ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (495-498), Month: April - June 2015, Available at: www.researchpublish.com

CHANGING THE CLOUD FROM VENDOR LOCK-IN TO THE META CLOUD

LAVANYA P.

Department of Computer Science and Engineering Visvesvaraya Technological University, Belagavi, Belagavi, India

Abstract: **The cloud offerings are emerged from a multitude of service providers calls for a Meta cloud that smoothens the jagged cloud landscape edges. The current public and hybrid cloud users face the problem of vendor lock-in problems that can be solved by the Meta cloud.**

Keywords: **Vendor Lock-In To the Meta Cloud, Changing the Cloud.**

1. INTRODUCTION

The cloud computing success is largely due to customers' ability to use services on demand with a pay-as-you go pricing model and it has achieved widespread adoption Low costs and high flexibility make migrating to the cloud compelling. Many companies hesitate mainly because of concerns related to service availability, data lock-in, and legal uncertainties to "move to the cloud". Lock-in is particularly problematic. For one thing, outages will occur even though public cloud availability is generally high. Businesses locked into such a cloud are essentially at a standstill until the cloud is back online. Moreover, service level agreements (SLAs) generally are not guaranteed by the public cloud providers- that is, the required quality of service (QoS) is not provided to the businesses that are locked into the cloud. Finally, most public cloud providers' terms of service let that provider unilaterally change pricing at any time. Hence, there is no mid- or long term control over its own IT costs for the businesses locked into a cloud.

At the core of all these problems, the need for businesses can be identified to permanently monitor the cloud they're using and be able to rapidly "change horses" – that is, we can switch over to the next cloud if we won't get security from the other cloud so we are using two clouds at a time. Myriad cloud providers are the one that are flooding the market with a lot of services, including compute services such as the Amazon Elastic Compute Cloud(EC2) and VMware vCloud, or Key-value stored, such as the Amazon Simple Storage Service(S3). These services don't follow standards but their own and some services are comparable to each other and others are vastly different, but they are all technically incompatible. To complicate this situation, many companies combine public offerings with their own private clouds, leading to hybrid cloud setups and they build on public clouds for their cloud computing needs.

We introduce the concept of a Meta cloud that incorporates design time and runtime components and also abstract away technical incompatibilities from existing offerings', thus reducing vendor lock-in. It helps to find out the right set of services for a particular use case and supports an application's initial deployment and runtime migration.

Problem Statement: Cloud providers are the one that are flooding the market with a confusing body of services such as the Amazon Elastic Compute Cloud, key-value stores. These services are comparable to each other, whereas others are vastly different. To further complicate this situation, many companies combine public offerings with private clouds and they also build on public clouds for their cloud computing needs.

Let's consider a web –based sport portal for an event , which allow the users to place bets. For this event cloud computing paradigm provides the necessary flexibility and elasticity as it requires an enormously efficient and reliable infrastructure. It lets service providers handle short-term usage spikes without needing respective dedicated resources available continuously. The problem is that once an application has been developed based on one particular provider's cloud services and using its specific API, that application is bound to that provider deploying it on another cloud would usually require completely redesigning and rewriting it. Such vendor lock-in leads to strong dependence on the cloud service provider.

2. OBJECTIVES AND MOTIVATION

The meta cloud promises transparent use of cloud computing and reduces the vendor lock-in. Meta cloud lack integration and most of the companies necessary to realize the meta cloud already exist. Thus, integrating state-of-the-art tools promises a huge leap toward the Meta cloud. The community must create a truly open meta cloud with added value for all customers and broad support for different providers and implementation technologies to avoid meta cloud lock-in.

3. SYSTEM ARCHITECTURE

Here we are using UML for describing the system architecture in detail using the blueprint.

UML Diagrams:

UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

UML is a method that is very important part of developing objects oriented software and the software development process. It uses mostly graphical notations to express the design of software projects.

Using the UML helps project teams communicate, validate the architectural design of the software and explore potential designs.

Definition: UML is a general-purpose visual modeling language that is used to specify, visualize, construct and document the artifacts of the software system.

UML is a language: It will provide rules for communication and function on conceptual and physical representation and it also provide vocabulary.

UML Specifying: Specifying means building models that are precise, unambiguous and complete. The UML address the specification of all the important analysis, design and implementation decisions that must be made in developing and displaying a software intensive system.

UML Visualization: UML makes easy to visualize the system and for better understanding. It includes both graphical and textual representation.

UML Constructing: UML models can be directly connected to a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.

UML Documenting:

UML provides variety of documents.

