A MAS TO MANAGE AND MONITOR SLA FOR CLOUD COMPUTING
A Draft Position Paper

Benjamin Gâteau
Centre de Recherche Public Henri Tudor, SISE Unit - Service Science & Innovation Dept
29 Av. John F. Kennedy, L1855 Luxembourg, Luxembourg

Keywords: Dynamic infrastructure, Normative organization, SLA management.

Abstract: The More than a technological solution, the Cloud Computing is also an economical advantage and will play an important roles in next years. However, in order to ensure a QoS commitment between a provider and a customer, Service Level Agreements (SLA) describe a set of non-functional requirements of the service the customer is buying. SLA is the best way to ensure QoS. In this paper, we use a MAS to manage SLA by monitoring the respect of service in the context of Cloud Computing.

1 INTRODUCTION

Evolution of high level ICT infrastructures in Europe brings some difficulties due by a complex management and the need of more and more energy. One current fashionable solution to this problem is the use of cloud computing services. Cloud computing is a ICT model allowing an easy access through networks to mutualised and configurable resources able to be quickly activated and deactivated.

Cloud computing has become a mainstream technology offering mutualisation of IT infrastructures as services along several paths such as Software (SaaS), Platform (PaaS), and Infrastructure (IaaS). Companies such as Amazon, Microsoft, IBM, and Google, to name but a few, offer such services, which rely on virtualization and pay-as-you-go business models. Flexibility and elasticity are also important features of Cloud Computing made possible by the concept of Dynamic Infrastructure.

Dynamic Infrastructure is an information technology paradigm concerning the design of data centers so that the underlying hardware and software can respond dynamically to changing levels of demand in more fundamental and efficient ways than before. The basic premise of Dynamic Infrastructures is to leverage server virtualization technology to pool computing resources wherever possible, and then to allocate these resources on-demand. This allows for load balancing and is a more efficient approach than keeping massive computing resources in reserve to run tasks that take place, for example, once a month. The potential feature benefits include enhancing performance, scalability system availability and uptime, and the ability to perform routine maintenance on either physical or virtual systems all while minimizing interruption to business operations and reducing cost for IT. Dynamic Infrastructures also provide the fundamental business continuity and high availability requirements to facilitate cloud or grid computing.

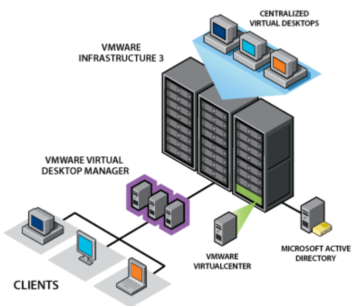

Figure 1: VMware Infrastructure example (source: VMware).

Fig. 1 illustrates an example of dynamic infrastructure based on VMware solution. Physical servers of the datacenter stores virtual machines that can be loaded, managed and configured remotly by the customer of the service (Iaas kind in this case). For that the customer previously specifies his needs in term of

Gâteau B..
A MAS TO MANAGE AND MONITOR SLA FOR CLOUD COMPUTING - A Draft Position Paper.
DOI: 10.5220/0003558206870692
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (MAS-2011), pages 687-692
ISBN: 978-989-8425-52-2

distribution (OS), CPU, memory, disk space, bandwith, etc. to define the virtual machine(s) he will pay for (pay on demand and based on the configuration he ask for).

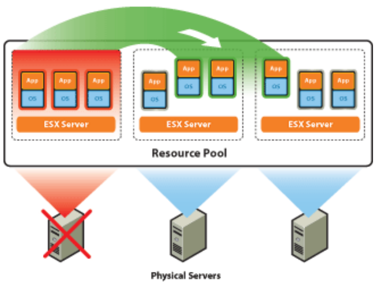

Figure 2: Moving VM in a VMware Infrastructure.

