Towards a Mobile Cloud Context_Aware Middleware

Manel Gherari
LAMIS Laboratory
University of Tebessa, Algeria
Gherari.manel@yahoo.fr

Abdelkrim Amirat
LIM Laboratory
University of Souk-Ahras,Algeria
abdelkrim.amirat@yahoo.com

Mourad Oussalah
LINA Laboratory
University of Nantes, France
mourad.oussalah@univ-nantes.fr

Abstract: **Due the fact that Mobile Cloud Computing (MCC) is a recent paradigm, it lacks tools and frameworks to model its architecture. Existing ADLs (Architectural Description Languages) do not support modelling Mobile Cloud Architecture (MCA), since the latter encompasses more complicated and rich architectural features -including contextual information- than traditional applications. In this paper we propose a Framework that allows: 1- Describing mobile Cloud applications architecture with the Mobile Cloud ADL (MC-ADL). 2- Provides context-aware middleware Smart Cloud Gate to ensure contextual configuration of the mobile cloud application at runtime and contextual evolution of Mobile Cloud Application's architecture**.

Keywords: Mobile context, Cloud context, Mobile Cloud Computing, Mobile Cloud Architecture, Mobile Cloud Architecture Evolution, Mobile Cloud Middleware, Context Awareness.

1 INTRODUCTION

Mobile devices have become an increasingly important part in human life, they are no longer a luxury but a must. Nowadays, mobile applications are becoming more and more ubiquitous and provide even richer services on mobile devices. As predicted by Walsh mobile devices will overtake pc as the most used means to acess the world wide web [1]. By the end of 2014 mobile Cloud applications will deliver annual revenues of 20 billion dollars [2], yet limitations of mobile devices (low bandwidth, small storage capacity, battery's short life, reliability, etc.) are a hindrance which must be outdone. Mobile technologies and Cloud's marriage is the optimal solution to the limitations mentioned earlier. This new technology (cf. Mobile cloud Computing) aims to bring the benefits of the Cloud to mobile devices to improve their performance (e.g. extends battery life, improve data storage, increase processing powers, etc.).

Mobile Cloud Computing (MCC), takes mobile devices to a whole new level where the latter rely on the Cloud to store and process data. Therefore, we can consider that mobile Cloud applications represent the next generation of mobile applications.

It is noteworthy that despite the benefits associated with the adoption of the Cloud by mobile technology the gate to the Cloud remains frozen, this means that the application often use the same services without having an update of the novelty in Cloud, thus applications lack awareness of new services that are more advantageous in terms of features and qualities than the currently used ones. Other hindrances figure in term of mobility, Cloud context awareness, real time requirements and so on.

Mobility has taken global computing infrastructure from static, homogenous, powerful desktop computer to a highly dynamic, heterogeneous, resource-constrained handled and wearable computer level [3]. Furthermore the Cloud has taken mobile computing to another higher level, where computational and storage is outsourced into the Cloud, this new paradigm demands a radically new software architectural paradigm that addresses challenges related to Cloud Mobile Application implementation, access schemes to hybrid Cloud by mobile application, networks heterogeneity and mobile Cloud application (MC-APPs) development.

The importance of context lies in the fact of that it provides important information about the resent status of people , places, things, and devices in the environment. [4]

Mobile Distributed computing systems is a new form of mobile computing that is concerned not just by mobile computing but also by mobile people. The main challenge of this kind of mobile computing is to take advantage of the changing environment with a new class of applications that are aware of the context in which they run, these applications adapt according to the location of the user. [5]

Mobile cloud applications will be more efficient if they rely on contextual information collected from the cloud and the mobile counterparts. Mobile Context encompasses more than just its location and time, but also include its intrinsic information (e.g. RAM, CPU,OS, Bandwidth and etc.). While Cloud Context regroups all the information concerning the cloud service and its provider (e.g. Type of Service, Pricing Model, and QoS, etc.).

In order to exploit the full potential of Cloud computing to enhance the mobile devices performances, the developer must have a complete awareness of both mobile's and Cloud's contexts. For this mobile Cloud applications need an architectural representation described in a manner that fosters a contextual reconfiguration and evolution.

