International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Application Portability: A Necessity

Suryanarayan Sudersanam†, Shon Bangale†*, GVA Sashank† and Srija Ganguly†

†Department of Computer Engineering, MPSTME of NMIMS University, JVPD Scheme, Vile Parle West, Mumbai -400056, India

Accepted 16 June 2016, Available online 22 June 2016, **Vol.6, No.3 (June 2016)**

Abstract

Cloud computing is a growing field. People have started relying upon this technology a lot more nowadays, yet some issues still remain which become hindrances to the development of this sphere. One such issue is the difficulty of portability of applications in clouds. There are various needs of portability of applications including prevention of vendor lockin, flexibility in services desired, more reliability on clouds, prevention of data loss and increasing the scope of the application in terms of usage and environment. Portability is the capacity of an application to run effectively on different systems without having to change its configurations. Our motive is to achieve the same in case of applications supported by Cloud Computing.

Keywords: *Cloud Computing, PaaS, API, Platforms, Standardization, intermediation.*

1. Introduction

1.1 What is CLOUD?

The word cloud (Communities and Libraries Online Union Database) substitutes for the statement the Internet, so the phrase Cloud Computing means a type of Internet-based computing, having different aspects such as server authentication, security, storage, data transfer and applications are delivered to an organization's computers and devices through the Internet.

1.2 Working of Cloud Computing

The main aspect of cloud computing is to apply traditional supercomputing, or high-performance computing power, to perform billions of computations per second, in consumer oriented applications such as financial businesses, to deliver information which is personalized , to provide data storage or to power large online computer games that provide an all-round 3D view to the user.

To perform this, virtualization and its techniques are used to maximize and enhance the power of Cloud Computing. Also cloud computing uses large servers and networks of large groups of such servers typically running low cost consumer PC technology with specialized connections to spread data processing capabilities across them. This shared information technology infrastructure contains large pools of systems that are linked together (eric griffith *et al*, 2016).

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (Giuseppina Cretella *et al*,2012).

This sphere has 3 service models and 5 deployment models. Service Model:

- Infrastructure as a Service (IaaS)- The capability provided is to access such resources that helps the consumer deploy his application. The consumer doesn't need to manage the underlying infrastructure but can work with the operating systems and deployment models. (NIST,2011)
- Platform as a Service(PaaS)- The capability provided is to use the resources provided by the provider and deploy the application. (NIST,2011)
- Software as a Service(SaaS)- The capability provided is to use the provider's application running in the Cloud. (NIST,2011) Deployment Model:
- On-demand self-service- No need of human interaction while demanding a service. (NIST,2011)
- Broad network access- Capabilities are available on the network itself.(NIST,2011)
- Resource pooling- The provider's computing resources are pooled to serve multiple consumers.
- Rapid elasticity- Capabilities can be scaled according to requirements. (NIST,2011)
- Measured service- Resource usage can be controlled and monitored providing transparency to both the provider and the consumer. (NIST,2011)

*Corresponding author: **Shon Bangale**

The particular problem statement is The portability of applications in Cloud Computing. Portability is the capacity of an application to run effectively on different systems without having to change its configurations. Our motive is to achieve the same in case of applications supported by Cloud Computing. Portability in this sphere is a problem because of various reasons like:

- Different policies of different cloud vendors which leads to difference in the environments in which the application will run.
- Lack of compatibility due to differences in cloud infrastructure and environment. Hence differences in the source and target clouds in multiple ways.
- Geographical location of the data stores or cloud vendor which leads to differences in vendor policies again.
- Lack of a standard language or protocol that can be followed by all the clouds in order to support the same application to run.

Most importantly, the term portability shouldn't be confused with the term inter-operability which means the ability of two or more systems to exchange information and to use that information effectively.

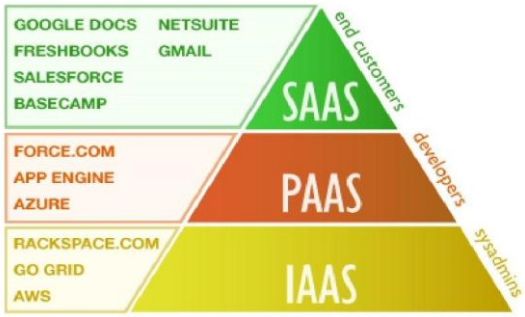

Fig .1 Layers of Cloud Structure (Vangie Beal)

1.3 Some Common Issues

There are various hindrances to Cloud Computing. Some of them are:

- Reliability
- Security
- Costs
- Privacy
- Bandwidth management

The other major ones which basically encompasses the above issues are being discussed below.

