

Mobile Cloud Middleware: A New Service for Mobile Users

K. Akherfi, H. Harroud

Abstract—Cloud computing (CC) and mobile cloud computing (MCC) have advanced rapidly the last few years. Today, MCC undergoes fast improvement and progress in terms of hardware (memory, embedded sensors, power consumption, touch screen, etc.) software (more and more sophisticated mobile applications) and transmission (higher data transmission rates achieved with different technologies such as 3Gs). This paper presents a review on the concept of CC and MCC. Then, it discusses what has been done regarding middleware in cloud and mobile cloud computing. Later, it shows the architecture of our proposed middleware along with its functionalities which will be provided to mobile clients in order to overcome the well known problems (such as low battery power, slow CPU speed and little memory...).

Keywords—Context-aware, cloud computing, middleware, mobile cloud.

I. INTRODUCTION

OVER the last few years and during the global economic crisis, businesses were looking for new ways to reduce costs while maximizing value which gave birth to cloud computing. The concept of cloud computing (CC) is composed of two terms: "Cloud" refers to Internet and it is a metaphor for the Internet network diagram (it is derived from network diagram), while "Computing" depicts the use of computer technology for task execution. As 'Cloud Computing', it means "When the task execution requiring computer is based on internet" without the need of physically installing software or implementing complex infrastructure [1]. CC provides highly scalable computing resources as a service through the Internet on a "pay-as-you-go" basis. Besides, it allows the virtualization of resources and facilitates the scalability over internet [2].

CC can be seen from two different angles: one is the applications sent as services through the Internet and the other one is the hardware that provides those services [3], [4]. Three major paradigms or models of cloud services are available in the cloud area:

- Infrastructure as a Service (IaaS): focuses on exposing the hardware resources to users as services and provides a complete control over the operating system. It is like having or getting access to a physical computer connected to the Internet. For instance, Amazon EC2 is a successful IaaS in the cloud market; it is the most known public cloud and the most widely used.

A. Akherfi and S. H. Harroud are with the School of Science and Engineering, Al Akhawayn University in Ifrane, 53000, Ifrane, Morocco (phone: 212-5358-62114; fax: 212-5358-62030; e-mail: k.akherfi@ aui.ma, h.harroud@ aui.ma).

- Platform as a Service (PaaS): Google Application Engine (Google App Engine) is an example of PaaS. It provides elastic platform for Java and Python applications with some limitations. Thanks to this service, a customer can get access to a specific application development platform hosted on infrastructure cloud. In other words, it provides the client with computational resources as high level application platform without bothering him/her with maintenance and system setup meaning that the client or the developer will focus on creating the wanted functionalities [4].
- Software as a Service (SaaS): represents software capabilities and functions as services. It provides the customer with software suites that are generally customized. Those software capabilities are offered as WSs by different service providers such as Google, Amazon, and Yahoo [4].

Fig. 1 represents the three paradigms or layers mentioned earlier.

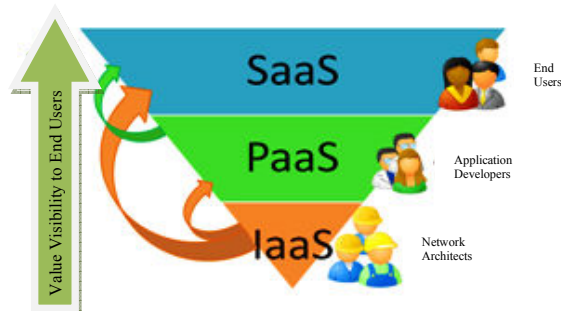


Fig. 1 Cloud Computing Layers [19]

In general we distinguish four types of cloud deployment models:

- Private Cloud: It is cloud infrastructure within the physical boundaries of an organization. Normally, private clouds run s cloud middleware for provisioning services on an internal network
- Public Cloud: It is a cloud that offers utility computing on the Internet as a service.
- Hybrid Cloud: Hybrid cloud is a grouping of private and public clouds. Hybrid clouds often are built to scale out from a private to a public cloud.
- Community Cloud: A community cloud which supports a community, where several organizations have similar needs and seek to share infrastructure in order to realize some of the benefits of cloud computing [5].

