Service Level Agreement in Cloud Computing

Pankesh Patel[1,2], Ajith Ranabahu[1], Amit Sheth[1]

[1] Knoesis Center, Wright State University, USA
{ajith,amit}@knoesis.org
[2] DA-IICT, Gandhinagar, INDIA
pankesh_patel@daiict.ac.in

Abstract. Cloud computing that provides cheap and pay-as-you-go computing resources is rapidly gaining momentum as an alternative to traditional IT Infrastructure. As more and more consumers delegate their tasks to cloud providers, Service Level Agreements(SLA) between consumers and providers emerge as a key aspect. Due to the dynamic nature of the cloud, continuous monitoring on Quality of Service (QoS) attributes is necessary to enforce SLAs. Also numerous other factors such as trust (on the cloud provider) come into consideration, particularly for enterprise customers that may outsource its critical data. This complex nature of the cloud landscape warrants a sophisticated means of managing SLAs. This paper proposes a mechanism for managing SLAs in a cloud computing environment using the Web Service Level Agreement(WSLA) framework, developed for SLA monitoring and SLA enforcement in a Service Oriented Architecture (SOA). We use the third party support feature of WSLA to delegate monitoring and enforcement tasks to other entities in order to solve the trust issues. We also present a real world use case to validate our proposal.

KEYWORDS: Cloud Computing, Web Service Level Agreement, Service Level Objectives, SLA monitoring, SLA enforcement, Cloud Security.

1 Introduction

Cloud computing [1] is the new trend of computing where readily available computing resources are exposed as a service. These computing resources are generally offered as pay-as-you-go plans and hence have become attractive to cost conscious customers. Apart from the cost, cloud computing also supports the growing concerns of carbon emissions and environmental impact since the cloud advocates better management of resources. We see a growing trend of off-loading the previously in-house service systems to the cloud, based primarily on the cost and the maintenance burden. Such a move allows businesses to focus on their core competencies and not burden themselves with back office operations.

Cloud is defined as *both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services* [1]. According to this definition delivery of application as services

(SaaS - Software as a Service) over the Internet and hardware services (IaaS - Infrastructure as a Service) are both parts of cloud computing phenomena. From hardware service(utility computing) point of view, there are few new aspects in cloud [1], the most prominent being the *illusion of infinite computing resources* and the *ability to pay for use of computing resources on a short-term basis as needed*.

As consumers move towards adopting such a Service-Oriented Architecture, the quality and reliability of the services become important aspects. However the demands of the service consumers vary significantly. It is not possible to fulfill all consumer expectations from the service provider perspective and hence a balance needs to be made via a negotiation process. At the end of the negotiation process, provider and consumer commit to an agreement. In SOA terms, this agreement is referred to as a SLA. This SLA serves as the foundation for the expected level of service between the consumer and the provider. The QoS attributes that are generally part of an SLA (such as response time and throughput) however change constantly and to enforce the agreement, these parameters need to be closely monitored [2].

Due to the complex nature of consumer demands, a simple "measure and trigger" process may not work for SLA enforcement. Four different types of monitoring demands made by consumers are mentioned in [3]. One scenario is *a consumer demands the data exposed by a service provider without further refinement* such as transaction count, which is a raw metric. Second scenario is *consumer requests that collected data should put into meaningful context*. This scenario creates the requirement for a process which collects data from different sources and applies suitable algorithms for calculating meaningful results. Such metrics include statistical measures such as average or standard deviation that need to be computed from a raw set of numbers. The third scenario is the *consumer requests certain customized data to be collected*. In the fourth scenario the *consumer even specifies the way how data should be collected*. Both the latter mentioned scenarios imply an advanced consumer who would have a knowledge of the inner workings of a provider and somewhat rare in practice. Other issues such as trust also need to be considered during SLA enforcement. For example consumers may not completely trust the certain measurements provided solely by a service provider and regularly employ third party mediators. These mediators are responsible for measuring the critical service parameters and reporting violations of the agreement from either party.