Data Leakage

Sensitive data tends to be leaked, or accessed by someone who isn't authorised to do so. This puts many Cloud dependant organisations at a great risk of their data confidentiality and integrity. The prime focus was to firstly detect data leakage through some methods like intelligent data allocation to various agents (Chandni Bhatt *et al*,2014), watermarking etc. Secondly, the prevention of the same was also the focus by studying techniques like CIA mechanism (P. Prasad *et al*, 2011) and BASE64 algorithms. Data allocation talked about how agents guilty of leakage can be pinpointed by strategically planning how data should be distributed amongst them. Watermarking is basically putting an identification of the owner in the data that it belongs to him. CIA mechanism scored the sensitivity of the data based on its Confidemtiality, Integrity and Availabilty and allocated it to a partivular ring of protection. BASE64 algorthm is an extension to the CIA mechanism but with encoding along with usage of JAR files (K Mythili *et al*,2013) .

Key Management

Key Management One of the problems of cloud computing is secure data transfer. One such method to ensure secure data transfer is key management using cryptography. Cryptography poses certain challenges as discussed in papers read before. (Kajal Chachapara *et al*, 2013; Sun Lei *et al*,2010; Amar Ramesh Buchade *et al*,2014).

Different solutions were found out based on how to securely transfer the data between cloud and client through encryption and decryption variations, keying operating variations and certain protocols of cryptographic key management. Some of them are:

- Decryption and encryption in asymmetric keying using the well-defined algorithms (i.e. RSA, AES).In this method the key length was maximised to overcome brute force attack.
- Combining symmetric and asymmetric keying and forming a hybrid key management system. In this key management we will be using two separate keys as in asymmetric/public key management but we wont have to worry about encrypting the public key. Also the key length will be large as in symmetric key management is being used. This offers better efficiency.
- By defining a protocol that can be used by any cloud cryptographic client, ranging from multitenant implementation to cloud storage, it addresses the critical need for a comprehensive key management protocol built into the cloud computing system, which can deploy effective unified key governance for all their encryption.
- The last but not the least solution was to combine all the above solutions and create a service which has the advantages of all the above three solutions.

Data Bottleneck

Data Bottleneck Bottleneck: Is a point in the enterprise where the flow of data is Impaired or, stopped entirely. The main problem of bottleneck arises with Big Data's like terabytes. Petabytes, Zettabytes (Network Bottleneck,Technopedia). Common Causes of Bottle neck:

- A new set of instance or keyword generation at server side increases the storage on server as well as network channel.(Danilo Ardagna,2015)
- Large amount of data (scientific data, social data) to be uploaded is quite often already present on server, constant synchronisation of this data creates multiple instances of data. Creating huge dataflow on channel.(Danilo Ardagna,2015)
- Other than data most of computing at times is also done on the cloud , this computations can usually cause increase the stress on cloud , resulting rise to bottleneck.(Lav R Varshney *et al*, 2014)
- Non optimal compression of data often rises to bottleneck.(Danilo Ardagna,2015)
- Homogenous flow of data often gives rise to more computation in order to maintain sequence which in turn could channels give rise to bottleneck too.

Data Bottleneck in Cloud Computing gives a rise to loss of data packets. This Bottleneck can be present at any part of network not only server or client. The loss of packets causes huge loss in scientific and financial industries. Common solution to these problems are concept of multi-Cloud (Danilo Ardagna,2015). Increasing the efficiency of the channel(Shabnam Sharma *et al*,2013).Bottleneck is a narrow section of channel or a node of network that impedes data traffic flow. A network bottleneck refers to a condition in which data flow is limited by computer or network resources. The flow of data is controlled according to the bandwidth of various system resources.

Bottlenecks are usually caused or formed at following locations Servers, Network infrastructure, Applications at user end.