In mobile cloud modelling field Cavalcante propose Cloud-ADL [6]. Cloud-ADL is a customized architecture description language (ADL) that takes advantage of ACME [7] elements to model the architecture of a Cloud application. Cloud-ADL provides constructors that enable to specify programmed (i.e. foreseen at the architectural level) dynamic reconfiguration actions to be taken at the runtime level.

MODACloud is a framework that tames the complexity of development and administration of multi-cloud systems by offering a domain-specific modelling language along with a run-time environment that facilitate the specification of provisioning, deployment, and adaptation concerns of multi-cloud systems at design-time and their enactment at run-time.

MODACloud offers an advance quality design, development and operation methods based on Model Driven Development (MDD) Paradigm. It start by providing a Cloud Application Modelling Language and framework CloudML to create application taking in consideration the difference in the underlying infrastructure[8].

Till this time and based on our bibliographic research in MCC field [9] [10], a framework for modelling a Mobile Cloud Application has not been proposed yet.

From this point, we present in this paper a framework for modelling mobile Cloud applications. Mobile Cloud ADL (MC-ADL) that addresses the aforementioned issues. MC-ADL does not only allows Mobile Cloud application's architecture description with EMF and GMF tools [11], but also implements a dynamic evolution mechanism as Atlas Transformation Language (ATL) rules [12]. To maintain the awareness on both Cloud's and mobile's context; Smart Cloud Gate Middleware is responsible of the dynamic contextual re-configuration of the MC-Apps at runtime level, and at the same time provoking the contextual evolution rules at architectural level in order to keep the consistency between the two levels (Architectural and runtime levels).

The rest of this paper is organized as follow: In Section 2 we will discuss Our contribution as a proposed approach, in section 3 we present a case study. Finally related works in section 4 and a conclusion will be given in section 5.

2 PROPOSED APPROACH

The popularity of cloud computing has led providers to offers an ever growing solutions to consumer. Exploiting the particularities of each cloud will enhance the performance, the availability and the cost of the mobile application. Unfortunately these cloud solutions are usually heterogeneous and incompatible which lead to many difficulties during the development and administration of multi cloud mobile applications (i.e. mobile applications that use services provided by hybrid clouds). Furthermore both mobile's and Cloud's contexts are managed separately, therefore the benefits of the Cloud upon mobile technology remains limited. Also in this new paradigm (i.e. Mobile Cloud Computing) most if not all researchers' and business actors' intention is focused on the underlying infrastructure of MCC, and more important on how to improve the performance of mobile Cloud applications and overcame issues related to battery use, storage capacity and bandwidth. As a result, considering that Cloud services provisioning is continuously operating activity, fostered by the competition between cloud vendors or cloud services providers. Services that have recently been deployed in the Cloud infrastructure remain unused or unknown at architectural level. Since this mechanism is usually implemented via middleware and its active at runtime without keeping a link with the architecture -if it does exist- of the application.

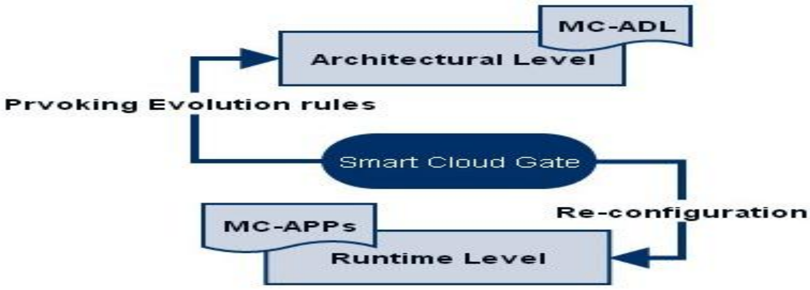

Figure 1: Smart Mobile Cloud Environnement.

An abstract view of mobile Cloud application facilitate the reasoning on the application before its implementation, furthermore foster its reusability, auto-adaptability and extends the mobile Cloud application's lifespan usability, since the change that could occur while deploying the application have been already foreseen in the architectural design phase and applied at runtime level dynamically.

Unfortunately traditional ADLs [13],[14] do not have means to represent mobile Cloud elements, since the latter are more complicated and have rich architectural features than a traditional applications.

