CLOUD MANAGEMENT ARCHITECTURE IN NGN/NGS CONTEXT
QoS-awareness, Location-awareness and Service Personalization

Rachad Nassar and Noëmie Simoni
TELECOM ParisTech - LTCI - UMR 5141 CNRS, 46, rue Barrault F 75634, Paris Cedex 13, France

Keywords: Cloud computing, NGN/NGS, E2E QoS-awareness, Location-awareness, Service personalization, SOA.

Abstract: Cloud computing has become one of today's hot topics. The major contribution of this Internet-based service delivery paradigm consists in offering computing, storage and network resources able to guarantee information technology externalization. In parallel to this novel trend, cloud users requirements are quickly emerging due to both network and service convergence. Therefore, beyond its externalization solution, cloud must also respond to users needs within this "Next Generation Networks/Next Generation Services" (NGN/NGS) context. Hence, it should offer service personalization for cloud users, take into consideration their mobile context, and guarantee an end-to-end QoS. In this paper, we propose a QoS-based cloud management architecture that overcomes the aforementioned challenges through several mechanisms. First, we surpass mobility and E2E QoS challenges by gathering ubiquitous elements into ubiquity-based virtual communities. Second, we ensure service personalization by proposing a seamless and dynamic service composition based on stateless services. Finally we take into consideration user's ambient context by using location-based virtual communities. Computing models for QoS-aware and location-aware clouds are also provided.

1 INTRODUCTION

Nowadays, cloud computing is a new buzzword around the globe. Many different definitions has been given for this concept. In this paper, we have chosen the one proposed by Prof. Ian Foster (Foster et al., 2008), since it combines the main key aspects that distinguish clouds from previous distributed computing paradigms. According to this definition, cloud computing is "a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet". Based on this definition, cloud computing is considered as a vast quilt on which externalized services are running. These services arise from heterogeneous domains (Telco, Web and IT) and are executed separately. Hence, we believe that the success of cloud computing should not be limited to externalization, but it should also be related to service convergence which is the heart of our Next Generation Services (NGS) context. The latter consists in seamlessly composing global Cloud Services (CSs) by using stateless service elements. Cloud Service Providers (CSPs) are then able to dynamically create and offer users any desired CS.

Moreover, based on this same definition, we strongly believe that clouds success also depends on the ability of CSPs to dynamically maintain and manage their resources in order to best answer the users requirements. However, with the rapid evolution of Next Generation Networks (NGN) and its network convergence aspect, users needs are quickly emerging. In fact, within the NGN context, all access networks converge into one IP core network. Cloud users are then able to access different CSs that are provided by different CSPs, while using different equipment technologies and access points. Thus, cloud users become more nomadic and want to access their services anywhere, anytime and anyhow. Therefore, to attract more users, CSPs should guarantee an End-to-End (E2E) Quality of Service (QoS) by taking into consideration real-time users preferences and contexts.

In essence, users can have functional and non-functional preferences. The former are characterized by personalized CSs that dynamically and transparently adapt themselves to users needs without session interruption; the latter are characterized by QoS requirements that appear in Service Level Agreements (SLAs) signed between CSPs and cloud users. In fact, this user-centric approach that is based on service per-

Nassar R. and Simoni N..
CLOUD MANAGEMENT ARCHITECTURE IN NGN/NGS CONTEXT - QoS-awareness, Location-awareness and Service Personalization.
DOI: 10.5220/0003390106290635
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 629-635
ISBN: 978-989-8425-52-2

sonalization and QoS preferences is highly desirable in future cloud environments since it increases the Return On Investment (ROI) for CSPs on one hand and allows to enrich users experiences on the other hand. However, to the best of our knowledge, this research area is not fully investigated within the context of cloud computing.

