

Model Based Migration of Cloud Systems: Review and Roadmap

Jaleleddine Hajlaoui[1](✉), Zied Trifa[2], and Zaki Brahmi[3]

[1] MARS Research Laboratory, University of Sousse, Sousse, Tunisia
hajlaouijalel.ig@gmail.com
[2] MIRACL Research Laboratory, University of Sfax, Sfax, Tunisia
trifa.zied@gmail.com
[3] College of Science and Arts at Al-Ola, Taibah University, Madina, Saudi Arabia
eibrahmi@taibahu.edu.sa

Abstract. Cloud computing has become a dominating trend in the IT industry and academia. In recent years, many Cloud providers have emerged in the market, each one has proper standards and service interface. Migration of legacy systems and their data to a Cloud platform or moving applications from one Cloud provider to another are complicated and high cost processes. In major legacy on-premise applications, they bring technical and business challenges. The choice of the best migration strategy is a hard task for Small and Medium-Sized Enterprises (SMEs) looking at best price and zero risk. Although guidelines offered by Cloud providers to support their users in migration tasks, the user implication is often mandatory to perform a conclusive migration. In this paper, we provide an overview of contemporary approaches for the model based migration of applications into Cloud computing. We highlight the Pros and Cons of each one and we present a summary comparison of these approaches. As well, we propose a new Cloud migration approach that renders the migration process more straightforward than with existent approaches and guarantees the services and data portability.

Keywords: Model based migration · Cloud computing · MDA · Multi-agent systems · Survey

1 Introduction

Cloud computing is a fundamental shift in delivering IT services with large impact on organizations. It responds to their requirements in terms of computational power, reduced cost of infrastructure, efficient resource usages and availability of applications. We recognize five main characteristics of Cloud computing namely: on-demand service, broad network access, resource pooling, rapid elasticity, and measured service.

The fast shift of Cloud Computing has contributed to a large market of Cloud services that provide infrastructure, platforms and software for multiple usages. Recently, the paradigm of multi-cloud has imposed itself for several purposes [1]. This paradigm allows, for example, to overcome the problems of elasticity of resources provided by

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022, LNCS 13375, pp. 249–264, 2022.
https://doi.org/10.1007/978-3-031-10522-7_18

only one Cloud provider. This promotes Cloud user independence to a unique provider (i.e., avoid the vendor lock-in) and offers the opportunity to optimize the quality of the resources offered by providers at a time. However, these providers generally use heterogeneous technologies which constitutes various challenges for the use of a multi-cloud ecosystem. Among the most inherent challenges, we find interoperability and portability of resources. Interoperability is related to the ability to deploy applications to different Cloud vendors. Portability is the ability for a user to easily move their data to another Cloud provider. Otherwise, the heterogeneous characteristics of a multi-cloud ecosystem are at the origin of difficulties of deploying applications in different environments. Heterogeneity can appear between the supported programming languages, libraries, services, and others. To solve these problems, standardization initiatives [2] have emerged including Unified Service Description Language (USDL), Topology and Orchestration Specification for Cloud Applications (TOSCA), Open Cloud Computing Interface (OCCI) Cloud Infrastructure Management Interface (CIMI), Cloud Data Management Interface (CDMI), just to cite some of them.

Cloud benefits like Cost reduction, automatic updates, high availability application attract Small and Medium-Sized Enterprises (SMEs) to move totally or partially their data, applications or other business elements from their onsite computers to the Cloud [3]. This process is known as *Cloud migration* which involves also migration of these elements from one Cloud to another. However, the migration process reveals many challenges: *i*) *Vendor lock-in problem and portability*; *ii*) *Complexity of Cloud deployment iii*) *Gap between Cloud stakeholders; iv) Lack of domain concepts and methodologies; vi) Choice of migration strategy*. The migration to the Cloud is not always so simple because it entails several interventions and modifications. The passage of the application from its classical architecture to a modern architecture needs in most cases the refactoring of the application [4]. Conversely, before passing to the refactoring, it's necessary to perform detailed feasibility studies in order to define the different constraints (technical, security, financial ...). These studies help to choose the type of migration (partial or total) or the Cloud provider(s) [5].

Several approaches have been proposed to ensure or facilitate application migration into single or multi-cloud environments. They proposed solutions for the different stages of migration processes. In this paper, we perform a deep study of existing approaches and we compare them according to several criteria such as data portability, degree of automation and type of migration. We propose a new Cloud migration approach based on three main blocks (migration strategy, migration execution, monitoring and maintenance). The rest of this paper is organized as follows. Section 2 identifies challenges of the migration process and defines its different steps. Section 3 surveys the contemporary approaches and compare them across identified criteria. Section 4 describes our proposed approach. Section 5 discuss related surveys. Finally, Sect. 6 concludes the paper.

2 Background

Migration to the Cloud computing is a critical process since there are many challenges to adopt for Cloud solutions. To migrate application to the Cloud environment, we must plan three steps: migration strategy, migration execution and administration.

