Mobile Service in Cloud of Things Using Model Driven Architecture

D Murugeswari
Department of Computer Science, Thassim Beevi Abdul Kader College for Women, Kilakarai
Email: murugeswari2412@gmail.com
Dr A R Nadira Banu Kamal
Department of Computer Science, Thassim Beevi Abdul Kader College for Women, Kilakarai
Email: nadirakamal@gamil.com

----------**ABSTRACT**----------
Cloud of Things (CoT) is a combination of Internet of Things (IoT) and cloud computing for intelligent and smart application particularly in mobile devices. CoT will provide more attention to intelligent and mobile applications in a business insight. Model Driven Architecture and Cloud computing are the most important paradigms in software service engineering. Model Driven Architecture (MDA) approach for development and maintenance becomes an obvious choice for provide software solutions that are robust, flexible and energetic for developing applications for mobile and it is used to develop Software as a Service (SaaS) therefore, to make possible mobile applications development by relieving the developers from technical details. Model Driven Architecture is a software development process that models the system at different levels of abstraction and uses transformation tools for model-model or model-code transformations. This paper aims to study the various mobile services that have been developed in cloud computing applications with a focus on using Model Driven architecture (MDA).

Keywords – **Cloud of Things, Computation Independent Model, Model-Driven Development, Platform Independent Model, Software as a Service**

1. Introduction

The number of devices and mobile applications has increased rapidly because of the rapid development of Internet of Things (IoT) applications. Mobile Devices are the Gateway to the Internet of Things. The Internet of Things (IoT) is all about devices: how we relate with them and the data they produce. One key piece of the IoT that is often overlooked is our mobile device. It is supposed that the number of devices has already topped the number of people on the Earth since 2011. And the number of devices is expected to grow to 24 billion by 2020. Thus the requirement of a strong and flexible environment for IoT application support has become a critical issue. Auspiciously, Cloud Computing provides a strong basis for resource sharing in a flexible way. IoT and cloud computing working in integration makes a new paradigm named Cloud of Things (CoT) [1]. IoT objects particularly mobile devices will be connected via cloud platforms for different business applications. Merging Cloud platform and IoT application, CoT will take a more and more significant role in different industries and research areas [2]. In CoT, IoT objects are extended from sensors to every front-end thing on the Internet. And distributed objects are connected as a whole system for complex and intelligent applications, such as smart house, smart factory, and smart city [3].Other than data and resources in a single view of cloud or current IoT applications, CoT will pay more attention to intelligent and mobile applications in a business insight. The issues are related to combination of IoT with cloud computing and involve smart gateway to perform the complex tasks and complete solution for a certain application requirement, other than the simple event acquiring and trans-forming as traditional sensors do. The construction of cloud-enabled mobile applications can take advantage of existing resources.

2. The Applications of the Integration

In this section we define an extensive set of applications that are made possible or considerably enhanced CloudIoT paradigm. The instruction is shown in Fig. 1.

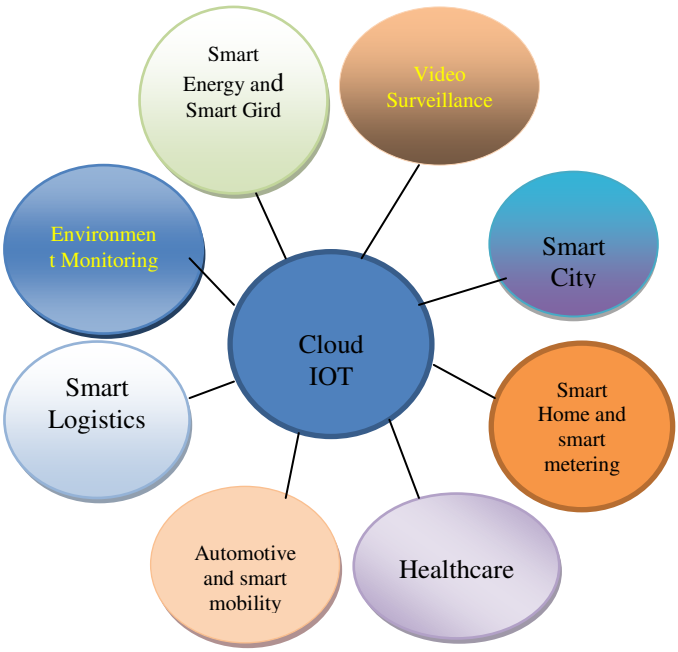


Fig. 1The Application Scenarios

IoT and multimedia technologies have made their entrance in the healthcare field ambient-assisted living and telemedicine [9]. The implementation of Cloud in this scenario leads to the abstraction of technical details, eradicating the need for expertise in, or control over, the technology infrastructure. And also makes the implementation of security multimedia health services, to defeat the problem, on the device running a large number of multimedia & security algorithms with partial computing power and small batteries.