Another main aspect that CSPs should take into consideration is users ambient context. The latter can be characterized by users locations or by their activities. In the scope of this paper, we treat the spatial mobility aspect and the influence of users locations on CSs provisioning. In fact, with the resource externalization trend into cloud systems, CSs sources are often far away from end devices. Thus, the latency - the time to send or receive one byte of data - is directly influenced, since it depends on the distance separating a user from his CS. For instance, network latency of few milliseconds becomes critical for highly interactive services such as immersive applications, since it can cause a noticeably degraded users experience. In order to minimize latency, CSPs must offer a more adapted scalable distribution of their resources. In addition, they should connect cloud users, specially those requiring highly immersive interactions, to the closest CSs sources.

After having introduced the main aspects that motivate our work, we propose in the following of this paper solutions to overcome the aforementioned challenges. The proposed solutions answer the following questions: how can CSPs dynamically and transparently guarantee personalized services for users? how can they maintain, in real-time, the required E2E QoS? and how can they deliver CSs while taking into consideration users locations?

The remainder of this paper is organized as follows. In Section 2, we discuss some of the related works. In Section 3, we propose a QoS model, an ubiquity-based virtual community concept and a dynamic service composition, in order to ensure QoS-aware clouds and personalized CSs. In Section 4, we allow CSPs to provide location-based CSs by proposing a location-based virtual community concept. The conceptual QoS and location-aware cloud architecture is presented in Section 5. Finally, conclusion and future perspectives are presented in Section 6.

2 RELATED WORK

Service personalization, E2E QoS management and location-awareness are highly important research topics for future cloud environments. In this section, we discuss some of the investigations in these aspects.

In the context of *Personalized Services*, the authors in (Guo et al., 2009) focus on how to provide personalized services for users by using a client side program that records user activities and computes models on personal devices. Afterwards, selected models are uploaded to service providers so they can have better knowledge of that user and consequently provide him personalized services. The drawback of this approach lies in the fact that pushing all models and computing programs to clients side contradicts the cloud computing externalization concept. Moreover, this previously mentioned paper investigates the commonly known description of service personalization as the ability of service providers to offer, or rather impose adequate services to users, while being based on their analytical records. In opposite to the previous description, our service personalization approach is based on a dynamic service composition where users can impose their own preferences.

Some of the papers found in the literature reveal a growing interest in the topic of *QoS-awareness* in cloud environments. In (Spillner and Schill, 2009), the authors present, according to the service run-time behavior, a technique to dynamically adjust SLA constraints after adjusting the values of service description's nun-functional properties. In addition, QoS-awareness is also discussed in (Korn et al., 2009) where SLA monitoring is delegated to a third independent party, namely Service Level Management Authority (SLMA). In opposite to these approaches, our proposition overcomes QoS violation without modifying any SLA parameter. Moreover, we propose auto-managed QoS-aware CSs.

Finally, to the best of our knowledge, the topic of *location-aware* clouds has not yet received much attention. Moreover, research communities who treat this subject, propose externalized services that provide location-based information. For instance, the authors in (Wang and Yang, 2009) propose a mobile information retrieve system based on GPS and Web2.0 applications. Through this platform, users can obtain location-based information and personalized recommendations. In our proposition, we guarantee location-based information for users, but we also apply the location-awareness concept on CSs. In this context, based on their users locations, CSPs offer distributed CSs. Hence, we overcome the latency problem by allowing users to access the closest CS.

3 QoS-AWARE CLOUD MANAGEMENT

In this section, we introduce in the first subsection our QoS model according to which we gather CSs into Ubiquity-based Virtual Communities. The latter concept is introduced in the second subsection. In addition, a QoS agent is integrated into each CS in order to ensure an autonomous management of these communities. All these concepts are merged into a QoS-aware computing model that is presented along with a QoS management architecture in the last subsection.