Mobile Cloud Computing (MCC) is defined as follows: "MCC is defined as cloud computing extended by mobility, and a new ad-hoc infrastructure based on mobile devices. It provides mobile users with data storage and processing services on a cloud computing platform." [6].

Mobile Cloud Computing is providing many benefits for cloud computing and network operators, such as reduced dependence on hardware and software equipment, and increased reach. Mobile cloud computing has many advantages among the few listed below:

- Possibility of Sharing information and applications without the need of costly hardware and software [7];
- Enhanced features and functionalities of mobile devices through new cloud applications [8];
- Ease of access and development since the access point to mobile cloud computing is through a browser [7];
- Extending battery lifetime for mobile devices [9]
- Improved data storage capacity and processing power [9]

Fig. 2 shows the general architecture of MCC. In this figure, mobile devices are connected to the mobile networks using base stations such as access point, or satellite. The role of these base stations is to establish and control the connections between the networks and mobile devices. The central processors receive the information and requests sent by the mobile users, and then send them to servers providing mobile network services. In this architecture, the mobile network operators provide services to mobile clients as AAA (stands for Authentication, Authorization and accounting) based in the home agent (HA) and subscribers' data stored in databases. In the cloud, cloud controllers execute the clients' requests to provide mobile users with the corresponding cloud services [10].

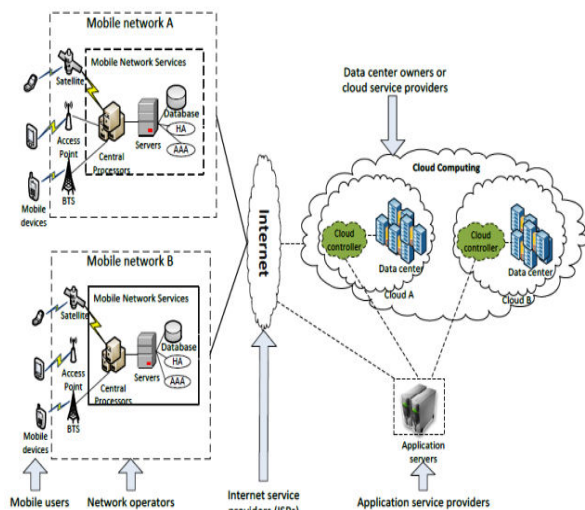


Fig. 2 Mobile Cloud Computing Architecture [10]

The importance of MCC can be seen among the few points listed below:

- MCC allows users access to cloud services.

- Enhanced features and functionalities of mobile devices through new cloud applications [11];
- Improved data storage capacity and processing power [11]

This paper introduces a mobile cloud middleware that permits the mobile clients to cope with the aforementioned limitations by providing set of features such as the adaptation and optimization of the cloud services results. In addition to the monitoring of resources and context of the device, and dynamically adapts the requests accordingly.

The paper is organized as follows. Section II highlights the ongoing research related to middleware and distributed systems. Section III shows the architecture of our proposed middleware along with the description of the interaction between its modules and components. Section IV describes a prototype scenario that illustrates how the middleware has been used to tackle the adaptation issue. Section V presents an example scenario to illustrate our middleware use. Section VI summarizes and points to future work.

II. RELATED WORK

The location of a device that always changes, connectivity fluctuation, service and host discovery are now the main characteristics of mobile systems and challenges at the same time. Moreover, mobile users are now demanding more and more services which are suitable to their current context.

In order to overcome the previous challenges, many research efforts have focused on designing new platforms that are able to support the requirements imposed by mobility.