In this area, the common problems with the management, technology, safety, and legal investigation: interoperability, system security, the streaming media quality of service (QoS) and dynamically increased storage is usually considered a barrier. Improvement of medical treatment is capture. It is comfort to generate national health records, in order to provide decision-making basis for regional disease through the comparison and analysis of medical information. Remote monitoring cloud platform of medical information containing body sensors, sensor networks, communication module, family gateway, medical information analysis and etc., Medical sensor network to gather information is different. Personal health record data is huge and growing rapidly. Therefore a huge amount of data need to automatic classification, analysis and processing as information service provider. For example, security service center to provide security services; Centers for disease control emergency organization disease detection and control status early.

These services according to the emergency first aid information extraction; the hospital provides remote diagnosis in medical information; Rehabilitation center provide remote rehabilitation guidance according to the variety of recovery information; Health service center to provide health instruction, through the extraction of the user's health information. So, the network resource sharing means should be required and they need an effective method to extract information. The following sections illustrate our proposed work.

3. Model Driven Architecture

Model Driven Architecture (MDA) supported by OMG (Object Management Group), is offered as the most interesting and the most common alternative. MDA has the similar standard as MDE (Model Driven Engineering), but it provides its own characteristics described in three levels of abstraction, defines some requirements to be respected, and also recommends the use of some standards.

Fig.2 shows MDA software Development using three tier applications. In this diagram, the first level of MDA is the CIM (Computation Independent Model) presented as models utilized by business managers and business analysts to illustrate the business process. The second level is the PIM (Platform Independent Model) which permits defining the models utilized by analysts and the software designers to understand an independent analysis and an impression of the developed software. The third level is the PSM (Platform Specific Model) is models of code utilized by software developers. These models are supposed to include all the information needed to operating an execution platform and used by software developers. The code is not a model of MDA, but it is the final result of the MDA process. In system configuration, MDA (Model Driven Architecture), EUD (End User Development) and SaaS are three manners for developing mobile applications with existing resources.