2.1 Challenges and Motivations

There are many challenges facing Cloud consumer such as:

Vendor lock-in problem: this is a major impediment to the adoption of Cloud computing. It's due to the lack of standardization for Cloud Application Programming Interfaces (APIs) and data models. The heterogeneity of Cloud provider interfaces makes it hard to migrate applications between different Cloud providers or combine different offerings [6]. When every Cloud provider uses his own APIs and platforms, the consumer hasn't the ability to adapt an application and/or its components between different platforms with minimal efforts and costs. Without portability, migrating application requires modifications on code which is hard and expensive.

Complexity of cloud deployment: the deployment in the Cloud is a cumbersome process considering the numerous tasks unavoidable to get a successful deployment (i.e., pre-migrating tasks, migrating tasks and post-migrating tasks). Moreover, the deployment, management and reconfiguration of sophisticated applications across multiple Clouds is even harder. For instance, after the successful multi-cloud deployment, and execution of different components in different Cloud platforms, we should run a monitoring process to detect possible requirements violations, which may oblige a reconfiguration.

Gap between cloud stakeholders: it's important to understand the requirements of diverse stakeholders (providers, developers, administrators, financial managers, etc) and to obtain models at the early stage of development to address their applications concerns [7]. Since variability [8] occurs in all levels of Cloud applications, the configuration and customization of the migrations processes should be considered to fulfill goals and requirements of all tenants. Software Product Line (SPL) and feature modelling [9] can be useful to address engineering complexity and manage variability induced by multi-tenant Cloud migration. In addition, handling security issues and requirements of tenants while defining their migration variants should also be taken into account.

Lack of domain concepts and methodologies: developers need to understand the identity of Cloud applications. Cloud applications are composed of a set of interconnected software components [7]. Each component can be deployed and executed on a different Cloud provider in multi-cloud environment. We need a Cloud modeling language that represent a Cloud architecture deployment model for multi-tenant geo-distributed applications.

Choice of migration strategy: in the first step of the migration process, it is important to define the different constraints (technical, financial, juridical) and details concerning applications, consumers and companies. In addition, it is fundamental to analyze information collected to make the right decisions. The definition of the migration strategy is considered as a complex optimization problem since it is necessary to satisfy all the requirements of customers and applications with zero risk, zero problem and at low costs.

2.2 Migration Steps

a. **Migration strategy**

According to [10], the migration process begins with feasibility and requirements analysis and defining the migration strategy. Basera et al., [5] define the steps to follow

to identify the strategy for migration of the legacy application. There are two categories of migration [11],

- Migration to Cloud or modernization of legacy applications: it's the transition of the application from traditional IT environment to the Cloud environment. After satisfying all constraints of the company, we choose the most suitable type of migration and the Cloud providers and we launch the code refactoring. This category contains 4 subcategories [12]:

  - Replace component(s) with Cloud offering: this type of migration requires adaptation and reconfiguration to address the problems of incompatibility between the Cloud service and the rest of the application. Example: using Google App Engine DataStore in the place of the local MySQL database server.
  - Partially migrate some of the application functionalities to the Cloud: this type of migration entails the migration of one or more application layers to the Cloud. Example: use a combination of Amazon SimpleDB and EC2 instance to host the auditing data and business logic.
  - Migrate the whole software stack of the application to the Cloud: this is the classic example of migration. The application is encapsulated in one or more virtual machines which are running in Cloud platforms [13]. For this type of migration, there is no need to make any changes to the application.
  - Cloudify the application: this is the migration of the entire application. All functionalities of the application are implemented as services which run in the Cloud. For this type, the data and the business layers are mondatory.

- The migration from one provider to another: the lack of standardization between Cloud providers represents a real obstacle hindering interoperability and portability. This problem is unfavorable to the desired objectives by companies. For some companies, the most important is insurance of portability services and data from one Cloud platform to another. By cons, others dream to the possibility of harmonizing the handling of different application components and the interoperability [2] in a standard way between different objects through orchestration [14] of Cloud resources distributed across multiple Clouds to ensure good interaction between the components of the application

b. **Migration execution**

This step includes modeling, development, test and validation and deployment of application components to migrate complicated tasks. Several approaches have been proposed in the literature to facilitate developer's jobs. These approaches are based on Model Driven Architecture, Cloud patterns and semantic modeling. They are more detailed in Sect. 3.

c. **Monitoring and maintenance**

This step follows the deployment of the application. Monitoring and maintenance of the application are mondatory and it is recommended to automate them. Mosaic [15] (Cloud agency) is one of the approaches which offers a semi-automatic solution based on the Multi-Agent Systems for the monitoring and administration.

3 Literature Review and Comparison

To deal with the above challenges (c.f., previous section), many approaches have been proposed. We propose to classify them as approaches based on: Multi-Agent Systems, Cloud patterns, Semantic Modelling, Model Driven Architecture, Cloud Standards and Software Product Line.