Authentication

Authentication in Clouds Identifying the users who have the permission to access the cloud is very important. Generally, in a company different users have access to different data, and this data should be kept highly confidential. This is where authentication comes into play. A cloud provider should provide strong and secure authentication measures in their offerings. Many researchers have worked upon this problem and came up with solutions: -

- Multifactor Authentication Systems In this the authentication of user is done based upon multiple factors, like OTP or captcha to increase the level of security (Rohitash Kumar Banyal *et al*, 2013).
- Bio-Metric systems Each individual has different iris/finger print structure and hence it is difficult to duplicate. The paper (Shabnam Sharma *et al*,2013) elaborated on how these can be incorporated in authentication process to increase the security.
- Use of Protocols like Kerberos, Diffie-Hellman combined with key distribution system, public key infrastructure were proposed to authenticate the user accessing the cloud (Shabnam Sharma *et al*,2013; Victor Boyko,2000).

1.4 Overview

The idea is to create a simple application and test it with respect to implementing portability in clouds. Our application is to implement a type convertor. The application converts a particular file in one type (ex-.pdf) into any other type (ex.docx) according to the users choice before downloading. The conversion is supposed to happen in the application itself. The main procedure that we intend to create in the application is:

- The user chooses a file to be converted.
- The application converts the file.
- The converted file is made available to the user.

The major advantage of such kind of application would be that the user wouldnt need to download the file into his computer in order to convert its type. This would not only save storage space in the hardware but also would make the conversion procedure a single step achievement rather than a series of steps to follow. Also the technology of cloud is being promoted more in this case.

The major steps we intend to follow in our action plan are:

- Cloud creation or hiring third party services.
- Coding the application.
- Testing the application for portability depending upon suitable features.

2. Literature Review

In this section we will be discussing about the different existing techniques given in three different research papers. These papers deal with the problem of portability within Cloud Platforms, and give the solution to the problem.

2.1 Towards Application Portability in Platform as a Service (Stefan Kolb et al,2014)

In this technique the writers have discussed various issues which occur while porting an application to different cloud. They have studies over 68 different cloud vendors, and derived a standardized profile with a common set of capabilities that can be found among PaaS providers and matched with one another to check application portability based on ecosystem capabilities. A general PaaS cloud can be divided into three layers. Each layer has some parameters and functions. Which play an important role in efficient working of cloud. Understanding and replicating these properties in the target cloud is necessary in order to achieve application portability.

Infrastructure layer

– This layer deals with the hardware part of cloud like CPU, RAM and Storage units. To achieve portability of application it is necessary to ensure that the hardware specifications of target cloud match to that of the source cloud.
– Another important factor is the geographical region the application will be deployed in. This is particularly interesting because of legal and performance reasons. As bandwidth capacities keep increasing on the customers end, latency is one of the main constraining factors for publicly hosted applications

Platform layer

– The platform is the main deliverable of a PaaS offering and includes the application hosting environment delivered as a service. It consists of two stacks, both stacks can be combined by the customers via bindings. Those bindings are generally environment variables that include important properties of the services like endpoint URLs, credentials, and other configuration information.
-The runtime stack includes the basic runtimes offered by the PaaS, i.e. the programming languages that applications can be written in. Furthermore, we see the vast popularity of language specific frameworks like Ruby on Rails which are leveraged to develop todays applications.
-The service stack includes mostly latency and performance critical core services like data stores. Add-on services are supplied by third party service providers that integrate with the PaaS.

Management layer

– It allows control over the deployed applications and the configuration settings of the platform. The management layer includes the abilities to deploy and manage the lifecycle of the applications.

This encompasses pushing, starting, and stopping of applications. Moreover, the provisioning of all native services and add-ons is initiated from the management tier.

From the above we can say that managing several layers and the possible manifestations of the inherited components and capabilities is complex, thus portability between PaaS is a difficult task. Nearly every vendor has its particular management API, platform configuration and restrictions, resulting in a strong dependency to a certain provider and significant costs when migrating from one vendor to another In order to make PaaS offerings comparable and match able the writer of this paper grouped a set of core properties of their business and ecosystem perspectives into a standardized machine-readable PaaS profile.