Mobile Cloud Architecture Description Language (MC-ADL), it enables the conception of mobile cloud applications as an eclipse plug-in with developed EMF and GMF tools.

In order to exploit the full potential of Cloud computing to enhance the mobile devices performances, the developer must have a complete awareness of both context mobile and Cloud. For this mobile Cloud applications need an architectural representation like any other application. Once the MC-Apps' architectural design is reasonably complete it is supposed to be realized faithfully in the implementation phase.

Addressing challenges aforementioned, the question that must be asked is how to take a full advantage of the Cloud in a smart manner? Our suggested answer is a smart environment composed of three levels : architectural level, the intermediate level that is represented by Smart Cloud Gate (SCG) contextual middleware, and finally the last level consists of application level (i.e. runtime level).

An overview of the proposed environment is shown in Figure 1, where we can notice that SCG is responsible for provoking evolution rules that will be applied at the MC-Apps' architecture at the same time, any evolution that occurs at architectural level will have an effect of triggering the contextual re-configuration of the MC-App at Runtime level and vice versa.

A further explanation of each level will be presented in the following subsections.

In this paper we will discuss the general view of the environment with a particular focus on the description and the evolution of MC-Apps' architecture (i.e. architectural level).

Architecture as defined by Garlan [15] is a level of design that foster the description of elements that compose the system, including the interaction among them and the pattern that guide their composition and construction.

Traditional Architectural Description Languages (ADL) represent computational components as abstract components related by connectors that model their interactions and finally a configuration that describe the way that components react and related with each other. If we try to represent mobile Cloud architecture with traditional ADLs; a simple component, connector and configuration won't do the perfect job, because still information related to provisioned Cloud services (e.g. SAL and Quality of Service) and mobile counterpart that need to be represented during the design process. To justify the need of a new ADL adapted for Mobile Cloud applications we propose MC-ADL dedicated only to Mobile Cloud Applications. More conception details of the proposed ADL will be discussed in the following subsections.

2.1 MCA's evolution and MC-Apps re-configuration mechanism

It is noteworthy that pure mobile Cloud services are based on Data Synchronization yet services at other Cloud layer (SaaS, PaaS, and IaaS) may increase mobile functional capabilities. We believe that the first step to exploit the Cloud in well-mannered way is to have a complete knowledge about Cloud and mobile in order to provide to

Consumers only the services that respond to their ever changing needs.

To fulfil this vision we propose Smart Cloud Gate Contextual Middleware (SCG). The general schema of SCG functioning is presented in Figure 2 as sequence diagram that shows in details how components of the middleware interact with each other.

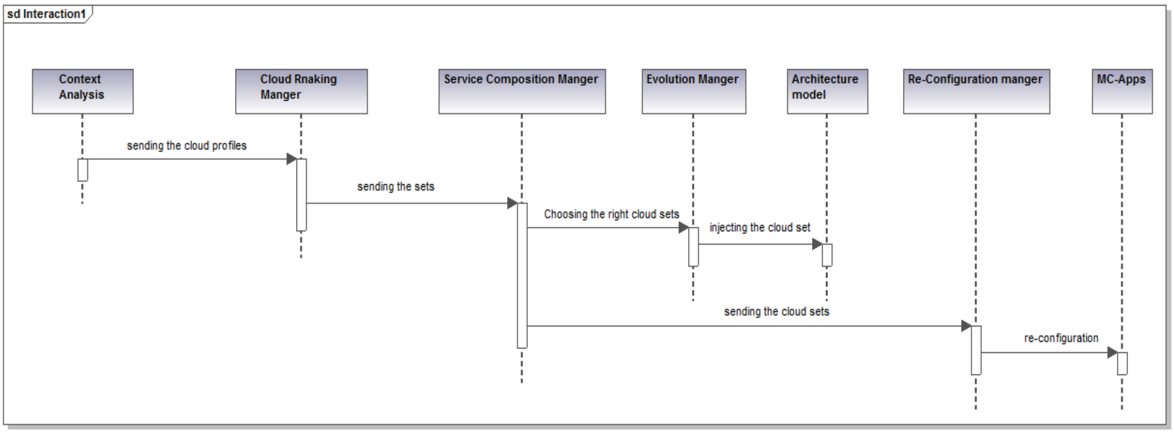

Figure 2: Sequence diagram showing the interaction between Smart Cloud Gate's components.