3.1 Multi-agent Systems Based Approaches

Multi-Agent Systems (MAS) help to solve complex problems using a system based on a set of smart, interactive, autonomous and dynamic agents. MAS offer techniques to solve problems and improve a number of activities such as brokering, negotiation, management, monitoring and reconfiguration in a multi-Cloud environments [16]. This family of approaches has some boundaries as it does not provide solutions for the step of adaptation and refactoring of application code and data in the migration process. MOSAIC (Cloud Agency) [15] offers a Cloud agency component to deploy and execute application. It helps user to discover and select the best Cloud solution that satisfies applications requirements. During the migration phase, MOSAIC proposes an agnostic interface based on MAS to manage resources of different providers, negotiate with other Cloud providers and reconfigure the MOSAIC application in the new chosen Cloud.

3.2 Cloud Patterns Based Approaches

Several alternatives are offered by different providers that help developers to identify the most viable architecture. For this, Cloud providers and researchers define many Cloud patterns that can organize a pre-configured architecture. The concepts of design patterns are used to implement them. These patterns describe proven solutions and the common aspects of Cloud environments. They help developers to understand the application architecture or the necessary changes to the application code for a successful migration. Many Cloud patterns catalogs are offered by researchers and Cloud providers like IBM Virtual Patterns and Azure Cloud Patterns. These catalogs contain general and specific description solutions to recurring problems in the Cloud.

This line of approaches gives developers patterns to facilitate the tasks of development and migration. Nevertheless, it is hard to define a large number of patterns for each provider and each problem especially with the lack of standardization. Also, patterns must be clear and well-detailed to reduce complexity. Moreover, the major problem of this class of approaches is the mapping between the pattern and the real implementation.

Andrikopoulos and al., [12] determine the three layers of application (presentation, business logic and data) and they focus on the two lower layers of the application (business logic and data). Authors identify three categories of Cloud data patterns: *i*) Confidentiality of patterns for data security *ii*) Functional Cloud data patterns (for data manipulation) *iii*) Non-functional Cloud data patterns (to ensure elasticity and scalability). Cai and al., [15] design an approach for code2code transformation to migrate applications in Cloud environment. The approach automatically transforms the source code to the target code using a set of patterns. The proposed approach contains 4 steps: code scanning, pattern matching, template design and code transformation.

3.3 Semantic Modelling Based Approaches

To develop a Cloud application, developers must have professional skills and knowledge about different programming models, API and the underlying infrastructure. This is due to the lack of standardization and the differences in the semantics of the resources offered by Cloud providers. In fact, every provider has his own terms and representations of available resources. Besides, all Cloud providers expose their operations via Web services. Conversely, each vendor offers its own service interface. For this, it's essential to create semantic models to have good interoperability between different components. These models are helpful in three aspects: functional and non-functional definitions, data modeling and service description enhancement. Technologies used in semantic Web can be useful to address these three aspects. The proposed approaches in this category focus more on the Cloud API and Cloud resources to provide access to heterogeneous resources. MOSAIC (Ontology) [17] provides an open source platform for developing multi-cloud applications. At design time, the developers use the APIs to create applications in the form of a set of multiple Cloud components. At run time, MOSAIC deploys each component on the Cloud platform that provides the best implementation. Thus, developers can choose on which Cloud provider they want to deploy each component.

3.4 MDA Based Approaches

MDA (Model Driven Architecture) is a model based approach for the development of software methodology that focuses on creating models or abstractions. It supports full lifecycle (analysis, design, implementation, deployment, maintenance, etc). The goal of the MDA is *"Design once, build it on any platform"*. MDA defines three main abstraction levels: *i) Computation Independent Model (CIM)* which presents exactly what the system is expected to do, but hides all information technology related specifications to remain independent of how that system will be implemented, *ii) Platform Independent Model (PIM)* is a "*formal*" specification of the structure and function of a system that abstracts away technical details, *iii) Platform Specific Model (PSM)* Specifies how the functionality specified in a PIM is realized on a particular platform.

MDA allows developers to design applications in a Cloud-agnostic way. In fact, it's possible to describe a system as models with high level of abstraction; these models are independent from any platform (PIM Model) of Cloud provider on which we will deploy and run applications. From one PIM model, several PSM models can be generated using model transformation techniques [16], each PSM is specific to a well-defined platform. In

fact, the key of MDA's success is the transformation techniques that provides automatic or semi-automatic transformations from a model to another or from a model to code.

The main benefits of MDA for Cloud applications are facility of portability, interoperability and reusability of application components which can be migrated easily from one platform to another [3]. The MDA approach is based on modeling in high level of abstraction and a transformation engine to reduce the complexity of the migration process especially the refactoring and the adaptation. Although the functional method of transformation engine is ambiguous, the degree of automation of this process is not explicit.