3.1 QoS Model

Nowadays, industrial providers are offering different types of Infrastructures as a Service (IaaS), Platforms as a Service (PaaS) and Software as a Service (SaaS). Therefore, heterogeneous CSs have been provided for cloud users. In parallel, the heterogeneity aspect also appears on the users access level, since they can choose among different types of terminals and access points. Consequently, if CSPs want to provide CSs with a respected E2E QoS, they must first overcome the heterogeneity challenge. For this purpose, we propose to provide an homogenous vision for all existing cloud elements by introducing a new QoS model. The latter is based on four criteria that are defined as follow:

- **Availability.** It represents the ability of a CS to be accessed at a certain time. It indicates the ratio of accessibility for a CS.
- **Reliability.** It represents the ability of a CS to be used without deteriorating the information and while respecting contract conditions. It indicates the percentage of unintentional modifications in the information caused by the CS.
- **Delay.** It indicates the duration of a request treatment by a CS.
- **Capacity.** It reflects the ability of a CS to treat normally a request while using all the possible treatment means. It indicates the charge rate of the CS.

We note that there is no need to combine these four criteria into one formulation, since each variation in one of the CS QoS characteristics will be solved by passing the request into a ubiquitous element as we will see in the next subsection.

3.2 Ubiquity-based Virtual Communities: VXCU

Due to the NGN evolution, cloud users are able to access different CSs while using different terminals, access points and networks. In addition, they become more nomadic and aim to conserve their continuous session despite their situations. To reach these goals and to guarantee an E2E QoS for their users, CSPs should overcome the mobility challenge consisting of four mobility types (Guo et al., 2008): terminal mobility, user mobility, network mobility and service mobility. In this subsection, we surpass the aforementioned problem by gathering ubiquitous CSs into Ubiquity-based Virtual Communities (VXCU); X={S,C,E} which represents different types of CSs: applicative Services (S), abstracted Connectivity networks (C), and Equipment (E) such as servers and platforms. We note that in our context, ubiquitous CSs correspond to those having the same functionality and an equivalent QoS. Therefore, throughout end-users situations, we manage the continuity of their sessions by provisioning VXCUs that respond to their functional and nonfunctional (QoS) preferences.

In many cases, specially in mobile context, a CS that is used in a user session might not continue to function normally or to fulfill the user QoS requirements. For this reason, we solve the problem by using the corresponding provisioned VXCU. In fact, we dynamically replace the current CS by an ubiquitous counterpart that belongs to the same VXCU. Therefore, we seamlessly adapt the user's session against any degradation and we guarantee the continuity of this session while maintaining the E2E required QoS. In the following, we divide the VXCU management process into two phases: the creation and the exploitation phases.

In the creation phase, each CS joins an existing VXCU or creates a new one. The creation process is based on three basic services:

- Ubiquity Inquiry Service: determines the functionality and QoS criteria of a CS while having its ID as input.
- Discovery Service: launches a search in order to discover CSs that verify the criteria indicated at the input. In the ubiquity case, a functionality and QoS criteria are considered as inputs.
- Presence Service: filters an obtained list according to CSs states. It selects among a list of CSs IDs the ones that are "Available" (Accessible by the end-user), "Activable" (Activated by the CSP but not yet used) and "Activated" (Activable and used).

The creation process is launched during each CS deployment. First, the Ubiquity Inquiry Service is invoked in order to get the functionality and the QoS criteria of this CS. Second, the Discovery Service uses the former output as searching criteria. Thus, we obtain a list of CSs having the same functionality and QoS as the CS in question. Finally, the Presence Service applies its filtering process on the obtained list. In consequence, the VXCU verifying the functionality and QoS of this CS is determined. Thus, the latter joins this community. We note that if no existing VXCUs verify the required QoS and functionality criteria, the CS in question creates a new VXCU having its own characteristics.

After explaining the creation process, we present the exploitation phase, where the CS is already a member of a VXCU. We mention that the VXCU concept is based on a Peer-to-Peer (P2P) self community management process in which each CS acts as an auto-managed peer. For this purpose, a QoS-Agent is integrated into each CS. During this dynamic VXCU auto-management process, each CS compares its current QoS and functionality with the ones of its current VXCU. If the result is positive, then the QoS-Agent sends a notification called “IN Contract” in order to inform the other VXCU members that the community contract is still respected. If not, the QoS-Agent sends a notification called “OUT Contract”. In this case, the CS leaves the VXCU and finds another one.