We believe the upcoming trend of cloud computing is an extension of the SOA paradigm and the above mentioned issue of *striking a balance* applies to the cloud as well. The process of managing the provider-consumer agreements in computing clouds closely resemble the generic provider-consumer agreement process we mentioned above. Hence we propose an architecture for managing cloud consumer and provider SLAs, based on the WSLA specification [3].
We highlight two reasons to justify the importance of this research.

1. The most prominent cloud provider, Amazon EC2, puts the burden of proving SLA violations on the consumer. i.e. the consumer should take steps to

enforce the SLA [4]. Having a formalized SLA enables the set up of the enforcement process to be automated and hence relieves consumers from that burden.
2. We believe the work that significantly intersects with ours is [5] where WSLA has been used as a base for grid service monitoring. However computing grids are very different from computing clouds in terms of 1) business model, 2) architecture, 3) resource management, 4) programming model, 5) application model and 6) security model [6]. Hence we believe applying WSLA to the cloud context would be a significantly different effort from the previous work. Some of the important aspects we discovered are detailed in section 4. To the best of our knowledge this is the first use of WSLA in the context of cloud computing.

In the rest of this paper we present the related work [section 2] and introduce the WSLA framework [section 3]. Then we present our architecture proposal [section 4] and provide a use case based on a real world cloud usage scenario [section 5]. We conclude the paper with a detailed discussion [section 6] on the architecture as well as the pros and cons of our proposal.

2 Related Work

Significant level of research in SLAs has been performed during standardizing efforts. There are two main specifications for describing a SLA for web services. 1)Web Service Agreement [7](WS-Agreement) from Open Grid forum (OGF) and 2) Web Service Level Agreement language and framework (WSLA) [3] from IBM. To the best of our knowledge, other most prominent ongoing research project for SLA specification is SLAng [8]. In other related work, Rule-based Service Level Agreement(RBSLA) [9] is highlighted. RBSLA follows a knowledge based approach and uses RuleML [10] to specify the SLA.

Another relevant specification in this context is WS-Policy[11] from the World Wide Web Consortium (W3C). By using WS-Policy, Service providers can advertise their policies. On the other hand service consumers can also specify their policy requirements. These *policies* primarily consist of non-functional properties. Web Service Offering Language[12](WSOL), Web Service Modeling Ontology [13](WSMO) and Web Service Management layer [14](WSML) all provide some level of description for non-functional properties. However all the above work is not in the direct context of SLA.

3 Background - WSLA framework

As described in [3], WSLA consists of a set of concepts and a XML language. It is designed to capture service level agreements in a formal way. WSLA comprises of mainly three entities.

1. **Parties**: WSLA contains the descriptions about 1) service provider, 2) service consumer and 3) third parties. The task of these third parties may

vary from measuring service parameters to taking actions on violations as delegated by either the service provider and service consumer.
2. **SLA parameters**: In WSLA, SLA parameters are specified by metrics. Metrics define how service parameters can be measured and are typically functions. There are at least two major types of metrics. 1)Resource metrics are retrieved directly from the provider resources and are used *as is* without further processing. For example, transaction count. 2) Composite metrics represents a combination of several resource metrics, calculated according to a specific algorithm. For example *transactions per hour* combines the raw resource metrics of transaction count and uptime. Composite metrics are required when the consumers need insightful and contextual information where raw numbers do not suffice. In [2], a third metrics referred to as a *business metric* has been defined. It relates SLA parameters to financial terms specific to a service customer.
3. **Service Level Objectives(SLOs)**: A set of formal expressions. These formal expressions have the well known *if...then* structure. The antecedent (*if*) contains *conditions* and the consequent (*then*) contains *actions*. An action represents what a party has agreed to perform when the conditions are met.