MODAClouds [3] provides decision support systems which guide to select Clouds based on cost and requirements. It offers an open source IDE for application design, development and deployment within a run-time environment for the high-level design, early prototyping, the semi-automatic translation of design artifacts into code and automatic deployment of applications on multi-cloud. REMICS (REuse and Migration of legacy applications to Interoperable Cloud Services) [18] provides a series of tools for software modernization using the *Model Driven Modernization* methodology. The process of modernization of legacy software starts with a recover activity to design the application's architecture. The designed architecture helps to analyze the legacy application and identify the best way for the migration to Cloud platforms. The source architecture will be reformed into target architecture. Herein, MDA is used to obtain the Cloud service implementation. In [19], authors argue how to transform the architecture of a legacy application into a Cloud native application architecture earlier than its deployment in a Cloud environment. Based on the concept of smart use case coupled with the *Architecture Driven Modernization* (ADM), they concretize the process of modernizing of legacy applications through transiting to a Cloud environment. The objective of this process is not only to raise the IT agility but also to increase its business agility. The process is pipelined in a semi-automatic manner especially transformation/upgrade and forward engineering stages implemented by dedicated transformation chain for a specific application. [20] focus on the migration of Cloud infrastructure services and develop a *Model-Based Round-Trip Engineering Approach* which combine generation techniques with runtime reflection. This approach deals with differential changes of IaaS models. After specification of a Cloud infrastructure service topology in terms of a *de jure* model, it will be compared with another model coming from the runtime. The interaction between a reflection service with the interfaces of the IaaS provider generates such a *de facto* model. After that, a model containing differences between the two pre-mentioned models is computed and model transformation is executed.

MoDAC-Deploy [21] is an approach designed to simplify Cloud service deployment and migration between different Cloud Service Providers (CSPs) without user interventions. Moreover, this framework is based on generative mechanisms to overcome APIs heterogeneity and avoid the vendor lock-in by providing application portability across multiple Cloud providers. However, this framework is based on a domain specific modeling language and is not aligned with any Cloud standard which limit its reuse and extension.

StratusML (Stratus Modeling Language) [7] is an architectural framework that grants for the Cloud stakeholders a user interface to model multi-cloud and multi-tenant applications, configure them and define its behavior at the run-time and estimate cost under different Cloud platforms. The CloudML language (*Cloud Modeling Language*) [22] aims to facilitate provisioning and adaptation of multi-cloud application. It is based on two concepts DSML (*Domain-Specific Modeling Languages*) and a Model at run-time. DSkyl [23] is a development platform (PaaS) to develop and deploy Cloud application components. It provides a modeling tool editor with application modeler template and deployment template. Recently, a systematic review of Cloud modeling languages and tools is described by [24].

3.5 Cloud Standard-Based Approaches

Standardization initiatives have emerged including [2]: USDL (*Unified Service Description Language*) for the (SaaS) Software as a Service which provide a generic description of the business and service-level components to make them exchangeable and consumable in a Cloud environment. TOSCA (*Topology and Orchestration Specification for Cloud Applications*) for (PaaS) Platform as a Service which describe how to define the components of an application and their relationships as a graph topology. Using management plans, these components can be deployed, configured, and managed at a Cloud provider. OCCI (*Open Cloud Computing Interface*) dedicated for the (IaaS) Infrastructure as a Service and provide consistent access to existing IaaS resources. CIMI (*Cloud Infrastructure Management Interface*) looks like OCCI and covers main features of IaaS (deployment and management of virtual machines and other artifacts such as volumes, networks, or monitoring. CDMI (*Cloud Data Management Interface*) that defines the interface to manage Cloud elements from storage.

[25] outlines the integration of configuration management with model-driven Cloud management realized through TOSCA and Chef which is a well-known configuration management tool. The mappings between TOSCA and OCCI are defined in [14] to build a model driven Cloud-provider agnostic Cloud orchestrator leveraging both of them for portable application and resource provisioning and deployment. [25] combines TOSCA and Chef, a popular configuration management tool to ensure portability when migrating services from one Cloud environment to another.

The Model-Driven Configuration Management of Cloud Applications with OCCI (MoDMaCAO) framework is developed by [6] to handle the issues of the current version of OCCI in terms of configuration management of Cloud applications. In this framework, the capabilities of the OCCIware tool chain are used to implement disparate configuration management tools. The proposed framework is illustrated by modeling, deploying and managing four different distributed Cloud applications and generating prototypical implementations for three configuration management tools. In overall, MoDMaCAO is claimed to be an extension of OCCI among others enumerated in [25] where semantic interoperability is achieved in multiclouds through FCLOUDS. This framework can support the Cloud consumer to migrate from one API to another. The illustrative example shows a migration from an OCCI Infrastructure VM (Virtual Machine) to Google Cloud Platform VM.

3.6 Software Product Line-Based Approaches

Both PLE (Product Line Engineering) and MDA are combined in several works to perform model based Cloud migration with variability modelling and analysis. PLE supports this by offering choice flexibility in the targeted Cloud environments.