Middleware is used to connect several machines in order to allow a smooth interaction between them and it is often used in applications or distributed systems, which can be defined as "a collection of autonomous computers that are connected through a network which enables computers to share the resources of the system, so that users perceive the system as a single, integrated computing facility" [12]. Applications and distributed systems such as mobile clients can be available thanks to middleware, independently from network services and transparently located over the network [13].

According to some research, middleware technology has shown that is a way of suitable utility of personal interaction with the cloud provider platforms. In order to bring the cloud to the mobile, many approaches have been proposed. For instance, Chun et al. introduce CloudClone [14] as a framework that allows executing a mobile platform on a virtual machine. This latter, handles the provisioning of cloud services once a real handset is synchronized to it. Similarly, Cuervo et al. propose MUAI [15] a framework that enables offloading mobile code to the cloud to save energy for mobile applications. Here, the idea consists of offloading code components to the cloud or nearby servers when processing is needed. The previous approaches emphasis on using a single cloud and they do not permit mobile application to benefit from multi-cloud operations. Another approach proposed by Wu et al. [16] presents an architecture that enables online, offline and mixed mode of operation for mobile applications with unified access to the business logic. The architecture

framework enables easy building and adapting mobile applications that run in a selected mode depending on the scenario and user requirements. However, this approach is limited, in sense that dynamically changing the operation mode is not possible which requires rebuilding the application.

Because the worldwide utilization of wireless, and Smartphone device, mobile users are now more and more requiring services which are suitable for their current context as long as they are moving. Furthermore, applications are becoming more complicated and context-aware to meet the users' requirements. In the literature, there are several definitions of the notion of context.

According to Dey [17], "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." Based on the previous definition, the context information is extremely broad. For instance, time, location, and mobile user activity are the most widely used context indicators by applications.

Reference [18] proposes a framework that provides context-aware mobile services based on an algorithm.

In a given context, the algorithm determines a kind of gaps that occurred. The authors defined a gap as a result of context changes. In the next step, the algorithm determines the cause of predefined gaps. Then, the algorithm will choose an appropriate adapter for the mobile user for the identified gap. Nevertheless, in the other contexts, the link cannot be defined when mobile users change to another context. Thus, this framework works with the predefined gaps and this may lack flexibility in practical usage.

III. MIDDLEWARE ARCHITECTURE

In the present attempt of this work, architecture is proposed. It is based on the middleware technology which is used for supporting the mobile clients in accessing the services of cloud platform. The middleware should provide an infrastructure for a transparent execution of elastic applications and a flexibility of adjusting the requirements for mobile clients to Cloud Services.

The middleware is software that provides a runtime environment the necessary support and organization of multiple applications. It binds together the applications, and the operating systems. Though, the main important functionality of the middleware is to achieve adaptive and efficient utilization of heterogeneous resources and services as well. Our system provides a unique interface that responds according to the cloud services requested.

The used architecture breaks down the system into independent elements where each one of them does specific functionalities in order to contribute to the achievement of our middleware's objectives.

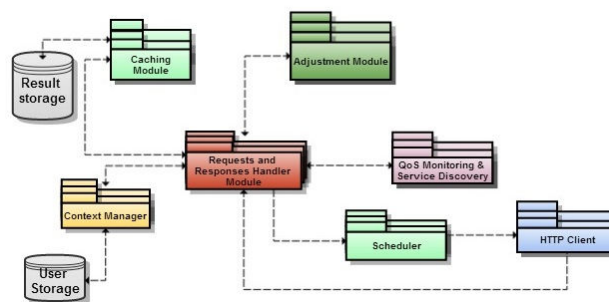


Fig. 3 Middleware General Architecture

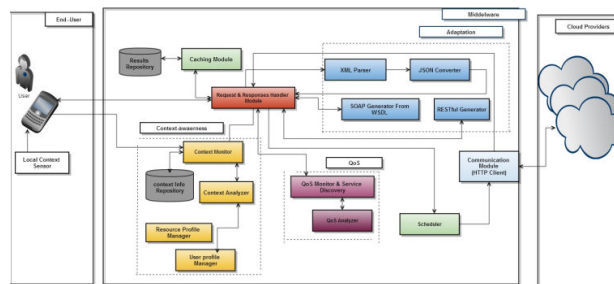


Fig. 4 Middleware Detailed Architecture

Fig. 3 shows the generic architecture of our system. It is composed of seven modules in addition to two different repositories. The objective of the architecture of this middleware is to allow the mobile clients to connect to cloud services. The system will execute all the essential adjustments to the mobile clients.