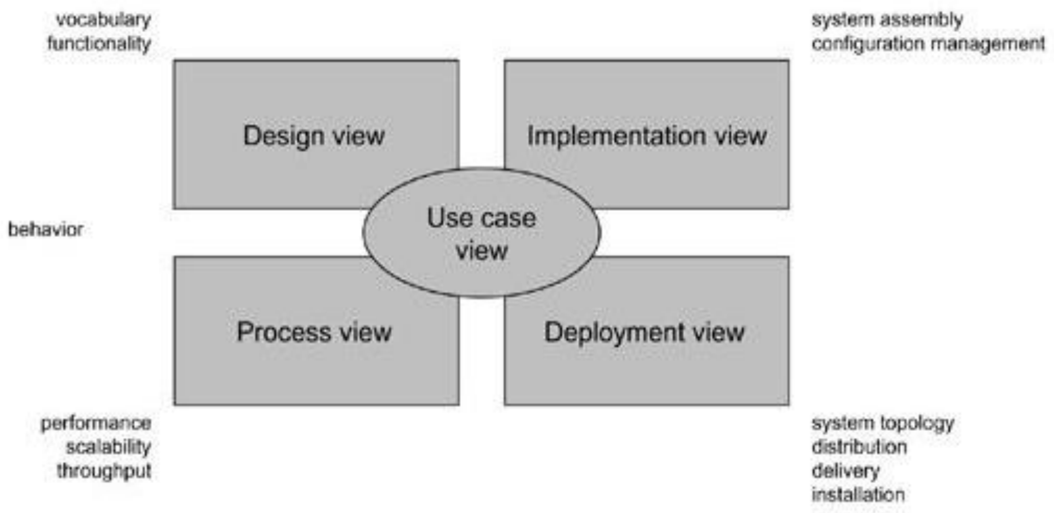

Fig: 3.1. Modeling a system architecture using views of UML

The use *case view* of a system encompasses the use cases as seen by its end users, analysts, and testers that describe the behaviour of the system.

The *design view* of a system encompasses the collaborations that form the vocabulary of the problem and its solution, the classes, interfaces.

The *process view* of a system that forms the system's concurrency and synchronization mechanisms encompasses the threads and processes.

The *implementation view* of a system that are used to assemble and release the physical system encompasses the components and files.

The *deployment view* of a system encompasses the nodes that form the system's hardware topology on which the system executes.

4. MODULE DESCRITION

1. Registration
2. Login
3. File Upload
4. Migrate Cloud
5. Send Mail

Registration:

In this module, user, owner, ttp(trusted third party), csp(cloud service provider) have to register first, then oly he/she has to access the database.

Login:

In this module, any of the above mentioned person have to login, they should login by giving their username and password.

File Upload:

In this module, owner uploads a file(along with meta data) into cloud, it subjects to validation by TTP before it gets uploaded. Then TTP sends file to CSP. CSP decrypt the file and tries to modify the data but he can't modify it and the alert will go to the owner and results in the cloud migration.

Migrate Cloud:

The advantage of meta cloud is that if we are not satisfied with one csp we can switch over to the next cloud as it will not provide security. Here we are using two clouds at a time. In second cloud, they can't corrupt the real data, if they made an attempt, it will fail.

Send Mail:

In this module, owner will send mail to the user along with file description key, so that user can download the file. Owner sends the mail to the users who are registered earlier while uploaded the file into the correct cloud.

5. RELATED WORK

In this model, standardized programming APIs are used to create cloud-neutral applications that aren't hardwired to any single provider or cloud service. Cloud provider abstraction libraries such as libcloud, fog and jclouds are used for accessing different vendors' cloud products and they provide unified APIs. Using these libraries, developers are relieved of technological vendor lock-in because they can switch cloud providers for their applications with relatively low overhead.

As a second ingredient, the meta cloud defines concrete features by using the resource templates that the application requires from the cloud.

6. CONCLUSION

The meta cloud promises the transparent use of cloud computing services and attenuates the vendor lock-in. It lacks integration. Integrating these state-of-the-art tools provide huge leap toward the meta cloud. To avoid meta cloud lock-in, the community must drive the ideas and truly create a open meta cloud with added value for customers and broad support for different providers and implementation technologies.

ACKNOWLEDGEMENT

I feel great pleasure to acknowledge the guidance and assistance of all those people who have made work on this project endeavor and I thankful anonymous references for helpful suggestion.

REFERENCES

[1] M. Armbrust et al., "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, 2010, pp. 50–58.

[2] B.P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," Proc. Int'l Conf. Networked Computing and Advanced Information Management, IEEE CS Press, 2009, pp. 44–51.

[3] J. Skene, D.D. Lamanna, and W. Emmerich, "Precise Service Level Agreements," Proc. 26th Int'l Conf. Software Eng. (ICSE 04), IEEE CS Press, 2004, pp. 179–188.

[4] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-of-the-Art and Research Challenges," J. Internet Services and Applications, vol. 1, no. 1, 2010, pp. 7–18.

[5] M.D. Dikaiakos, A. Katsifodimos, and G. Pallis, "Minersoft: Software Retrieval in Grid and Cloud Computing Infrastructures," ACM Trans. Internet Technology, vol. 12, no. 1, 2012, pp. 2:1–2:34.