We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Cloud Application Portability: Issues and Developments

Isaac Odun-Ayo, Chinonso Okereke and
Hope Orovwode

Additional information is available at the end of the chapter

Abstract

Cloud computing is a standard that is fast gaining momentum in the IT world. The availability of storage capacity that can be accessed and increased as the need arises makes computing easier. Applications can also be deployed using services provided by a cloud service provider. Portability allows utilization of applications and services across various domains. Portability could be in the area of programming language, application programming interface, data storage or data migration. Clearly, the easier it is to move services across various providers, the more attractive cloud computing becomes. The study was executed by means of review of some literature available on cloud application portability. This chapter examines current trends in cloud application portability area and gives focus for future research. In the present work, the objective is to answer the following question: what is the current trend and development in cloud application portability? Papers published in journals, conferences, white papers, and reputable magazines were analyzed. Some core topic facets were used in this review for the identification of trends in cloud application portability. The finding is that discussions on virtualization and API-specific issues are not adequate. This will be of benefit to prospective cloud users and even cloud providers.

Keywords: cloud computing, Internet of Things, portability, data storage, APIs

1. Introduction

According to [1], cloud computing is a paradigm for enabling universal, easy-to-use, on-demand network access to a shared pool of configurable computing resources such as networks,

storage, servers, applications, and services, which can be quickly provided and released with very little management effort. Cloud service portability can be well-defined as the capacity to move cloud services from one provider to another in an easy manner. Cloud computing can also be described as "a large scale distributed computing model motivated by frugalities of scale in which a pool of abstracted, virtualized, dynamically scalable managed storage, computing power, services, and platforms are distributed on demand to external customers over the Internet" [2]. Important features of cloud computing are scalability, on-demand service, and virtualization. The virtualized resources are provided through an Application Programming Interface (API) [3]. Cloud computing offers three major services, namely, Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). Cloud deployment types are public, private, community, and hybrid clouds. Enterprises through cloud computing are able to focus on core activities and training instead of spending time and money on infrastructure.

Procurement of infrastructure is reduced or eliminated; it also implies little or no maintenance cost on infrastructure. On-demand benefit of the cloud ensures that enterprises utilize what they need per time, using state of the art technology. Importantly, only a resource utilized is paid for. In SaaS, a software that is owned, delivered, and managed remotely by one or more cloud service providers is offered on a pay-per-use manner [3]. SaaS usually is the most noticeable layer of cloud computing for cloud users. This is because it is majorly about actual software applications that are used and accessed on the cloud. IaaS refers to computing services like processing or storage that can be gotten as a service. The process of virtualization is the primary means of providing IaaS services. The PaaS referred to an abstraction layer between the IaaS and SaaS. PaaS is concerned with software development. Users develop their applications based on a platform and upload the code to the platform provided by the CSP. The user does not need to worry about infrastructure and the CSP manages the scaling of the applications. Applications deployed on the cloud including data are difficult to move between clouds, leading to portability issues. Vendor lock-in occurs when a cloud user cannot move data and application from a particular cloud service provider [4]. The lock-in could be termed vertical or horizontal heterogeneity [5]. Vertical heterogeneity occurs with IaaS that provides infrastructure to consumers. PaaS allows development and deployment of user's applications. Such a platform uses specialized libraries and routines and moving application to another platform will require extensive changes [5]. In view of this, cloud users are forced to remain with a particular CSP and porting becomes very expensive [5]. Portability will therefore mean that cloud users will write codes that work in different clouds irrespective of the nature of the clouds. However, this should not be strictly so. Consumers should be able to change CSPs and move to any CSP that offers a better service at possibly a lower cost. Unfortunately, different cloud application platform offerings are categorized by considerable heterogeneity [6]. In view of these irreconcilabilities, users who deploy an application to a specific platform may find it difficult moving such an application to another CSP.

The focus of this chapter is to scrutinize data and application portability in cloud computing. The issues involved in application portability are discussed. Current industry trends are also be examined. This chapter contributes to enhancing developments in the area of cloud application

portability. The rest of the chapter is structured as follows: Section 2 examines related work. Section 3 discusses issues related to cloud portability. Section 4 highlights the current development in cloud portability with reference to developments in the industry. Section 5 concludes the chapter and suggests the future area of research.