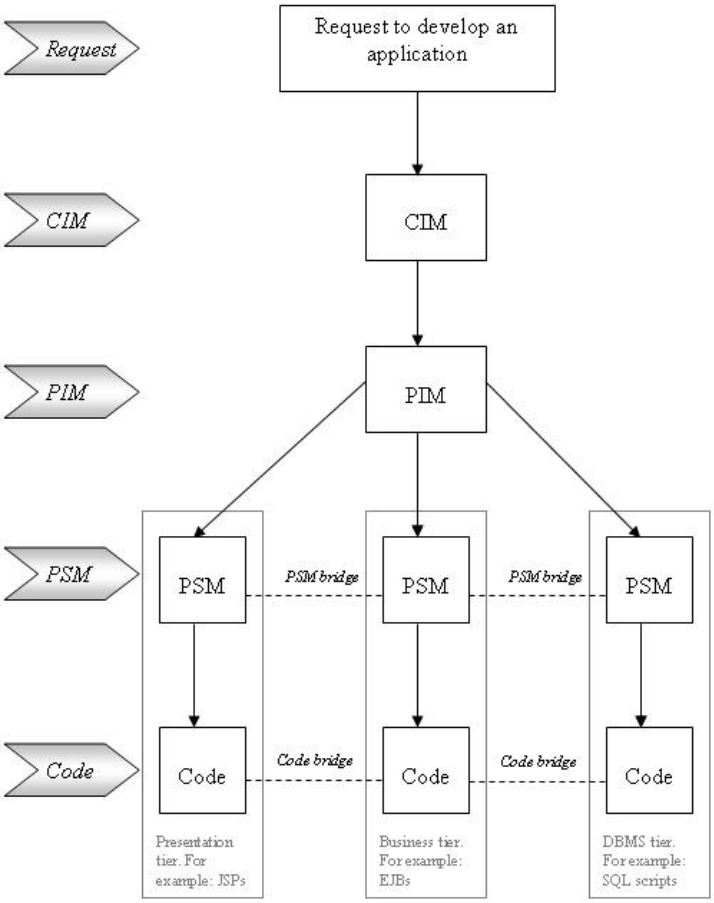


Fig.2 MDA Software Development

MDA [4] promote software development by means of providing several different abstraction levels of the software development process. It permits developers to pay more concentration to the business logic instead of technical details.

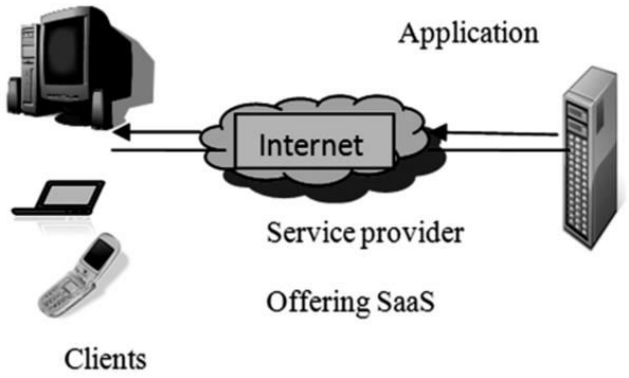
In recent times, the idea of ontology has been initiated into MDA software development for semantic disposing. A major fault of existing MDA approaches is their lack of mechanisms to produce model transformation between models such as CIM (Computation Independent Model), PIM (Platform Independent Model) and PSM (Platform Specific Model) [5]. CIMs belong to the business area while PIMs and PSMs are some type of system components. The business model is moderately different from executable system components, and logical structures

of business processes are separated from the complete implementation of system functions. The approach target on software flexibility and rapid integration that most web-based software are failed to provide. On the other hand, most existing approaches focus on function or data composition, but insufficiently support EUD for infrequent, situational, and ad-hoc integrations and cooperation.

4. Cloud SaaS Based on MDA

4.1. Software as a Service

SaaS[6]services realize the scalability, agility and reliability of the cloud platforms. Service technologies provide flexible design principles used in the phases of system development and integration. Among them, Huang W. [7] proposed a service model which supports multi-tenant in cloud computing. In order to reuse existing resources in a new combination way, Ketter [8] introduced an agile service composition by means of discovering and sharing high-level components. On the highest level of abstraction, it facilitates the inserting and removing of pre-built components as well as accessible services and other resources.



Aim to develop or move around application with non unified and incompatible services into Cloud platforms, a unified semantic representation of services so as to perform automatic service discovery. However, only exist services but not composed services are involved, which make the application of low flexibility.

In Software as a Service (SaaS) model, the client would depend on the service provider for appropriate security measures. The provider makes sure that the multiple users don't get to see each other's data. Therefore it becomes important to the user to ensure that right security measures are in place and also difficult to obtain an assurance that the application will be available. When using SaaS model, the cloud customer will, by definition, be alternate a new software applications for existing ones. So, the focus is not upon portability of applications, but on preserving or enhancing the security functionality afforded by the legacy application and achieving a successful data migration. The SaaS software vendor may host the application on his own private server or set up it on a cloud computing infrastructure service provided by a third-party provider (e.g. Amazon, Google, etc.). The use of cloud computing joined with the 'pay-as-you-go' approach facilitates the application service provider decrease the investment in infrastructure services and able it to focus on providing better services to the customers. Enterprises today view data and business transactions as strategic and safeguard them with access control and compliance policies. But, in the SaaS model, enterprise data is stored at the

SaaS provider's data center, along with the data of other enterprises. Furthermore, if the SaaS provider is controlled a public cloud computing service, the enterprise data may be stored along with the data of other unrelated SaaS applications. The cloud provider may duplicate the data at multiple locations across countries for the purposes of maintaining high availability. Most enterprises are well-known with the traditional on- promise model, where the data continued to reside within the enterprise boundary, subject to their policies. Cloud computing providers required to solve the common security challenges being faced by conventional communication systems.

