

# A Survey of Cloud Migration Methods: A Comparison and Proposition

Khadija SABIRI<sup>1</sup>, Faouzia  
BENABBOU<sup>1</sup>

Information Technology and  
Modeling Laboratory<sup>1</sup>  
University of science in Ben M'sik  
Casablanca, Morocco

Mustapha HAIN<sup>2</sup>, Hicham  
MOUTACHAOUK<sup>2</sup>

Information Technology and  
Modeling Laboratory<sup>2</sup>  
ENSAM  
Casablanca, Morocco

Khalid AKODADI<sup>3</sup>

LAROSERI Laboratory<sup>3</sup>  
Faculty of Sciences, CHOUAIB  
DOUKALLI<sup>3</sup>  
El Jadida, Morocco

**Abstract**—Along with the significant advantages of cloud computing paradigm, the number of enterprises, which expect to move a legacy system towards a cloud, is steadily increasing. Unfortunately, this move is not straightforward. There are many challenges to take up. The applications are often written with the outdated technologies. While some enterprises redevelop applications with a specific Cloud provider in mind, others try to move the legacy systems, either because the organization wants to keep the past investments, or because the legacy systems hold important data. Migrating the legacy systems to the Cloud introduces technical and business challenges. This paper aims to study deeply and to compare existing Cloud migration methods, based on Model Driven Engineering (MDE) approach to highlight the strengths and weaknesses of each one. Finally, we have proposed a Cloud legacy system Migration Method relied on Architecture Driven Modernization (ADM), and explained its working process.

**Keywords**—Application on premise; Migration methods; Cloud migration Method; PIM; PSM; ADM

## I. INTRODUCTION

Cloud computing has lately attracted full attention of many stakeholders due to its economic, business and technical benefits. Almost IT enterprises regard Cloud computing as a new way to adapt their business strategy to their business goals [1]. Clouds are the model of the utility computing based on resources virtualization in terms of hardware, development platforms, and services [2]. Cloud computing has three service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [2–4]. The pay-as-you-go is the Cloud business model, which the enterprises have the ability to decrease capital expenditure [5]. Additionally, Cloud computing provides plenty of advantages such as costing flexibility, high scalability, rapid elasticity, investment cost saving and so on [6]. The deployment models of Clouds are public, private, hybrid and community [7].

Many enterprises expect to move legacy systems to the Cloud to exploit the benefits of Cloud. However, many questions, as well as various challenges, are considered. Thus, through Cloud migration, the enterprises either have to start over or to adapt existing systems to Cloud. Despite outdated legacy systems technologies, it still works for the user's needs and is adaptable to business logic. In one hand, the cost of replacing them with the newly designed systems from scratch

is often too high. In the other hand, every provider has their own standards that may impact on the way that applications are developed. As a result, users may be bound such a provider which hinders the decision of many customers to move to Cloud.

Many types of research in Cloud migration have been achieved. However, only a few have proposed a generic solution, even though there are more solutions to specific types of applications or at a given Cloud type. In this paper, we propose a comparative study between Cloud migration methods, notably those based on MDE, considered models in the centre of the software engineering process. The objective is to propose a Cloud migration Method based on ADM to ensure processes migration from legacy system on-premise to Cloud environment regardless of features applications and Cloud providers.

The rest of the paper is structured as follows: Section 2 illustrates literature survey of related works. A comparison of existed Cloud migration methods is presented in Section 3. Section 4 displays the proposed Cloud Migration Method. Section 5 contains a comprehensive conclusion of the paper findings.

## II. SURVEY OF RELATED WORKS

In the literature, there are several studies on cloud migration, such one has looked from a particular point of view which prevents to have a global vision of from the start of the cloud migration process. Basing on [8], those works are classified from the point of view discussed in the migration issue:

- Decision making support;
- Migration methods;
- Developments tools.

This section presents some work that focuses on decision-making support, migration methods and development tools for moving existing system to the Cloud services.

### A. Decision Making Support

The works in references [9–10] present a support of a cloud migration decision in enterprises, analyzing the key factors to evaluate the application in terms of technical and non-technical

constraint that must be considered before migration. Further aspects should be covered such as a study of migration feasibility, whether the application could be moved and choosing suitable Cloud provider.

#### Migration Methods

Other researchers have tackled migration methods [11–14]. the paper herein referenced, [11] proposes taxonomy of the migration tasks involved, which is based on experiences of migrating PetShop DotNet to Windows Azure, a cloud platform type, as well as Java PetStore to Amazon EC2, a Cloud infrastructure type, in order to enhance the taxonomy of migration tasks in a way to sum up a generic cloud migration process.