2. Related work

In [5], cloud application portability: an initial view is presented. The focus is on examining the heterogeneous nature of cloud platforms. Approaches to tackling the identified challenges were proposed through a cross-platform application process. In [4], the need for portability and interoperability in cloud computing is proposed. The approach is an open source model for enhancing portability and interoperability in the various cloud platforms. This will eventually reduce the cost of cloud migration. In [7], semantics-centric solutions for application and data portability in cloud computing is presented. The focus is on the problem of data and application portability in the cloud. The semantic web community proposed some solutions to overcome some aspects of cloud portability. In [6], portable cloud services using Topology and Orchestration Specification for Cloud Applications (TOSCA) is proposed. The focus is on the possibility of moving services between cloud providers to avoid vendor lock-in. This was done at the managerial and operational level using TOSCA. In [8], a portable cloud application—from theory to practice—is presented. The approach was to design an open source application capable of handling multiple cloud usage. This helps to deal with the issues of application, data, and service portability on the cloud.

In [9], a comparative analysis of legislation on how to attribute the right to data portability in Europe is presented. It discussed how competition is prevented by cloud service providers through lock-in. Thereafter, it examines legislation that makes portability possible for cloud consumers. In [10], a new cloud services portability platform is proposed. The approach is to design a cloud portability framework using an adapter model. This approach leverages on TOSCA for application portability. In [11], application mobility in pervasive computing: a survey is presented. The focus is to examine different migration methods for application portability. A framework was proposed along four dimensions to allow easy mobility of applications. In [12], challenges emerging from future cloud application scenarios are presented. Various aspects of cloud computing were examined. The issues of integration and interoperability were also discussed and suggestions made on the way forward. In [13], enabling portability in advanced information-centric services over structured peer-to-peer systems is proposed. The approach is a three-layer architecture to enhance portability. The middleware in the framework is versatile over a wide variety of applications. In [14], streamlining DevOps automation for cloud application using TOSCA as standardized metamodel is presented. The chapter proposes a framework for integrating different applications on the cloud. The framework is also implemented and validated. In [15], toward application portability in platform as a service is presented. The approach was to discuss portability options in terms of PaaS provider and the ecosystem capabilities. The proposed model was implemented with a data set and different PaaS vendors.

3. Cloud portability issues

3.1. Layers of platform as a service

"Application portability is the capability of program to be implemented on numerous types of data processing systems without changing the program to a dissimilar language and with little or no modification" [4]. Portability is the capacity to use mechanisms or system lying on several software or hardware environments [8]. The primary focus of portability is in the PaaS. PaaS is divided into three layers namely, infrastructure, platform, and management. Since PaaS hides most of its physical properties, users are left with concepts like dynos, workers' units, gears, or app cells that can be used for specific instances in PaaS, while CPU and disk utilization are negligible [4]. The CPU usage is the main factor for billing in IaaS, but instances count and RAM size are used in PaaS. Geographical location of application deployment is another issue. Based on latency considerations, applications are also subject to regional regulations [4]. The platform constitutes the hosting environment with two stack components: the runtime stack and service stack [4].

The runtime stack in PaaS constitutes the programming language used for developing the application such as Ruby on Rails which is fairly popular [4]. Higher stack leads to the more specificity in the application dependency; hence increasing lock-in risk. The service stack is divided into add-one and native and add–on services. The latter services, which are native services, are hosted by PaaS vendors, and it includes latency and performance services like data store [4]. While Add-on service is provided by third parties: they include data store and services like analytics, search engines, and messaging services. Another aspect is extensibility involving build packs that enable developments add own packages of service. The management layers allow control of the deployed application which encompasses pushing, starting and stopping an application, including the provision of native and add-on services. Importantly, resource usage and monitoring necessary for billings and scaling decision is carried out at the management layer. Even though the mentioned functionalities are collectively used by all various PaaS to a large extent, commands and procedures are not standardized and vary broadly between providers [4].