Fig. 4 shows the instantiated architecture of our middleware which is derived from the generic one. The architecture modules along with their interactions are described below. The middleware is composed of set of components and adapters, which manage the connection and communication between different cloud providers and the mobile clients.

- **Requests & Responses Handler Module:** As a manager, this module handles the different requests issued by the clients as well as the responses received from the cloud provider via the communication module.
- **User Profile Manager:** It monitors the user profile. This module allows the user to create his profile and register in our middleware as well as modifying his information.
- **Mobile Profile Manager:** It monitors the mobile profile.
- **Caching Module:** this module stores the service results into the result repository. It is responsible for reading and writing data from and into the Result database.
- **Context Monitor:** It receives and interprets context information. It stores context information in the context information repository. Moreover, it accepts or discards mobile client requests based on the resources available (power aware, resource aware...).
- **Context Analyzer:** it works by analyzing the received context information and sending the results back to the context monitor.

- Scheduler: It schedules the accepted requests and makes a plan (Schedule queue).
- QoS Analyzer: It analyzes the cloud services and stores the set of providers suitable for the client requests.
- QoS monitoring & Service discovery: It discovers the cloud provider via the Universal Description Discovery and Integration (UDDI). This module interacts with the others via Remote Procedure Call (RPC), and exchanges service registration information. It allows the middleware to communicate with different services provided by different cloud providers using different standards for the specification of the service functionalities.
- XML Parser: The parser extracts only the needed and required data that should be processed by to "JSON Converter" (JavaScript Object Notation). And this is the first step to get optimized results.
- JSON Converter: The second step to achieve the optimized results is performed by this module. It converts the received xml data to JSON format before sending this new format to the clients.
- SOAP (Simple Object Access Protocol) Generator from WSDL: this module will generate SOAP WS (Web Service) from WSDL provided by requested service. And this in case the requested service is SOAP WS.
- RESTful Generator: This module transforms the SOAP WS into RESTful WS in case the result service is SOAP WS. This module is needed because mobile clients use a RESTful WS interface.
- Context Information Repository: a database that stores the end users' profiles. It includes all the information about users' context information along with their preferences.
- Result Repository: this repository is dedicated for saving requests along with their responses.

IV. WALK THROUGH THE ARCHITECTURE

The objective of this section is to demonstrate how our system architecture will perform in a specific situation by providing some scenarios that will illustrate the interaction between the different modules that composed our middleware. The used web service in these scenarios is WeatherBug. It provides the client with weather cameras, forecasts, and weather alerts.

Scenario 1: The WeatherBug WS Request Is Not stored in our system.

In this scenario the client wants to get a weather forecast for a specific location. WeatherBug, the provider, gives access to both REST and SOAP web services.

- The request is received by the Requests & Responses Handler Module. This module checks if the wanted request is already saved in our system by sending the request to the Caching Module.
- If not, the Requests & Responses Handler Module will forward the request to the QoS monitoring & Service discovery in order to find the cloud provider, then it will send the request to the QoS Analyzer.

- The QoS Analyzer will analyze the cloud services and send the result to the Requests & Responses Handler Module.
- Since this provider uses both SOAP and REST, there is no need to send the request to the SOAP Generator. So, the request is sent immediately to the Scheduler which will put the request in the queue and send it to the communication module.
- The communication module will send the request to the WS provider in the cloud.
- Once the response is received by the middleware via the HTTP Client module, it is sent to the Requests & Responses Handler Module in order to be sent to the XML Parser.
- Once the XML Parser extracts the needed data, it forwards it to the Requests & Responses Handler Module. Then, this latter will forward the response to the JSON Converter.
- JSON Generator forms a new format of the response and sends it back to the Requests & Responses Handler Module.
- The optimized response is then sent to the Caching Module to be saved in the system repository.
- The Requests & Responses Handler Module sends the optimized results to the client.