[27] presents a model-driven approach to manage, instantiate configurations and deploy images for Virtual Machine provisioning in Cloud computing. Virtual images are represented as product lines with feature model configurations. Their deployment is automated by model-based techniques in a flexible way. This approach is applied in [28] to minimize the amount of not required software installed in VMIs. This allows to reduce the power consumption of VMI provisioning and the data transfer via network. The approach of [29] aims to find the best Cloud IaaS configuration required to migrate or to optimize the choice for internal systems migration to a Cloud environment. Since authors perceive this need as a variability management and analysis concerns, they are based on FM to design the IaaS variability of the different *Amazon Web Service*[1] virtual machine offerings. Authors claim that their proposal can be generalized to providers other than AWS, like Rackspace/OpenStack or Microsoft.

Table 1. Comparison between the proposed approaches

Approaches	Portability		Techniques	Degree of automation	Type of migration		Steps
	DL	BL			LA	C2C	
[12]	–	–	Cloud patterns	Manuel	+	–	Migration execution
MOSAIC (Cloud agency) [15]	–	+	MAS	Semi-automatic	+	+	Migration strategy-monitoring & maintenance
[15]	–	+	Cloud patterns	Automatic	–	+	Migration execution
[23, 30]	–	+	MDA	Manuel	–	+	Migration execution
Mosaic ontology methodology and framework [17]	–	+	Semantic modeling	Semi-automatic	–	+	Migration execution
MODAClouds [3]	–	+	MDA	Semi-automatic	+	+	Migration strategy-Migration execution-monitoring & maintenance

(*continued*)

[1] http://aws.amazon.com/products/.

Table 1. (*continued*)

Approaches	Portability		Techniques	Degree of automation	Type of migration		Steps
	DL	BL			LA	C2C	
Remics [18]	–	–	MDA	Semi-automatic	+	–	Migration execution
StratusML [7]	–	+	MDA	Semi-automatic	–	+	Migration execution
CloudML [22]	–	+	MDA	Semi-automatic	–	+	Migration execution
MLSAC [41]	+	+	MDA	Semi-automatic	+	–	Migration execution

Azure. However, they counted no less than 1758 possible configurations just for an Amazon instance without considering middleware layer options. In addition, they proceed by matching the searched configurations against all enumerated offered configurations which really time consumable.

In [23, 30], the focus is on facility of migration to multiple Clouds. Indeed, FM are used to conduct the migration planning process considering SLA constraints like VM properties (size of RAM, No of processors, CPU processing speed etc.). In this work, the nodes of a Feature model represent the application components or features, whose edges define the relations between these application components. A model-driven multi-cloud configuration approach is developed by [31]. The implemented framework which is named SALOON combines feature model and ontologies to explicit infrastructure and platform variability, and give semantic descriptions of Cloud applications. SALOON involves four PaaS providers and the authors affirm that it can cover provider models that match the metamodel they design. Nonetheless, this may be not easy since this framework has not standard interfaces or formal specifications.

The above mentioned approaches discuss different aspects. For example, some approaches have focused on SLA negotiation, vendor selection. Other approaches focused on the migration of legacy application. To make a comparison between the different approaches, we choose the following criteria:

- The portability of Data Layer (DL): is the ability to easily transfer data into and out of the Cloud service environment.
- The portability of Business Layer (BL): is the ability to easily transfer an application or application components between Cloud providers.
- Techniques: define the techniques used in the approach such as MAS or MDA.
- Degree of automation: define the degree of automation of the approach (automatic, semi-automatic and manual).
- Type of migration: There are two categories:

  - Migration of legacy application (LA): approaches that address the migration of legacy applications.
  - Migration from a Cloud to Cloud (C2C): approaches that address the migration of application or application components from Cloud provider to another.

- Steps: define in which steps of the migration process the approach works.

In the Table 1, the majority of approaches involve legacy application migration. The only automatic approach is [15] which only processes the code2code transformation. The unique contribution which present an evaluation process is proposed in [41] to get a feedback from users via reusing through MLSAC metamodel method fragments to ensure the quality of the reengineering process. However, Security is not well considered in the migration process. Data portability issue is still ambiguous and the proposed approaches do not give a clear solution, excluding [41] where data confidentiality and sensitivity requirements are present for simple data structure. It is essential to concentrate on data portability. There are a variety of data storage models of Cloud providers. Each Cloud provider supports its own set of data types.

4 Cloud Migration Approach

In this section, we propose a new Cloud migration approach to define the different steps of the migration process. It contains the steps to follow once we decide to migrate an application to or between Cloud providers. Our proposal, also, addresses the steps that follow the moving of the application to the Cloud as the maintenance and the monitoring. It covers the case when the customer wishes to migrate application again to another provider. In all steps of migration process, it is necessary to consider the security aspect.

a. **Migration strategy**

This step is composed of a set of blocks such that its functioning is based on MAS:

- Negotiation block: for the negotiation of SLA (Service Level Agreement)
- Discover services block: to search for the most suitable Cloud service when the user want to replace component(s) with Cloud offering.
- Discover providers block: to find a list of Cloud providers in the market that offers the desired services with the cost estimation.
- Service composition block: to select the best structure of the set of services that build the application and define the services which are interoperable.
- Monitoring reports block: this block is executed from the second iteration of component migration. This block defines the problems encountered and the behavior of service during its execution.