3.2. Need, levels and advantages of application portability

Cloud portability guarantees that applications, data, or services work in the same way in different cloud services with a common programmable interaction [8]. A primary reason for portability is migrating all or part of existing services between clouds. This could be because of the need for optimal utilization, to reduce cost, technology, or SLA issues. Most challenging are situations in which a user's cloud application is distributed between two or more providers and administrative domain simultaneously [9]. Portability can be categorized into three ways: functional portability, data portability, and service enhancement. Functional portability is realized by describing the application's functionality details in a vendor–agnostic manner [8]. Device agnosticism allows hardware and software components to work with various systems without requiring any special adaptations. Hence, there is compatibility across most

common systems. Data portability is reached when a customer is able to access and salvage application data from one provider and to input this into a corresponding application hosted by another provider. Service enhancements metadata is added through annotation [8]. The metadata provides information about other data. Also, control APIs permit infrastructure to be added and reconfigured instantaneously, based on traffic and other factors [9]. The requirement for portability at PaaS focuses on lessening the amount of rewriting when application is ported. Data portability permits users to transfer their personal information from one service to another, choosing a service suitable to their needs [10]. Another advantage of porting is that it allows a user to be released from lock-in to a provider. Ironically, data portability is also in the interest of the CSP. This is because when users are allowed to freely move data, trust will be enhanced between CSPs and users. In addition, more information can be used by CSPs in the area of advertising.

3.3. Areas of challenge in application portability

There are four areas of concern in application portability, namely programming language and framework, platform-specific services, data store, and platform-specific configuration files [11]. The specific programming language that is used to build an application is a major determining factor in cross-platform deployment. All cloud platforms have certain languages and framework which they support. For instance, Google AppEngine makes available support to Java, but does not extend such support to Java class libraries which are supported by OpenShift [11]. Cloud platform provides services through specific APIs. A service is a high–level functionality that the provider can use with no need for from-scratch implementation [6]. Analytical tools that are used in data sets and APIs for image manipulation are examples in this area. Reduction of application development time can be achieved when developers use such platform services. Developers can join in functionalities from platform services by binding it to the respective platform APIs instead of programming those functionalities from scratch [8].

Two types of data storage exist on the cloud platform. These are: data base store and file store. Data base store is for strong structured data, while file store is like a hard disk on the cloud. There are two kinds of data base store: SQL and NoSQL. The SQL database is the traditional relational database available on all major cloud providers. The NoSQL is a new system that groups all database systems that do not adhere to the SQL relational structure. NoSQL has simple operations, faster access, and the ability to distribute data over many services among other characteristics [6].

Configuration files are used by traditional software applications to instruct the environment on how to execute on the application. Platform-specific configuration files also exist on the cloud. Google AppEngine, for instance, uses the "app engine – web xml" file. Adapting the configuration file to each target cloud platform affects application portability [8]. Solution to cloud portability issues are through standardization and intermediation. Standardization indicates the definition of common set of standards for PaaS offerings. The adaption of such standards by all CSPs will enable developers design, deploy, and manage their applications independent of a cloud platform.

The Distributed Management Task Force (DMTF) evolved the open virtualization format (OVF) that enables standardization of virtual machines' formats and portability. The Storage Network Industry Associations (SNIA) evolved the Cloud Data Management Interface (CDMI) to standardize the issue of cloud storage services [5]. The Topology and Orchestration Specification for Cloud Applications (TOSCA) is supported by Organization for the Advancement of Structured Information Standards (OASIS) for standardizing applications to enable portability across platforms [5]. Intermediation frees application development from platform-specific APIs and supported formats. jCloud and mOSAIC are open source platforms meant to resolve portability issues based on platform-specific APIs [5]. The issue of data storage is also being addressed by the Bridge Query Language (BQL).

4. Industry perspective in cloud portability

As already discussed, several organizations have emerged to deal with the issue of portability. To move workloads between clouds, there is a need for standardized programming interfaces, layers of abstraction, and management capabilities [4]. Making programming interfaces standard allows movement of programs between interfaces. These standards allow companies to leverage on toolsets when preparing IT tasks for deployment on a cloud service [4]. Portability between cloud resources is also made more feasible by insulation of cloud services from underlying infrastructure through layer of abstraction. Layer of abstraction also increases speed of switching cloud services as well as decreases the cost and stress of such activity because of a reduction in the dependency on underlying infrastructure like processors, operating system and virtualization software [4]. Management capability is necessary to enable IT personnel handle the cloud services to which they subscribe without the introduction of new layers of capability. For success, these management capabilities must work well with already existing management products, both for managing virtual and physical services.

4.1. Support for multiple hypervisor technology

Multiple hypervisor technology such as VMware, Hyper-V, KVM Ven should be supported appropriately [4]. The virtual machine has become an essential unit of work and encapsulation especially for IaaS and some PaaS clouds [4]. Support for multiple hypervisor technology should enable customers choose their preferred VM and quickly transfer to it irrespective of the underlying platform. This is because there is likely to be mixed virtualization environment in the enterprise of a customer.