Smart Cloud Gate is a dynamic gateway to the Cloud that provides a personal perspective to mobile users based on Cloud service ranking. Service ranking is done as follow:

First the **context analysis** component: is responsible for profiling Cloud Crowd-Sourcing and mobile Crowd-Sourcing to extract information that will be considered later as ranking parameters. Information that should figure in each Cloud profile are: QoS, Popularity of the service among users, pricing metric and execution platform's information. The list is a subject of extension in the future. Concerning Mobile profile it should include, history about service usage, time and location, and other intrinsic characteristics.

After information concerning cloud's and mobile's profiles are stored in *Profile Data warehouse*, a selection algorithm that has been implemented in the **Cloud Ranking manager** component will be executed by *DB-server* to choose the appropriate set of service that compose the Cloud View to adequate user at the specific time and location and of course that satisfy their requirements.

Cloud **service composition manger** : is the component that manages the composition of the cloud view provided to the MC-App.

Cloud View is the final mobile Cloud architecture that has passed several evolutions according to Cloud service ranking, and the last evolution is done to meet the context's changes.

The second phase of the evolution is applied according to context's changes. At this phase we can understand the importance of the context awareness of mobile Cloud systems. As the context of the user change, this prompts invocation of different Cloud services based on the current context (time + location). Having this kind of context awareness, the invocation of Cloud services will be bound to user, since the request of service by the consumer will be accompanied by the user location , the device time, and other intrinsic characteristic, so the most adequate service will be selected according to that information. With Mobile context change the set of service change and the architecture evolves.

The last step consists of choosing the optimal evolution path, since each set of services generate a unique evolution path, and only one is needed to be followed. If the application is already implemented and in execution, than the evolution at the architecture level will provoke a re-configuration at the runtime level done by re-**configuration manager**

In the following section, we explain the evolution's mechanism with real example using EMF architecture models and ATL evolution rules.

3 CASE STUDY

Relying on Smart Mobile Cloud Reference Architecture proposed in previous work [16],[17] we have developed a framework for modeling mobile Cloud architecture.

After finalizing the EMF model, we need to generate a graphical editor in order to be able to edit the architecture.

A conception of example architecture is given in Figure 3. It should be noted that if no single Cloud service can satisfy the functionality required by the user, there should be a possibility to combine several existing services together in order to fulfill the consumer's request and answer vendor-lock in issues. Thus MC-ADL fosters service composition (e.g. Facebook app use different services located at different level like PaaS and SaaS and provisioned by different providers)

.

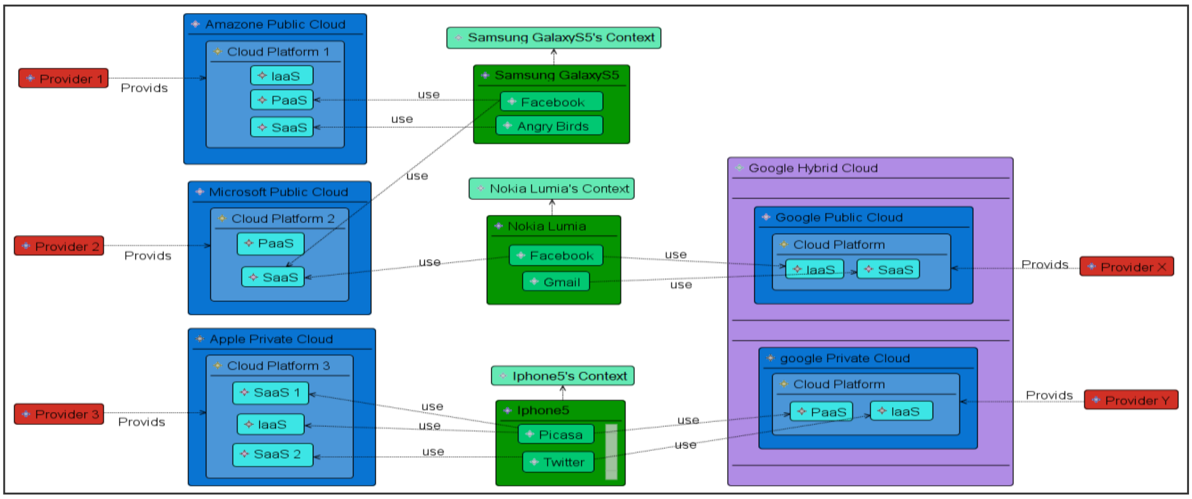

Figure 3: Mobile Cloud Architecture's meta-model and its prototype instance representing an architecture model.