Scenario 2: WeatherBug WS Request Is Already stored in the system

- In case the response of the requested WS is saved within our proxy. The following steps will be performed.
- The Requests & Responses Handler Module receives the request, and sends it to the Caching module.
- A search is performed at the level of the Caching module to look for the appropriate response in the data repository and retrieve it.
- The response is sent back to the Requests & Responses Handler Module.
- Finally, the response is delivered to the mobile client via the Requests & Responses Handler Module.

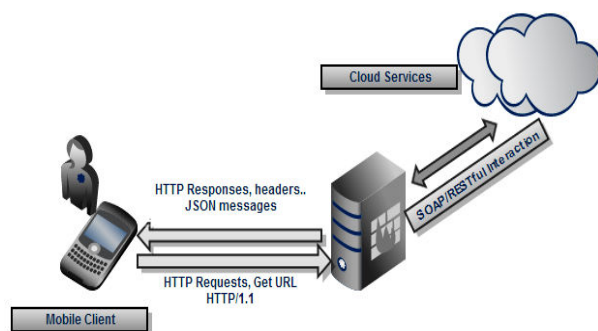


Fig. 5 Middleware and Web Services Interaction

Fig. 5 illustrates how the mobile client requests and consumes a WS from a service cloud provider. First, it starts with an HTTP GET request. Then, the middleware extracts the resource identifier from the URL path through the HTTP GET

request. Next, the system performs the appropriate tasks according to the current situation.

V. ILLUSTRATIVE SCENARIO

Let us consider the following scenario that demonstrates the adaptation of the user context information. Mobile applications become sophisticated and mobile users use applications binding on cloud services such as audio or video streaming applications.

- The user starts the video streaming application, and requests a video from the cloud service provider.
- The device starts the service discovery using the 3G or other network bandwidth.
- In case of the bandwidth changes because of the mobility of the users:
 - The user will not get the expected QoS.
 - The binding cost remains the same since it is based on the primarily requested bandwidth.
 - User satisfaction won't be as expected, meaning it will be low.

Any changes at the level of the context parameter, for instance the bandwidth will push our middleware to search and bind to a service that will provide the suitable QoS level for the current context, and satisfy the user needs. And this will lead to the appropriate cost and charges which are proportional to the consumed bandwidth and the volume of the data transferred.

VI. CONCLUSION

As widely known, mobile devices are personal, portable, small and with different features. However, these devices have some restrictions, such as less process power and small bandwidth that do not allow them to satisfy from the wanted services in addition to the reason that most of services are designed for stationary clients. The growing success of mobile computing devices and networking technologies have pushed researchers for the investigation of new middleware that deal with mobile cloud computing challenges.

In this paper, we have highlighted the work that has been done in the field of distributed systems, and we have focused on presenting our middleware, its instantiated architecture along with its functionalities that are mainly: the adaptation of the services in order to fit the mobile devices, the context awareness, the mobility, and the interoperability.

The middleware behavior changes in the context of execution, therefore to achieve the best quality of service and optimal use of resources. We believe that research in the field of mobile computing middleware still faces many challenges such as security issues. Mobile devices are exposed to security attacks and security for mobile applications should be taken into consideration in the design of a mobile middleware system.