3.3 Computing Model for QoS-aware Clouds

In this subsection, we explain how the aforementioned QoS management model and concepts are able to dynamically and seamlessly adapt users sessions against any variation in users preferences. For this purpose, we propose a solution that allows cloud users to create their own personalized sessions by combining different CSs provided by different CSPs.

Our personalization vision is based on a dynamic seamless composition that is provisioned and tailored according to users preferences. To reach this goal, a new model is introduced, namely the Virtual Private Network (VPXN) (Guo et al., 2008); X={S,C,E} since for each cloud user we provision VPXNs on the applicative Services layer (VPSN), Connectivity networks layer (VPCN) and Equipment layer (VPEN). In fact, these personalized VPXNs gather CSs that best answer the cloud user’s preferences. Consequently, the user’s session is dynamically created by combining different CSs from these VPXNs. To support this dynamic composition, management services are introduced in (Nassar and Simoni, 2010):

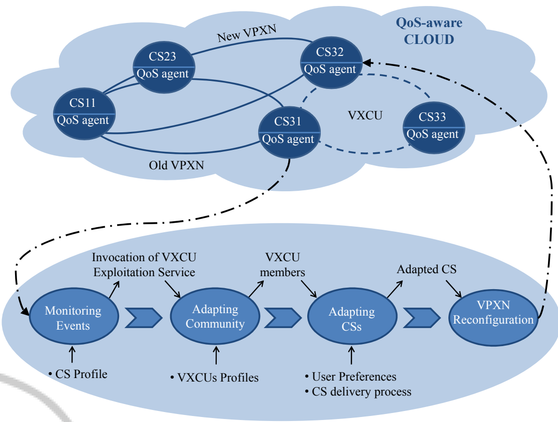

Figure 1: Computing Model for QoS-aware Clouds.

- Service Logic: represents the service composition workflow. By analogy, there are Network Logic and Equipment Logic.
- Semantic Routing: routes user’s request to the next SE in the Service Logic. To support this function-based routing, a QoS Routing Table is used. By analogy, on the network and equipment layers, there are the well-known routing mechanisms (e.g. MPLS, OSPF, etc.) and handover techniques.
- Service Selection: selects the chosen SE. By analogy to this service, we introduce the Network and Equipment Selection services.
- VPXN Configuration: modifies and configures cloud users VPXNs.

In addition to these management services, we introduce in this paper a QoS Policy Agent that verifies the integrity of the independent modifications that are caused, on each layer, by each of the aforementioned services. It is a decision table that monitors E2E QoS and is considered as a part of the AmbientGrid (Simoni et al., 2008). The latter is an information inference driven by the profiles defined in the knowledge base, namely “Infoware” (Simoni et al., 2008). It dynamically manages ambient resources in a personalized way.

In our proposition, we favor an event-based approach. For this reason, this QoS management architecture is associated with a computing model that analyzes the QoS-based event and consequently adapts the VPXN configuration. As shown in Figure 1, a provisioned CS in the user’s Old-VPXN (e.g. CS31) could not maintain the required QoS. Its QoS-Agent detects this QoS degradation and notifies the cloud management system. The latter first monitors the event. Since it is a QoS degradation, the VXCU Exploitation service is invoked, and the “Adapting Com-

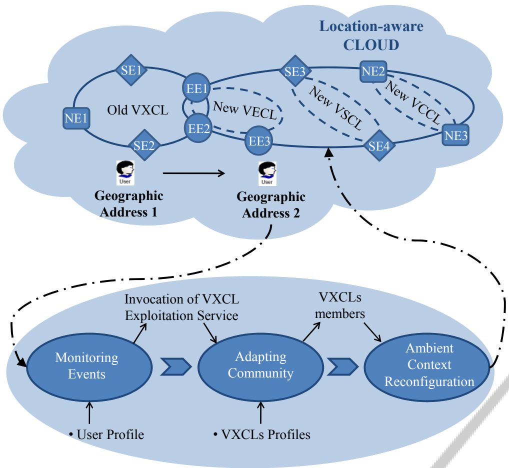

Figure 2: Computing Model for Location-aware Clouds.