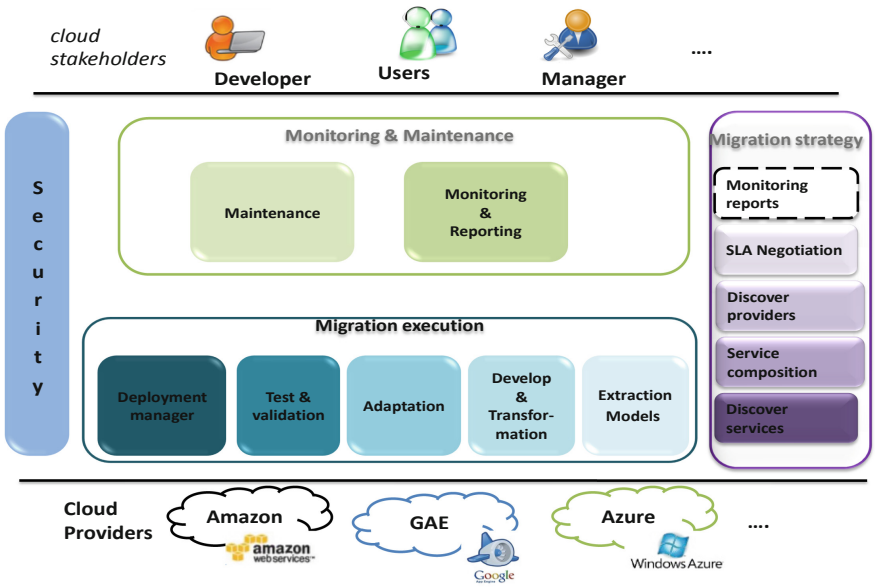

Fig. 1. Comprehensive cloud migration approach architecture

b. **Migration execution**

This step is composed of a set of blocks such that its functioning is based on Model Driven Architecture:

- Extraction Model block: extract the model from a piece of code or from a database.
- Develop & transformations block: apply the necessary operations (development, transformation…) in order to obtain the new version of code or data.
- Adaptation block: define the necessary changes to the rest of the application and apply them to ensure interoperability between the new component and the rest of the application.
- Test & validation block: test the application with the new version of the migrated component and validate it.
- Deployment manager block: deploy the migrated component or data in the new Cloud platform.

c. **Monitoring & maintenance**

This step is composed of two blocks:

- Monitoring and reporting block: this block is dedicated to monitor the functioning of the application and generating performance reports.
- Maintenance block: this block is responsible for the maintenance of the application and apply the necessary updates.

5 Research Implication

Prior surveys have been devoted to identify taxonomies, examine approaches and reviews tools of Cloud migration. [32] gives a survey about the Cloud architectures and Cloud decision making approaches and frameworks. This work address the steps, key concerns of a Cloud migration decision framework and the key decisions that the system should maintain. Similarly, authors of [33] attempt to distill Cloud migration approaches, identify their common characteristics and their empirical findings. This survey allows to show a global view of the state of the art from the methodological point of view of Cloud migration comprising key concerns, activities, and criteria to be considered. [34] identifies a set of relevant process elements included in the moving legacy systems from on-premise to Cloud platforms. The presented legacy-to-Cloud process model aims to clarify how organizational legacy systems can be transformed by Cloud services. The comparative study between Cloud migration methods given by [35] concentrate especially on those based on MDE (Model Driven Engineering) and takes models in the heart of the software engineering process. It identifies the key factors to evaluate the application in terms of technical and non-technical considerations in pre-migration, migration and post-migration phases. In a very recent survey [36], these phases are discussed from the scope of processes of migration to open source software in the world but without considering Cloud particularities.

Our work highlights a comprehensive taxonomy of Cloud migrations proposals including SPL based approches which offer the customizability of the Cloud migration decisions and the configurability of components. As well, we can argue that Cloud standard based approaches are interesting to promise interoperability and portability across multiple Clouds.

Also, we consider the survey of [37] dealing with security concerns for live virtual machine migration. It confirms the lack of comprehensive approaches for task migrations in mobile Cloud computing aligned with large scale deployment environment. [38] compares different industrial solutions proposed by Cloud vendors while [39] reports a systematic literature review about both academic and industrial tools for different migration types.