Dynamic infrastructures allow the use of resource when it is needed. For instance, Fig. 2 represents the case when clients of the service provider don't use their virtual machine at a certain time (night for instance). It could happen that only few virtual machines are running on each physical machine. So physical servers are not fully used and gathering virtual machines on one physical server could permit to save energy and use less resource. But by moving virtual machines, the provider must ensure that the customers pay for the service he previously specified and that the Quality of Service (QoS) is still the same. QoS delivery affects the value of the service for the client and significantly depends on IaaS or PaaS provider's infrastructure. Therefore there is need to divide responsibility/risk between *X*aaS provider and customer. To describe the responsibilities a formal description of non-functional requirements from the client's point of the view is required.

Service Level Agreements (SLA) permit to specify the needs and offers of customer and the service provider. To summarize the have to define some SLA in order to .

The aim of our work is to define a SLA infrastructure that provides the same guarantees and proofs that we can find e-contract management, with more flexibility. In this paper we are considering a datacenter providing IaaS remotly an automatically (such as GoGrid[1] or Cloud.com[2] for instance) through a web-based application. This application mediates the interactions between customers and providers. Agreement between parties are represented as SLAs and are specified with a set of web interfaces defining value of KPI that the provider has to achieve (king of rules or obligations). To this aim, we will use an e-contract model to enable an agent-based execution and management of them on one hand, and, on the other hand, we will add a monitoring system to control and enforce the SLAs between provider and customers.

This paper is organized as follows. We make a quick survey on SLA management in the next section, then we present our normative multi-agent system solution before proposing a solution to monitore SLA with an electronic institution based on multi-agent system for the cloud computing.

2 SLA MANAGEMENT

SLA describes a set of non-functional requirements of the service the customer is buying. The agreement usually contains also penalties when the requirements are not met. An example of a non-functional requirement would be "RTO - Return to Operation time for a service in case of a failure. To describe a non-functional requirement we needed an objective to be achieved (e.g. RTO under two minutes) and a set of indicators that prove the objective is met (e.g. new instance bootstrap time). The objective to be achieved is called "Service Level Objective (SLO) and the indicators are called "Key Performance Indicators (KPIs). Service Level Objective (SLO) is the objective of service quality that has to be achieved. It is represented by a set of measurable KPIs with thresholds to decide if the objective is fulfilled or not. The fulfillment of an SLOs describes a state of service when all of the SLOs key performance indicators are within a specified thresholds. KPIs usually consist from one or more raw monitored values including min, avg and max specifying the scale. They can also represent some aggregated measurement (e.g. average output) within a sliding window that is combined from one or more monitoring outputs. The provider has to be able to measure and affect the KPIs otherwise it would not make sense to guarantee them. The Cloud Computing infrastructures are usually large scale, therefore SLAs need to be formally described to enable their automated handling and protection.

Automated SLA protection is based on a set of policy rules. Each policy rule is formed by one or more conditions (KPI's value matching pattern) and one or more actions. KPIs are periodically evaluated according to defined policies. If one or more conditions are met, then appropriate actions are triggered.

An example of the policy action can be increasing number of service instance. The action is triggered by a server load KPI matching the policy rule. This enables to automatically keep the load KPI under certain value specified by the SLO and avoid violating

[1] www.gogrid.com
[2] www.cloud.com

the general SLA. These kinds of rules are named *elasticity rules*.

Specification of the policies (rules and action) is a complex task in large scale dynamic system. Analysis of historical KPI data and triggered actions can be used to specify new or modify existing policies. This enables system adaptation and automated SLA protection evolution. System can for example learn from historical data about periodical service load peaks and generate specific rules to keep the system with throughput specified in SLA . Another research challenge is the specification of models able to describe and simulate these kinds of dynamic systems.

The SLA can also contain certain penalties if one or more of the SLOs is broken. The amount of a penalty usually differs with the SLO guarantee class (e.g. Bronze, Silver, Gold) . This allows to overbook resources with a certain risk that some of the SLOs will be broken (e.g. Some of the Bronze SLO are not going to be fulfilled in favor of some Gold SLOs that are connected with higher penalties). The system can be then optimized to minimize the penalties and maximize the utilization.

With the development of the Internet and Virtual Enterprises appearance, e-Services and e-Contracting in general have gained an increasing interest in the business domain. While some B2B applications handle electronic contracts (e-contract) and digital signatures as digitalized paper contracts (Laurikkala and Tanskanen, 2002), an increasing number of them aim at increasing their enactment and management (e.g. (Koetsier et al., 2000)). We consider in this paper that the management of a SLA defining a set of SLO which have to be respected is the same that managing an e-contract composed by a set of clauses represented by obligations which have to be also respected.