Figure 1 illustrates the major components of WSLA in a UML diagram [3]. An established WSLA contains the following major sections:

1. **Parties**:This section comprises of two parties : supporting parties and signatory parties. Signatory parties are the service provider and the service consumer. Supporting parties are the third parties that come into picture when signatory parties decide to delegate certain tasks such as measuring SLA parameters.
2. **Service Definitions**: A service definition contains the description of the service providers interface. Services are represented by service objects. Each service object associates with one or more SLA parameters.
3. **Obligations**: This section contains the conditions and the action guarantees.

4 Architecture

Now we present our cloud WSLA architecture. We have realized the following aspects of the cloud that effects the direct use of WSLA.

1. The cloud is inherently dynamic and the resource usage changes dynamically. Hence any system that tries to enforce a SLA need to embrace this dynamic nature. We suggest that all measuring tasks in a cloud context be performed through *functions*[3]. We identify certain measurements the cloud providers must provide and discuss them later in this section.
2. Due to the mounting concerns of privacy and data security, consumers may be hesitant to disclose certain details to cloud providers. We identify a set of tasks that can be delegated to trusted third parties to cater for better security.

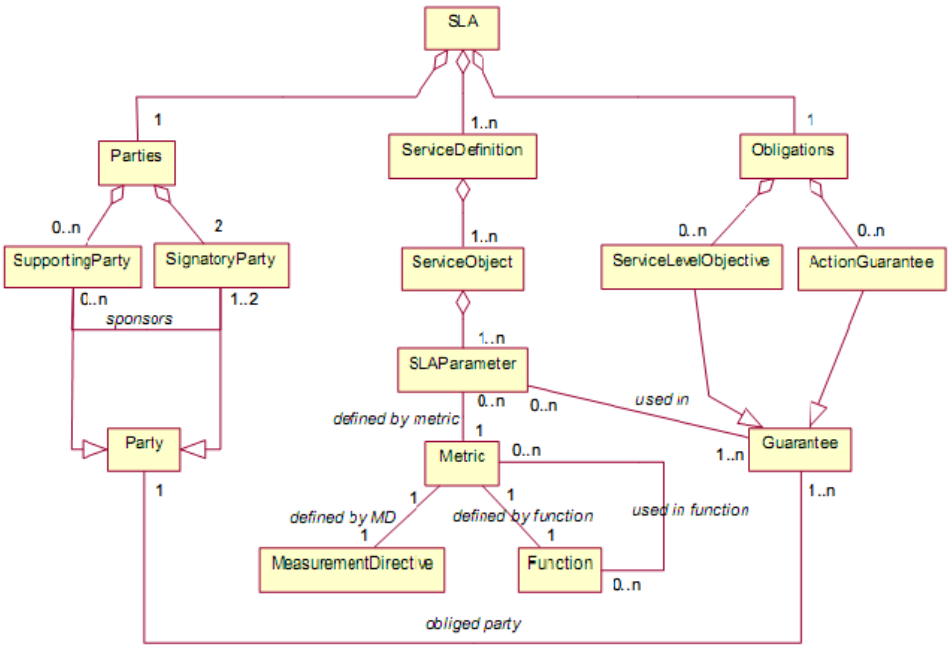

Fig. 1. Main Concepts of WSLA

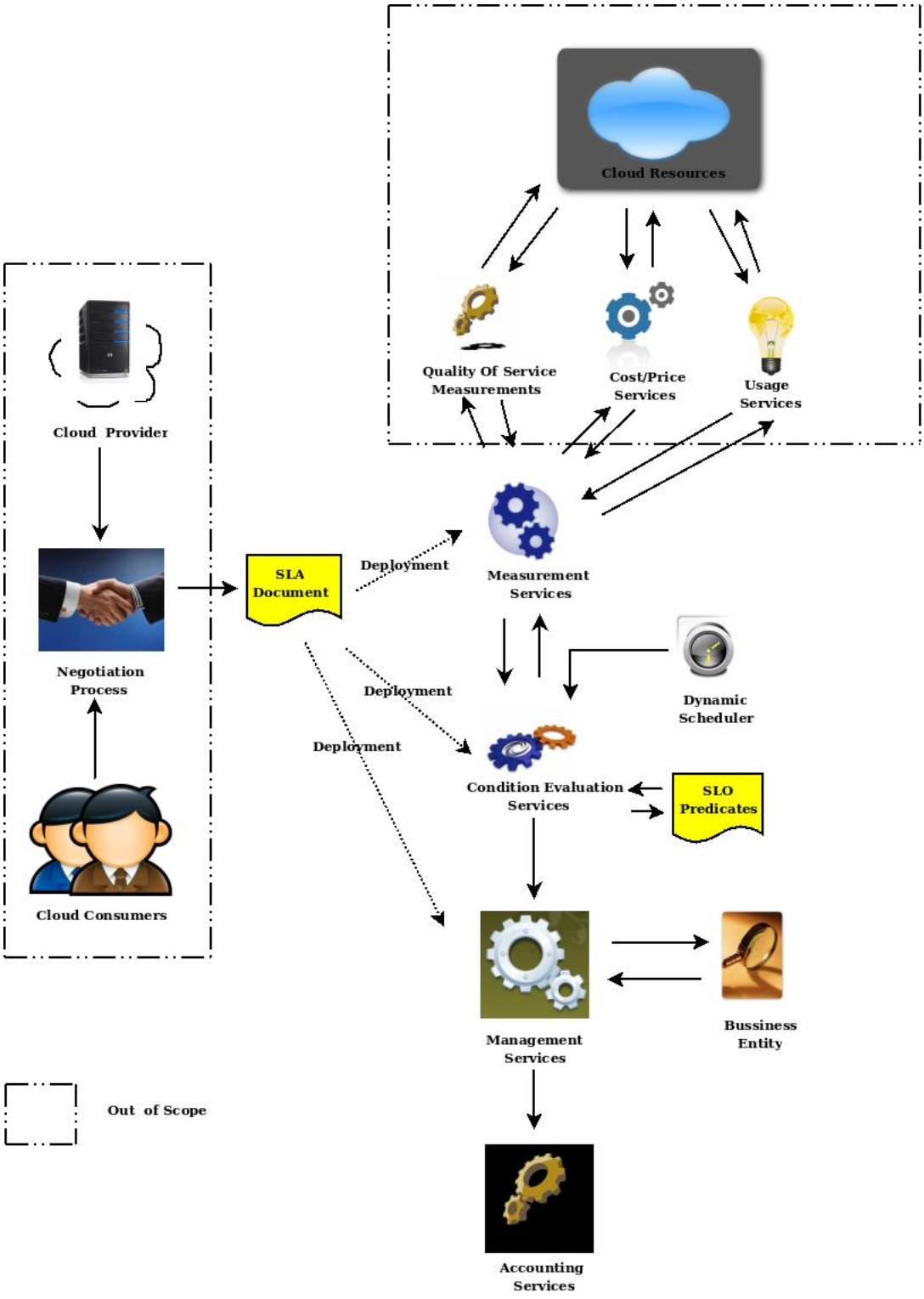

Fig. 2. Architecture

3. Cloud services are subjected to load fluctuations and SLA violations are more likely to happen during these transitions. The nature of these fluctuations are unpredictable and hence a static schedule for evaluating conditions may not suffice. We suggest that SLAs in the cloud context use a dynamic schedule for condition evaluations.

We present our proposed architecture in figure 2. In this architecture, we have assumed that the cloud provider and the cloud consumer already participated in the negotiation process and have an agreed set of service parameters, i.e. *the negotiation and SLA establishment steps are considered out of scope for this work*. Once the SLA document is established, it needs to be *deployed*. In [2] the term *SLA deployment* is defined as the process of validating and distributing the SLA, in part or full, to the involved parties. The work [2] identifies that the provider and the consumer may not want to share the complete SLA document with supporting parties due to security considerations. We describe three common WSLA services [2] and some of their adaptations required in the cloud context.