Cloud services composition is a highly complex task, since the composition process can't be handled manually by humans. Generally the complexity of this mechanism is due the fact that the number of Cloud services available across the web are dramatically increasing, and it impossible to search manually Cloud services repository to choose from it, there does not exist a unique language that define and evaluate web service in identical means since the latter are developed by different Cloud vendors using different APIs and platforms, and most important challenge is that Cloud service are created and updated on the fly, so the composition system must detect the updating and the decision for using such services should be made based on up to date information. Since the evolution of MC-Apps architecture and their re-configuration at runtime level is based on Cloud ranking done by SCG, the same Cloud service information provided by SCG will serve as criteria to Cloud service composition in future work presenting Smart mobile Cloud Middleware that implements Cloud service composition for mobile cloud applications.

Studying both contexts (profiling the Cloud and the mobile) Cloud services have been ranked based on the user requirement, QoS, Cloud profile and mobile profile, and several Cloud services' sets has been categorized. Next by executing a selection algorithm the appropriate set is chosen, services composing the set will be either added to the architecture or serve as alternative for existing services. The evolution mechanism is implemented via ATL rules and it is illustrated in figure 4.

After analyzing and studying mobile's and Cloud's profiles, a new platform (i.e. Google Platform 2) has been added to Google Public Cloud, offering a new Cloud service that is more advantageous to Facebook application more than services provided in Google platform 1 (i.e. SaaS2). In very particular circumstances (Time+Location) the application uses well specified services in the architectureV2.1 we notice that the set of services used by SkyDrive App has changed along while context changes.

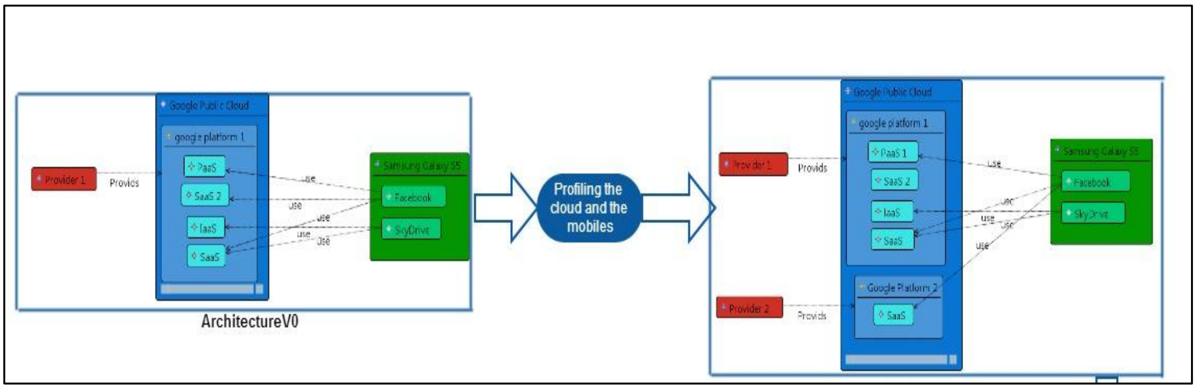

Figure 4: Mobile Cloud application's Architecture example of an Evolution.

4 RELATED WORK

In this section we will present, the work related of the subject of this paper that regards modeling MC-Apps' architecture, managing their contextual evolution and runtime contextual reconfiguration. First we will provide a classification of the state of art on the subject of cloud solutions. This classification is based on architectural representation, MC-Apps modeling, context awareness, contextual architecture evolution, and contextual at runtime level re-configuration (Cf. Table 1). Then we will argue whether or not these solution can handle mobile cloud application, and manage the complexity of administrating and developing this new generation of mobile applications. Finally we introduce our proposed vision to manage the complexity of modeling and administration the development of mobile cloud applications, that we realized via the framework that we propose .