Indeed, in his PhD thesis relating to E-contract modeling and e-enactement (Krishna, 2010), P. Radha Krishna describe the clauses of an electronic contract as obligations being part of a SLA. He also says: "*E-contract management solutions should maintain, monitor and manage contract rules derived from these SLAs. Contract parties should verify QoS parameters by performing an SLA monitoring, which involves monitoring the performance status of the offered service. The e-contract management system could assess the SLA requirements and apply penalties if there is any deviation.*"

The SLA4D-Grid project (Wieder et al., 2009) defines a SLA management layer on top of an existing infrastructure providing e-contracting capabilities. The infrastructure specifies, implemnts and deploys a SLA-based service stack for e-Contracting. The authors say: "*The SLA4D-Grid project is designing and implementing an SLA management layer. The functions of the developments cover the complete SLA lifecycle, including SLA design, contract establishment, SLA provisioning, and SLA monitoring.*".

The SLA@SOI project (Comuzzi et al., 2010) is a FP7 project dealing with the definition of a SLA management framework. The consortium defined a reference architecture, specified a SLA template and the methodology to translate SLAs inot monitoring specifications (for the EVEREST environment). With the RESERVOIR project, they the defined how using cloud standards for the interoperability of cloud frameworks.

In this global trend, multi-agent technologies have been introduced to achieve the automation in creation, execution and monitoring of e-contracts by agents on behalf of users. Lots of work deal with negotiation of the issues related to the content of contract and to their execution(e.g. (Dignum and Sierra, 2001)). The resulting contracts consist in digital agreement between contractual parts where rights and duties in terms of deliverables, costs and delays of the participants are explicitly represented. However such contracts often lead to inflexible relations between participants. The obtained result is in contrary to the requirements of open and dynamic system that are stressed by the actual business paradigms aiming at improving the competitiveness of companies like dynamic virtual enterprises and dynamic service outsourcing (Hoffner et al., 2001). Moreover, few of the existing research works take into account the monitoring of contracts clauses (Padovan et al., 2002) that bind agents together.

In (Boissier and Gâteau, 2007) we proposed an Electronic Institution model based on MAS to manage electronic contract by specifying obligations. In this paper we aim at doing the same for the management of SLA for Cloud Computing. In the next section we present $\mathcal{M}$OISE^{Inst} and $\mathcal{S}$YNAI, normative organization modelling and middleware.

3 NORMATIVE MULTI-AGENT SYSTEM

In this work we propose a multi-agent support for the enactment and monitoring of the different SLO specified in the SLA. SLAs specify the agreement between customers and the provider concerning their participation to the distributed execution of the job. A SLA must describe both the functioning and the structure organizing this functioning. Moreover it contains explicit legal dimensions bearing on the involved participants. In order to take this into account we propose to use a "normative organizational model", called $\mathcal{M}$OISE^{Inst}, to express SLA. This normative organizational model is accompanied by a specialized "normative middleware", called $\mathcal{S}$YNAI[3] to monitor and enforce legal aspects expressed in the SLAs.

3.1 Normative Organization Modelling

$\mathcal{M}$OISE^{Inst} (Gâteau et al., 2007) is founded on the $\mathcal{M}$OISE^{+} organizational model[4](Hubner et al., 2002). It is composed of the following components that are used to specify an organisation of agents in terms of structure, functioning, evolution and norms (OS of the Fig. 3):

- A *Structural Specification* (SS) defines: (i) the *roles* that agents will play in the organization, (ii) the *relations* between these roles in terms of authority, communication or accointance, (iii) the *groups*, additional structural primitives used to define and organize sets of roles;
- A *Functional Specification* (FS) defines global *business processes* that can be executed by the different agents participating to the organization according to their roles and groups;
- A *Contextual Specification* (CS) specifies, a priori, the possible evolution of the organization in terms of a *state/transition graph*;
- A *Normative Specification* (NS) defines the deontic relations gluing the three independant specification (SS, FS, CS). This NS clearly states rights and duties of each roles/groups of SS on sets of goals (missions) of FS, within specific states of CS.

