service provider. These details are of different data and data types and saved in the object as a list of "quality_performance_parameter" associated with its "value". For example, "response time=3.5", "price=low", "bad communication", "customerSupport=FALSE" and so on. Thus, the provider’s reputation is now represented as the collection of his reputation regarding several quality parameters. The consumer who uses his services is required to rate him in some detailed way (e.g. his service availability, price, response time, reliability, technical support, etc.)

From this detailed profile, the reputation service is able to cross reference the quality parameters required by the consumer (retrieved from his priority list) and the performance parameters extracted from the providers’ reputation objects. The priority list contains the quality parameters ordered from the consumer’s most important parameter to the least important one. So, once the service receives the list of potential providers, it refines it based on the providers’ reputation objects. The priority list contains the quality parameters ordered from the consumer’s most important parameter to the least important one. So, once the service receives the list of potential providers, it refines it based on the providers’ reputation objects. The priority list contains the quality parameters ordered from the consumer’s most important parameter to the least important one. So, once the service receives the list of potential providers, it refines it based on the providers’ reputation objects. The priority list contains the quality parameters ordered from the consumer’s most important parameter to the least important one. So, once the service receives the list of potential providers, it refines it based on the providers’ reputation objects.

The refine process in the reputation service is based on a simple algorithm where:

```
get priority P1 from the ConsumerList
For all Providers_reputation in the ProviderList
    select the providers with the highest P1 values
    save in FilterList1
get priority P2 from the ConsumerList
For all Providers_reputation in FilterList1
    select the providers with the highest P2 values
    save in FilterList2
Finally, the consumer receives the filtered list produced by a trade off between the best QoS parameters and the user require-
```
ments. The consumer therefore takes the decision on selecting his provider.

6. RELATED WORK

The future of distributed computing has been a subject of interest for various research in the recent years. The authors in [7] propose an architecture for marked-oriented allocation of resources within clouds. They discuss some existing cloud platforms from the marked-oriented perspective and presents a vision for creating a global cloud exchange for trading services. However, this work does not address legal or security issues. In our architecture the third parties do not allocate resources from cloud providers to sell them to the customers. They act as an abstraction layer between service vendors and service users. Our approach enables also service providers to establish the compliance of security policies without the fear of disclosing sensitive information about internal security measures.

Other approaches [30] [17] use marked-based mechanisms for resource allocation, allowing users to differentiate the values of their jobs and rely on the scheduling of CPU cycles. Bellagio in [5] operates on environments where different QoS may exist (e.g disk space, memory, bandwidth, etc) and allows users to express their interests in particular sets of resources. This work focuses on resource allocation in federated distributed computing infrastructures more than on the negotiation of QoS requirements. Our approach selects suitable services based on user’s requirements.

In [9] the authors present a service-oriented desktop grid system. It provides an infrastructure that supports multiple application models, security, and communication protocols. It provides SLA support by enabling the user to specify QoS requirements such as deadline and budget. The aim was to provide a set of services that facilitate grid construction and development of applications. Thus, the core value of this system is a service oriented runtime environment that is deployed on virtual infrastructures. There are also different research in the field of SLAs, during the standardization efforts like in [18] [16] [3]. Some approaches use rule based techniques to specify electronic contracts, e.g RuleML language [14] RBSLA [22]. However, these approaches do not completely meet the requirements of a generic cloud environment.

7. CONCLUSION AND FUTURE WORK

In this paper we introduced an architecture where a third party acts as an abstraction layer between users and cloud vendors with the services at their disposal. The three main components of the architecture are: Trusted third Center (examines cloud providers compliance of legal and security measures, and negotiates QoS parameters), Requirements Formalization Service (formalizing mutual expectations and requirements, and enabling automated SLA definition and negotiation process), and Reputation Service (enables the selection of cloud providers based on users expectations and providers’ reputation). We also presented an approach to describe consumers requirements at the business process layer. It also enables third parties to use a reputation based algorithm to determine the list of the most suitable providers for the customer.

As a future work, we intend to define a domain-independent model that eliminates ambiguities of terminologies’ definitions such as response time or downtime. It can also map general requirements to formal QoS parameters in any SLA language (we are using only WSLA so far). We will extend this work and our work in [12] by defining trust relationships between the interacting parties.

8. REFERENCES