A PaaS profile contains the following information

a) Meta Information are stored in order to keep track of the profile itself, whether it is updated or not, is it vendor verified, when was the profile last verified.
b) Business Properties include the official name of the PaaS offering, URL to the PaaS webpage, stage of lifecycle like Beta or Alpha, pricing model, SLA etc.
c) Ecosystem Properties contains information regarding horizontal vertical scaling, hosting properties, information regarding infrastructure, runtime, middleware, frameworks and services, languages supported, version of the system etc. They created profiles for over 68 PaaS offerings by various vendors with their core properties and functionalities. In order to access and evaluate these profiles they developed a Web Application through which one can check various characteristics of a PaaS offering and compare them with other PaaS offerings.

This will help in identifying how the source and target cloud are similar / different, which framework / language the source and target cloud support, it will also help the user identify the target cloud which best suits the needs. But the main issue with this paper is that it doesnt focus / consider low level portability issues like standardization of the management interface, management of API differences etc... The researches of this paper stated that they are conducting studies on the above issue and are trying to come up with a solution soon.

2.2 Automatic cloud vendor API analysis for supporting cloud application portability (Giuseppina Cretella et al, 2012)

Cloud is an emerging field in todays world and it is becoming important by the day as new providers keep arising every day. It is very difficult to find a firm that can provide all the services under one roof. The issue that arises are invoking and describing services to the specified requirements and to communicate. Cloud application portability a concept as explained earlier is one which refers to the ability to move applications between cloud vendors with minimum level of integration issues.

This paper offers an API mapping approach commonly known as API alignment technique. This technique shines light on the concept that to provide support to cloud application portability, is to understand functionalities offered by APIs and map them with other provider APIs. Alignment and porting of the application is done through the extraction of knowledge from APIs (i.e. their functionalities). To put all of this under one roof one must have good knowledge on two things, they are -

- Reverse engineering- It is taking apart an object to see how it works so that duplication or enhancement of the object is possible.
- Program comprehension and its stages/directions- It is a domain of computer science concerned with the

ways software engineers maintain existing source code. The cognitive and other processes involved are identified and studied. The results are used to develop tools and training.

API alignment techniques

Points that need to be looked into for this technique are

- Mapping of APIs with APIs of other providers.
- Combination of API, UML design, manuals and documentation are used to extract information on the functionalities, which are then ported along with semantic annotations.
- Analysis of the workflow, the architecture and prototypical implementation.

This technique has three ways in which it can be implemented, they are

a) API alignment with neutralization Terms to be known in this methodNeutral API- It also called as the leader API which contains all the functionalities without the architectural details.
Graph- Assume that an API for a given domain is available which can be semantically described using a graph.

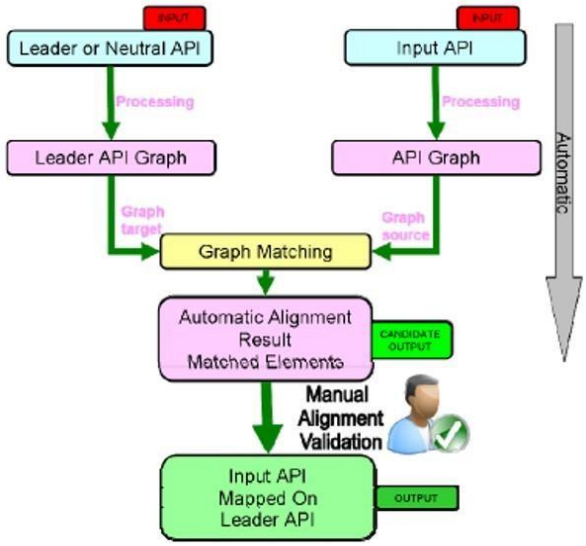

Fig .2 API Alignment (Giuseppina Cretella *et al*, 2012)

- The input and the leader API are converted to their respective graphs which need to be matched.
- Then automatic matching and element result alignment is carried out which is also called as the candidate output.
- Now that the API alignment is done, a manual alignment validation is carried to verify the automatic process and then the output is achieved. It is to be noted that here mapping of functionalities of the services provided by cloud vendors takes place which in turn makes inter portability easy. b) API alignment using semantic approach Terms to be known in this method are the ones learnt in the previous method as well as the following - Ontology- It is an explicit specification of conceptualization in which every node is a concept in itself.