A BNF[5] definition of OS is available in (Gâteau et al., 2004) for SS and FS and in (Gâteau et al., 2005) for CS and NS and in (Gâteau, 2007) for the whole model.

3.2 Normative Organization Middleware

In human societies, institutions define rules (North, 1990) that enclose all kinds of formal or informal constraints that human beings use to interact. These last years, electronic institutions have been introduced in multiagent domain. They propose to model these rules with normative systems (Jones and Carmo, 2001). Institutions are defined as a set of agents,

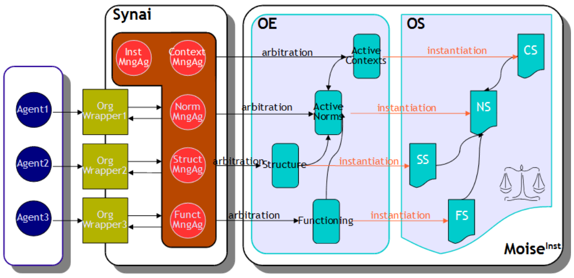

Figure 3: $\mathcal{M}$OISE^{Inst}, normative organization specification model, and $\mathcal{S}$YNAI, normative middleware.

which behave according to norms taking into account their possible violation. The functioning of the agents is supervised and controled with a set of *institution services* (e.g. (Dellarocas, 2000)). The institution services that support and use $\mathcal{M}$OISE^{Inst} are regrouped in a specific "normative middleware" called $\mathcal{S}$YNAI on which the agents execute. This layer is in charge of : (i) managing the life cycle of SS as entering/exiting of agents within the organization, or requesting/leaving of roles or groups by the agents, (ii) coordination of the concurrent execution of FS as commitment to missions or achievement of goals, etc, (iii) dynamic and evolution of the organization state through the CS, (iv) the monitoring and supervision of norms of NS activated/deactivated by the evolution of the organization.

Using both $\mathcal{S}$YNAI and $\mathcal{M}$OISE^{Inst}, we are able to define a SLA management system as a set of agents whose behavior is ruled by an organization expressed in $\mathcal{M}$OISE^{Inst} and controlled by an arbitration system, implemented with $\mathcal{S}$YNAI, that has the possibility to reward or to punish agents in case they respect or not their agreements as expressed in SLAs. Using $\mathcal{M}$OISE^{Inst}, agents used on the provider infrastructure are able to reason and to take into account the different SLAs described with $\mathcal{M}$OISE^{Inst}. The $\mathcal{S}$YNAI platform on its side take into account this specification in order to supervise and control virtual resources. Both layers are based on an agent execution platform.

These four specifications form the Organizational Specification (OS). The Organizational Entity (OE) is then built from the set of agents that have adopted a role according to the SS of the OS. From this time the $\mathcal{S}$YNAI middleware manages and controls the functioning of this OE (composed of active contexts, active norms, current structure and current functioning) by the way of different events corresponding to the entry/exit of agents of the OE, adoption/leaving roles or groups, change of context, commitment to missions, achievement of goals.

[3] $\mathcal{S}$YNAI: SYstem of Normative Agents for Institution.

[4] $\mathcal{M}$OISE^{+}: Model of Organization for multI-agent System.

[5] BNF: Backus-Naur Normal Form.

4 SLA MANAGEMENT ARCHITECTURE PROPOSAL

The framework we propose is based on classic VMware or other virtualisation architecture used by a IaaS provider. In our solution, each physical server is represented by an agent belonging to a $\mathcal{M}$OISE^{Inst} organization. An organization specified with $\mathcal{M}$OISE^{Inst} represents a SLA that a customer could defined. For instance, with help of a web interface, the customer build his virtual network composed of database server, web server, storage server and application server linked between them. Each component has its own characteristic as seen before (CPU, memory, etc.). By drawing its configuration, the customer specifies the Structural Specification of the Organization (a machine selected is a role that a physical server will play through a virtual instance) and the Functional and Normative Specification (each role will have the Obligation to achieve the characteristics defined for each virtual machine which could be considered as the SLO of the SLA).