So, among these methods, code generation based on MDA is somewhat infeasible due to complicated relations and the always-changing execution environment. Model transformation from CIMs to PIMs, which is used to drive service-level configuration, can be promising way for end users to develop their personal or customized web-based application.

4.2 Cloud SaaS

A Cloud Software-as-a-Service (SaaS) is a capability provided to the consumer to use the provider's services running on a cloud platform. The services are accessible from various client devices through a thin client interface like web browser. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of restricted user-specific application configuration settings [9]. The applications in the cloud may be as simple as a time zone converter performing a single discrete function, or as complex as a holiday packaging system performing a set of linked business functions.

Obviously, the technologies are constantly developing. Instead of directly developing these cloud software services using available technologies, modeling them at a higher level of abstraction will decouple them from the undesired effects of technology change and enhance their longevity. An MDA based development of cloud SaaS will allow to define these services in a technology-independent way and will play an important role in improving the quality of cloud software services, making them more robust, flexible and lively. Encapsulating business logic in a way that is independent of the technical mechanisms will properly capture the fundamental nature of the applications; and will also make it possible to reuse them in a variety of framework [10].

We demonstrate our approach using an example of software application running as a service in the cloud – the Credit Card Verification System (CCVS). The CCVS may be accessed by anyone connected through the Internet, with web browser interface. For Example, a customer could purchase products or services from an online or offline store and make payment through his credit card. In this situation, the CCVS can be accessed from the merchant site for validating the card.

We assume a simple approach to credit card verification where the various steps in the business process may be listed as:

- The consumer purchases goods or services from a merchant (online/offline) through his credit card.
- In case of online shopping, the credit card details are submitted at the customer's browser. Here these details are securely transmitted to the merchant site and to the CCVS at last. In an offline merchant store, the credit card details are submitted to CCVS through the Electronic Funds Transfer Point of Sale (EFTPOS) terminal by swiping the card.
- After received by the CCVS, the credit card details and the bill amount is sent to the Clearing House.
- The Clearing House submits the transaction to the bank that issued the card.
- The bank verifies the card details and transaction amount, and sends the information, approving or refusing the transaction, to the Clearing House.
- The Clearing House then forwards these details to the merchant.
- If the transaction is approved, the bill amount is refund to the merchant by the Clearing House.
- The bank pays the amount to the Clearing House afterwards.
- The bank deducts the transaction amount from the credit available to the customer.
- The bank, receives the payment from the card holder at a later date. Any processing fee charged by the Clearing House or the card problem bank is documented on the card statement as expense on part of the customer.
- So, The CCVS application needs to interact with other entities which are external to the system, the merchant, the bank (the bank that issues the card) and the Clearing House, so as to accomplish the task.

The following assumptions have been made by the authors, with regard to the system under consideration

- Only one person can hold an account in a bank
- Only one clearing house can process a credit card.
- A customer cannot have a credit card that is not issued by a bank. The payment through Credit Cards contains two main processes 1) Credit Card validation, and 2) Settlement of payment. For simplicity, we are only focusing on the first process, the validation of the credit card. Even though MDA does not limit itself to Unified Modeling Language (UML) for modeling the system, we are using UML for the purpose of diagram. A use case diagram capture the functionality of the system is represented in Fig.3