Other specialists [12] have analyzed experiences from moving Hackystat, an Open Source Software (OSS), to a Cloud. It aims to run the application on IaaS which will be consumed as services. Migrating to an SOA is required. Finally, the work referenced in [13] proposes Cloud Motion Framework (CMotion), which supports migrating composite applications into Clouds. The aim of this framework is to enable the different technologies to be compatible with each, by using adapters and runtimes to host each component.

#### B. Development Tools

Some works have been addressed to provide development tools for cloud migration process [15–17].

MoDisco [15] is an extensible framework defined as a support for several legacy technologies by providing a set of models to make easier the design and building of model-based solutions suited to legacy systems. Modisco is based on MDE paradigm to enhance the existed model-driven reverse engineering approaches.

Blu Age [16] is an agile solution with the aim of modernizing application, starting by extracting the business code of the legacy system to transform it into a PIM presentation independent of any technologies and regenerates it to a modernized system through MDA approach. Blu Age consists in designing a platform for supporting existing software modernization based on a model-driven development (MDD). It is focused on business level translation rather than translating legacy code at a technical level to a given technology.

Modelio is an extended open source tool supported UML standard. It is designed to support mechanisms extensibility by offering a set of modules to meet the specific needs [17]. Modelio starts with knowledge discovery from legacy system to recover legacy model in order to generate SOA components thanks to SOAtization methods.

#### C. Discussion

To summarize, this section enumerates current researches which are divided in decision-making support, migration methods and development tools. The first approach considered as the first step in migration process covers migration tasks with the aim of facilitating decision makers such as feasibility study, decision to move towards Cloud and the identification of the suitable Cloud provider. The second one is about migration

methods that focused mainly on refactoring application code to leverage with such Cloud platform. The third approach defined some tools devoted to migrating a legacy system to Cloud.

The aim of this paper is to run comparative analysis tackling other Cloud migration methods that are based on MDE approach.

### III. PRESENTATION OF CLOUD MIGRATION METHODS

This section illustrates the existing Cloud migration methods, based mainly on MDE approach, its motivation, and its overall solution.

#### A. CloudMIG

The aim of CloudMIG [18] is summarized by the following points:

- Reduce over and under resources' provision;
- Evaluate additional expenditure;
- Discuss scalability issues;
- Align existing systems with a Cloud environment.

This approach consists of six steps to move a legacy system to a Cloud:

- Extraction: Reverse engineering step is needed to better understand legacy system for the purpose of generating result a software's representation, a current architecture of software, and software system's utilization model acting;
- Selection: aims at selecting a target platform for a given Cloud provider-specific during migration process, thanks to Cloud Environment Model (CEM), it describes shared characteristics of different Cloud environments;
- Generation: This activity produces three output models such as architecture of targeted platform, mapping model, and violations model of the target architecture which describes the constrains of Cloud environment;
- Adaptation: is a manual rearrangement of the target architecture and deployment adaptation;
- Evaluation: this step analyzes the produced target architecture using dynamic and static metrics;
- Transformation: comprises manual migration towards target architecture.

#### B. REMICS

The aim of REMICS (Reuse and Migration of legacy systems to Interoperable Cloud Services) is to support cloud migration of existing systems with using model-driven methodology and tools [19]. This work focuses on:

- Outdating technologies of legacy systems;
- Replacing the legacy system with systems designed from scratch;
- Reusing and integrating the legacy applications;

- Adapting existing services and increasing their interoperability to cope with the migration requirements.

To tackle these issues, this methodology helps a migration of the existing IT systems to Cloud through model-driven technologies, based on ADM which is the concept of modernization. Starting with recovering the architecture of the legacy system through the architectural model, in order to analyze the legacy system about the feasibility of modernizing different components parts of the legacy applications, and migration to SOA, to obtain the autonomous components. The migration activity identifies the different ways to modernize them by:

- The replacement of components by the online services;
- The utilization the Model-Driven Interoperability (MDI) helps to adapt the existing services using automatically generated mediators;
- The redevelopment of components by wrapping into SOA packages.

### C. ARTIST

The project ARTIST (Advanced Software-based Service Provisioning and Migration of Legacy Software) covers a process to modernize legacy systems by providing migration method suitable to the Cloud [20]. The major outcomes have been treated by ARTIST:

- Tailor a legacy system to Cloud features;
- Migration of a legacy system is tied to using such technology and explicating the architecture of legacy systems.