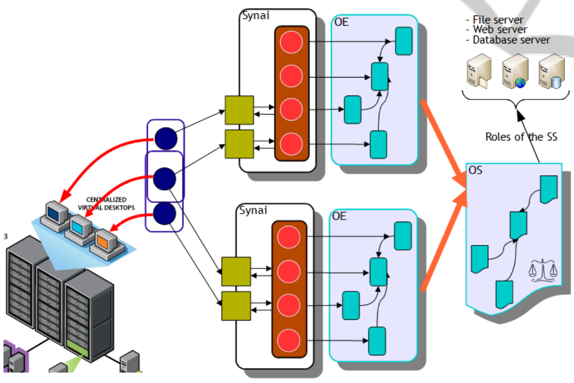

Figure 4: $\mathcal{M}$OISE^{Inst} as SLA template in Cloud Computing architecture.

On the Fig. 4, we represents on the left the infrastructure coming from the datacenter. The agents (in blue) are launched on each physical server and represent them. By creating specific virtual machines, they instantiate the current structure of an organization (a topology buyed by a customer, i.e. a OE instantiating an OS).

The agent act via the $\mathcal{S}$YNAI layer which monitor the agent. The monitoring is used for:

- detecting that a virtual machine violates the SLA for any reason and triggering the agent playing the role (i.e. the physical machine hosting the virtual machine) have to modify the VM configuration in order to respect a specific SLO of the SLA (a norm of the OE)
- showing to the customer the state of the OE regarding the OS (transparency purpose)

5 CONCLUSIONS

Service Level Agreement (SLA) describes agreement on non-functional requirements between provider and customer. SLA consists of service level objectives (SLOs) that are evaluated according to measurable Key Performance Indicators (KPIs). Automatic SLA protection enables further increase of the system utilization and system profit. In currently available systems only some basic SLAs like "uptime over a time period guarantee are available.

This part of cloud computing is target of intensive research now. Formal ways of SLA description have to be standardized. Algorithms and models for the resources allocation, automated SLA protection and evolution has to be found.

These areas are for example in the scope of RESERVOIR (Resources and Services Virtualization without Barriers) - EU FP7 funded project [6] backed by consortium of industry and academy partners.

Future works will concern the specification of the OS with $\mathcal{M}$OISE^{Inst} and its implementation with Utopia (Schmitt et al., 2010; Schmitt et al., 2011). A national project funded by the Fond National of Research[7] in Luxembourg will permits to also work on security and risk through SLA management in cloud computing. We aim at gathering monitoring QoS and security in the same way.

REFERENCES

Boissier, O. and Gâteau, B. (2007). Normative multi-agent organizations: Modeling, support and control, draft version. In G. Boella, L. v. d. T. and Verhagen, H., editors, *Normative Multi-Agent Systems. Dagstuhl Seminar Proceedings 07122, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI)*, Schloss Dagstuhl, Germany.

Comuzzi, M., Kotsokalis, C., Rathfelder, C., Theilmann, W., Winkler, U., and Zacco, G. (2010). A framework for multi-level sla management. In Dan, A., Gittler, F., and Toumani, F., editors, *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*, volume 6275 of *Lecture Notes in Computer Science*, pages 187–196. Springer Berlin / Heidelberg.

Dellarocas, C. (2000). Contractual agent societies: Negotiated shared context and social control in open multi-agent systems. In *Proceedings of the Workshop*

[6]http://www.reservoir-fp7.eu
[7]FNR: www.fnr.lu.

on Norms and Institutions in Multi-Agent Systems, 4th International Conference on Multi-Agent Systems (Agents-2000), Barcelona, Spain.

Dignum, F. and Sierra, C., editors (2001). *Agent-Mediated Electronic Commerce - The European AgentLink Perspective*, volume 1991 of *Lecture Notes in Artificial Intelligence*. Springer Verlag. ISBN 3-540-41671-4.

Gâteau, B. (2007). *Modlisation et Supervision d'Institutions Multi-Agents*. PhD thesis, Ecole Nationale Superieure des Mines de Saint Etienne. defended at CRP Henri Tudor, Luxembourg.