1. **Measurement Services:** These services are responsible for measuring the runtime parameters of cloud providers resources. As discussed previously, service parameters like response time, throughput are constantly changed due to variability in service request from consumer side. In the context of the cloud however the usage and cost parameters are also dynamic. This is due to the pay-as-you-go nature and the elasticity of the cloud. Hence we identify 1) usage and 2) cost / price data as two major additional services that should be added to the set of measurement services in the context of clouds.
2. **Condition Evaluation Service:** This service is responsible of getting the results from measurement services and evaluating the Service Level Objectives. If there are violations the Management service will be contacted. We believe that due to the dynamic nature of the cloud, the condition evaluation needs to be performed more frequently than in a traditional service framework. Traditionally there is little attention on the complexity of conditions. In the cloud context, we propose that conditions be simpler for faster evaluation cycles.

   We add a dynamic scheduler that depends on a metric like the transaction rate. This ensures that when the load is high, the enforcement check runs more frequently since its most likely the violations happen during such transitions.
3. **Management Service:** This service is responsible for taking corrective actions on violation of the Service Level Objectives. We anticipate that since the cloud represents utility type computing resources, the management service would be primarily handling financial penalties similar to the real world utility industry practices.

5 A Scenario From The Real World

Now we present a use case, influenced by the use of cloud computing in the computer gaming industry. Compute clouds are being increasingly utilized by on-line games vendors due to the cost benefits and the flexibility [15]. Our use case tries to highlight the SLA aspect of such a gaming vendor that want to utilize the cloud. As mentioned in section 1, X Inc has significant benefits in formalizing this SLA.

X Inc, a creator of on-line multi player games wants to utilize a computing cloud to deploy the core gaming process for their latest game. X Inc is not very sure how this game will be accepted by the public and they do not want to make an upfront commitment on the resources they allocate for this game. Hence X Inc decided to choose a cloud computing platform that supports automatic scaling. However X Inc wants a set of guarantees on the response time in order to retain the interested gamers. Although X Inc is ready to maintain a decent response time, they have a threshold for the maximum hourly cost in order to maintain their budgetary constraints. If the response time constraints are not met, X Inc is likely to loose some of their gamers and hence will penalize the cloud provider in case of such a violation. Z Inc provides cost/price services and quality measurements services for cost and resource usage calculation of consumers. For verification purposes X Inc also hires Y-accounting, a trusted third party for resource usage and cost calculation. Y-accounting is responsible for handling finances on behalf of X Inc and SLA violations are directly reported to Y-accounting.

This typical scenario requires that X Inc, with their cloud provider Z Inc, create a SLA with the above mentioned constraints. This SLA include the following SLOs.

1. The maximum hourly cost (C_{max}) needs to be below $CThr_{max}$
2. The average response time (RT_{avg}) needs to be below Thr_{avg} subjected to condition 1.
3. The maximum response time (RT_{max}) needs to be below Thr_{max} subjected to condition 1.

Z Inc. is not capable of calculating any of the composite metrics mentioned in the SLA. Z Inc, however is capable of providing the running time and unit costs. Hence during the SLA negotiation process, Z Inc agreed to let a third party measurement service to measure the response times, and allow yet another third party service to use their time and cost services on behalf of X Inc.

Composite metrics like RT_{avg} and RT_{max} are calculated by third party services using the collected values. C_{max} is also calculated using the data from cost/price services and usage services. Condition evaluation service queries the appropriate measurement services for the relevant values according to attached dynamic schedule. On receiving the values, condition evaluation service evaluates the SLO predicates. In case of SLA violations, management service is notified and Y-accounting is contacted to process the financial penalties.