REFERENCES

- [1] Sharma, P. (2008). It's Probable that You've Misunderstood "Cloud Computing" Till Now. Retrieved from <http://www.techpluto.com/cloud-computing-meaning/>
- [2] Lovell, R. (n.d.). Introduction to Cloud Computing. Retrieved from <http://www.thinkgrid.com/docs/computing-whitepaper.pdf>
- [3] Armbrust, M., Fox, A., Grith, R., Joseph, D., Katz, H., Konwinski, A., Lee, G., Patterson, A., Rabkin, A., Stoica, I., & Zaharia, M. (2009). Above the clouds: A Berkeley view of cloud computing. Retrieved from <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>
- [4] Ranabahu, A., & Maximilien, E.M. (n.d.). A Best Practice Model for Cloud Middleware Systems. Retrieved from <http://knoesis.wright.edu/library/download/cloud-oopsla-09.pdf>
- [5] Cloud Deployment Models. Retrieved from: <http://www.conres.com/cloud-computing-deployment-models>
- [6] http://www.en.zte.com.cn/endata/magazine/ztecommunications/2011Year/01/articles/201103/t20110318_224532.html
- [7] M. Tantow, "Cloud Computing and Smartphones," article in Cloud Times, 01 March 2011. Retrieved from <http://cloudtimes.org/cloud-computing-and-smartphones/>
- [8] P. K. McKinley, F. A. Samimi, J. K. Shapiro, and C. Tang, "Service Clouds: A Distributed Infrastructure for Constructing Autonomic Communication Services," in Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, pp. 341, December 2006
- [9] Emmirech, W. (1997). Distributed Systems Principles. Retrieved from <http://www.cs.ucl.ac.uk/staff/uca/wxe/lectures/ds98-99/dsee3.pdf>
- [10] C. Lee, D. Niyato, H. T. Dinh, P. Wang. "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches"; Retrieved from http://www.eecis.udel.edu/~cshen/367/papers/survey_MCC.pdf
- [11] P. K. McKinley, F. A. Samimi, J. K. Shapiro, and C. Tang, "Service Clouds: A Distributed Infrastructure for Constructing Autonomic Communication Services," in Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, pp. 341, December 2006
- [12] Emmirech, W. (1997). Distributed Systems Principles. Retrieved from <http://www.cs.ucl.ac.uk/staff/uca/wxe/lectures/ds98-99/dsee3.pdf>
- [13] Defining Technology (2008). What is Middleware? Retrieved from <http://www.middleware.org/whatis.html>
- [14] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic Execution between Mobile Device and Cloud," in 6th Int. Conf. on Computer Systems. ACM, 2011, pp. 301-314.
- [15] E. Cuevo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making Smartphones Last Longer with Code Offload", in 8th int. Conf on Mobile Systems, Applications, and Services. ACM, 2010, pp. 49-62.
- [16] H. Wu, L. Hamdi, and N. Mahe, "TANGO: A Flexible Mobility-Enabled Architecture for Online and Offline Mobile Enterprise Applications," in Proceedings of 11th International Conference on Mobile Data Management (MDM 2010). Kanas City, MO, USA: IEEE, 2010, pp. 230-238.
- [17] A.K. Dey, —Understanding and Using Context, Journal of Pervasive and Ubiquitous Computing, vol. 5(1), pp. 4-7, 2001.
- [18] H. H. La and S. D. Kim, "A Conceptual Framework for Provisioning Context-aware Mobile Cloud Services," in Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD), pp. 466, August 2010.
- [19] SaaS, PaaS, IaaS, Cloud: Definitions [Online] Retrieved from: <http://icp.ge.ch/sem/cms-spip/spip.php?article962>

Khadija Akherfi is currently a PhD student at the Fakultät für Informatik, faculty of information, of Technical University of Munich. She received her master degree in Master of Science in Software Engineering from Al Akhawayn University in Ifrane (AUI), Morocco in 2012.

Hamid Harroud is an associate professor in Computer Science at the School of Science and Engineering (SSE), Al Akhawayn University in Ifrane (AUI), Morocco. His research interests include context-aware systems, mobile computing and policy-based management. H. Harroud received his MSc degree from the Mohammadia School of Engineering (EMI), Morocco in Electrical Engineering, and has a Ph.D. in computer science from the University of Ottawa, Canada. Contact him at H.Harroud@au.ma.