OWL- Web Ontology Language is a semantic mark-up language for sharing and publishing ontologies on the World Wide Web. OWL was developed as a vocabulary extension of RDF.

Annotated API-APIs that are self-explanatory and are more descriptive because of their add-on functionalities.

Process of alignment

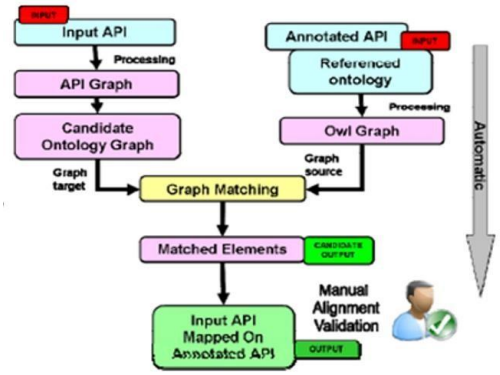

Fig .3 API Automatic Alignment (Giuseppina Cretella *et al*, 2012)

- The input and the annotated API are converted to their respective graphs which need to be matched. The input API is converted to the API graph which then is converted to the candidate ontological graph as it has to be mapped with the annotated APIs graph and the annotated API with the help of a referenced ontology is converted into an OWL graph.
- Then automatic graph matching and element result alignment is carried out which gives the candidate output.
- Now that the API alignment is done, a manual alignment validation is carried to verify the automatic process and then the output is achieved. There are general things to be noted here that this method is better than the 1st method as here all the semantics of both the functionalities are mapped which increases the mapping efficiency

c) API alignment using semantic annotations Terms to be known in this method are the ones learnt in the previous method as well as the following: Base Ontology- This ontology is declared in this document both in human-readable form (Figure - 4) and machine-readable simple HTML ontology extension i.e. SHOE form.

Process of alignment?

- The domain specific API and the application domain ontology are converted to their respective graphs

which need to be matched. The inputted domain specific API are converted to API graphs and the application domain ontology along with an external input (functional ontology) is converted to the ontology graph.

• The API graph is then converted to a candidate graph as well as into an API syntax ontology (base ontology) with the help of an ontological API representation. The candidate and the ontology graphs are matched and then automatic element result alignment is carried out which gives the candidate output.

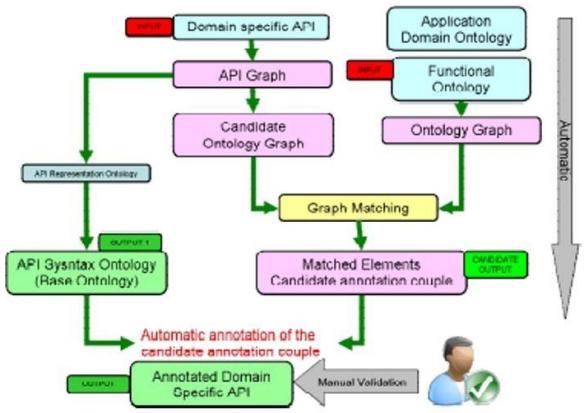

Fig .4 API Automatic Semantic Annotation (Giuseppina Cretella *et al*, 2012)

• The matched elements or the candidate output undergoes automatic annotation along with the API base ontology to form annotated domain specific API.

• Now that the domain specific API alignment is done, a manual alignment validation is carried to verify the automatic process and then the output is achieved.

In this approach it is imperative to note that the matched and aligned API that we receive is domain specific and is more accurate than both the previous methods.

*2.3 Cloud Application Portability: An Initial View (*Fotis Gonidis *et al)*

One of challenge in Cloud Computing is application portability between two or more cloud environments. The application developed in this paper was ported on four different platform Open Shift, Google App Engine, Heroku, and Amazon Elastic Beanstalk.

PaaS, a key benefit is that users can develop and deploy applications without the burden of setting up and maintaining the necessary programming environment and infrastructure that the application is executed on. Different cloud application platform offerings are characterized by considerable heterogeneity. Because of incompatibilities, users that develop applications on a specific platform may encounter significant problems when trying to deploy their application in a different environment. This gives rise to the familiar problem of vendor lock-in. Consumers need to be able to easily change between cloud providers and should be free to choose the one that better serves their needs in terms of quality and/or cost.