To overcome these issues, this methodology composes of three main steps:

- Pre-migration phase consists of analyzing both technical and business aspects of legacy system which are considered as the result for the feasibility analysis , to identify the maturity-level assessment as well as target platform expected characteristics;
- Migration phase uses reverse engineering and forward engineering techniques to run the legacy system on Cloud environment. The migration process of the legacy system is tailor to the specific result gotten from the pre-migration analysis and the particular legacy application requirements. Additionally, this step takes into account the business model aspect. Lastly , checking and validating of resulted software are introduced in this phase;
- Post-migration phase relies on the applications components resulted, which are running in the cloud environment, in order to verify whether the technical and business objectives already defined in the pre-migration phase have been reached. Model-based testing and both of end-user functional and non-functional testing are the methods to approve the validation step.

### D. Based on Services

Ref [21] presents a generic methodology, and displays the way to migrate legacy applications to Cloud service, to respond to these limitations:

- Outdated technologies and methodologies;
- Maintain system is hard;
- Tied migration methods to specific technologies;
- Less modification of legacy code.

This work consists of these following seven steps:

- Reconstructing architectural model of the legacy application: analyzes the legacy system and reconstructs a model of the legacy application architecture;
- Redesigning the architecture model: helps to determine services by redesigning the model of architecture's legacy system;
- MDA transformation: transforms the architecture to target;
- Generating Web-services: the Web service which are implemented in WSDL or JEE Annotation are generated as targeted web service;
- Invoking web services: The functionalities of the legacy system are identified in order to be invoked as the web service-base of legacy's functionalities;
- Selecting the suitable Cloud platform: it is chosen to be conformed to particular target platform needs , to support the web-services executed;
- Deploying web-services on Cloud platform: End user consumes the Web services running on Cloud. Service.

### E. PaaS Migration

This work aims [22] at proposing a migration method between Cloud platforms, to deal with the vendor lock-in issue, resulted of locking application to specific Cloud features platform. Each cloud platform uses his specific tools, libraries, technology ...etc.

In order to overcome this issue, this work has cited above attempts by offering a set of tools to re-engineer software, using the MDA and refactoring approach. This approach contains, besides Pattern Definition with Additional transformation rules, three steps:

- The first step is Discovery which aims at extracting the source platform to have a high level of sources representation.
- Transformation step aims at conforming the system to cloud platform through mapping rules, made to bridge the two representations.
- Migration step aims to test the resulted software deployed in cloud platform and to generate replies at runtime. The step checks the deployment at runtime and

the original platform utilizing rollback, used the old setting and code of the legacy system.

#### IV. COMPARATIVE STUDY OF MIGRATION METHODS

In this section, the analysis is on the previous migration methods in terms of characteristics, drawing the comparative tabular to show strengths and weaknesses of each one. Firstly, we define some criterions used in this study.

##### A. Comparison criterions

- The Business and Technical Constraints: to show if the business and technical aspect take into consideration during the migration process.
- Level of automation: means the degree of migration methods automation: semi- or fully-automated approach;
- Architecture adaption: describes how to adapt legacy system architecture to Cloud environment, it composes of the following sub criterion:

a) *Legacy system to PSM*: explains process of recovering models with knowledge discovery;

b) *Legacy PSM → PIM*: means transformation process from PSM to PIM through Reverse Engineering techniques;

c) *Upgrade PIM*: means describes the way architecture application was redefined, with aim of obtaining the autonomous components;

d) *PSM → PSMCloudified*: is the process of adapting the software regarding the targeted platform through mapping rules;

e) *PIMUp → PSMCloudified*: explains the transformation processes from PIMUp to PSMCloudified. It is the result of the compatibility with the cloud platform chosen and legacy platform specific provided as the migrated software.

- Migration layer: points out which layer takes place;
- Migration validation: aims to check whether the resulted system has worked as supposed to do, either by case study or by experimental experience.

##### B. Comparison

Each symbol “+” represents level of automation methods and describes the way used in it, namely:

<sup>1</sup>: Transforming legacy system to PSM;

<sup>2</sup>: Transforming legacy system to PSM and applying Forward Engineering techniques through PIM4Cloud;

<sup>3</sup>: Transforming a legacy system to PSM and PSM to PIM;

<sup>4</sup>: Transforming a legacy system to PSM and PIM to web services;

<sup>5</sup>: Transforming a legacy system to PSM and PSM to PSMCloudified.