To the best of our knowledge capturing and describing conceptual elements of Mobile Cloud Computing (MCC) has not been proposed in the literature yet.

Cavalcante in his PhD work [6] describe Cloud applications, via Cloud-ADL that is a customized architecture description language (ADL) that takes advantage of ACME [7] elements to model the

Framework	Architectural Representation	Architectural Evolution	Supporting MC-Apps Modeling	Runtime Reconfiguration	Context Awareness
Cloud ADL[6]	Customized ACME ADL	Foreseen Evolution	/	Via a cloud based middleware	Cloud context awareness
ModaCloud[18],[19]	Via CloudML	Based on runtime information	Possible	- Monitoring, - Data synchronization -Runtime adaptation	Cloud context awareness
CELAR [20]	Elastic cloud Application description (C-Eclipse)	Dynamic change in application topology	/	-Cost-Evaluation -Cross platform monitoring via JCatascopia	Cloud context awareness
CloudMF[21, 22]	Via ClouML	Via @Runtime Environment	Possible	Via @Runtime Environment	Cloud Awareness
	At Migration phase	Model2 model transformation	Possible migration	software updates or cloud provider changes	Cloud Awareness

ARTIST[23, 24]		during the migration phase	from native mobile apps to cloud based mobile apps		

Table 1: Cloud application framework classification

architecture of Cloud application. Cloud-ADL provides constructors that enable to specify programmed (i.e. foreseen at the architectural level) dynamic reconfiguration actions to be taken at the runtime level. But the customized ADL do not support modeling MC-Apps, cause in have more feature that can't be presented by only components and connectors

Ardagna and *al.* in [18] [19] argue that Model-Driven Development can be helpful in this context as it would allow developers to design software systems in a Cloud-agnostic way and to be supported by model transformation techniques into the process of instantiating the system into specific, possibly, multiple Clouds. their proposed framework, provides the following main components: CloudML a cloud application modeling language, Decision Support System that guide the selection of cloud services, Runtime Layer that supports monitoring and self-adaptation of the cloud application, and finally Model-Driven-Development Layers.

CELAR [20] provides two open sources tools for managing and monitoring elastic cloud applications. first: c-eclipse that describe elastic cloud applications by adopting TOSKA [25]. Jcatascopia is a completely automated, independent and interoperable cloud monitoring system that takes in consideration the changes in the topology of the application due elasticity actions.

Rather than focusing only on the design and adaptation of the application like Dynamic Adaptive System (DAS), Cloud computing focus also on the underlying infrastructure of these systems. Like traditional DAS , cloud based DAS need to adopt a model based approach to facilitate the reasoning on this later at the design time and their adaptation at the runtime. Based on this vision Ferry et *al.* propose in [21, 26] CloudMF a model based framework hat consist of a domain specific language tools that model the provisioning and deployment of multi cloud systems, and a model @runtime environment to enact the provisioning deployment and adaptation of these systems.

Offering a legacy systems as Services via the cloud is a challenging task. The modernization and adaptation of these kind of applications is difficult not only from a technical perspective but also in business level that requires the adaptation of the business process. Answering this challenge, Menychtas and *al.* propose in [23] Model Driven modernization and migration approach, that is part of the a ARTIST Project [24] that aims to provides a model based methodology to describe application as well as cloud infrastructure, and platform.

4.1 Discussion

we conclude from the previous Classification of related works in term of framework and tools the following

They only target Traditional Software and legacy system , this interest is justified but as stated by Medvidovic et *al.* in [3] Mobility also important and mobile applications are the next generation in software and cloud era. and that explain how Mobile Context in not mentioned since these framework do not deal with mobile applications.

from our point of view and based on that classification we can state that some Project like ARTIST[24],MODACLOUD[19], CLOUDMF[26] offer a possibility for adapting the development and management of mobile cloud applications based on their vision and approaches.

5 CONCLUSION

Contextual Information extracted from both cloud and mobile will foster a better exploitation of the Mobile Cloud Computing paradigm. Starting from this point we have proposed in this proceeding research a framework based the reference architecture, Smart Mobile Cloud Architecture that offers a generic solution which enable managing both Cloud's and mobile's contexts simultaneously.