A real example to illustrate this argument is the case of Coghead, an online application development platform supporting the development and hosting of data-driven applications. The platform had managed to attract hundreds of developers before it suddenly announced that it would stop operating, calling all customers to export the data that was stored in their applications, but not giving them the option to port the actual applications to some other platform. a) Cloud platforms:

This section talks about different cloud developing platforms, this will also help to understand what type or what cloud platform one should use for portability.

Table 1 Comparison of different Cloud Platforms.

Parameter	Open Shift	Google App Engine	Zoho Creator
Language	Java,PHP,Ruby,Python	Java, Python, Go	Scripting language.
Database	MySQL,mangoSQL	Google Cloud SQL.	No database.
Category	Provide and Support to Standard Method	Standard Programming Language.	Adapt to Application paradigm.
File Storage Support	No support	Supports	Supports
Local Library	No local Library.	In forms of API	As Toolkit.
Advantage	Flexibility.	Less time required ,simple.	Easy to code.
Disadvantage	Time consuming, complex.	Predefined APIs make codes tough.	Limited scope.

b) Portability issues in cloud applications

• Programming languages and/or frameworks the specific programming languages and frameworks that an application has been built with is obviously a major determinant for cross-platform deployment. Each cloud platform supports certain languages, frameworks, and versions thereof.

• Platform specific services, an important characteristics of several cloud platforms is that they provide certain services via specific APIs. A service can be considered as high-level functionality that the provider can use without the need to implement it from scratch. A portability issue arises when the application needs to be ported to a different cloud platform. There are two cases:

– The target platform doesnt provide the full set of services that the application uses. For example,

SMS services and monitoring services are not supported. In this case the developer would need to recreate the missing functionality from scratch on the new target platform.

– The target platform supports the services that the application uses but provides different APIs in order to use them. In this case the developer would need to modify the application code and align it with the APIs of the new target platform.

Data Storage is an essential part of an application.

There are two types of data storage

– Database stores: used for storing structured data while the second one could be perceived as an analogy to a hard disc drive on the cloud.
– File Stores : used for storing data in files like txt, excel, xml etc.
Therefore the following conflicts may arise when trying to port an application across various platforms:
– Incompatible data structures: As it became clear there is a wide range of available databases where each one of them adopts a different data structure.

Portability issues are bound to arise when trying to move data from a SQL to a NoSQL database, but also between different types of NoSQL databases, e.g. when moving from a key-value store to a document store. – Different query languages: Apart from the incompatibility due to different data structures, conflicts may occur at the way of querying data. Databases use their own APIs or query languages.

Therefore even when data is moved across databases of the same category (e.g. a document store), portability issues may arise concerning the way the database is accessed.
– Data migration (export/import formats): Another issue that should be considered is data migration. It may happen that an exported database cannot directly be imported to another database engine due to incompatible data formats.

Use of graphical interface. Human users of the cloud application can manually perform operations on the file storage space.
– Platform specific configuration files Similar to the configuration files that traditional software applications require in order to instruct the hosting environment on how to execute the applications, cloud platforms may require analogous configuration files. For example Google App Engine uses the appengineweb.xml file. The process of adapting the configuration files to each target cloud platform adds to the overall overhead of cross-platform deployment of a cloud application.

c) Analysis of this technique

High degree of heterogeneity between cloud application platforms, any approach to tackle application portability needs to target a specific set/class of platforms that present similar characteristics.

Problems: such as programming languages and frameworks, database offerings, file storage service, platform specific services and configuration files. The paper presented related work aimed at addressing the challenge of cloud application portability and distinguished between two generic approaches to tackle the issue of application Portability: standardization and intermediation. Standardization addresses cross-platform portability through the adoption of common standards by cloud providers. Alternatively intermediation enables developers to create applications independently of a specific platform and then bind them to particular target platforms through some form of automatic translation.