According to this comparative study, we have noticed that each Cloud migration method has advantages and disadvantages, notably:

- The works in references [18, 21] didn't take business aspect into account rather they focus on running a legacy system in IaaS, PaaS Cloud. However, the other works stress out the need of adding the business concerns to technical aspects.
- Reference [19] is based on migrating the SOA autonomous components, either by replacing one or more legacy component by Cloud services, or wrapping legacy component.

As conclusion, this migration method presents three techniques to move to Cloud such as:

a) *Replacing component(s) by Cloud offerings*: This is the least invasive type of migration, where one or more (architectural) components are replaced by Cloud services [5]. Such migration type aims to resolve incompatibility issue through a set of reconfiguration to take a profit of the supported layer [23] and may be more expensive than rewriting.

b) *Outsourcing and wrapping*: the application functionality which is based on exposing them as service, hosting and running on Cloud environment. It requires the modernization of the legacy system by presenting them as autonomous components SOA in order to take a plenty of benefits of SOAs paradigm. Although the migration of a legacy system towards SOA is not an easy task, it must answer two major questions such as: what part of the legacy application can be moved[24], and how can be executed the migration process [25]. As a result, not all legacy system are enough mature to take up this transformation, and two studies required moving to Cloud, start with establishing a study to migrate to SOA, followed by another one to Cloud.

c) *Cloudifying*: is focused on transformation model to model based on Cloud blueprints, patterns, and best practices.

- In this study, the maturity aspect of the methods is measured by experiment experiences or case studies.
- In the Transformation PSM to PSMCloudified, users are not allowed to add more functionality or custom application responding to their needs.
- Automation level is referred to measure the level of automated cloud migration process. However, in this study, a fully automated process isn't recorded.

TABLE I. A COMPARATIVE TABULAR OF MIGRATION

Criteria Methods	Constraints on migration		Architecture Adaptation					Migration Evaluation			Migration layer	Automation degree
	Technical	Business	Legacy system → PSM	Upgrade PIM	Transformation		Tools	Approach Migration validation	Test			
					PIM → PSM Cloudified	PSM → PSM Cloudified						
<b>CloudMIG</b>	Yes	No	Yes	Yes	No	No	No	Cloud SIM	Rating	Experiment experience	IaaS, PaaS	+ <sup>1</sup>
<b>REMICS</b>	Yes	No	Yes	Yes	- Migration to SOA - Redesign architecture application by Soaml	- Replace legacy component - Wrap legacy component - Forward Engineering by PIM4Cloud	No	-	model-based testing model-based validation techniques	Case Study	IaaS, PaaS,	++ <sub>2</sub>
<b>ARTIST</b>	Yes	Yes	Yes	Yes	No	- Transformation optimization patterns	No	-	Equivalence Test	Case study	IaaS, PaaS,	++ <sub>3</sub>
<b>Based on services</b>	Yes	No	Yes	Yes	- Redesign architecture application - Identified service	- Transformation to web services - Hosting web service in environment Cloud	No	-	Not Mentioned	Case Study	IaaS, PaaS	++ <sub>4</sub>
<b>PaaS migration</b>	-	-	Yes	No	No	No	Transformation rules	-	Test cases	Example	PaaS	++ <sub>5</sub>

- Transformations focus on at the level of architectural models by redesigning the architecture of legacy system within a service-oriented architecture.

### C. Discussion

As a conclusion, we find out that the underestimation of legacy system complexity is referred to the lack of architecture documentation or unknown internal structure. Additionally, restriction technically related to the legacy systems nature such as huge of data and code, outdated legacy technologies and absence of separation concerns are considered as the main challenges to overcome so to better understand the legacy system. As shown, each migration process is tailored to a specific requirement of the legacy system.

- Moving legacy systems with keeping functionalities and data of the application in order to improves the past investments in software;
- Give users the possibility to add new business or Cloud functionalities if necessary;
- Target system becomes easy to understand, to maintain, and to make changes.

This method aims at presenting each process as a model to explore the benefits of Model Driven Engineering paradigm.