The mobile Cloud application development as we propose is done as follow: first describing MCC architecture using the proposed ADL MC-ADL. Then instantiating the architecture will result, executable mobile Cloud application. While monitoring and analyzing both the Cloud and the mobile devices, the Smart Cloud gate middleware (SCG) apply a selection algorithm in order to rank the Cloud services that enter into the composition of the set that will be used later during the evolution mechanism. Finally the optimal evolution path is chosen and applied.

REFERENCES

1. Walsh, M., *Gartner: Mobile to outpace desktop web by 2013.* Online Media Daily, 2010.
2. Beccue, M. and D. Shey, *Mobile Cloud Computing.* ABI Research, 2009.
3. Medvidovic, N. and G. Edwards, *Software architecture and mobility: A roadmap.* Journal of Systems and Software, 2010. **83**(6): p. 885-898.
4. Jani, M., J. Kela, and E.-J. Malm, *Managing context information in mobile devices.* IEEE pervasive computing, 2003. **2**(3): p. 42-51.
5. Schilit, B., N. Adams, and R. Want. *Context-aware computing applications*. in *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*. 1994. IEEE.
6. Cavalcante, E., *Architecture-Driven Engineering of Cloud-Based Applications.* 2013.
7. Garlan, D., R. Monroe, and D. Wile. *Acme: an architecture description interchange language*. in *CASCON First Decade High Impact Papers*. 2010. IBM Corp.
8. Team, M., *MODA CLouds Underpining the leap to DevOps Movementon Cloud Environement.* july 2013.
9. Dinh, H.T., et al., *A survey of mobile cloud computing: architecture, applications, and approaches.* Wireless communications and mobile computing, 2011.
10. Fernando, N., S.W. Loke, and W. Rahayu, *Mobile cloud computing: A survey.* Future Generation Computer Systems, 2013. **29**(1): p. 84-106.
11. EMF, G., *http://fr.wikipedia.org/wiki/Graphical_Modeling_Framework.* 2009.
12. Group, A., *Atlas transformation language.* URL http://www. eclipse. org/m2m/atl, 2007.
13. Amirat, A. and M. Oussalah. *Towards an UML Profile for the Description of Software Architecture*. in *Proceeding if International Conference on Applied Informatics (ICAI'09)*. 2009.
14. Medvidovic, N. and R.N. Taylor, *A classification and comparison framework for software architecture description languages.* Software Engineering, IEEE Transactions on, 2000. **26**(1): p. 70-93.
15. Garlan, D. *Software architecture: a roadmap*. in *Proceedings of the Conference on the Future of Software Engineering*. 2000. ACM.
16. Gherari, M., A. Amirat, and M. Oussalah, *Towards Smart Cloud Gate Middleware: An approach based on Profiling Technique.* Cal 2014, France paris, 2014: p. France.
17. Gherari, M., et al., *Towards a Smart Cloud Gate for Smart Devices.* in proceeding IEEE Cist'14 morocco, tetouan, 2014.
18. Ardagna, D., et al. *Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds*. in *Modeling in Software Engineering (MISE), 2012 ICSE Workshop on*. 2012. IEEE.
19. ModaClouds, *MODACLOUDS EU Project [Online] available: http://www.modaclouds.eu/.*
20. Project, C., *CELAR EU project [Online], available at; http://www.celarcloud.eu*
21. Brandtzæg, E., P. Mohagheghi, and S. Mosser. *Towards a domain-specific language to deploy applications in the clouds*. in *CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDs, and Virtualization*. 2012.
22. Ferry, N., et al. *Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems*. in *CLOUD 2013: IEEE 6th International Conference on Cloud Computing*. 2013.
23. Menychtas, A., et al. *ARTIST Methodology and Framework: A novel approach for the migration of legacy software on the Cloud*. in *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2013 15th International Symposium on*. 2013. IEEE.
24. PROJECT, A., *Artist Project eu [Online] available at: http://www.ARTIST project.eu.*
25. OASIS, *OASIS [On Line] available at: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca.*
26. Ferry, N., et al. *Managing multi-cloud systems with CloudMF*. in *Proceedings of the Second Nordic Symposium on Cloud Computing & Internet Technologies*. 2013. ACM.