[40] scrutinizes aspects of Cloud computing that leads to vendor lock-in and proposed solutions for this issue. Moreover, the literature on proprietary lock-in risks in Cloud computing is explored in terms of its principal causes, consequences, mitigations strategies, and encountered challenges by company stakeholders on their Cloud migration exercises. The proposed taxonomy of Cloud lock-in perspectives is performed at the light of reports about real experiences on migration allowing a good knowledge concerning all Cloud SaaS migration issues. Moreover, authors describe measures to enhance the portability, security and interoperability of Cloud (and on-premise) applications in hybrid environments. In our current survey, we also recognize the vendor lock-in challenge and recapitulate proposed solutions to avoid this pitfall. Being agree with [37] about the lack of migration approaches for mobile Cloud computing, our proposal can be easily extended to ambient systems while benefiting from our previous research. Furthermore, we endeavor to align our proposal with variability concerns and Cloud standards.

6 Conclusion

Cloud migration is an interesting topic. Several approaches have been proposed, but there are still other challenges that have not been well treated. There are several obstacles for an easy and successful migration as the Lack of standardization and the choosing of the best migration strategy. In this context, we propose a new Cloud migration approach to describe the different functionality in the migration process, the first step of this process Migration strategy based on MAS techniques and the second step Migration execution based on MDA technologies. The proposed approach is under development. Our approach proposes to automate some features which saves time and effort, and ensures services and data portability between Cloud providers.

References

1. Ferrer, A.J., Pérez, D.G., González, R.S.: Multi-cloud platform-as-a-service model, functionalities and approaches. Procedia Comput. Sci. **97**, 63–72 (2016)
2. Stravoskoufos, K., Preventis, A., Sotiriadis, S., Petrakis, E.G.: A survey on approaches for interoperability and portability of cloud computing services. In: CLOSER, pp. 112–117, April 2014
3. Cretella, G., Di Martino, B.: An overview of approaches for the migration of applications to the cloud. In: Caporarello, L., Di Martino, B., Martinez, M. (eds.) Smart Organizations and Smart Artifacts. LNISO, vol. 7, pp. 67–75. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07040-7_8
4. Zhao, J.F., Zhou, J.T.: Strategies and methods for cloud migration. Int. J. Autom. Comput. **11**(2), 143–152 (2014)
5. Beserra, P.V., Camara, A., Ximenes, R., Albuquerque, A.B., Mendonça, N.C.: Cloudstep: a step-by-step decision process to support legacy application migration to the cloud. In: 2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), pp. 7–16, September 2012
6. Muñoz, V.M., Ferguson, D., Helfert, M., Pahl, C. (eds.): CLOSER 2018. CCIS, vol. 1073. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29193-8
7. Hamdaqa, M., Tahvildari, L.: Stratus ML: a layered cloud modeling framework. In: 2015 IEEE International Conference on Cloud Engineering (IC2E), pp. 96–105. IEEE, March 2015
8. Hajlaoui, J.E., Omri, M.N., Benslimane, D.: Multi-tenancy aware configurable service discovery approach in cloud computing. In: 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 232–237. IEEE, June 2017
9. Jumagaliyev, A., Whittle, J.N.D., Elkhatib, Y.S.S.A.: Evolving multi-tenant SaaS cloud applications using model-driven engineering (2016)
10. Rai, R., Sahoo, G., Mehfuz, S.: Advancements and approaches towards moving from legacy application to cloud. Int. J. Commun. Netw. Distrib. Syst. **16**(2), 114–139 (2016)
11. Dombrowski, S., Ermakova, T., Fabian, B.: Graph-based analysis of cloud connectivity at the internet protocol level. Int. J. Commun. Netw. Distrib. Syst (2018)
12. Andrikopoulos, V., Binz, T., Leymann, F., Strauch, S.: How to adapt applications for the cloud environment. Computing **95**(6), 493–535 (2013)
13. Pahl, C., Xiong, H., Walshe, R.: A comparison of on-premise to cloud migration approaches. In: European Conference on Service-Oriented and Cloud Computing, pp. 212–226. Springer, Berlin, Heidelberg, September 2013