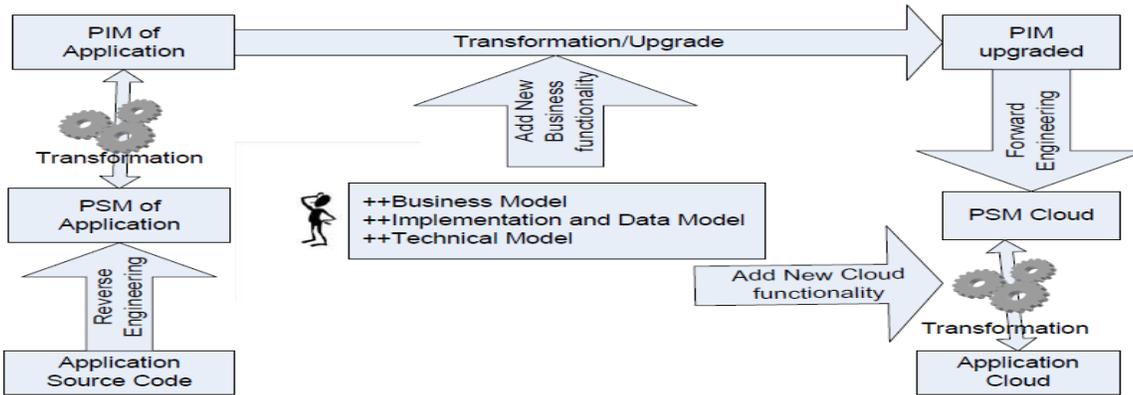


Fig. 1. Proposed Cloud Migration

*b) Implementation and Data Model:* Identifies business requirement through business process defines the architectural model of the application and their interrelationships and to identify the several components of the application in order to define which component of the architecture is hosted in which of the containers.

*c) Infrastructure Model:* describes the usage of the physical resource in a way to determine which resources are not properly used resulting in additional costs such as energy, maintenance, etc...

To sum up, most of the presented Cloud migration methods are mainly based on migrating the SOA autonomous components, as it has potential advantages but also some significant drawbacks, mainly not all applications are enough mature to transform it into services and it may not be always possible to reuse legacy system's functionality by offering them as services due to technical concerns. In conclusion, replacing them could be less expensive than reused. In the next section, we propose our Cloud Migration Method and describe the processes migration independently of the type of legacy systems and Cloud Provider.

### V. PROPOSED CLOUD MIGRATION METHOD

We summarize the objectives of this work showed in fig 1 in the following points:

The results of extracting Application Source Code, the PSM of Application, is modeled in three views, which enables to separate out the concerns in way to identify enterprise need and to match with Cloud model services such as:

*a) Business Model:* Identifies business requirement through business process of legacy application. In this step, the motivation for migration and the goals to be achieved is elaborated. It also describes how the application will be behaved towards a cloud computing environment.

The difference between PIM Application and PIM Upgrade is that the first one describes the current functionality offered by the legacy application regardless of any technical details, and the second one enables users the possibility to add new business functionality if necessary. As a consequence, a new design artifact evolves from the original design. PSM Cloud is the results of Meta-model transformation between a Meta-model legacy system and a Meta-model Cloud application through transformation rules, which have to be conformed transformation Meta-model

## VI. CONCLUSION AND PERSPECTIVES

In this paper, Cloud migration methods were reviewed based on Model Driven Engineering (MDE). As a result, we obtained a comparative tabular, which identifies weak and strong points for each one according to own comparison criteria. We also proposed own Cloud Migration Method, relied on ADM approach, which is focused on modernization activities on the architectural models rather than code artifact.

The future work aims to elaborate Meta models of the legacy system and Cloud application, which will be an abstract model to define Meta model transformation, in order to explore the benefit of models paradigm. Thus, legacy system model and Cloud application model have to be conformed to own Meta models to run Model-to-model transformations which translate between source and target models regardless of Cloud provider and requirement legacy system.