munity" phase is launched. Based on the CS's functionality and QoS, this phase provides all possible ubiquitous members. Thus, according to the aforementioned CS delivery process (service composition according to user preferences: CS logic, semantic routing, etc.), the "Adapting CSs" phase chooses the adequate counterpart (e.g. CS32) to replace the current unwanted CS (CS31). Finally, the "VPXN Reconfiguration" phase is launched and a New-VPXN is then created. In consequence, this computing model demonstrates how the VXCU concept and the personalized service composition process are used to dynamically and seamlessly manage users sessions while conserving their E2E QoS and taking their preferences into consideration.

4 LOCATION-AWARE CLOUD MANAGEMENT

In order to reduce latency influence on cloud users experiences, CSPs are supposed to distribute their resources according to their users specific locations (home, work, hotels abroad, etc.). In this section, we propose a management mechanism that allows the provisioning of location-based CSs. Therefore, throughout cloud users movements, we anticipate their demands by providing the nearest resources.

Our location-awareness proposition considers the perviously discussed virtual community concept. In fact, we sense user's location and use it to create and manage Location-based Virtual Communities (VXCLs, with X={S,C,E}). The latter gather all CSs that have the same location as the user. Consequently, for each users location, three VXCLs are provisioned (VSCL, VCCL and VECL). In the following, we divide the VXCL management process into two phases: the creation and the exploitation phases.

In order to create VXCLs, we use the following basic services:

- Geo-Location Service: determines an element's geographic location while having its Element ID as input.
- Discovery Service: previously defined in subsection 3.2. However, in this case, the discovery is based on the user's location.
- Presence Service: previously defined in subsection 3.2.
- Sorting By Type Service: sorts by type (applicative Services, Connectivity networks, Equipment) a list of Element IDs received as an input.

The creation process is executed in advance for user's specific locations. Hence, it allows to anticipate user's movements. First, the Geo-Location Service determines user's specific location. Second, the Discovery Service uses the obtained result as its searching criterion. Thus, we get a list of CSs that have this same location. Third, the Presence Service applies its filtering process on this list. The obtained VXCLs verify user's location. At the end of this process, the Sorting By Type Service filters the obtained VXCLs according to CSs types. In summary, the aforementioned steps provision the VSCL, VCCL and VECL corresponding to user's specific location.

After explaining the creation phase, we present the exploitation phase during which the user is moving. Instantly, we offer the user the three provisioned CSs lists that correspond to his current location.

According to our event-based approach, we introduce a computing model that analyzes the location-based event and consequently adapts the user's ambient context configuration. As shown in Figure 2, a user moves towards a new location (Geographic Address2). Consequently, a notification is sent to the cloud management system. The latter first monitors the event. Since, it is a modification in user's location, the VXCL Exploitation service is invoked and the "Adapting Community" phase is launched. This phase provides all the location-based communities (VSCL, VCCL, VECL) that are provisioned to this user's specific location. Finally, the "Ambient Context Reconfiguration" phase is launched and a new ambient context (New-VSCL, New-VCCL, New-VECL) is then created. Hence, this computing model shows how the VXCL concept can be used to dynamically and instantly enrich user's experience and context-awareness.

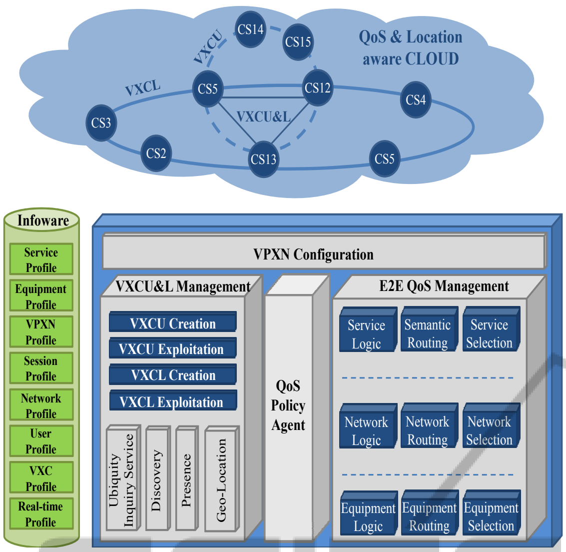

Figure 3: Cloud Management Architecture.