4.2. Choice of operating system

The coming of application virtualization continues to break the ties between the operating system and the application but a large number of applications are still dependent on specific versions of an operating system [4]. Enterprises have built several types of operating systems in their data centers overtime; which implies that it is impossible to limit users of the cloud to only a particular type of OS (Operating System) and enterprises would not be able to attend to large number of workloads.

4.3. Programming models

Data centers were chosen by most enterprises a long time ago and some of these programming models may not be right for the cloud. IT staff could also show reluctance to change these programming models because of their familiarity with them [4]. Cloud providers must utilize current programming models such as. Net and Java, and web models like PHP and Ruby. When clouds do not support these languages, more importance is placed on hybrid cloud management.

4.4. Application programming interface

PaaS cloud is simply programming to a given cloud platform. PaaS has a lot of functionalities such as massive scalability. On the other hand, APIs are useful only to new applications. There is a need for the modification of existing applications to benefit from these APIs. This modification is usually difficult and expensive procedure [4]. In effect, customers must examine the complete API set in such a way as to be able to migrate and also move out later if the need arises. Although cloud service providers are excited about migrating customers to their platform and usually provide tools for this purpose, it is sometimes difficult for customers to move their applications from cloud to cloud [4].

4.5. Unified management

Dedicated management console is a vital part of every cloud technology including public clouds. This management must aim for more unification. This is especially because of the need for tighter coordination between on-premises and off-premises resources in a hybrid cloud [4]. API and management data accessibility would allow unified cloud console to support access control, scheduling, resource management, and billing.

4.6. Data portability

The data on application must also be portable, not just in terms of copying files from one location to another. Data can exist in the form of database or file with varying access methods. Data portability means application would be able to access and move production data which they need to operate upon [4]. Enterprise data requires appropriate standards for the support of data export and conversion from one format to another.

4.7. License flexibility

Software licensing has always been bound to location and hardware, and this has not changed with cloud computing. Customers must know of the restriction that software license has on cloud portability [4].

4.8. Topology and orchestration specification for cloud applications

TOSCA defines composite applications and how they are managed using modules and for portability [6]. It characterizes the service template that contains a cloud service topology and

its operations. For example, it defines an application hosted on an server which is hosted on an operating system, and method of deployment, method of termination, and management of the services provided [6]. Typically, managing services requires a lot of manual effort by the customer. Each enterprise learns how to operate an application, acquires management knowledge, and automates certain aspects in the scripts. TOSCA solves this challenge by enabling application development and operators to model management best practice and reoccurring tasks into clear plans [6]. With these plans being por between different environments and providers, reusability and automation of service management is achieved and this reduces the total ownership cost by a significant percentage. Automated service management will also enhance rapid elasticity and self-service. TOSCA plan is portable specifically because of its workflow language and engines it uses [6]. Example of workflow languages portable between dissimilar engines are Business Process Model and Notation (BPMN) and Business Process Expression Language (BPEL).

5. Analysis and discussion

Cloud portability is a topic with huge importance to cloud computing. This section analyses some of the most recent research papers based on some core areas namely virtualization

References	Virtualization issue	Data issues	Applications issues	API issues	Identity management and access control	Security concerns	Interoperability
[17]			x	x			
[18]	x	x	x			x	
[21]	x	x	x	x		x	x
[26]	x		x				x
[27]			x				x
[28]							
[22]	x		x			x	x
[23]	x		x			x	x
[19]	x	x		x		x	x
[24]	x	x	x			x	x
[29]	x		x	x			x
[30]			x	x			x
[16]			x				
[31]			x	x			x
[20]		x		x	x		

Table 1. Comparative analysis of cloud portability issues.

issues, data portability, application/application portability issues, API issues, security concerns, identity management/access control, and interoperability issues. **Table 1** shows the comparative analysis of the core areas in cloud portability. From the analysis of the reviewed papers, application portability issues is seen to be the most discussed as it is mentioned in all reviewed papers except [16]. In contrast to application portability, only 33% of the reviewed papers specifically dealt with data portability issues. This is because application portability is more significant that data portability. Next highly mentioned with 87% among the reviewed papers are interoperability issues. These are because cloud portability and interoperability issues usually go hand in hand. Only [17–20] did not discuss interoperability issues. Virtualization and API-specific issues were averagely discussed with 53% each. However only [16, 18, 21–25], that is, 40% of the reviewed papers, addressed security concerns. The lowest researched area from recently reviewed papers is identity management/access control issues which was only mentioned in [22]. It shows its minimal significance in cloud portability.