6 Discussion

As indicated in this proposed architecture, We see a very legitimate need for a clear and formal methodology to handle SLAs in the context of cloud computing. WSLA, which suggests a very flexible architecture for managing SLAs between providers and consumers, seem to be the most suitable candidate. In applying WSLA however, the need for a host of support services arise. Some cloud computing providers may provide these support services but WSLA does not strictly mandate such provisions and hence third parties can step into provide the necessary services. We see this as one of the strong points of WSLA where, true to the paradigm of SOA, every functionality is provided as a service that may not necessarily come from the same provider.

One important observation we make in the context of clouds is the lack of standardization. This is specially important when we attempt to apply monitoring across multiple clouds. Even though it is possible to cater for different cloud interfaces through a middleware, there is no universal set of metrics that can be monitored across cloud providers. There are attempts to standardize the clouds [16] and we underscore the importance of such efforts in the light of monitoring capabilities. As a part of these standardization efforts, we also suggest a set of basic metrics and best practices for measurements be established.

7 Future Work

We see many avenues of future research in this area. One such avenue is based on scalability, which is considered an important aspect of cloud computing. Clouds however may not be able to scale indefinitely and when a resource limitation is encountered, a service provider may decide to delegate the tasks to other cloud providers, transparent to the consumer to avoid significant SLA violation penalties. Such a scenario creates research opportunities in SLA management. We anticipate to investigate SLA aspects like accounting, monitoring of QoS parameters and condition violation in similar scenarios as future work.

The current WSLA framework is based on XML and therefore limits the ability of matching in composition metrics to syntactical. Semantic Web technologies can be used to enhance the descriptions and hence improve the quality of these matches. We believe that work done in [17] is relevant in this regard and can be extended to the cloud context.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley (Feb 2009)
2. Keller, A., Ludwig, H.: The wsla framework: Specifying and monitoring service level agreements for web services. J. Netw. Syst. Manage. **11**(1) (2003) 57–81

3. Ludwig, H., Keller, A., Dan, A., King, R., Franck, R.: Web service level agreement (WSLA) language specification. IBM Corporation (2003)
4. Amazon: Service level agreement for ec2 [http://aws.amazon.com/ec2-sla/] (2008)
5. He, C., Gu, L., Du, B., Li, Z.: A WSLA-based monitoring system for grid service-GSMon. In: 2004 IEEE International Conference on Services Computing, 2004.(SCC 2004). Proceedings. (2004) 596–599
6. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud Computing and Grid Computing 360-Degree Compared. In: Grid Computing Environments Workshop, 2008. GCE'08. (2008) 1–10
7. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web services agreement specification (WS-Agreement). In: Global Grid Forum. (2004)
8. Lamanna, D.D., Skene, J., Emmerich, W.: Slang: A language for defining service level agreements. (2003) 100–106
9. Paschke, A.: Rbsla a declarative rule-based service level agreement language based on ruleml. In: CIMCA '05: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-2 (CIMCA-IAWTIC'06), Washington, DC, USA, IEEE Computer Society (2005) 308–314
10. Boley, H., Tabet, S., Wagner, G.: Design rationale of ruleml: A markup language for semantic web rules. (2001) 381–401
11. Authors, V.J., Ibm, F.C., (editor, C.K., Microsoft, D.L., Ibm, A.N., Ibm, N.N., Bea, M.N., Riegen, C.V., Microsoft, J.S.: Web services policy framework (wspolicy)
12. Tosic, V., Patel, K., Pagurek, B.: Wsol-web service offerings language. Lecture notes in computer science (2002) 57–67
13. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web service modeling ontology. Applied Ontology **1**(1) (2005) 77–106
14. Cibran, M.A., Verheecke, B.: Modularizing web services management with aop
15. Kennedy, S.: Denis dyack's head is in the clouds [http://tinyurl.com/n2gg2w] (2009)
16. Government, U.S.: Federal cloud computing initiative overview [http://tinyurl.com/nbrmgo] (2009)
17. Oldham, N., Verma, K., Sheth, A., Hakimpour, F.: Semantic WS-agreement partner selection. In: Proceedings of the 15th international conference on World Wide Web, ACM New York, NY, USA (2006) 697–706