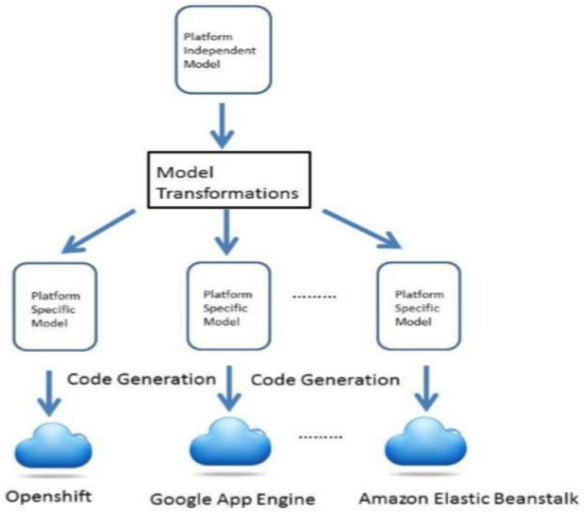

Fig. 5 Intermediation (Fotis Gonidis *et al*)

3. Proposed Work

In this section we will talk about how we plan on implementing and executing the project. Here the discussion is on the different modules and classes:

3.1 Basic Flow Chart of proposed work

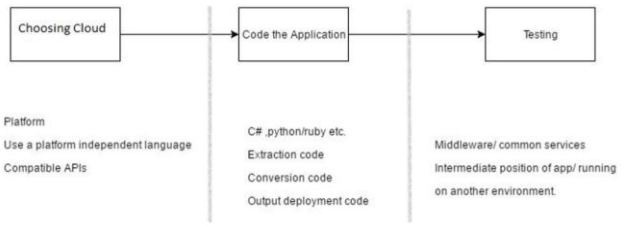

Fig .6 Flow Chart of the Proposed System

The Basic flowchart as given above will be executed in following way: a) Step 1 – Cloud creation
– Revising and using platform independent Application.
– Listing out the compatible APIs b) Step 2
– Coding language like C, python to make the portable application. – Writing code of:
- Extraction
- Conversion – Output deployment c) Step 3
– Using concept of intermediation : Placing the application in business logic in a three tier system. – Middle-wear environment: Using Platform that provide common services and have equivalent rules and regulations.

3.2 Class Diagram

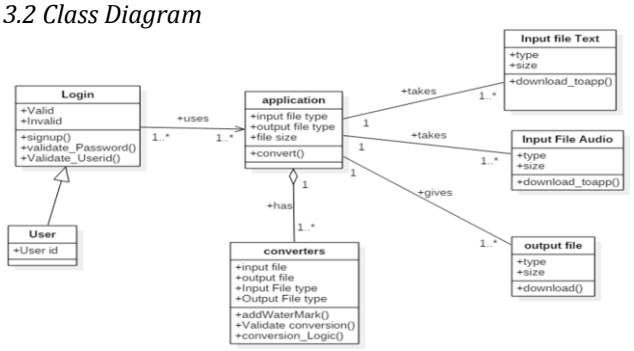

Fig.7 Class Diagram of Proposed application

The above class diagram is the basic classes and there functionalities and attributes, and the correlation amongst them.

– User: User must Provide User-Id.
– Login: Validates User id/Password.
– Application: Takes in the Input file, asks for Output file type. Fetches the right algorithm from converters class.
– Input file Text: to take in file types like Pdf, doc, docx. Validate the input and download it to the application for conversion.
– Input file audio: to take in file types like mp3, mp4, avi. Validate the input and download it to the application for conversion.
– Converters: Check the input file. Validate if the input file can be converted to required output file. E.g Invalid: mp4 to Pdf, Valid: Mp4 to mp3.The return the right logic and algorithm to application class. – Output: The required file, can be downloaded by user.

Conclusions

Cloud is one of the most frequently used structure in computing industry today with the booming needs of Cloud the problems in it have become crucial and need to be solved. The increase in demand of Cloud Computing in industry and the necessity to solve the problem of Portability. Multiple providers use some common basic functions which along with extra bundles can help us to achieve this. Though being complex, its not impossible. Many approaches have already been recorded.It is necessary to ensure that there are at least reasonable ways of providing the same feature/function on multiple platforms, even if the same APIs won't work across them all. Gives us the very reason to continue and begin with experimentation on portability. Lack of Standardization makes the problem of portability tougher. Not only being the area of our interest but also an important change for all Cloud user is needed. Finding either a standard solution or another method using APIs would be our approach to our problem. We will be trying to solve this problem by developing our own Cloud Application. This Application will do the following:

– Type Conversion, e.g. PDF to Word Doc. – Download media file in various forms like mp4, 3gp, mp3, etc.