Gâteau, B., Boissier, O., Khadraoui, D., and Dubois, E. (2005). $\mathcal{M}$OISE^{Inst}: An organizational model for specifying rights and duties of autonomous agents. In van der Torre, L. and Boella, G., editors, *1st International Workshop on Coordination and Organisation (CoOrg 2005) affiliated with the 7th International Conference on Coordination Models and Languages*, Namur - Belgium.

Gâteau, B., Boissier, O., Khadraoui, D., and Dubois, E. (2007). Controlling an interactive game with a multi-agent based normative organizational model. In Vzquez-Salceda, J., Boella, G., Boissier, O., and Matson, E., editors, *"Coordination, Organizations, Institutions, and Norms in Agent Systems II"*, volume 4386 of *LNCS*, pages 86–100. Springer Berlin / Heidelberg. ISBN: 978-3-540-74457-3.

Gâteau, B., Khadraoui, D., and Dubois, E. (2004). Architecture e-business sécurisée pour la gestion des contrats. In *3ème Conférence sur la Sécurité et Architectures Réseaux (SAR)*, La Londe, Cote d'Azur - France.

Hoffner, Y., Field, S., Grefen, P., and Ludwig, H. (2001). Contract-driven creation and operation of virtual enterprises. *Comput. Networks*, 37(2):111–136.

Hubner, J. F., Sichman, J. S., and Boissier, O. (2002). $\mathcal{M}$OISE^{+}: towards a structural, functional, and deontic model for mas organization. In *Proceedings of the first International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 501–502, Bologna, Italy. ACM Press. ISBN 1-58113-480-0.

Jones, A. and Carmo, J. (2001). *Handbook of Philosophical Logic*, chapter Deontic logic and contrary-to-duties, pages 203–279. Kluwer.

Koetsier, M., Grefen, P. W. P. J., and Vonk, J. (2000). Contracts for cross-organizational workflow management. In *EC-WEB '00: Proceedings of the First International Conference on Electronic Commerce and Web Technologies*, pages 110–121, London, UK. Springer-Verlag. ISBN 3-540-67981-2.

Krishna, P. R. (2010). *E-contract modeling and e-enactement*. PhD thesis, International Institute of Information Technology, Hyderabad - 500 032, INDIA.

Laurikkala, H. and Tanskanen, K. (2002). Managing contracts in virtual project supply chains. In *PRO-VE '02: Proceedings of the IFIP TC5/WG5.5 Third Working Conference on Infrastructures for Virtual Enterprises*, pages 93–100, Deventer, The Netherlands, The Netherlands. Kluwer, B.V. ISBN 1-4020-7020-9.

North, D. C. (1990). *Institutions, Institutional Change and Economic Performance*. Political Economy of Institutions and Decisions. Cambridge University Press, 1st edition edition. ISBN 0521397340.

Padovan, B., Sackmann, S., Eymann, T., and Pippow, I. (2002). A prototype for an agent-based secure electronic marketplace including reputation tracking mechanisms. *International Journal of Electronic Commerce*, 6(4):93–113.

Schmitt, P., Bonhomme, C., Aubert, J., and Gâteau, B. (2011). Programming electronic institutions with utopia. In Aalst, W., Mylopoulos, J., Sadeh, N. M., Shaw, M. J., Szyperski, C., Soffer, P., and Proper, E., editors, *Information Systems Evolution*, volume 72 of *LNBIP*, pages 122–135. Springer Berlin / Heidelberg. ISBN 978-3-642-17722-4.

Schmitt, P., Bonhomme, C., and Gâteau, B. (2010). Easy programming of agent based electronic institution with utopia. In *In proceedings of the 10th international conference on New Technologies of Distributed Systems (NOTERE 2010)*, Tozeur - Tunisia.

Wieder, P., Hasselmeyer, P., and Koller, B. (2009). Enhancing a national academic computing infrastructure with e-contracting capabilities. In Cunningham, P. and Cunningham, M., editors, *Proceedings of eChallenges e-2009 Conference*. ISBN: 978-1-905824-13-7.