5 CLOUD MANAGEMENT ARCHITECTURE

After introducing the novel QoS-aware and location-aware cloud management concepts, we structure in this section all the aforementioned mechanisms and basic services into a cloud management architecture. As previously stated in the proposed computing models, we adopt an event-based approach. In addition, for an efficient and flexible usage of clouds, we adopt a Service Oriented Architecture (SOA) approach. The latter organizes reusable, autonomous and interoperable services into the service layer. However, according to our NGN/NGS context, we do not limit our proposition to the aforementioned SOA characteristics. In contrary, the added-values of our approach consists in adding the following architectural aspects:

- Mutualization: It represents the ability of a service to be not only reusable, but also shareable. Consequently, depending on its QoS criteria, the mutualized service is shared among several requests. The untreated requests are stocked in a queue that is associated to the service. If the Timeout of a request is less than its waiting delay in the queue, our service delivery system reroutes the request towards a ubiquitous CS.
- Auto-Management: It represents the ability of a service to be not only autonomous, but also auto-managed. For this reason, we have introduced a QoS-Agent as described in subsection 3.2.
- Interconnection: It represents the ability of a service to be not only interoperable, but also interconnected with others in order to dynamically and seamlessly compose personalized services. For this purpose, we have already presented the VPXN concept in subsection 3.3. The latter introduces "Virtualization" as another characteristic of our architecture. It allows to transparently offer services for the users and to let them use these services as if they were dedicated to them.

The cloud management architecture is shown in Figure 3. It manages the cloud while taking into consideration all the NGN and NGS challenges. For this purpose, this global structure gathers all the previously proposed management mechanisms. The "VXCL Creation" and "VXCL Exploitation" parts support the user-centric approach by enriching users context-awareness. The "VXCU Creation" and "VXCU Exploitation" parts allow to overcome the NGN mobility challenge by gathering ubiquitous CSs into auto-managed communities. The VXCU concept is combined with the "QoS Policy Agent" and the "E2E QoS Management" parts in order to guarantee an E2E QoS for cloud users. The "VPXN Configuration" part supports the NGS context by ensuring personalized, dynamic and seamless CSs compositions. Moreover, a knowledge base is necessary in order to provide all needed information. For this purpose, we use the Infoware (Simoni et al., 2008) which contains several XML profiles (for CSs, cloud users, real-time, sessions, communities, etc.). It efficiently and dynamically manages the decisional and reactive information. It is not a simple data base like the Home Subscriber Server (HSS) in the IP Multimedia Subsystem (IMS) architecture. On the contrary, it is a well structured knowledge base acting as a real-time informational inference. It is worth noting that within the project in which our work is situated, security aspects are also investigated by another research group.

Furthermore, as previously discussed, E2E QoS is guaranteed by creating VXCUs that gather ubiquitous CSs which are deployed all over the world. However, creating VXCUs with huge number of ubiquitous CSs is not efficient, since it makes the stored information more bulky and takes longer time to decide the adequate substitute for the degraded CS in user's session. In order to minimize the time and information clarity costs, we propose to filter ubiquitous CSs by using specific locations. Consequently, we create virtual communities that are simultaneously based on ubiquity and location. In comparison with normal VXCUs, the former communities are able to manage clouds in more efficient way. These communities are named Ubiquity and Location based Virtual Communities (VXCU&L).