Once the application is developed we will try to run this application on more than three clouds. Like Google, Microsoft Azure and Amazon etc. Other important feature of the application will be adding is the user can directly save it on google drive and use google Docs to edit the Doc or other Google extension. In this way the user may not have to have a local storage of the document.

This application will be useful for those member of society who have older versions of software with simple net connection and a computing device one can access and download the Document in the required format.

References

Recommendations of the National Institute of Standards and Technology(September 2011) ,The NIST Definition of Cloud Computing, September 2011 NIST Special Publication 800-145.

Danilo Ardagna(2015), Cloud and multi-cloud computing: current challenges and future applications. In Proceedings of the Seventh International Workshop on Principles of Engineering ServiceOriented and Cloud Systems, IEEE Press, pp. 1-2.

Rohitash Kumar Banyal, Paril Jain, and Veerendra Kumar Jain(2013), Multi-factor authentication framework for cloud computing,Computational Intelligence, Modelling and Simulation (CIMSim), 2013 Fifth International Conference ,IEEE, pp. 105–110. [4] Chandni Bhatt and Richa Sharma(2014), Data leakage detection. International Journal of Computer Science and Information Technologies, pp. 2556–2558.

Victor Boyko, Philip MacKenzie, and Sarvar Patel(2000), Provably secure password-authenticated key exchange using diffie-hellman. Advances in CryptologyEurocrypt 2000, Springer,pp. 156–171.

Amar Ramesh Buchade and Rajesh Ingle(2014), Key management for cloud data storage: Methods and comparisons, Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference ,IEEE, pp. 263–270.

Kajal Chachapara and Sunny Bhadlawala(2013), Secure sharing with cryptography in cloud computing, Engineering (NUiCONE), 2013 Nirma University International Conference,IEEE , pp. 1–3.

Giuseppina Cretella and Beniamino Di Martino(2012),Towards automatic analysis of cloud vendors apis for supporting cloud application portability, Complex, intelligent and software intensive systems (CISIS), 2012 sixth international conference ,IEEE, pp. 61–67.

H. A. Dinesha and V. K. Agrawal(2012), Multi-level authentication technique for accessing cloud services, 2012 International Conference on Computing, Communication and Applications, pp. 1–4.

Fotis Gonidis, Anthony J. H. Simons, Iraklis Paraskakis, and Dimitrios Kourtesis, Cloud application portability: An initial view. In Proceedings of the 6th Balkan Conference in Informatics, ser. BCI 13, pp. 275– 282. Online. Available: http://doi.acm.org/10.1145/2490257.2490290.

Stefan Kolb and Guido Wirtz(2014), Towards application portability in platform as a service, Service Oriented System Engineering (SOSE), IEEE 8th International Symposium , pp. 218– 229.

Sun Lei, Dai Zishan, and Guo Jindi(2010), Research on key management infrastructure in cloud computing environment, Grid and Cooperative Computing (GCC), 2010 9th International Conference ,IEEE, pp. 404–407.

K Mythili and H Anandakumar(2013),Trust management approach for secure and privacy data access in cloud computing, Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference ,IEEE, pp. 923–927.

P. Prasad, B. Ojha, R. R. Shahi, R. Lal, A. Vaish, and U. Goel(2011), 3 dimensional security in cloud computing, Computer Research and Development (ICCRD), 3rd International Conference ,vol 3, pp. 198– 201.

Shabnam Sharma and Usha Mittal(2013), Comparative analysis of various authentication techniques in cloud computing, International Journal of Innovative Research in Science, Engineering and Technology, pp. 994–998.

Eric Griffith(2016), What is Cloud Computing?, Online, Available: http://in.pcmag.com/ networking communications-software/38970/feature/what-iscloud computing.

Network bottleneck, Technopedia.Online. Available: https://www.techopedia.com/definition/24819/network-bottleneck.

Vangie Beal,Cloud Computing :The cloud,Webopedia

Lav R Varshney and Krishna C Ratakonda(2014), An information theoretic view of cloud workloads, Cloud Engineering (IC2E), 2014 IEEE International Conference , pp. 466–471.