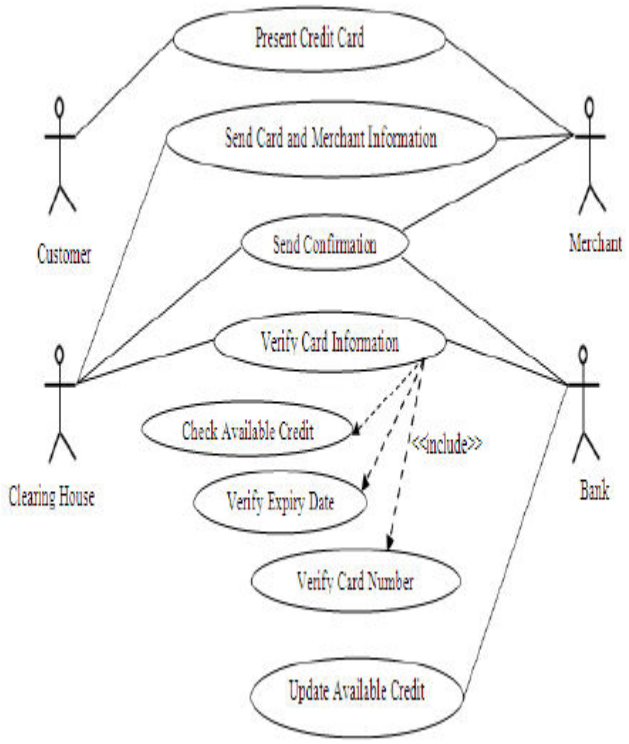


Fig. 3 Use Case Diagram for CCVS

The characteristics of the actors in the system are:

- Customer having credit card to make payments for purchases.
- Merchant receives the payment from the customer for the goods or services sold.
- Clearing House is a firm that has contract with the merchant's bank to clear charges in exchange for a flat fee and a percentage of every charge processed.
- Bank has issued the credit card to the customer. A PIM specifies the system at a higher level of abstraction as compared to a PSM. Because, the PIM is independent of any technology used for implementing the system, it can be reused for several different platforms.

5. Conclusion

Cloud computing idea at a future where the computations would be performed on centralized facilities operated by third party compute and storage resources, instead of local computers. A technology-specific software development is not feasible in the long run, since with the appearance of newer technologies, the older ones may become old and/or may get replaced. The illustration of the PIM and PSM in this paper support that developing software applications in the cloud in a manner that is independent of the specific technologies, using MDA, will enable to gather the benefits of MDA based software development. Above all, it will enhance the inflexibility, durability and reusability of the cloud service developed.

REFERENCES

[1] Zhou, J., Leppanen, T., Harjula, E., Ylianttila, M., Ojala, T., Yu, C., & Jin, H. Cloud things:"A common architecture for integrating the internet of things with cloud computing", 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), (2013). pp. 651-657.

[2] Aazam, M., Khan, I., Alsaffar, A. A., & Huh, E. N.: "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved", 2014 11th International Bhurban Conference on Applied Sciences and Technology (IBCAST), (2014) pp. 414-419.

[3] Tei, K., & Gurgen, L. :"ClouT: Cloud of things for empowering the citizen clout in smart cities. 2014 IEEE World Forum on Internet of Things (WF-IoT), (2014) pp. 369-370

[4] Zhang, H., Liu, J., Zheng, L., Wang, J.: "Modeling of Web Service Development Process Based on MDA and Procedure Blueprint". 2012 IEEE/ACIS 11th International Conference on Computer and Information Science (ICIS), pp. 422-427 (2012)

[5] Mukhtar, M.A.O., Hassan, M.F.B., Jaafar, J.B., Rahim, L.A.: "Enhanced Approach for Developing Web Applications Using Model Driven Architecture". International Conference on Research and Innovation in Information Systems (ICRIIS 13), pp. 145-150 (2013)

[6] Du, J, Dean, D, Tan, Y, Gu, X, Yu, T: "Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds". IEEE Transactions on parallel and distributed systems, Vol. 25, Iss. 3, pp. 730-739 (2014)

[7] Huang, W., Wei, X., Zhao, Y., Wang, Z., and Xiao, Y.: "A Multi-tenant Software as a Service Model for Large Organization". 2013 International Conference on Cloud and Service Computing, pp. 112- 119 (2013) [21]

[8] Ketter, W., Banjanin, M., Guikers, R., Kayser, A.: "Introducing an agile method for enterprise mash-up component development". Proceedings of the 12th IEEE conference on commerce and enterprise computing, Washington (2009)

[9] Peter Mell and Tim Grance, "The NIST Definition of Cloud Computing", Version 15, 10-7-09, [Online] Available:http://thecloudtutorial.com/nistcloudcomputingdefinition.html

[10] David Frankel, John Parodi, "Using MDA to develop Web Services", IONA Technologies PLC, Second Edition, April, 2002 [Online] Available: http://www.iona.com/archwebservice/WSMDA.pdf

[11] http://www.gartner.com/it-glossary/cloud-computing

[12] http://www.mobilecloudcomputingforum.com/

[13] Portio Research Mobile Fact book, Portio Research http://www.portioresearch.com/resources.html (2013)