As shown in Figure 3, one possible technique to create VXCU&Ls is to filter the VXCUs by using users specific locations. For this purpose, we combine the VXCU that contains ubiquitous CSs with the VXCLs that are provisioned for user's specific location. In this way, we create the VXCU&Ls that gather ubiquitous CSs found around the user's geographic location. An example is given in Figure 3 where the VXCU contains {CS11,CS12,CS13,CS14,CS15} and the VXCL contains {CS11,CS12,CS13,CS2,CS3,CS4,CS5}. Consequently, the VXCU&L is dynamically created and contains {CS11,CS12,CS13}. Moreover, we consider that CS11 is part of user's session and it suffers from a QoS degradation. Hence, we neglect CS14 and CS15 and choose the substitute between CS12 and CS13. This technique allows to react faster against QoS degradation events and to manage more efficiently the E2E QoS.

6 CONCLUSIONS AND PERSPECTIVES

Through its externalization aspect, cloud computing provides a combination of cost and agility savings. In fact, organizations prefer saving time and energy on managing services that can rather be moved to the cloud. However, externalizing services induces a growing need for ways to manage and enhance cloud environments, in order to best answer the cloud users requirements. The latter, within an NGN/NGS context, have become more mobile and more greedy. They aim to access any type of CS anytime, anywhere and anyhow. Thus, an efficient cloud environment that guarantees users preferences and needs is required. For this reason and in order to overcome the previous challenges, we have proposed in this paper a cloud management architecture composed of innovative mechanisms. In this ameliorated SOA architecture, we have favored service personalization over monolithic applications. For this purpose, we have introduced a seamless and dynamic service composition that combines stateless and mutualized service elements. In fact, this CS composition takes into consideration users preferences, such as the required E2E QoS. In consequence, we have proposed the ubiquity-based virtual communities concept which is based on a QoS model. By gathering ubiquitous elements having the same functionality and an equivalent QoS, we have conserved the continuity of cloud users sessions while guaranteeing their required E2E QoS. Moreover, in order to reduce the latency, we have provisioned location-based virtual communities gathering CSs close to users specific locations. Both ubiquity and location based community concepts are supported by event-based computing models. Finally, for a more efficient management, we have proposed to filter the VXCUs provisioning according to specific locations. For this reason, we have created several VXCUs in different locations in order to best manage the users required E2E QoS.

However, for different users locations, different VXCUs would be used to guarantee the users session continuity. In our future work, we must investigate the transition of the management process between two VXCUs when the user is moving. By analogy to the handover on the access networks level, this future discussion subject is named “Semantic Handover”.

ACKNOWLEDGEMENTS

The authors would like to thank all the participants in the UBIS project, financed by the French ANR VERSO 2008 in which is situated our work.

REFERENCES

Foster, I., Zhao, Y., Raicu, I., and Lu, S. (2008). Cloud computing and grid computing 360-degree compared. In *GCE'08, Grid Computing Environments Workshop*.

Guo, H., Chen, J., Wu, W., and Wang, W. (2009). Personalization as a service: the architecture and a case study. In *CloudDB'09, International Workshop on Cloud Data Management*.

Guo, W., Simoni, N., and Yin, C. (2008). Automated management of user centric session in ngn. In *GLOCOMW'08, IEEE GLOBECOM Workshops*.

Korn, A., Peltz, C., and Mowbray, M. (2009). A service level management authority in the cloud. In *HP Laboratories Technical Report*.

Nassar, R. and Simoni, N. (2010). Ngn/ngs components for service personalization in a mobile and heterogeneous context. In *IWUSP'10, International Workshop on Ubiquitous Service Platforms*.

Simoni, N., Yin, C., and Chene, G. D. (2008). An intelligent user centric middleware for ngn: Infosphere and ambientgrid. In *COMSWARE'08, International Conference on COMmunication System softWAre and MiddlewaRE*.

Spillner, J. and Schill, A. (2009). Dynamic sla template adjustments based on service property monitoring. In *CLOUD'09, IEEE International Conference on Cloud Computing*.

Wang, Z. and Yang, F. (2009). A multiple-mode mobile location-based information retieve system. In *ICWMC'09, International Conference on Wireless and Mobile Communications*.