Based on the analysis, most of the mentioned core areas were researched based on their perceived significance. However, the suggestion is for more focused research on security concerns in cloud portability-related researches. It is also noted that more research can be carried out to address virtualization and API issues. Identity management/access control issues in cloud portability can be of importance going forward and needs to be focused on a little more.

6. Conclusion

Cloud computing and application portability are vital issues in the cloud. Users are exposed to lock-in because of the inherent challenges of data and application portability. This is usually more pronounced in PaaS cloud, especially in terms of infrastructure and platform. Portability issues in term of APIs, storage, programming languages, and configuration files were discussed. There are organizations creating solutions to allow for standardization and intermediation in cloud portability. Suggestion from the industry was also examined and TOSCA (meant for application portability) was briefly discussed. In view of its relevance to cloud portability, discussions on virtualization and API-specific issues must continue to evolve.

Acknowledgements

Author details

Isaac Odun-Ayo*, Chinonso Okereke and Hope Orovwode

*Address all correspondence to: isaac.odun-ayo@covenantuniversity.edu.ng

Covenant University, Ota, Nigeria

References

[1] Mell P, Grance T. The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. National Institute of Standards and Technology, Information Technology Laboratory. 2011;**145**:7. https://doi.org/10.1136/emj.2010.096966

[2] Foster I, Zhao Y, Raicu I, Lu S. Cloud computing and grid computing 360-degree compared. In: Grid Computing Environments Workshop (GCE'08); 2008. DOI: 10.1109/GCE.2008.4738445

[3] Stanoevska-Slabeva K, Wozniak T. Grid and Cloud Computing: A Business Perspective on Technology and Applications. Heidelberg/Dordrecht/London/New York: Springer; 2010. ISBN: 978-3-642-0 e-ISBN 978-3-642-05193-7. DOI: 10.1007/978-3-642-05193-7

[4] Bozman J. Cloud Computing: The Need for Portability and Interoperability. IDC Executive Insights. www.idc.com/gms; 2010

[5] Gonidis F, Paraskakis I, Simons AJH, Kourtesis D. Cloud application portability: An initial view. BCI'13; September 19-21, 2013. ACM. 978-1-4503-1851-8/13/09

[6] Binz T, Breiter G, Leyman F, Spatzier T. Portable Cloud Services Using TOSCA. IEEE Internet Computing. Vol. 16. no. 3. pp. 80-85. May-June 2012. DOI: 10.1109/MIC.2012.43 http://doi.ieeecomputersociety.org/10.1109/MIC.2012.43

[7] Ranabahu AH, Sheth AP. Semantics centric solutions for application and data portability in cloud computing. In: Proceedings of the IEEE Second International Conference on Cloud Computing Technology and Science; 2010. pp. 234-241. http://corescholar.libraries.wright.edu/knoesis/766

[8] Petcu D, Macariu G, Panica S, Cráciun C. Portable cloud application-from theory to practice. Future Generation Computer Systems. 2012;**29**(2013):1417-1430

[9] Van der Auwermeulen B. How to attribute the right to data portability in Europe: A comparative analysis of legislations. Computer Law & Security Review. 2017;**33**(2017):57-72

[10] Kostoska M, Gusev M, Ristov S. A new cloud services portability platform. In: 24th DAAAM International Symposium on Intelligent Manufacturing and Automation 2013; 2014

[11] Yu P, Ma X, Cao J, Lu J. Application mobility in pervasive computing: A survey. Pervasive and Mobile Computing. 2012, 2013;**9**:2-17

[12] Jeferry K, Kousiouris G, Kyriazis D, Altmann J, Ciuffoletti A, Maglogiannis I, Nes P, Suzic B, Zhao Z. Challenges emerging from future cloud application scenarios. Procedia Computer Science. 2015;**68**(2015):227-237

[13] Pujo-Ahulló J, López PG. Enabling portability in advanced information centric services over structured peer-to-peer systems. Journal of Network and Computer Applications. 2010;**33**(2010):556-568

[14] Wettinger J, Bietentúcher U, Kopp O, Leymann F. Streamlining DevOps automation for cloud application using Tosca as standardized metamodel. Future Generation Computer Systems. 2016;**56**(2016):317-332