### REFERENCES

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia "A view of cloud computing", ACM Communication. vol. 53, 2010.
- [2] C. Y. Low, Y. Chen, M. C. Wu. Understanding the determinants of cloud computing adoption. *Industrial Management & Data Systems*, vol. 111, no. 7, 2011.
- [3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, M. Zaharia, "Above the clouds: A Berkeley view of cloud computing" Technical Report, California University, 2009.
- [4] L. M. Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner. "A break in the clouds: towards a cloud definition", ACM Communication, vol. 39, 2008.
- [5] V. Andrikopoulos, T. Binz, F. Leymann, S. Strauch, "How to adapt applications for the cloud environment: challenges and solutions in Migrating Applications to the Cloud Computing", vol. 95, no. 6, 2013.
- [6] Q. Zhang, L. Cheng, R. Boutaba. "Cloud computing: State-of-the-art and research challenges". *Journal of Internet Services and Applications*, vol. 1, no. 1, 2010, pp. 7–18.
- [7] C. Zhili, D. Zhihui, "A service-oriented virtualization management system with automated configuration". *IEEE International Symposium on Service-Oriented System Engineering*, 2008, pp 251–256. IEEE.
- [8] J.F. Zhao, J.T. Zhou. "Strategies and methods for cloud migration", *International Journal of Automation and Computing*, DOI: 10.1007/s11633-014-0776-7, 11(2), 143-152, April 2014.
- [9] A. K. Hosseini, D. Greenwood, J. W. Smith, L. Sommerville. "The Cloud Adoption Toolkit: Addressing the Challenges of Cloud Adoption in the Enterprise". 2012.
- [10] M. Menzel, R. Ranjan. "CloudGenius: A Hybrid Decision Support Method for Automating the Migration of Web Application Clusters to Public Clouds". *Proceeding on IEEE Transactions on Computers*, DOI: 10.1109/TC.2014.2317188, 2014.
- [11] V. Tran, J. Keung, A. Liu, A. Fekete. "Application migration to cloud: taxonomy of critical factors". *2nd International Workshop on Software Engineering for Cloud Computing*, ACM, USA, 2011.
- [12] M. A. Babar, M. A. Chauhan. "A tale of migration to Cloud computing for sharing experiences and observations". *2nd International Workshop on Software Engineering for Cloud Computing*, ACM, USA, 2011, pp. 50–56.
- [13] T. Binz, F. Leymann, D. Schumm. "CMotion: A framework for migration of applications into and between Clouds". *IEEE International Conference on Service-Oriented Computing and Applications*, USA, 2011.
- [14] Q.H. Vu, R. Asal. "Legacy Application Migration to the Cloud : Practicability and Methodology". *IEEE Eighth World Congress on Services*, 2012.
- [15] H. Bruneliere, J. Cabot, F. Jouault, F. Madiot. "MoDisco: A model driven reverse engineering". *Information and Software Technology Vol 56*, 2014, pp. 1012–1032.
- [16] Modernization legacy system Blue age Available: [http://www.omg.org/mda/mda\\_files/Blu\\_Age\\_datasheet\\_en.pdf](http://www.omg.org/mda/mda_files/Blu_Age_datasheet_en.pdf), August 12, 2015
- [17] Modelio. Available: <http://www.modelio.org>, August 12, 2015
- [18] S. Frey, W. Hasselbring. "Model-Based Migration of Legacy Software Systems to Scalable and Resource-Efficient Cloud-Based Applications: The CloudMIG Approach". *The First International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2010*.
- [19] M. Parastoo, A. J. Berre, A. Henry, F. Barbier, A. Sadovykh "REuse and Migration of Legacy Applications to Interoperable Cloud Services", *REMICS Third European Conference, ServiceWave*, pages 195–196, 2010.
- [20] A. Menychtas, C. Santzaridou, L. O. Echevarria, J. Alonso "ARTIST Methodology and Framework: A novel approach for the migration of legacy software on the Cloud". *15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2013.
- [21] W. Q. Zhang, A. J. Berre, D. Roman, H. A. Huru. "Migrating legacy applications to the service Cloud". *14th Conference Companion on Object Oriented Programming Systems Languages and Applications*, 2009.
- [22] A. Beslic, R. Bendraou, J. Sopena, J.-Y. Rigolet, "Towards a solution avoiding Vendor Lock-in to enable Migration Between Cloud Platforms", *2nd International Workshop on Model-Driven Engineering for High Performance and Cloud computing (MDHPCL)*, 2013.
- [23] P. Jamshidi, A. Ahmad, C. Pahl, Member, IEEE. "Cloud Migration Research: A Systematic Review", *Cloud Computing*, *IEEE Transactions on* (Volume: 1, Issue: 2), October 2013, pp 142 – 157, 08.
- [24] Sneed, H. "Integrating legacy software into a Service Oriented Architecture". *10th European Conference on Software Maintenance and Reengineering*. IEEE CS Press, 2006, pp. 3–13.
- [25] Lewis, G., Morris, E., Smith, D., L. O'Brien "Service-Oriented Migration and Reuse Technique". *13th IEEE International Workshop on Software Technology and Engineering Practice (STEP'05)*. 2005, pp. 222-229