14. Glaser, F., Erbel, J.M., Grabowski, J.: Model driven cloud orchestration by combining TOSCA and OCCI. In: 7th International Conference on Cloud Computing and Services Science (CLOSER), pp. 644–650 (2017)
15. Cai, Z., Zhao, L., Wang, X., Yang, X., Qin, J., Yin, K.: A pattern-based code transformation approach for cloud application migration. In: 2015 IEEE 8th International Conference on Cloud Computing (CLOUD), pp. 33–40. IEEE, June 2015
16. Di Martino, B., Cretella, G., Esposito, A.: Methodologies for cloud portability and interoperability. In: Cloud Portability and Interoperability, pp. 15–44. Springer, Cham (2015)
17. Moscato, F., Di Martino, B., Aversa, R.: Enabling model driven engineering of cloud services by using mosaic ontology. Scalable Comput. Pract. Exp. **13**(1), 29–44 (2012)
18. Sadovykh, A., Srirama, S., Jakovits, P., Smialek, M., Nowakowski, W., Ferry, N., Morin, B.: Deliverable D4. 5 REMICS Migrate Principles and Methods (2010)
19. Sabiri, K., Benabbou, F., Khammal, A.: Model driven modernization and cloud migration framework with smart use case. In: Proceedings of the Mediterranean Symposium on Smart City Applications, pp. 17–27. Springer, Cham, October 2017
20. Ta'id, H.: Facilitating migration of cloud infrastructure services: a model-based approach. In: CloudMDE@ MoDELS, pp. 7–12 (2015)
21. Alili, H., Drira, R., Ghezala, H.H.B.: Model driven framework for the configuration and the deployment of applications in the cloud. Cloud Comput. **2016**, 73 (2016)
22. Ferry, N., Rossini, A., Chauvel, F., Morin, B., Solberg, A.: Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In: Cloud Computing (CLOUD), pp. 887–894. IEEE, June 2013
23. Vijaya, A., Neelanarayanan, V.: A model driven framework for portable cloud services: proof of concept implementation. Int. J. Educ. Manag. Eng **4**, 27–35 (2015)
24. Bergmayr, A., et al.: A systematic review of cloud modeling languages. ACM Comput. Surv. (CSUR) **51**(1), 22 (2018)
25. Wettinger, J., et al.: Integrating configuration management with model-driven cloud management based on TOSCA. In: CLOSER, pp. 437–446, May 2013
26. Challita, S., Zalila, F., Merle, P.: Specifying semantic interoperability between heterogeneous cloud resources with the FCLOUDS formal language. In: CLOUD 2018–11th IEEE International Conference on Cloud Computing, July 2018
27. Le Nhan, T., Sunyé, G., Jézéquel, J.M.: A model-driven approach for virtual machine image provisioning in cloud computing. In: European Conference on Service-Oriented and Cloud Computing, pp. 107–121. Springer, Berlin, Heidelberg, September 2012
28. Le Nhan, T., Sunyé, G., Jézéquel, J.M.: A model-based approach for optimizing power consumption of IaaS. In: Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on 2012 Dec 3, pp. 31–39. IEEE (2012)
29. Ivanov, I.I., van Sinderen, M., Leymann, F., Shan, T. (eds.): CLOSER 2012. CCIS, vol. 367. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-04519-1
30. Vijaya, A., Dash, P., Neelanarayanan, V.: Migration of enterprise software applications to multiple clouds: a feature based approach. Lecture Notes on Software Engineering **3**(2), 101 (2015)
31. Quinton, C., Romero, D., Duchien, L.: SALOON: a platform for selecting and configuring cloud environments. Softw. Pract. Exp. **46**(1), 55–78 (2016)
32. Wan, Z., Wang, P.: A survey and taxonomy of cloud migration. In: 2014 International Conference on Service Sciences (ICSS), pp. 175–180. IEEE, May 2014
33. Gholami, M.F., Daneshgar, F., Low, G., Beydoun, G.: Cloud migration process—a survey, evaluation framework, and open challenges. J. Syst. Softw. **120**, 31–69 (2016)
34. Gholami, M.F., Daneshgar, F., Beydoun, G., Rabhi, F.: Key challenges during legacy software system migration to cloud computing platforms—an empirical study (2017)

35. Akodadi, K.: A survey of cloud migration methods: a comparison and proposition
36. Sobhani, R., Seifzadeh, H., Gandomani, T.J.: A review of migration processes to open source software. Int. J. Open Source Softw. Process. (IJOSSP) **9**(1), 20–31 (2018)
37. Balobaid, A., Debnath, D.: An empirical study of different cloud migration techniques. In: 2017 IEEE International Conference on Smart Cloud (SmartCloud), pp. 60–65. IEEE, November 2017
38. Balobaid, A., Debnath, D.: Cloud migration tools: overview and comparison. In: World Congress on Services, pp. 93–106. Springer, Cham, June 2018
39. Da Silva Filho, H.C., de Figueiredo Carneiro, G., Costa, E.S.M., Monteiro, M.: Tools to support SMEs to migrate to the cloud: opportunities and challenges. In: Information Technology-New Generations, pp. 159–165. Springer, Cham (2018)
40. Opara-Martins, J.: Taxonomy of Cloud Lock-in Challenges (2018)
41. Fahmideh, M., Grundy, J., Beydoun, G., Zowghi, D., Susilo, W., Mougouei, D.: A model-driven approach to reengineering processes in cloud computing. Inf. Softw. Technol. **144**, 106795 (2022)
42. Khemili, W., Hajlaoui, J.E., Omri, M.N.: Energy aware fuzzy approach for placement and consolidation in cloud data centers. J. Parallel Distrib. Comput. **161**, 130–142 (2022)
43. Mokni, M., Yassa, S., Hajlaoui, J.E., Chelouah, R., Omri, M.N.: Cooperative agents-based approach for workflow scheduling on fog-cloud computing. J. Ambient Intell. Human. Comput. 1-20 (2021)
44. Trifa, Z., Hajlaoui, J.E., Khemakhem, M.: Pollution attacks identification in structured P2P overlay networks. In: International Conference on Information and Communications Security, pp. 674–686. Springer, Cham, December 2017