[15] Kolb S, Wirtz G. Towards Application Portability in Platform as a Service. In: Proceedings of the 2014 IEEE 8th International Symposium on Service Oriented System Engineering (SOSE '14). Washington, DC, USA: IEEE Computer Society; 2014. pp. 218-229. DOI=http://dx.doi.org/10.1109/SOSE.2014.26

[16] Martino BD, Cretella G, Esposito A. Classification and Positioning of Cloud Definitions and Use Case Scenarios for Portability and Interoperability. Rome: 3rd International Conference on Future Internet of Things and Cloud; 2015. pp. 538-544. DOI: 10.1109/FiCloud.2015.119

[17] Munisso R, Chis AE. CloudMapper: A Model-Based Framework for Portability of Cloud Applications Consuming PaaS Services. 25th Euromicro International Conference on Parallel. St. Petersburg: Distributed and Network-based Processing (PDP); 2017. pp. 132-139. DOI: 10.1109/PDP.2017.94

[18] Pozdniakova O, MaZeika D. A Cloud Software Isolation and Cross-Platform Portability Methods. Open Conference of Electrical, Electronic and Information Sciences (eStream). Vilnius; 2017. pp. 1-6. DOI: 10.1109/eStream.2017.7950315

[19] Martino BD, Cretella G, Esposito A. Advances in Applications Portability and Services Interoperability among Multiple Clouds. In: IEEE Cloud Computing; Vol. 2. no. 2. pp. 22-28. Mar.-Apr. 2015. DOI: 10.1109/MCC.2015.38

[20] Brinkley J, Hoffman D, Tabrizi N. A Social Networking Site Portable Profile System for Blind and Visually Impaired Users Based on Cloud and Semantic Web Technologies. Honolulu, HI: IEEE International Conference on Cognitive Computing (ICCC); 2017. pp. 104-111. DOI: 10.1109/IEEE.ICCC.2017.21

[21] Parameswaran AV, Asheesh C. Cloud Interoperability and Standardization. In: SETLabs Briefings. 2009;**7**(7):19-27

[22] Moravčík M, Segeč P, Papán J, Hrabovský J. Overview of Cloud Computing and Portability Problems. 15th International Conference on Emerging eLearning Technologies and Applications (ICETA). Stary Smokovec; 2017. pp. 1-6. DOI: 10.1109/ICETA.2017.8102511

[23] Antoniades D et al. Enabling Cloud Application Portability. IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC). Limassol; 2015. pp. 354-360. DOI: 10.1109/UCC.2015.56

[24] Scandurra P, Psaila G, Capilla R, Mirandola R. Challenges and Assessment in Migrating IT Legacy Applications to the Cloud. IEEE 9th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Environments (MESOCA). Bremen, Germany: 2015. pp. 7-14. DOI: 10.1109/MESOCA.2015.7328120

[25] Atayero AA, Ilori OA, Adedokun MO. Cloud security and the Internet of Things: Impact on the virtual learning environment. In: Edulearn15 7th International Conference on Education and New Learning Technologies; 2015. pp. 3857-3863

[26] Martino BD, Esposito A, Cretella G. Semantic representation of cloud patterns and services with automated reasoning to support cloud application portability. IEEE Transactions on Cloud Computing. 2017;**5**(4):765-779

[27] Martino BD, Cretella G, Esposito A. Semantic and Agnostic Representation of Cloud Patterns for Cloud Interoperability and Portability. IEEE 5th International Conference on Cloud Computing Technology and Science. Bristol; 2013. pp. 182-187. DOI: 10.1109/CloudCom.2013.123

[28] Yangui S, Glitho RH, Wette C. Approaches to end-user applications portability in the cloud: A survey. IEEE Communications Magazine. 2016;**54**:138-145

[29] Markoska E, Kostoska M, Ristov S, Gusev M. Using P-TOSCA to prevent vendor lock-in for cloud-based laboratories. 23rd Telecommunications Forum Telfor (TELFOR). Belgrade; 2015. pp. 982-985. DOI: 10.1109/TELFOR.2015.7377629

[30] Kolb S, Röck C. Unified Cloud Application Management. IEEE World Congress on Services (SERVICES). San Francisco, CA; 2016. pp. 1-8. DOI: 10.1109/SERVICES.2016.7

[31] Markoska E, Chorbev I, Ristov S, Gusev M. Cloud Portability Standardization Overview. 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Opatija; 2015. pp. 286-291. DOI: 10.1109/MIPRO.2015.7160281